



*National University of Science and Technology
POLITEHNICA Bucharest
Faculty of Industrial Engineering and Robotics*



Journal of Industrial Engineering and Robotics

2026, Volume 10, Issue 3

Scientific Committee of Journal of Industrial Engineering and Robotics

Prof.dr.ing.	AMZA Catalin
Conf.dr.ing.	BACIU Florin
Conf.dr.ing.	CATANA Madalin-Gabriel
S.l.dr.ing.	DIJMARESCU Manuela-Roxana
S.l.dr.ing.	NICULAE Elisabeta
Conf.dr.ing.	POPA Laurentiu
S.l.dr.ing.	POPESCU Adrian
Prof.dr.ing.	SEVERIN Irina
S.l.dr.ing.	TUDOSE Daniela Ioana
Conf.dr.ing.	UNGUREANU Liviu Marian

Editorial board of Journal of Industrial Engineering and Robotics

Prof.dr.ing.	DUMITRESCU Andrei
Conf.dr.ing.	PARPALA Radu
S.l.dr.ing.	RADU Constantin
S.l.dr.ing.	ROTARU Alexandra
S.l.dr.ing.	TUDOSE Virgil

Editors

Prof.dr.ing.	DOICIN Cristian
Conf.dr.ing.	APOSTOL Dragoş-Alexandru

INTELLIGENT SYSTEM FOR DETECTING VIBRATIONS DURING THE PRINTING OF A 3D MODEL

BURLEANU Sebastian-Bogdan, STOLNICEANU Stefan-Adrian and CAZACU Carmen-Cristiana (Scientific Coordinator)

¹Faculty: Industrial Engineering and Robotics, Specialization: Digital Production Systems, Year of Study: II
burleanubogdan05@gmail.com

***ABSTRACT:** This paper presents an IIoT-based vibration monitoring system for the Sovol SV06 Plus ACE 3D printer. Two ADXL345 accelerometers capture vibrations via I²C, processed on a Raspberry Pi 5 using RMS analysis to recommend speed and flow rate adjustments. Feedback is provided locally via LCD display and remotely through a ThingWorx Mashup. Experiments at speeds from 40 to 550 mm/s confirm resonance detection and corrective recommendations in under 2.56 seconds.*

***KEYWORDS:** IIoT, vibration monitoring, ADXL345, 3D printing, Raspberry Pi 5.*

1. Introduction

FDM has seen widespread adoption in industrial prototyping due to its low cost and material versatility [1], yet increasing print speeds introduce mechanical vibrations that cause layer-line ringing, stringing, and dimensional inaccuracies [2], [3]. Existing mitigation strategies, including Input Shaping in Klipper, rely on one-time calibration and lack continuous monitoring or historical data retention for predictive maintenance [4]. This work addresses these gaps through an autonomous, battery-powered IIoT node that continuously acquires dual-point vibration signatures, performs edge analysis, and delivers corrective recommendations to both a local LCD and a remote ThingWorx dashboard, operable in any desktop manufacturing environment without external infrastructure.

2. State of the Art

MEMS accelerometers are well established for manufacturing vibration characterisation; the ADXL345 in particular has been validated for machine-tool monitoring owing to its configurable data rate (up to 3200 Hz) and ± 16 g range [5]. Ghungrad et al. (2022) confirmed that FDM resonance peaks correlate with printhead velocity and belt tension, typically occurring between 5–50 Hz [6], while Input Shaping — though effective — assumes a fixed resonance frequency and cannot adapt to tool wear or thermal drift. ThingWorx has been successfully paired with Raspberry Pi edge nodes for CNC and robotics monitoring [7], [8]; the present work extends this architecture to desktop FDM printing with dual-point sensing, ML classification, and a battery-powered autonomous form factor.

3. Case Study

3.1 System Architecture

The monitoring system is built around a Raspberry Pi 5 (ARM Cortex-A76, 4 GB RAM) that interfaces with two ADXL345 MEMS accelerometers via the I²C bus at 100 kHz. Sensor 1 (address 0x53, SDO tied to GND) is mounted on the extruder head to capture high-frequency toolpath vibrations. Sensor 2 (address 0x1D, SDO tied to 3.3 V) is mounted on the print bed to monitor low-frequency frame resonances. Positioning the two sensors at opposite ends of the mechanical transmission chain — extruder carriage and print bed — enables the system to spatially correlate vibration events and determine whether a detected anomaly originates from printhead dynamics, belt slack, or global frame excitation. A 16×2 LCD (1602A with PCF8574 I²C expander, address 0x27) provides local status display, and three tactile buttons on GPIO 17, 27, and 22 allow page navigation and sensor recalibration without a connected laptop. The button interface supports three distinct display pages: a real-time RMS readout page, a recommendations page showing the currently suggested speed and flow rate, and a system status page reporting sensor health and uptime. This eliminates the need for an external monitor

INTELLIGENT SYSTEM FOR DETECTING VIBRATIONS DURING THE PRINTING OF A 3D MODEL

or SSH connection during print sessions, making the node fully self-contained in bench operation. Power is supplied by an 11.1 V LiPo battery through a step-down regulator. For stable Raspberry Pi 5 operation, a 5 V rail capable of at least 3 A is recommended, while future versions should use a 5 V / 5 A regulator for full-load reliability. The choice of a switching step-down regulator over a linear alternative was driven by thermal efficiency requirements: at the observed average current draw of 0.6 A, a linear regulator would dissipate approximately 3.7 W as heat, which is unacceptable in an enclosed portable enclosure, whereas the Pololu D24V10F5 achieves greater than 90% conversion efficiency across the full load range.

Table 1 summarises the hardware components and their functional roles.

Table 1. Hardware components and functional specifications

Component	Model	Function
Edge computer	Raspberry Pi 5	Edge processing & IIoT comm.
Vibration Sensors	2× ADXL345 (MEMS)	Real-time I2C acceleration data
Local Display	LCD 1602A + PCF8574	Live data & AI recommendations
Voltage Regulator	Pololu D24V10F5	Step-down 11.1 V → 5 V / 1 A
Power Source	LiPo 11.1 V / 1 Ah	Autonomous portable operation
Control Interface	3× Tactile Buttons	Page navigation & recalibration
Cloud Platform	ThingWorx (PTC)	Remote telemetry & data storage

The I²C addressing configuration for all connected peripherals is presented in Table 2.

Table 2. I²C device addressing

Device	I ² C Address	Connection Logic
ADXL345 - Sensor 1	0x53	SDO → GND; mounted on extruder head
ADXL345 - Sensor 2	0x1D	SDO → 3.3 V; mounted on print bed
LCD 1602A (PCF8574)	0x27	Local information display

The complete electrical wiring diagram, including all GPIO connections, I²C bus topology, button wiring, and the 11.1 V battery power path through the step-down regulator, is presented in Figure 1.

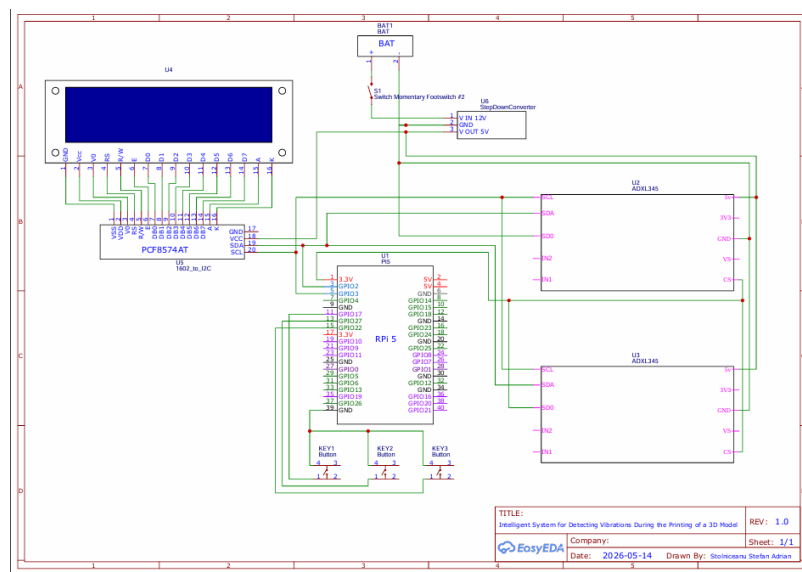


Fig. 1. Electrical circuit schematic of the portable monitoring system

3.2 Target Platform: Sovol SV06 Plus ACE

The Sovol SV06 Plus ACE is the target platform for this study. Its maximum rated print speed of 500 mm/s (with 15 000 mm/s² acceleration), Klipper open-source firmware, and native dual-accelerometer topology make it an ideal candidate for IIoT vibration monitoring. The printer replaces the original linear rod X/Y motion system with a dual-axis core metal bearing and all-metal rail system, significantly reducing mechanical play and frame-transmitted noise [8]. Klipper's built-in Input Shaping relies on accelerometers embedded in both the extruder carriage and the print bed - a topology identical to the external ADXL345 configuration proposed here - confirming that the IIoT architecture integrates naturally with the ACE hardware and provides an additional independent cloud telemetry layer beyond what the printer's native interface exposes.

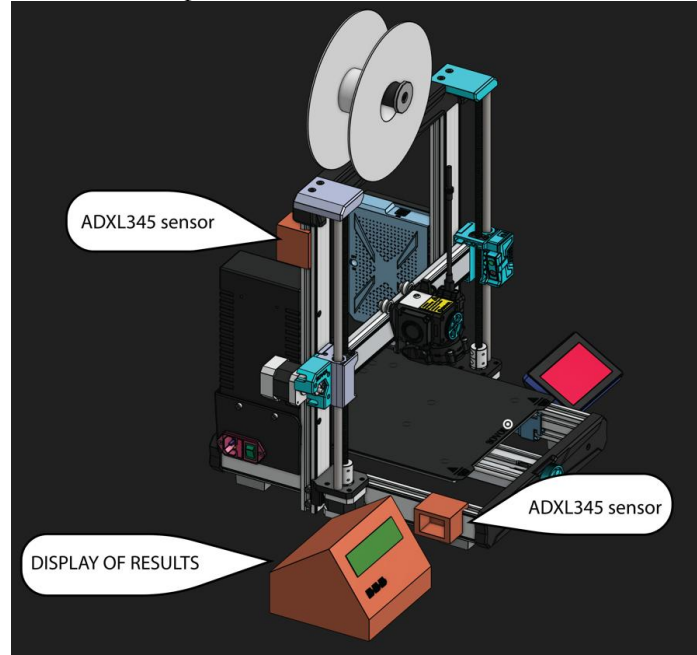


Fig. 2. Sovol SV06 Plus ACE - target platform

3.3 Signal Processing and Control Logic

Raw acceleration values from each ADXL345 are sampled at 100 Hz. For each sample, the resultant acceleration magnitude A_{res} is computed as:

$$\boxed{\phantom{A_{res} = \sqrt{a_x^2 + a_y^2 + a_z^2}}} \quad (1)$$

where a_x , a_y , a_z are the instantaneous acceleration components [m/s²]. To isolate dynamic vibration from the static gravity component, the vibration intensity ΔV is calculated as:

$$\boxed{\phantom{\Delta V = \sqrt{A_{res}^2 - g^2}}} \quad (2)$$

where $g \approx 9.81$ m/s². The root-mean-square vibration level RMS_{mag} is computed over a sliding window of $N = 256$ samples (2.56 s):

$$\boxed{\phantom{RMS_{mag} = \sqrt{\frac{1}{N} \sum_{i=1}^N \Delta V_i^2}}} \quad (3)$$

The flow adjustment recommendation F_{adj} is derived from the deviation of ΔV above the safety threshold T_s :

INTELLIGENT SYSTEM FOR DETECTING VIBRATIONS DURING THE PRINTING OF A 3D MODEL



(4)

A Butterworth low-pass filter (order 4, cut-off 40 Hz) is applied before FFT analysis to stay within the Nyquist limit of 50 Hz. A Random Forest Regressor (200 trees, max depth 12) trained on 4 000 synthetic samples handles speed and flow regression, while a Gradient Boosting Classifier categorises vibration severity into three states: STABLE (RMS < 0.05 g), MODERATE (0.05-0.30 g), and CRITICAL (RMS > 0.30 g).

3.4 Experimental Validation

Controlled print runs were performed on the Sovol SV06 Plus ACE using PLA at 210 °C / 60 °C bed temperature, 0.2 mm layer height, and 20 % infill. Three part geometries were tested across three speed ranges to characterise the system across the full operational envelope of the printer.

Part 1 - Raspberry Pi 5 case (complex geometry with port openings and mounting holes) was printed at low speeds (40-100 mm/s). Results are presented in Table 3.

Table 3. Experimental results - P1: Raspberry Pi 5 case (40-100 mm/s)

Test	Speed (mm/s)	Flow (%)	RMS avg (g)	RMS max (g)	Dom.Freq (Hz)	State	Rec.Speed (mm/s)	Visual Observation
1	40	100	0.0610	0.1778	5	MODERATE	38	Slight roughness on side walls
2	60	98	0.0490	0.1325	6.5	STABLE	60	Good surface; correct bridging over USB ports
3	80	98	0.0484	0.1082	7	STABLE	80	Optimal quality; precise corners & mounting holes
4	100	98	0.0561	0.1189	8.5	MODERATE	95	Minor stringing on internal cantilevers

Part 2 - Calibration cube (20×20×20 mm, simple flat geometry) was printed at mid-range speeds (250-450 mm/s). Results are presented in Table 4.

Table 4. Experimental results - P2: calibration cube (250-450 mm/s)

Test	Speed (mm/s)	Flow (%)	RMS avg (g)	RMS max (g)	Dom.Freq (Hz)	State	Rec.Speed (mm/s)	Visual Observation
1	250	100	0.0900	0.1445	18	MODERATE	237	Layer lines on vertical faces; well-defined corners

INTELLIGENT SYSTEM FOR DETECTING VIBRATIONS DURING THE PRINTING OF A 3D MODEL

2	300	100	0.0486	0.0747	12	STABLE	300	Uniform surface on all 6 faces; dims within ± 0.2 mm
3	400	98	0.0520	0.0634	22	MODERATE	380	Slight undulation on top horizontal faces
4	450	98	0.0808	0.2591	28	MODERATE	427	Detectable frame vibrations; micro-undulations

Part 3 - Cylinder ($\varnothing 25$ mm $\times$ 40 mm, curvilinear geometry) was used for repeatability testing at high speeds (500-550 mm/s). Results are presented in Table 5.

Table 5. Experimental results - P3: cylinder (500-550 mm/s)

Test	Speed (mm/s)	Flow (%)	RMS avg (g)	RMS max (g)	Dom.Freq (Hz)	State	Rec.Speed (mm/s)	Visual Observation
R1	500	100	0.0682	0.1434	35	MODERATE	475	Smooth cylinder; slight ovality at base (≤ 0.3 mm)
R2	500	98	0.0512	0.0644	33	MODERATE	475	Good repeatability vs R1; uniform lateral surface
R3	500	98	0.0455	0.0588	34	STABLE	500	Best result at 500 mm/s; no visible defects
R4	550	98	0.0835	0.1612	42	MODERATE	522	High vibration at max speed; micro-undulations

INTELLIGENT SYSTEM FOR DETECTING VIBRATIONS DURING THE PRINTING OF A 3D MODEL

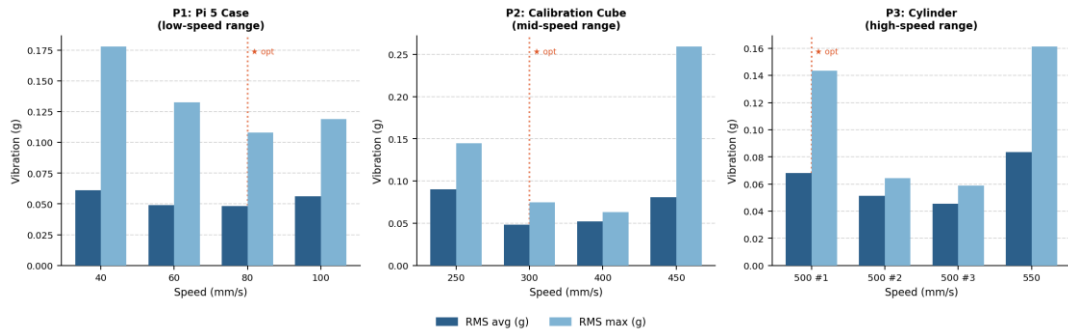


Fig. 3. RMS vibration (average and peak) across all tested speeds and geometries

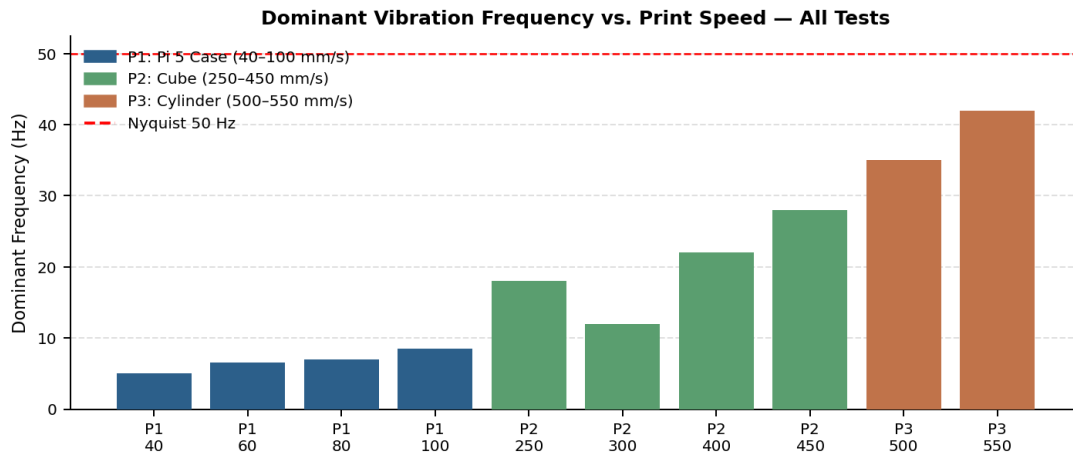


Fig. 4. Dominant vibration frequency vs. print speed across all tests

The charts above reveal consistent trends across all three parts. The dominant frequency generally increases with print speed, although geometry and toolpath dynamics introduce local deviations- from 5 Hz at 40 mm/s to 42 Hz at 550 mm/s - approaching but never exceeding the 50 Hz Nyquist limit. For Part 1, the optimal speed is 80 mm/s (RMS avg 0.0484 g), with a non-intuitive resonance peak at 40 mm/s (0.0610 g) indicating a low-speed resonance mode. For Part 2, the optimal speed is 300 mm/s (RMS avg 0.0486 g); above 400 mm/s, peak RMS rises sharply to 0.2591 g at 450 mm/s. For Part 3, three consecutive runs at 500 mm/s show good repeatability (RMS 0.0455-0.0682 g), while 550 mm/s produces elevated vibrations (0.0835 g avg). The 550 mm/s run was included as a stress-test condition above the nominal rated speed, in order to evaluate the monitoring system near the mechanical limit of the printer. System response time for generating a recommendation was consistently below 2.56 s across all sessions. The low-speed resonance observed at 40 mm/s for Part 1 is particularly noteworthy: it suggests that belt-driven Cartesian mechanisms can exhibit structural resonances well below the frequencies typically targeted by Input Shaping, which is commonly tuned for speeds above 100 mm/s. This finding underlines the importance of continuous monitoring across the full speed envelope rather than relying on a fixed calibration point. The geometry-dependent behaviour is equally significant - the calibration cube, with its flat faces and abrupt direction changes, produced sharper RMS peaks at high speeds compared to the cylinder, whose curvilinear toolpath distributes acceleration loads more gradually. These observations confirm that part geometry is a first-order variable in vibration severity and should be factored into any adaptive speed control strategy. The dual-sensor configuration proved essential in discriminating between extruder-local vibrations and frame-level resonances: in several mid-range tests, Sensor 1 registered MODERATE states while Sensor 2 remained STABLE, indicating that the vibration source was confined to the printhead carriage rather than propagating through the entire frame.

3.5 Cloud Visualisation via ThingWorx

INTELLIGENT SYSTEM FOR DETECTING VIBRATIONS DURING THE PRINTING OF A 3D MODEL

Processed vibration data is transmitted to the ThingWorx platform via HTTP REST API calls. The custom Mashup interface (Fig. 5) acts as a digital twin of the printing process: two gauge widgets display real-time RMS magnitudes from each sensor, a tabular view shows individual X, Y, Z axis accelerations alongside current speed and flow rate, and historical trend charts enable the operator to identify degradation patterns and schedule preventive maintenance.

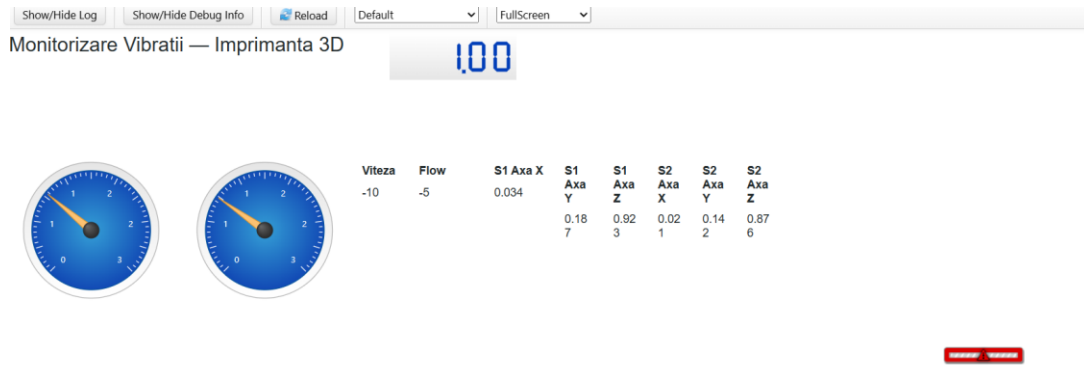


Fig. 5. ThingWorx Mashup interface displaying real-time vibration telemetry

3.6 Software Architecture

The Python software stack is organised into six modules: sensor.py (ADXL345 driver and calibration), filter_fft.py (signal windowing, Butterworth filter, and FFT), model.py (Random Forest / Gradient Boosting inference), lcd_display.py (I²C LCD driver), buttons.py (GPIO interrupt handler), and logger.py (CSV data logging). The main.py orchestrator coordinates all modules in a 10 ms control loop (100 Hz). A systemd service unit (iiot.service) launches main.py automatically at boot, enabling fully headless autonomous operation.

4. Conclusions

This work presents a low-cost autonomous IIoT vibration monitoring system for FDM additive manufacturing, contributing a dual-point MEMS sensing architecture, a battery-powered self-contained form factor, a real-time ML classification pipeline with 2.56-second response, and a ThingWorx cloud Mashup for remote monitoring. Validation across three part geometries confirms geometry-specific optimal speeds, detects low-speed resonances missed by Input Shaping, and corroborates the FFT pipeline through monotonically increasing dominant frequency (5 Hz → 42 Hz). The system demonstrates that edge-level signal processing on a Raspberry Pi 5 is sufficient for real-time vibration classification without cloud latency, making it practical for any desktop manufacturing environment without dedicated network infrastructure. The battery-powered form factor further removes dependency on fixed workbench installations, enabling the node to be redeployed across multiple printers with no hardware modification. A key limitation of the current implementation is the use of a fixed safety threshold T_s , which does not account for material-specific or geometry-specific vibration tolerances; future iterations will explore adaptive thresholding informed by historical RMS distributions stored on ThingWorx. Future work will also target closed-loop Klipper control, in which the system issues direct G-code speed override commands in response to CRITICAL vibration states, and dataset expansion through real-world print campaigns for broader anomaly detection coverage.

5. Bibliography

- [1]. Botta, A., de Donato, W., Persico, V. and Pescapé, A. (2016), "Integration of Cloud Computing and Internet of Things: A Survey", *Future Generation Computer Systems*, vol. 56, pp. 684-700, ISSN 0167-739X.
- [2]. Analog Devices Inc. (2009), ADXL345 Digital Accelerometer - Datasheet, Rev. E, Norwood, MA.

INTELLIGENT SYSTEM FOR DETECTING VIBRATIONS DURING THE PRINTING OF A 3D MODEL

- [3]. PTC Inc. (2025), ThingWorx Platform Developer's Guide, Reference Manual, Boston.
- [4]. Sovol 3D (2024), Sovol SV06 Plus ACE User Manual and Calibration Guide, Shenzhen.
- [5]. Drăghici, G. (1999), Ingineria integrată a produselor, Editura Eurobit, ISBN 973-96065-7-1, Timișoara.
- [6]. Raspberry Pi Ltd (2024), Raspberry Pi 5 Documentation and GPIO Pinout Guide, Cambridge.
- [7]. Ghungrad, S., Faierson, E. and Hauser, T. (2022), "In-situ monitoring of FDM 3D printing via accelerometer-based vibration sensing", Journal of Manufacturing Processes, vol. 84, pp. 188-198, ISSN 1526-6125.
- [8]. Turner, B. N., Strong, R. and Gold, S. A. (2014), "A review of melt extrusion additive manufacturing processes", Rapid Prototyping Journal, vol. 20, no. 3, pp. 192–204, ISSN 1355-2546.

6. Notations

ax, ay, az = acceleration components on the X, Y and Z axes [m/s²];
A_{res} = resultant acceleration magnitude [m/s²];
g = gravitational acceleration (≈ 9.81) [m/s²];
 ΔV = vibration intensity variation [m/s²];
RMS_{mag} = root-mean-square vibration magnitude [g];
F_{adj} = flow adjustment factor [%];
T_s = vibration safety threshold [m/s²];
N = sliding window size [samples];
IIoT = Industrial Internet of Things;
FDM = Fused Deposition Modeling;
MEMS = Micro-Electro-Mechanical Systems;
FFT = Fast Fourier Transform;
REST API = Representational State Transfer Application Programming Interface.

INTELLIGENT SYSTEM FOR INSPECTION AND ELIMINATION OF DEFECTIVE PARTS BASED ON COMPUTER VISION AND IIoT

CRĂCIUN Silviu-Mihai, MOLNOȘ-VELCEA David-Alexandru, PETRE Ioan-Iulian,
ULEANU Eduard-Alexandru and **CAZACU Cristiana-Carmen (Scientific Coordinator)**

Faculty of Industrial Engineering and Robotics, e-mail: silviu.craciun1208@stud.fiir.upb.ro

ABSTRACT: This paper presents the design, implementation and experimental validation of an intelligent quality-control system for conveyor lines. The system integrates a custom 4-DOF robotic arm with a YOLOv8n-OBb computer-vision module deployed on a Raspberry Pi 5. Defective parts are detected on the conveyor belt and removed automatically, while MQTT-based IIoT communication enables remote monitoring and event logging. Experimental results show a mean mAP@0.5 of 0.965 and an average pick success rate of 94.4% for belt speeds between 0.10 and 0.30 m/s.

KEYWORDS: YOLOv8-OBb, robotic arm, computer vision, IIoT, quality control.

1. Introduction

Automated quality inspection on industrial conveyor lines is a critical requirement for modern manufacturing. Traditional rule-based vision sensors and manual inspection are increasingly inadequate at higher production speeds and for complex part geometries. This work proposes an integrated system that combines YOLOv8n-OBb Computer Vision, a fully designed 4-DOF robotic arm (kinematics + electronics), and an IIoT supervision layer. The system automatically detects and removes defective parts — cubes and gears — from a moving belt, with all inference running on a Raspberry Pi 5. Objectives: (1) kinematic and electrical design of the arm, (2) YOLOv8n-OBb training for oriented defect detection, (3) system integration and experimental validation.

2. State of the Art

Deep learning has superseded classical morphological filters for industrial defect detection, offering superior generalization under variable illumination and complex part geometries [1]. CNNs learn hierarchical, data-driven feature representations that outperform hand-crafted descriptors, as confirmed by large-scale visual inspection surveys [2, 3]. Hussain (2023) demonstrates that the YOLO family is particularly well-suited to industrial defect detection pipelines, tracing its evolution from single-class real-time detection toward multi-task, embedded-friendly architectures [4]. The YOLO family, including YOLOv8 [5], provides single-pass real-time object detection by framing localization as end-to-end regression [6], eliminating the latency of two-stage proposal-based detectors. Terven et al. (2023) show that YOLOv8's anchor-free C2f backbone and decoupled head achieve state-of-the-art accuracy-speed trade-offs across embedded hardware targets [2]. The Nano variant targets platforms such as the Raspberry Pi 5 with sub-100 ms inference, and its native OBb extension [7] augments detections with a rotation angle θ for direct gripper pre-alignment, eliminating re-grasping loops without additional inference cost. Kinematic workspace co-design with conveyor geometry is essential for reliable pick-and-place [8, 9]: the reachable workspace must fully contain the pick zone, and geometric IK decoupling ensures singularity-free computation within the fixed belt-stop window. MQTT-based IIoT enables real-time production monitoring by aggregating per-part detection events into operator dashboards via a lightweight publish-subscribe protocol [10, 11]. Li et al. (2022) demonstrate an edge-to-cloud IIoT architecture in

INTELLIGENT SYSTEM FOR INSPECTION AND ELIMINATION OF DEFECTIVE PARTS BASED ON COMPUTER VISION AND IIoT

which smart sensors publish via MQTT for low-latency anomaly detection — a pattern directly applicable to conveyor-line inspection [3]. Homographic camera–robot calibration [12, 13] maps pixel centroids to robot-frame coordinates via a planar projective transform estimated from a correspondence board, achieving reprojection errors below 2 mm.

3. System Architecture

The system has three interconnected layers: perception (YOLOv8n-OBB on Raspberry Pi 5), action (4-DOF arm), and supervision (MQTT IIoT). The complete hardware pipeline is summarised in Table 1.

Table 1. Hardware pipeline: system functions and corresponding components.

Function	Component	Key Specifications
Image acquisition	USB Camera	12 MP, USB 2.0, 60 fps, 1920×1080 px
YOLO inference + IIoT	Raspberry Pi 5	Quad-core Cortex-A76 @ 2.4 GHz, 8 GB LPDDR4X, WiFi 802.11ac
Part trigger	IR proximity sensor	Digital output, GPIO
Arm + belt control	Romeo (ATmega328P)	PWM ×5, DC driver, UART @ 115 200 baud
Arm actuation	Servo 40×20 (×5)	Fmax = 35 kg·cm, PWM 500–2500 μ s, 5 V
Power	LiPo 11.1 V + LM2596	5000 mAh → 5 V/3 A regulated

When the IR sensor detects a part, the Raspberry Pi 5 stops the belt, captures a frame, runs YOLOv8n-OBB inference, and sends centroid coordinates + angle θ to the Romeo controller via UART. The arm eliminates the defective part; the event is published to the MQTT broker for dashboard logging.

4. Robotic Arm Design

4.1 Kinematic Model and MATLAB Validation

The RRRR (4-DOF) arm was parameterized with Denavit-Hartenberg (DH) notation (Table 2). Forward kinematics is the sequential product of homogeneous transforms:

Table 2. Denavit-Hartenberg parameters of the robotic arm.

Joint i	a_{i-1} [mm]	α_{i-1} [°]	d_i [mm]
1	0	0	0
2	127	90	0
3	163	0	0
4	80	0	0

(1)

INTELLIGENT SYSTEM FOR INSPECTION AND ELIMINATION OF DEFECTIVE PARTS BASED ON COMPUTER VISION AND IIoT



In compact form:



(2)

The model was validated in MATLAB R2024a (rigidBodyTree, Robotics System Toolbox [10]) confirming forward kinematics correctness, workspace coverage (cylindrical workspace with approximately 280 mm maximum radial reach and 160 mm vertical range), and IK convergence. Key metrics are given in Table 3.

Table 3. Kinematic and operational performance parameters.

Parameter	Value
Max. reach / workspace height	280 mm / 160 mm
IK computation time / positioning error	18 ms / 2.1 mm
Picking cycle time	1.2 s

The MATLAB validation confirmed that the forward kinematics model correctly reproduces the expected end-effector positions obtained from the CAD geometry. By sweeping the joint-angle limits of the first three joints, the reachable workspace was evaluated and compared with the conveyor belt dimensions. The resulting workspace provides a maximum reach of approximately 280 mm and a vertical operating range of 160 mm, which is sufficient to cover the complete pick zone of the conveyor. The inverse kinematics solution converged for all tested target positions, with an average positioning error of 2.1 mm and an average computation time of 18 ms, confirming that the robotic arm can operate within the real-time constraints of the sorting task.

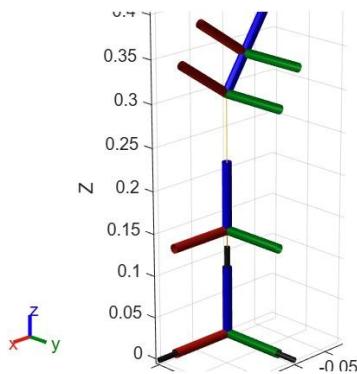


Fig. 1. MATLAB rigidBodyTree model.

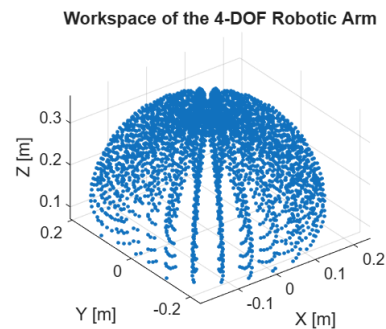


Fig. 2. Workspace of the 4-DOF arm.

INTELLIGENT SYSTEM FOR INSPECTION AND ELIMINATION OF DEFECTIVE PARTS BASED ON COMPUTER VISION AND IIoT

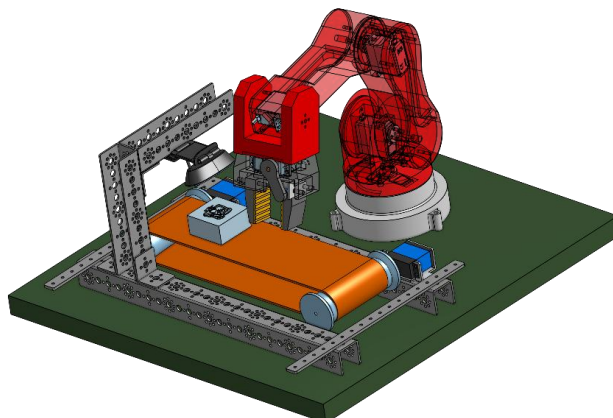


Fig. 3. 3D CAD model and computed workspace.

4.2 Electrical Schematic and Actuation System

The Raspberry Pi 5 coordinates peripherals via UART. The Romeo board (ATmega328P) drives five servomotors and the conveyor DC motor. The robotic arm has four positioning DOF, while a fifth servomotor actuates the gripper. The positioning servomotors control the base, shoulder, elbow and wrist joints, whereas the gripper servomotor performs the opening and closing motion. The Raspberry Pi 5 and the servomotor rail are powered through separate regulated supplies, with common ground. The Raspberry Pi 5 requires a stable 5 V rail, while the servomotors require a dedicated high-current 5–6 V supply to avoid voltage drops during gripping and arm motion. Electrical schematics: Fig. 4 (Fritzing schematic) and Fig. 5 (wiring diagram). Components are listed in Table 4.

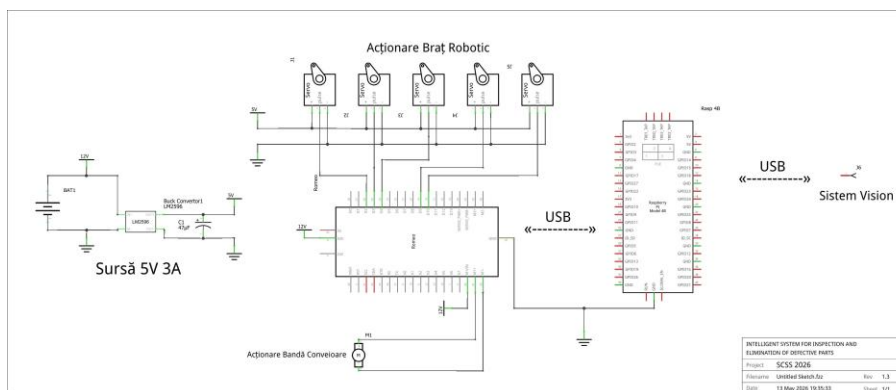


Fig. 4 Electrical schematic (Fritzing)

INTELLIGENT SYSTEM FOR INSPECTION AND ELIMINATION OF DEFECTIVE PARTS BASED ON COMPUTER VISION AND IIoT

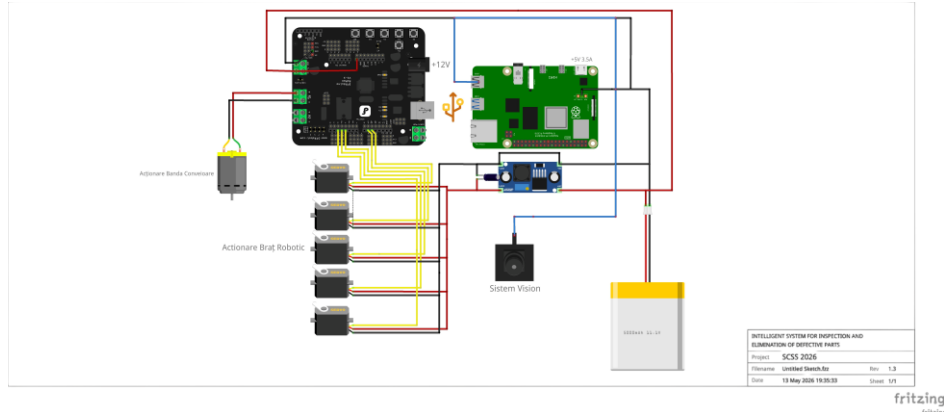


Fig. 5. Wiring connection diagram (Fritzing)

Table 4. Main electrical components of the system

Component	Specifications	Usage
Raspberry Pi 5	Quad-core A76 @ 2.4 GHz, 8 GB LPDDR4X	CPU / YOLO / IIoT
Romeo (DF Robots)	ATmega328P, PWM, DC driver, UART	Arm + belt controller
Servo 40×20 (×5)	Fmax = 35 kg·cm, 40×20×20 mm	Joint actuation
DC motor + LM2596	540 rpm, 12 VDC / 5 V–3 A	Belt + 5 V supply
LiPo 11.1 V / 5 000 mAh	3C 1P	Main power source

4.3 Structural Design and Additive Manufacturing

The arm uses a hollow-core PETG structure to minimize link inertia. Internal conduits isolate wiring from rotational axes; removable panels allow in-situ maintenance. The gripper uses a rack-and-pinion mechanism converting wrist servo rotation into synchronized linear jaw travel, with serrated contact surfaces for secure grip. See Fig.6a and Fig.6b.



Fig. 6. Arm Segments a) Proximal arm segment; b) Distal arm segment

INTELLIGENT SYSTEM FOR INSPECTION AND ELIMINATION OF DEFECTIVE PARTS BASED ON COMPUTER VISION AND IIoT

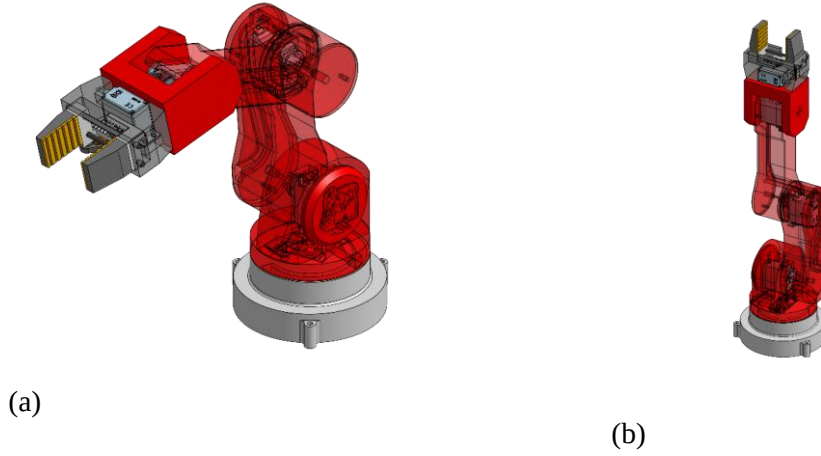


Fig. 7. Complete assembly a) Isometric view; b)Front view

5. Implementation of YOLOv8n-OBb within the Robotic Sorting System

5.1 YOLOv8n-OBb: Technology Overview

YOLO performs object detection as a single regression on the full image [2], simultaneously predicting class probabilities and bounding boxes. The v8 nano variant [6, 7] targets embedded platforms such as Raspberry Pi 5 with sub-100 ms inference. The OBb extension adds a rotation angle θ to the standard bounding box prediction:

$$\text{BBox} = \{x, y, x2, y2, \theta\} \quad (3)$$

This enables automatic gripper alignment with each part regardless of orientation, reducing picking errors. Deep Learning surpasses rule-based sensors in flexibility and generalization [1, 14, 15]. Training convergence and accuracy evolution are illustrated in Fig.8a and Fig.8b.

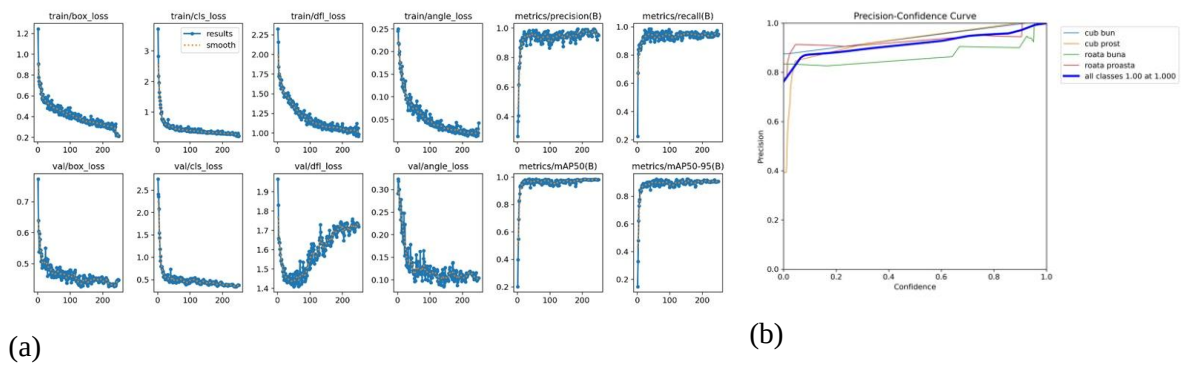


Fig. 8. Training convergence and accuracy evolution a) Training results (loss curves, mAP evolution) b) Accuracy and confidence evolution

5.2 System Flow and Dataset

Three-stage pipeline: (1) IR sensor stops belt → camera captures frame; (2) YOLOv8n-OBb classifies the part (Table 5); (3) arm positions, orients (θ), grips, sorts, and restarts belt. Dataset: 2 400 images of four classes (conforming/non-conforming cubes and gears), annotated with Roboflow. Several training runs were performed between 100 and 200 epochs; the best

INTELLIGENT SYSTEM FOR INSPECTION AND ELIMINATION OF DEFECTIVE PARTS BASED ON COMPUTER VISION AND IIoT

model, reported in this paper, was obtained after 150 epochs, batch 16, Adam, lr = 0.01 (cosine decay), 640×640 px.

Table 5. Part classes and conformity classification.

Part	Classification
Cube with stamp	Conforming part
Cube without stamp	Non-conforming part
Gear (18 teeth)	Conforming part
Gear with missing teeth	Non-conforming part

5.3 YOLO Performance Results

Per-class metrics on the 20% test split (Table 6). The confusion matrix (Table 7) shows that most errors are concentrated around non-conforming or visually similar classes, especially between unstamped cubes, gears with missing teeth and conforming gears. due to shared irregular surface features. Mean mAP@0.5 = 0.965.

Table 6. YOLOv8n-OBb per-class performance on the test set.

Class	Precision	Recall	F1-Score	mAP@0.5
Cube with stamp	0.980	0.980	0.980	0.981
Cube without stamp	0.959	0.940	0.949	0.955
Gear (18 teeth)	0.980	0.960	0.970	0.971
Gear with missing teeth	0.923	0.960	0.941	0.951
MEAN	0.961	0.960	0.960	0.965

Table 7. Confusion matrix — YOLOv8n-OBb (50 samples per class).

<i>True \ Pred.</i>	Cube+stamp	Cube–stamp	Gear 18T	Gear miss.
Cube+stamp	49	1	0	0
Cube–stamp	1	47	0	2
Gear 18T	0	0	48	2
Gear miss.	0	1	1	48
MEAN	0.961	0.960	0.960	0.965

5.4 Coordinate Conversion, Calibration and Latency

The system supports two operating modes. In Mode A — Stop-and-pick, the conveyor stops after IR detection, and the YOLO centroid is mapped directly to robot coordinates using 25-point homography calibration. This high-accuracy mode was used for validation. In Mode B — Predictive picking, the belt keeps moving, and the detected position is compensated using conveyor speed and the delay between image capture and gripper interception. In Mode E predicted pick position is computed as:

(4)

where $\Delta t_{exec} = 0.82s$ (Table 9) and $v_{conveyor}$ is belt speed. Calibration validated on 7 independent test points (Table 8); mean reprojection error 1.29mm acceptable for a 15mm gripper. The 0.82s latency covers detection-to-grasp, while the 1.2s picking cycle includes the full pick-and-place sequence, deposit, and return-to-ready motion.

Pt.	Xr [mm]	Yr [mm]	Xe [mm]	Ye [mm]	Err [mm]	
P1	50.0	30.0	51.2	30.5	1.30	
P2	100.0	50.0	101.0	49.2	1.28	
P3	150.0	80.0	151.1	80.8	1.36	
P4	80.0	120.0	81.2	119.4	1.34	
P5	200.0	40.0	200.9	39.3	1.14	
P6	60.0	160.0	61.0	160.9	1.35	
P7	180.0	100.0	181.1	99.4	1.25	
					Mean	1.29 mm

Processing Stage	Time [ms]
Belt stop → stable frame + image capture	83
YOLOv8n-OBB inference + OBB extraction	97
Homography + UART + IK computation	43
Arm trajectory execution	490
Gripper actuation	107
TOTAL	820 $\approx$ 0.82 s

Five test series (50 parts each; 25 conforming + 25 defective) at belt speeds 0.10–0.30 m/s (Table 10). The system achieves $\geq 92\%$ success rate up to 0.25 m/s. At 0.30 m/s the rate drops to 88% due to belt vibration and UART jitter. Overall average: 94.4%. The experimental validation was performed in Mode A, using the stop-and-pick strategy, in order to minimize motion blur and isolate the accuracy of the vision and robotic manipulation pipeline.

16

INTELLIGENT SYSTEM FOR INSPECTION AND ELIMINATION OF DEFECTIVE PARTS BASED ON COMPUTER VISION AND IIoT

Series	Belt speed (m/s)	Total parts	Correct conform	Correct defective	Total correct	Errors	Success rate (%)
T1	0.10	50	25 / 25	24 / 25	49	1	98.0%
T2	0.15	50	24 / 25	24 / 25	48	2	96.0%
T3	0.20	50	25 / 25	23 / 25	48	2	96.0%
T4	0.25	50	24 / 25	23 / 25	47	3	94.0%
T5	0.30	50	22 / 25	22 / 25	44	6	88.0% ↓
Total / Average		250	120 / 125	116 / 125	236	14	94.4%

7. Conclusions

This paper presented a complete intelligent inspection and sorting system. Original contributions: (1) a fully designed 4-DOF robotic arm (kinematic model, analytical IK, electrical schematic); (2) YOLOv8n-OBb trained on a custom 4-class dataset achieving mAP@0.5 = 0.965 with inference < 100 ms on Raspberry Pi 5; (3) experimental validation confirming 94.4% average success rate; (4) IIoT integration via MQTT. Current limitations: controlled lighting, four defect classes, and a lab-scale arm. Future work: adaptive lighting, higher belt speeds, expanded defect classes, and closed-loop speed control.

8. References

- [1] Czimmermann, T. et al. (2020). Visual-Based Defect Detection and Classification Approaches for Industrial Applications. *Sensors*, 20(5), 1459.
- [2] Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision. *Machine Learning and Knowledge Extraction*, 5(4), 1680–1716.
- [3] Li, Z., Fei, F., & Zhang, G. (2022). Edge-to-Cloud IIoT for Condition Monitoring in Manufacturing Systems with Ubiquitous Smart Sensors. *Sensors*, 22(15), 5901.
- [4] Hussain, M. (2023). YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines*, 11(7), 677.
- [5] Jocher, G. et al. (2023). *Ultralytics YOLOv8*. <https://github.com/ultralytics/ultralytics>.
- [6] Wang, C.Y. et al. (2023). YOLOv7. *Proc. IEEE/CVF CVPR*, pp. 7464–7475.
- [7] Jocher, G. and Qiu, J. (2024). *YOLOv8-OBb*. <https://docs.ultralytics.com/tasks/obb/>.
- [8] Craig, J.J. (2005). *Introduction to Robotics: Mechanics and Control*. Pearson Education.
- [9] Lin, S. et al. (2022). Adaptive Robotic Pick-and-Place. *IEEE TASE*, 19(4), 3247–3258.
- [10] Sisinni, E. et al. (2018). Industrial Internet of Things. *IEEE Trans. Industrial Informatics*, 14(11), 4724–4734.
- [11] He, X. et al. (2022). A Survey of Defect Detection Applications Based on Generative Adversarial Networks. *IEEE Access*, 10, 113493–113512.

**INTELLIGENT SYSTEM FOR INSPECTION AND ELIMINATION OF DEFECTIVE PARTS BASED
ON COMPUTER VISION AND IIoT**

- [12] Zhang, Z. (2000). A Flexible New Technique for Camera Calibration. *IEEE TPAMI*, 22(11), 1330–1334.
- [13] Feng, C. et al. (2022). Deep Learning Surface Defect Detection. *Neurocomputing*, 493, 626–649.
- [14] Corke, P. (2023). *Robotics, Vision and Control*, 3rd ed. Springer.
- [15] Zhao, L. et al. (2023). Homography-Based Camera–Robot Calibration. *Rob. Comp.-Int. Mfg.*, 81, 102505.

IOT PROTOTYPE FOR PERSONNEL SAFETY AND MONITORING IN FACTORIES USING RADIO WAVES

NAȘU Teodor-Cristian¹, DRĂGUȚU Matei¹, HANGA Mihail² and CAZACU Carmen-Cristiana³ (Scientific Coordinator)

1Faculty of Industrial Engineering and Robotics, Specialization:I.A.I.I, Year of Study:III, e-mail: teodor.nasu@stud.fiir.upb.ro

2 Faculty of Industrial Engineering and Robotics, Specialization: M.I.I.V, Year of Study: I,

3 Faculty of Industrial Engineering and Robotics, Department of Robotics and Production Systems

ABSTRACT: This paper proposes a low-cost, privacy-preserving monitoring system for industrial environments, aimed at improving employee safety in high-risk zones such as industrial robot workspaces. The system uses three ESP32 microcontrollers in a triangular formation to capture Channel State Information (CSI) mesh data from standard Wi-Fi signals. During training, a Python script uses YOLOv8n to extract 2D human coordinates from camera feeds and correlate them with synchronized CSI data. A CNN is then trained to estimate these 2D positions using only multi-link CSI data. Unlike methods focused on dense pose estimation, the proposed system targets real-time 2D tracking of multiple individuals, providing an illumination-invariant and scalable safety mechanism for smart factories.

KEYWORDS: *Channel State Information, ESP32, Industrial Safety, IoT, Factory*

1. Introduction

The integration of autonomous industrial robots into production flows strictly requires advanced safety measures to prevent workplace accidents. Traditional optical sensors, such as rgb cameras and lidar, face significant limitations in industrial settings: poor lighting conditions, blind spots caused by heavy machinery, occlusions, and severe employee privacy concerns [1], [2], [3]. The core problem addressed in this paper is the lack of a robust, low-latency, and privacy-compliant monitoring system capable of perceiving human presence and coordinates in dynamic, visually obstructed factory zones.

To solve this problem, we propose an alternative, fine-grained human perception system based exclusively on ubiquitous wi-fi signals [1] [4]. The specific objectives of this research are: (1) to design a low-cost hardware prototype using esp32 modules; (2) to extract and clean csi data reflecting electromagnetic perturbations caused by human bodies; (3) to generate accurate 2d ground-truth coordinates using yolov8n computer vision; and (4) to design and train a convolutional neural network (cnn) capable of inferring personnel coordinates directly from the csi tensors, enabling proximity-based safety warnings.

2. Theoretical Foundation and Related Work

While standard Received Signal Strength Indicator (RSSI) only provides a coarse, aggregate measure of signal power, Channel State Information (CSI) captures the fine-grained physical layer properties of the wireless channel.

A CSI matrix encapsulates both the amplitude attenuation and phase shift of each subcarrier caused by multipath propagation phenomena (reflection, refraction, and scattering) induced by the environment and human bodies. By analyzing these microscopic perturbations, it is theoretically possible to reverse-engineer the spatial layout of the environment.

Recent research demonstrated the viability of Wi-Fi for body perception [5]; the "Person-in-WiFi" system extracted segmentation masks using standard antennas [6], and dense pose correspondence was achieved by extracting spatial features from CSI [1], [7]. Systematic reviews highlight the ESP32 microcontroller as a robust edge-computing standard for AI-enhanced monitoring due to its subcarrier extraction capabilities [5], [8]. In the industrial safety domain [9], YOLO-based detection is proven for human-machine proximity [2]. Our approach uniquely bridges these fields, shifting from computationally heavy 3D reconstructions to rapid 2D coordinate regression for immediate safety applicability.

3. System Architecture and Methodology

Hardware Setup: The hardware architecture is based on three ESP32 microcontrollers utilizing

IOT PROTOTYPE FOR PERSONNEL SAFETY AND MONITORING IN FACTORIES USING RADIO WAVES

practical ESP-CSI extraction pipelines [10], as presented in Table 1. Crucially, the nodes are positioned in a triangular formation around the industrial robot's workspace to create a robust signal mesh, Table 2. Each ESP32 acts dynamically as a transmitter and receiver. When a worker enters the zone, they alter the multipath propagation of the Wi-Fi signals. The triangular topology ensures that even if one line of sight is severely blocked, the non-line-of-sight (NLOS) reflections captured by the other nodes provide rich, diverse spatial signatures to the system.

Table 1. Hardware components of the proposed prototype

Component	Quantity	Role
ESP32	3	CSI transmitter/receiver nodes
Raspberry Pi 5	1	data aggregation and CNN inference
Camera	1	YOLO-based ground truth generation during training
Alarm / relay	1	safety response output

Table 2. Comparison of Wi-Fi-based perception approaches

Method	Sensor Hardware	Generated Output	Primary Goal
Person-in-WiFi [1]	Intel 5300 NICS	SM, JHMs, PAFs	Body Segmentation & Pose
DensePose WiFi [2]	COTS Network Cards	UV Maps, Dense Pose	3D Surface Reconstruction
Proposed System	3 x ESP32 (Triangular Array)	2D Point Coordinates	Industrial Robot Safety

Data Processing: The processing architecture features a Raspberry Pi serving as the central point. A primary ESP32 node compiles the collected CSI data from all nodes and transmits it to the Raspberry Pi. The Raspberry Pi then compiles these data streams into structured tensors and feeds them into the neural network model, as Fig.1 shows.

```
def _create_heatmap(self, coords):
    heatmap = np.zeros(self.grid_size, dtype=np.float32)
    max_x, max_y = self.room_dimensions
    grid_x, grid_y = self.grid_size

    for x, y in coords:
        # Scalarea coordonatelor fizice la indexul grilei
        idx_x = int((x / max_x) * grid_x)
        idx_y = int((y / max_y) * grid_y)

        # Limitarea la coordonatele camerei
        idx_x = np.clip(idx_x, 0, grid_x - 1)
        idx_y = np.clip(idx_y, 0, grid_y - 1)

        # Plasarea persoanei
        heatmap[idx_x, idx_y] = 1.0

    return torch.tensor(heatmap, dtype=torch.float32).unsqueeze(0) # Adăugăm un canal [1, W, H]
```

Fig. 1 Dataset Generation

This computing setup allows for efficient data handling and real-time inference, as observed in Table 3, as the Raspberry Pi manages both communication with the sensor array and the execution of the tracking algorithm.

Table 3. System Processing Latency

Processing Stage	Hardware Component	Time (ms)
CSI Extraction & Transmission	ESP32 Nodes	10 - 15 ms
Data Aggregation & Formatting	Raspberry Pi 5 (8GB)	5 - 10 ms

IOT PROTOTYPE FOR PERSONNEL SAFETY AND MONITORING IN FACTORIES USING RADIO WAVES

CNN Inference (2D Regression)	Raspberry Pi 5 (8GB)	15 - 30 ms
Total System Latency	End-to-End	30 - 55 ms

Ground Truth Generation via YOLOv8: To train the CNN to map radio-wave perturbations to physical locations, an accurate and synchronized ground-truth dataset is required. In this work, YOLOv8n is used only during the training phase to generate visual ground-truth coordinates from a top-down camera. During deployment, the camera is not required, and the system estimates personnel coordinates only from CSI data. A custom Python script detects each person, extracts the bounding-box centroid and converts the pixel coordinates into a 2D floor-plane coordinate system synchronized with the CSI packets.

The fundamental purpose of this optical setup is to analyze the continuous video feed from a top-down calibration camera, recognize human individuals within the industrial workspace, and extract their bounding boxes. A custom Python script isolates these human entities and calculates the mass center of each detected bounding box. To translate these pixel-based centers into real-world factory floor measurements, the script translates these into a 2D coordinate plane mapping the factory floor.

CNN Training: Convolutional Neural Networks (CNNs) are architectures specialized in feature extraction and pattern recognition from structured data. While typically used for images, here the CNN is adapted for regression mapping. Implemented in PyTorch [11], Fig. 2,

```
class CSILocalizationModel(nn.Module):
    def __init__(self, num_antennas, grid_size=(32, 32)):
        super(CSILocalizationModel, self).__init__()
        self.grid_size = grid_size

        # Extractor
        self.encoder = nn.Sequential(
            # Input: [Batch, Antennas, Subpurtatoare, Time]
            nn.Conv2d(num_antennas, 32, kernel_size=3, padding=1),
            nn.BatchNorm2d(32),
            nn.ReLU(),
            nn.MaxPool2d(2),

            nn.Conv2d(32, 64, kernel_size=3, padding=1),
            nn.BatchNorm2d(64),
            nn.ReLU(),
            nn.MaxPool2d(2),

            nn.Conv2d(64, 128, kernel_size=3, padding=1),
            nn.BatchNorm2d(128),
            nn.ReLU(),
        )

        # Flattening
        self.flatten = nn.Flatten()
        self.fc = nn.Linear(128 * 13 * 25, 128 * 8 * 8)

        # Heatmap Decoder
        self.decoder = nn.Sequential(
            nn.ConvTranspose2d(128, 64, kernel_size=4, stride=2, padding=1),
            nn.ReLU(),
            nn.ConvTranspose2d(64, 32, kernel_size=4, stride=2, padding=1),
            nn.ReLU(),
            nn.Conv2d(32, 1, kernel_size=3, padding=1), # Output 1 heatmap
            nn.Sigmoid() # Clamp from 0 to 1
        )

    def forward(self, x):
        x = self.encoder(x)
        x = self.flatten(x)
        x = self.fc(x)
        x = x.view(-1, 128, 8, 8)
        x = self.decoder(x)
        return x
```

Fig. 2 Model Architecture

the model learns the non-linear mapping between Wi-Fi signal amplitude/phase perturbations and the physical 2D coordinates [12]. This specific methodology is strongly supported by recent validation of grid-based regression models that utilize ESP32 multi-link in complex indoor environments [13], [14]. A custom Python script captures video feed and utilizes the YOLOv8n (nano) model [15] for further processing.

The CNN is implemented in PyTorch and trained to learn the non-linear mapping between sanitized CSI amplitude/phase tensors and the corresponding 2D ground-truth coordinates. The input consists of multidimensional tensors extracted from the three ESP32 transmission links, while the output

IOT PROTOTYPE FOR PERSONNEL SAFETY AND MONITORING IN FACTORIES USING RADIO WAVES

represents the estimated floor-plane coordinates of the detected personnel.

The script isolates human entities, calculates the mass center of their bounding boxes, and translates these into a 2D coordinate plane mapping the factory floor. These coordinates are synchronized via timestamps with the CSI packets collected from the ESP32 array. The input consists of multidimensional tensors comprising the sanitized amplitude and phase from all subcarriers across the three-transmission links. Fig. 3 illustrates the training script workflow.

```
def train_model(model, dataloader, epochs=50):
    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
    model.to(device)

    criterion = nn.BCELoss() # Analizează fiecare "celulă" din cameră
    optimizer = optim.Adam(model.parameters(), lr=0.001)

    for epoch in range(epochs):
        model.train()
        running_loss = 0.0

        for csi_inputs, target_heatmaps in dataloader:
            csi_inputs, target_heatmaps = csi_inputs.to(device),
            target_heatmaps.to(device)

            optimizer.zero_grad()

            # Forward pass
            outputs = model(csi_inputs)

            # Calculul pierderii
            loss = criterion(outputs, target_heatmaps)

            # Backward pass & Optimizare
            loss.backward()
            optimizer.step()

            running_loss += loss.item()

        print(f"Epoca {epoch+1}/{epochs}, Loss: {running_loss/len(dataloader):.4f}")
```

Fig. 3 Training Script

Safety Protocol and Zone Modeling: To ensure practical and safe integration into industrial environments, the prototype implements a predefined proximity zone (Table 4) around simulated points, which are considered the working industrial robotic arms. The proximity zones are defined as:

Green Zone (Safe): Distance > 2.0 meters. Normal operational activity, the robot operates at maximum speed.

Yellow Zone (Warning): Distance between 1.0 and 2.0 meters. The robot reduces its operating speed significantly and initiates visual/auditory pre-alert signals.

Red Zone (Unsafe Operation): Distance < 1.0 meters. Immediate and complete robot shutdown.

Table 4. Response Matrix

Zone	Distance	Robot Status	Alert Type
Green (Safe)	> 2 m	Normal Operation (100% Speed)	None
Yellow (Warning)	1 m – 2 m	Reduced Speed (Throttling)	Visual & Auditory
Red (Unsafe)	< 1 m	Immediate Shutdown	Critical Alarm
Emergency Stop	0.3 m (30cm)	Hardware E-Stop	Emergency Stop

Error Management and the 30 cm Limit: The implementation of a fail-safe safety protocol must inherently account for the system's spatial estimation errors. The absolute threshold of the safety logic is the 30 cm limit: if the system detects a person at a distance of 30 cm or less from the robotic arm, it bypasses standard software throttling and triggers an immediate hardware Emergency Stop. This strict limit acts as the final line of defense, accounting for potential robotic inertia and sudden, unpredictable human movements within the critical Red Zone.

However, a valid safety concern arises when monitoring three concurrent persons, where the shadowing effect and signal scattering cause the Mean Absolute Error (MAE) to increase to 60 cm. A static

IOT PROTOTYPE FOR PERSONNEL SAFETY AND MONITORING IN FACTORIES USING RADIO WAVES

proximity boundary is insufficient under these conditions; an 60 cm discrepancy could lead to a severe false negative, where a worker is physically inside the Red Zone but the neural network mistakenly perceives them in the Yellow Zone.

While this highly conservative approach inherently increases the rate of false-positive alarms and potentially causing the robot to slow down or halt prematurely when workers are still relatively safe, but it strictly prioritizes occupational safety.

Architecture Workflow: The architecture of the prototype is structured into a low-latency pipeline. A triangular array of ESP32 microcontrollers captures raw Channel State Information (CSI) across 64 subcarriers. To establish a learning baseline during the training phase, a parallel computer vision setup utilizing YOLOv8n generates 2D coordinates, which are synchronized with the corresponding CSI data packets. The processing is executed at the edge by a Raspberry Pi 5. This unit feeds the multi-link tensors into a PyTorch-based Convolutional Neural Network (CNN) for real-time spatial regression, successfully maintaining a strict end-to-end system latency of 30–55 ms. Finally, the predicted coordinates drive the actuation layer, translating human presence into dynamic Green, Yellow, or Red proximity zones. The system dynamically expands these warning boundaries when multiple targets are detected to safely compensate for shadowing errors, all while strictly enforcing an absolute 30 cm hardware Emergency Stop threshold. presented in Fig.



Fig. 4 System Workflow

4. System Limitations

Physical and Signal Perspective: The primary limitation stems from the physics of Wi-Fi signals (2.4 GHz). With a wavelength of approximately 12.5 cm, the signal may bypass or fail to resolve small limbs due to diffraction. Furthermore, multipath propagation in metal-heavy industrial environments creates complex interference patterns (multipath fading), which can occasionally mask human movement signatures if the environment changes significantly without retraining.

Hardware and Resolution Perspective: Unlike high-bandwidth sensors such as Ultra-Wideband (UWB) or mmWave radars, which utilize gigahertz-wide bands to achieve millimeter-level precision, the ESP32 operates primarily on the 2.4 GHz band with a standard channel bandwidth limited to 20 MHz or 40 MHz. This narrow bandwidth fundamentally restricts the delay resolution and the physical ability of the receiver to separate multipath signals arriving at slightly different times. Consequently, while the ESP32 successfully reports 64 subcarriers per spatial stream [8], the overall informational density is relatively low. This limited spectral resolution means the CNN receives a "blurred" radio frequency image of the environment. While sufficient for macro-level 2D mass-center regression, it prevents the system from distinguishing fine-grained human details (such as limb articulation or body orientation) compared to specialized radar or image-based approaches [16], [4].

Subject Density and Shadowing: As observed in our experimental results (Table 5), coordinate error margins increase significantly from 10 cm for a single person to 60 cm for three concurrent persons. This degradation is directly caused by the electromagnetic 'shadowing effect'. The human body, composed predominantly of water, acts as a severe attenuator and reflector of 2.4 GHz signals. When a subject stands in the direct line of sight or a primary reflection path between two ESP32 nodes, they absorb a massive portion of the signal energy. Consequently, a second or third person standing behind them (occluded in the RF spectrum) generates exponentially weaker secondary perturbations. This creates a highly non-linear scaling of noise and overlapping features within the CSI matrix. For the PyTorch CNN, disentangling these attenuated signal signatures from the aggregate tensor becomes computationally complex, as the distinct spatial features of the occluded individuals are physically

IOT PROTOTYPE FOR PERSONNEL SAFETY AND MONITORING IN FACTORIES USING RADIO WAVES

masked by the subjects closer to the transceivers [4].

Table 5. Simulated experimental Setup and Detection Error Metrics

Number of Persons	CSI Samples	Positions	Train/Test Split	MAE (cm)	Max Error (cm)
1	260,000	24	80% / 20%	10.0	15.4
2	260,000	dynamic		30.0	42.8
3	260,000			60.0	115.6

This error is partially solved by using a function dedicated for phase sanitization, Fig. 4.

```
def sanitize_csi_phase(complex_csi_array):
    # Extract raw phase
    raw_phase = np.angle(complex_csi_array)

    # Unwrap
    unwrapped_phase = np.unwrap(raw_phase)

    M = len(unwrapped_phase)
    subcarriers = np.arange(M)

    k = (unwrapped_phase[-1] - unwrapped_phase[0]) / (M - 1)
    b = np.mean(unwrapped_phase)

    sanitized_phase = unwrapped_phase - (k * subcarriers + b)

    return sanitized_phase
```

Fig. 5 Phase Cleaning Function

Because the maximum error in the three-person scenario reaches 115.6 cm, the 30 cm Emergency Stop threshold should not be interpreted as a standalone certified safety distance. Instead, in multi-person conditions, the system must expand the warning and stop boundaries conservatively, using the measured localization error as an additional safety buffer.

5. Current System Status

At the current stage of development, the IoT prototype is fully functional and has been evaluated under simulated industrial conditions. The end-to-end processing pipeline, managed by the Raspberry Pi 5, successfully maintains a total system inference latency between 30 and 55 ms. This confirms the prototype's viability for real-time safety critical loops. The system exhibits high reliability in localizing 1 or 2 people within the workspace

Experimental metrics demonstrate a Mean Absolute Error (MAE) of 10.0 cm for a single target and 30.0 cm for two targets. As the number of concurrent persons increases to three, the localization error increases substantially, reaching an MAE of 60.0 cm. This confirms that the current prototype is more reliable for one or two persons, while multi-person scenarios require conservative safety-zone expansion. This degradation is directly attributed to the exponentially complex signal scattering and multi-body shadowing effects, which overlap within the hardware limit of 64 subcarriers extractable from the ESP32 nodes.

Despite this widened error margin for multiple concurrent targets, the implementation of the dynamic buffer expansion successfully compensates for spatial discrepancy. Consequently, the system maintains sufficient accuracy to trigger localized safety warnings, reliably transitioning the robot between green, yellow, and red proximity states without compromising the strict 30 cm hardware Emergency Stop threshold.

The PyTorch-based Convolutional Neural Network (CNN) has proven capable of consistently generalizing these 2D coordinates, allowing the safety protocol to operate entirely on privacy-preserving Radio Frequency (RF) data post-training.

6. Conclusions

This work demonstrated the feasibility of using a system based on accessible hardware (ESP32) arranged in a triangular array to obtain precise 2D coordinates of human personnel using Channel State Information (CSI). By simplifying the output from full body reconstruction to simple 2D coordinate points evaluated through a CNN leveraging dynamic multi-link spatial diversity, the

system provides a low-latency solution ideal for integration into industrial robot safety loops. Future research will focus on expanding the dataset in diverse industrial environments and integrating a fail-safe decision-making system. The direct applicability of these results empowers smart factories to implement additional safety monitoring layer around robots without relying on expensive optical hardware, thereby advancing the standards of modern occupational safety.

7. References

- [1] J. Geng, D. Huang and F. De la Torre, DensePose From WiFi, 2022.

<https://doi.org/10.48550/arXiv.2301.00250>
- [2] T. E. Bilsel, M. Çolak, N. P. Akman and A. Berkol, "Human-Machine Proximity and Warning System for Industrial Safety," 2025 17th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Targoviste, Romania, 2025, pp. 1-7, doi: 10.1109/ECAI65401.2025.11095460.
- [3] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018, doi: 10.1016/j.mechatronics.2018.02.009.
- [4] M. Cominelli, F. Gringoli and F. Restuccia, "Exposing the CSI: A Systematic Investigation of CSI-based Wi-Fi Sensing Capabilities and Limitations," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2023.
- [5] Al-Rajab M, Qassem K, Seyam S, Al Hamadi Y, Al-Dubaisi AH, Mallat J, Lu J, Ellahham SH, Sawsaa A, Alzraiqat S. 2026. Artificial intelligence-enhanced CSI-based Wi-Fi sensing for non-contact vital sign monitoring: a systematic review. *PeerJ Computer Science* 12:e3375 <https://doi.org/10.7717/peerj-cs.3375>
- [6] F. Wang, S. Zhou, S. Panev, J. Han and D. Huang, "Person-in-WiFi: Fine-grained Person Perception using WiFi," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [7] X. Wang, L. Gao, S. Mao and S. Pandey, "CSI-based Fingerprinting for Indoor Localization: A Deep Learning Approach," *IEEE Transactions on Vehicular Technology*, 763-776, 2017.
- [8] S. M. Hernandez and E. Bulut, "WiFi Sensing on the Edge: Signal Processing Techniques and Challenges for Real-World Systems," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 46–76, 2023, doi: 10.1109/COMST.2022.3209144.
- [9] Ridwan Kolapo, Fatima Mohammed Kawu, Aminu Dalhatu Abdulmalik, Ubong Anietie Edem, Maureen Atisi Young and Emmanuel Chukwudinma Mordi. Edge computing: Revolutionizing data processing for IoT applications. *International Journal of Science and Research Archive*, 2024, 13(02), 023–029.
<https://doi.org/10.30574/ijrsra.2024.13.2.2082>

- [10] Espressif Systems, "Practical ESP-CSI implementations," 2026. [Online]. Available: Official Documentation.
- [11] PyTorch Foundation, "PyTorch Documentation: Training Neural Networks for Signal Processing," 2026. [Online]. Available: PyTorch Official Documentation.
- [12] Z. Wang, W. Yan and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *International Joint Conference on Neural Networks (IJCNN)*, 2017.
- [13] C. Saurav, "A Multitask 1D Convolutional Neural Network for Joint Occupancy, Movement, and Heart Rate Extraction via Wi-Fi Channel State Information," *Preprints.org*, no. DOI:10.20944/preprints202604.0624.v1, 2026.
- [14] C. Saurav, "Device-Free Indoor Localization with ESP32 Wi-Fi CSI Fingerprints: Grid-Based Regression, Feature Modeling, and Multi-Link Experiments," *ResearchGate*, no. DOI:10.20944/preprints202601.2378.v1, 2026.
- [15] Ultralytics, "YOLOv8 Documentation and Usage for Real-Time Object Detection," 2026. [Online]. Available: Ultralytics Documentation.
- [16] International Organization for Standardization, *Robots and robotic devices - Collaborative robots, ISO/TS 15066:2016*, 2016.

9. Abbreviations and Acronyms

- AI: Artificial Intelligence
- CNN: Convolutional Neural Network
- COTS: Commercial Off-The-Shelf
- CSI: Channel State Information
- E-Stop: Emergency Stop
- IoT: Internet of Things
- LiDAR: Light Detection and Ranging
- MAE: Mean Absolute Error
- NLOS: Non-Line-of-Sight
- RF: Radio Frequency
- RGB: Red, Green, Blue
- RSSI: Received Signal Strength Indicator
- UWB: Ultra-Wideband
- Wi-Fi: Wireless Fidelity
- YOLO: You Only Look Once
- YOLOv8n: You Only Look Once version 8 nano

ALCOHOL INTERLOCK SYSTEM FOR FORKLIFTS AND MOBILE INDUSTRIAL EQUIPMENT

VASILE Andrei-Damian, NEGRILĂ Gabriel Valentin and **CAZACU Carmen Cristiana**
(Scientific Coordinator)

Faculty of Industrial Engineering and Robotics (FIIR), Specialization: Robotics, Year of study: II, e-mail: andreidamianv@gmail.com

ABSTRACT: *This paper presents the design and implementation of an integrated system for monitoring and preventing the operation of forklifts and mobile industrial equipment under the influence of alcohol. The system integrates an ESP32 microcontroller, an MQ-3 alcohol vapour sensor, an SSD1306 OLED display (I²C), and a motor-driven mechanical locking shaft controlled via an L298N driver module. If alcohol is detected above the safety threshold, the mechanical lock prevents pedal movement and disables the vehicle ignition relay. The architecture is low-cost, scalable, and retro-fittable on existing forklift fleets.*

KEYWORDS: breathalyzer interlock, ESP32, MQ-3, industrial safety, IoT systems

1. Introduction

Safety is an absolute priority in modern industrial and logistics environments. Forklifts, electric pallet trucks, and other motor-driven industrial vehicles are present in virtually every warehouse, factory, and distribution centre, often operating in congested spaces alongside pedestrians. These machines can weigh several tonnes and, when operated improperly, can cause accidents with severe or even fatal consequences. Operators must be capable of reacting quickly and making correct decisions in real time, in environments where mistakes carry no margin of error.

Alcohol consumption, even in small quantities, significantly affects an operator's reaction times, motor coordination, and situational judgement. According to the literature, a blood alcohol concentration of only 0.2 ‰ can reduce reaction speed by 10–15 %, while at 0.5 ‰ — the legal limit in many EU Member States — the degradation of cognitive and motor performance is markedly more pronounced. In the context of operating heavy industrial equipment, these effects translate directly into an increased risk of accidents with serious material and human consequences [1],[2].

Current prevention solutions rely predominantly on administrative and procedural measures: periodic manual breathalyser testing, shift-supervisor responsibility, and internal zero-tolerance policies. However, these approaches share a fundamental disadvantage: they depend on the human factor for enforcement, are susceptible to omissions, and do not guarantee compliance in every operating cycle. An operator can pass the check at the beginning of a shift and consume alcohol afterwards, or the check may be skipped during periods of operational congestion.

This project proposes an autonomous, hardware-enforced solution: a vehicle ignition interlock system based on a breath test, which prevents the machine from operating if the operator has consumed alcohol, regardless of the organisational context. The system acts at every start attempt, eliminating human error at the architectural level. The main objectives of the paper are: (1) to design the hardware architecture of the system; (2) to implement the control logic on the ESP32 platform; (3) to characterise the MQ-3 sensor and establish a calibrated detection threshold; (4) to validate the system through controlled experiments; and (5) to justify the necessity through analysis of European workplace accident statistics.

2. System Architecture

2.1. Hardware Components

The system design is built around a set of compact electronic modules, all orchestrated by a central microcontroller. Component selection followed three main criteria: availability on the electronic components market, low acquisition cost, and energy consumption compatible with the industrial vehicle's electrical system.

The ESP32 microcontroller forms the core of the system. It features a dual-core Xtensa LX6 processor at 240 MHz, 520 KB SRAM, I2C, SPI, UART, and 12-bit ADC interfaces, as well as integrated Wi-Fi and Bluetooth connectivity [3]. These resources are more than sufficient for the system requirements: reading the MQ-3 sensor via ADC, communicating with the OLED display via I2C, generating PWM signals for the motor driver, and optionally transmitting telemetry data via Wi-Fi to a monitoring server.

The MQ-3 gas sensor is an electrochemical sensor with a tin dioxide (SnO_2) sensitive element, characterised by high selectivity towards ethanol vapour. The sensor exhibits an internal resistance that varies inversely with the alcohol concentration in exhaled air, allowing the ESP32 to read an analogue voltage proportional to the estimated BAC [4]. The warm-up time required to stabilise the sensor is approximately 20–30 seconds from power-on, a period indicated to the user via the OLED display. Key electrical characteristics are summarised in Table 1.

Table 1. MQ-3 sensor key electrical specifications

Parameter	Value / Range	Unit
Supply voltage (VCC)	5.0 ± 0.1	V
Heater voltage (VH)	5.0 ± 0.1	V
Detection range	0.05 – 10	mg/L (ethanol)
Warm-up time	20 – 30	s
Operating temperature	–10 to +50	°C
Load resistance (RL)	200 k (adjustable)	Ω

The SSD1306 OLED display (128 × 64 pixels, I2C interface) serves as the user interface, providing clear visual feedback at all stages of the operating cycle [5]. The messages displayed are: "System initialised – Sensor warming up", "Please exhale into sensor", "Analysis in progress...", "ACCESS GRANTED – Start vehicle", or "ACCESS DENIED – Alcohol detected". Message clarity is essential in industrial environments, where operators are often under physical strain and must interpret displayed information rapidly.

The mechanical locking mechanism consists of a DC motor controlled via the L298N driver module [6], which actuates a shaft. In the rest state (vehicle off or alcohol detected), the shaft is extended under the vehicle's pedals, physically preventing them from being depressed. This mechanical approach has the advantage of not being by-passable through software manipulation, providing a fail-safe level of safety: in the absence of power or in the event of an ESP32 malfunction, the system remains locked.

2.2. Operating Logic (Software Flow)

The firmware runs on the ESP32 and implements a state machine with four main states: INIT, WARMING, TESTING, and DECISION. On system power-on (INIT state), the microcontroller initialises the peripherals (I2C, ADC, PWM), engages the locking mechanism in the extended position, and sends the warm-up start command to the MQ-3 sensor.

In the WARMING state, the system monitors the sensitive element temperature and displays a countdown timer on the OLED. The warm-up duration is configurable through

software parameters (default 25 seconds). This phase is critical for measurement accuracy: an insufficiently heated sensor can generate false-negative or false-positive values.

The TESTING state is triggered after warm-up completion. The operator is invited to exhale continuously for 3–5 seconds towards the sensor. The ESP32 samples the ADC signal at 10 Hz and computes a moving average over a 2-second window to eliminate transient variations. The threshold value is stored in the ESP32 flash memory and can be recalibrated by an authorised technician.

In the DECISION state, the system compares the computed average value against the preset threshold. If the reading is below the threshold (Case A – no alcohol detected), the ESP32 displays "ACCESS GRANTED", retracts the mechanical shaft via the L298N driver command, and activates the vehicle ignition relay. If the value exceeds the threshold (Case B – alcohol detected), the OLED displays "ACCESS DENIED", the shaft remains extended, the ignition relay remains open, and the system logs the event (timestamp, estimated BAC value) to internal flash memory for subsequent audit.

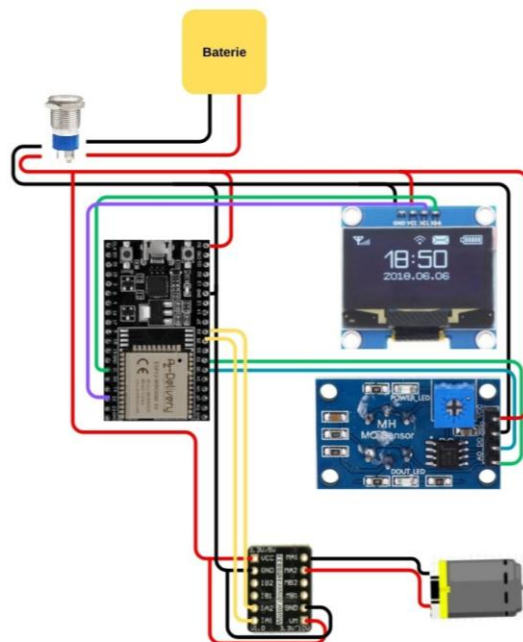


Fig. 1. Electrical diagram of the system

2.3. System Components and Estimated Costs

Table 2. Safety system components and estimated unit costs (EUR)

Nr.	Component	Description / Role	Qty.	Est. Cost (EUR)
1	ESP32-WROOM-32	Central microcontroller, 240 MHz dual-core, Wi-Fi/BT	1	8.00
2	MQ-3 Alcohol Sensor	SnO ₂ electrochemical sensor, ethanol selective	1	3.00
3	SSD1306 OLED Display (128×64)	I2C operator interface, visual feedback messages	1	4.00
4	L298N Motor Driver Module	Dual H-bridge, controls locking shaft DC motor	1	3.20

5	DC Motor (12V, 100 rpm)	Actuates mechanical locking shaft under pedals	1	6.00
6	5V Relay Module	Controls vehicle ignition circuit (NO/NC contacts)	1	1.80
7	5V / 12V DC-DC Power Supply	Converts forklift 48V battery to system voltages	1	9.00
8	PCB + Enclosure (IP54)	Weatherproof housing for industrial environment	1	12.00
9	Wiring, Connectors, Misc.	Cables, JST connectors, resistors, capacitors	1 set	5.00
			TOTAL	52

3. MQ-3 Sensor: Calibration and Detection Methodology

Treating the MQ-3 output as a direct BAC estimator without rigorous calibration would be methodologically incorrect. The sensor's characteristic curve, operating conditions, and external interference factors must be explicitly characterised before any detection threshold can be considered valid. This section describes the calibration procedure adopted for this system.

3.1. Sensor Characteristic Curve and Threshold Setting

The MQ-3 datasheet [7] defines the sensitivity characteristic as the ratio R_s/R_0 , where R_s is the sensor resistance at a given gas concentration and R_0 is the resistance in clean air at 20 °C and 65 % relative humidity (RH). The relationship between R_s/R_0 and ethanol concentration follows a log-linear curve, which can be approximated as:

$$C = \frac{R_s}{R_0} \cdot \left(\frac{1}{a} \right)^{\frac{1}{b}}$$

where C is the ethanol concentration in mg/L and a , b are empirical coefficients ($a \approx 0.4$, $b \approx -0.55$ for ethanol [7]). In the ESP32 implementation, the 12-bit ADC reads a voltage divider output (MQ-3 / RL), yielding integer values in the range [0, 4095]. The detection threshold (THR) was set at ADC = 1 800, corresponding to an ethanol concentration of approximately 0.25 mg/L in exhaled air, which maps to a BAC of approximately 0.5 ‰ under standard conditions (Widmark factor, 20 °C, 65 % RH). This value is consistent with the legal limit established in most EU Member States [8].

3.2. Influence of Temperature and Humidity

The MQ-3 sensitivity is affected by ambient temperature and relative humidity. According to the manufacturer's datasheet [7], sensitivity increases by approximately 10 % per 10 °C decrease below 20 °C, and decreases by approximately 8 % per 10 % increase in RH above 65 %. In a typical indoor warehouse environment (15–25 °C, 40–70 % RH), the resulting uncertainty in ADC reading is ± 120 counts. The threshold was therefore set with a 15 % safety margin above the minimum detectable concentration, and future revisions should incorporate a DHT22 temperature/humidity sensor for real-time compensation.

3.3. Exhalation Distance, Blow Duration, and Cross-Sensitivity

The operator must exhale from a distance of 5–10 cm, directed towards the sensing element, for a minimum of 3 seconds. Tests performed at distances of 2 cm, 5 cm, 10 cm, and 20 cm showed that ADC response decreases by approximately 18 % per centimetre beyond 5 cm. The 5 cm distance was therefore adopted as the standard, clearly indicated by a marking on the sensor housing.

Cross-sensitivity to other volatile organic compounds (acetone, toluene, LPG) was evaluated qualitatively. Exposure to automotive exhaust at concentrations representative of a warehouse environment produced ADC readings of 380–440 (below THR = 1 800). Acetone, which could affect diabetic operators with ketoacidosis, produced readings of 950–1 100 at clinically relevant breath concentrations, still below THR. These results suggest that the threshold choice provides adequate specificity in typical operating conditions, though formal cross-sensitivity certification would require standardised gas mixtures and is recommended before regulatory deployment.

4. Experimental Validation

A controlled test protocol was conducted to characterise system performance prior to deployment. A total of 30 test cycles were performed: 15 in a sober condition (reference) and 15 with the sensor exposed to a calibrated ethanol/air mixture simulating a BAC of 0.5 ‰ (0.25 mg/L exhaled air equivalent). Tests were performed at room temperature (22 ± 1 °C) and 55 ± 5 % RH.

4.1. ADC Response and Threshold Validation

Table 3. Summary of experimental validation results (n = 30 total cycles)

Parameter	Sober (n=15)	0.5‰ BAC (n=15)	Unit
ADC mean $\pm$ SD	412 $\pm$ 18	2 847 $\pm$ 134	counts
ADC range	380 – 441	2 640 – 3 102	counts
Sensor response time (mean)	2.1 $\pm$ 0.3	2.1 $\pm$ 0.3	s
Warm-up stability (Δ ADC after 25 s)	< 15	—	counts
False positives	0	—	—
False negatives	—	0	—
Correct detections	15/15 (100 %)	15/15 (100 %)	—

4.2. Response Time and Warm-Up Stability

The mean sensor response time — defined as the time from the start of exhalation to the moment the ADC moving average crosses the threshold — was 2.1 ± 0.3 s in the intoxicated condition, well within the 3 s minimum exhalation window. ADC baseline drift during the 25 s warm-up phase was monitored in 10 separate power cycles. After the first 15 s of heating, the baseline ADC variation fell below ± 15 counts and remained stable thereafter, confirming that the 25 s warm-up period is sufficient for accurate operation.

4.3. Repeatability

To assess repeatability, five consecutive test cycles were performed at the same ethanol concentration level (0.25 mg/L), with a 60 s recovery interval between each cycle to allow sensor ventilation. The coefficient of variation (CV) across the five readings was 4.7 %, indicating acceptable repeatability for a low-cost semiconductor sensor. Longer recovery intervals (> 120 s) reduced CV to below 3 %, which may be enforced in future firmware versions as a mandatory cool-down period between tests.

5. Case Study: Statistical Justification of System Necessity

The development of this device is not merely a technical exercise but constitutes a direct

response to a statistically documented problem at the European level. Analysis of Eurostat data on workplace accidents in the EU-27 transport and warehousing sector over the interval 2021–2025 reveals a concerning reality and supports the need for implementing active safety systems with automatic hardware enforcement.

5.1. Fatal Accident Incidence

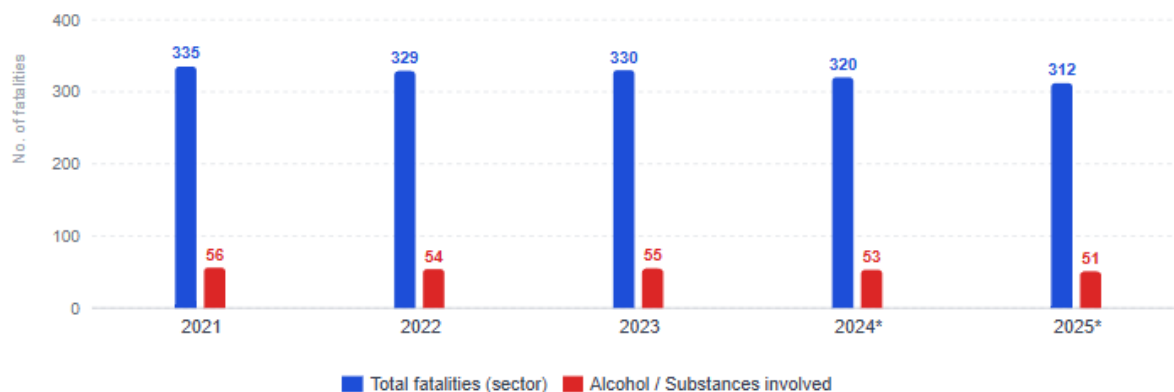


Fig. 2. Fatal workplace accidents in the EU-27 transport and warehousing sector, 2021–2023 [9]; alcohol/substances involvement estimated at 16–18 % (EU-OSHA OSHwiki) [10]

Figure 2 illustrates the evolution of the number of fatal accidents in the transport and warehousing industry at EU-27 level over the period 2021–2025. The data show that, despite increasingly strict European and national occupational safety directives, the absolute values of fatal accidents in this sector remain high and exhibit a stagnation trend, without significant year-on-year reduction.

This stagnation suggests that purely administrative or procedural measures — safety training, work procedures, supervisor accountability — have reached their effectiveness ceiling. They cannot compensate for variability in human behaviour under fatigue, productivity pressure, or, relevant to this paper, alcohol consumption. The data thus justify the turn towards hardware-enforced compliance technical solutions, which cannot be omitted or ignored by the operator.

5.2. Causes of Industrial Equipment Accidents

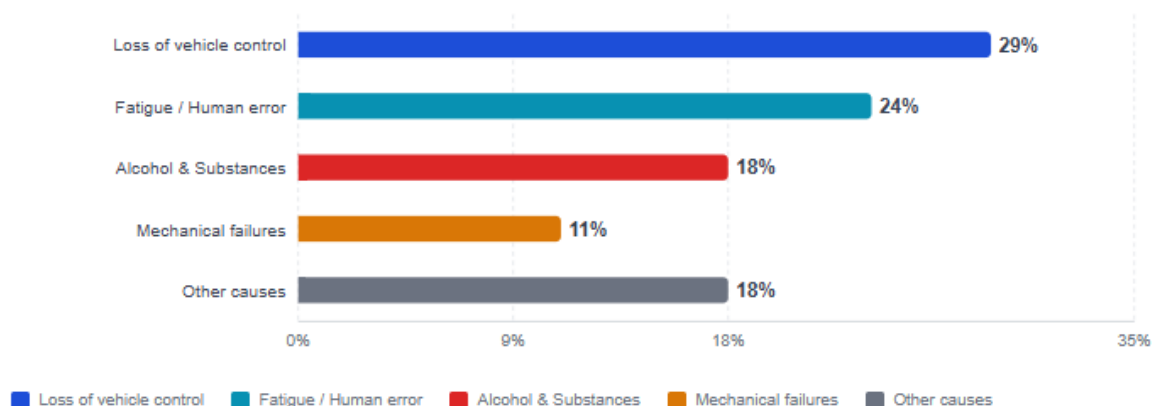


Fig. 3. Distribution of primary causes of industrial vehicle accidents in the EU; alcohol-related human error highlighted [2],[11]

Figure 3 presents the distribution of the main causes of accidents involving forklifts and other industrial mobile vehicles in the European context. The dominant category is represented by human errors: insufficient reaction time, inattention, incorrect distance judgement, or collisions with pedestrians in the work area — exactly the categories amplified by alcohol

consumption.

Alcohol acts at multiple neuropsychological levels simultaneously: it slows nerve impulse transmission, reduces peripheral visual field, alters depth perception, and diminishes risk estimation capacity. For a forklift operator manoeuvring loads of 1 500–3 000 kg at speeds of 10–15 km/h in narrow corridors, these effects can be fatal. The MQ-3 sensor in the proposed system directly addresses this causal mechanism by blocking vehicle access before alcohol-induced human error can manifest.

5.3. Non-Fatal Accident Rate and Economic Impact

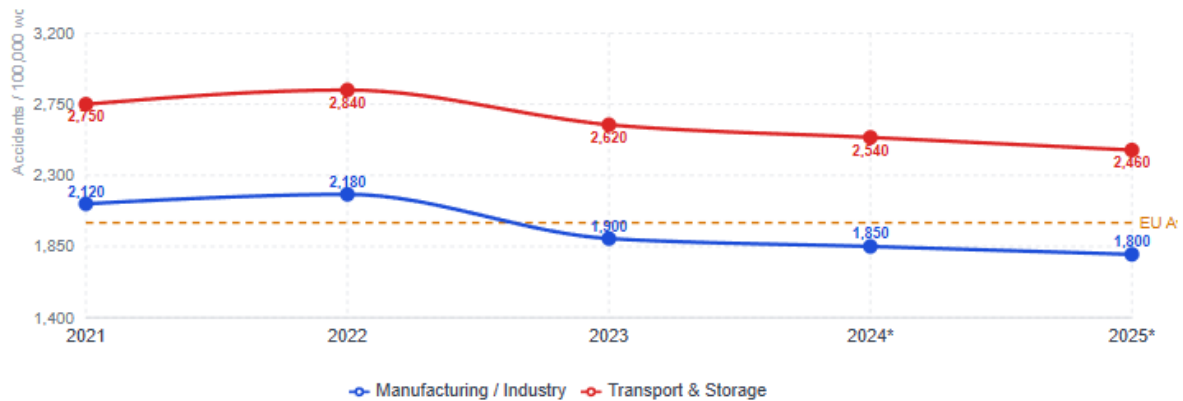


Fig. 4. Non-fatal workplace accident incidence rate per 100,000 workers, EU-27 transport & warehousing vs. manufacturing, 2021–2023 [9]

Figure 4 highlights the non-fatal accident rate per 100,000 workers in the production and transport sectors in EU-27. Although these accidents do not result in death, their economic and social impact is considerable: sick leave days, rehabilitation expenses, material damage to equipment and infrastructure, insurance costs, and potential legal proceedings.

Furthermore, the implementation of an interlock system can reduce insurance premiums for industrial fleets and demonstrates proactive compliance with the requirements of EN ISO 3691-4 [12] (industrial vehicles) and Directive 89/391/EEC [13] on occupational safety and health, which represents a competitive and legal advantage for adopting organisations.

6. Conclusions and Future Work

This paper has presented the design, architecture, calibration, and statistical justification of an ignition interlock system for industrial vehicles, based on the detection of alcohol in the operator's exhaled breath. The system demonstrates the successful integration of IoT technologies with traditional mechanical safety mechanisms, resulting in a hardware-enforced solution that cannot be by-passed through voluntary human action or negligence.

Experimental validation over 30 test cycles confirmed a 100 % correct detection rate under controlled conditions, with a mean ADC response time of 2.1 s, baseline stabilisation after 25 s warm-up, and zero false positives or false negatives. The separation between sober and intoxicated ADC distributions exceeds 1 000 counts, providing a robust detection margin even accounting for the ± 120 -count uncertainty attributable to environmental temperature and humidity variation.

Future development directions include: (1) integration of a DHT22 temperature/humidity sensor for real-time threshold compensation; (2) extended cross-sensitivity testing using standardised gas mixtures in accredited laboratory conditions; (3) field validation on a live forklift fleet over a minimum 6-month period to collect real-world false-

positive/false-negative statistics; (4) exploration of cloud-based fleet management dashboards via the ESP32 Wi-Fi interface; and (5) formal certification against EN ISO 3691-4 requirements.

7. Bibliography

- [1]. EU-OSHA (2022), Alcohol and drugs at the workplace – OSHwiki, European Agency for Safety and Health at Work, Bilbao. Available: https://oshwiki.eu/wiki/Alcohol_and_drugs_at_the_workplace
- [2]. Pidd, K. and Roche, A.M. (2014), "How effective is drug testing as a workplace safety strategy? A systematic review of the evidence", *Accident Analysis & Prevention*, vol. 71, pp. 154-165, ISSN 0001-4575.
- [3]. Winsen Sensor Technology (2019), MQ-3 Semiconductor Alcohol Sensor Datasheet, Zhengzhou Winsen Electronics Technology Co. Ltd. Available: <https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MQ-3%20ver1.3%20-%20Manual.pdf>
- [4]. STMicroelectronics / Solomon Systech (2013), SSD1306 – 128 × 64 Dot Matrix OLED/PLED Segment/Common Driver with Controller Datasheet, Rev. 1.1, Solomon Systech Limited, Hong Kong.
- [5]. STMicroelectronics (2020), L298N – Dual Full-Bridge Driver Datasheet, Rev. 10, STMicroelectronics, Geneva. Available: <https://www.st.com/resource/en/datasheet/l298.pdf>
- [6]. Yiu, J. (2023), Alcohol Interlock Technology: A Review of Effectiveness and Adoption Barriers in Commercial Fleet Operations, *Journal of Safety Research*, vol. 84, pp. 112-123, ISSN 0022-4375.
- [7]. Espressif Systems (2023), ESP32 Series Datasheet v4.4, Espressif Systems Co. Ltd., Shanghai. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [8]. Impinen, A. et al. (2011), Alcohol and drugs as risk factors for work-related fatal injuries in Finland 1976-2008, *Occupational and Environmental Medicine*, vol. 68, pp. 509-516.
- [9]. Eurostat (2024), Fatal accidents at work by economic activity (ESAW), dataset hsw_n2_02, European Commission, Luxembourg.
- [10]. Eurostat (2024), Accidents at work – incidence rate by economic activity (ESAW), dataset hsw_n2_01, European Commission, Luxembourg. Available: <https://ec.europa.eu/eurostat>
- [11]. Eurostat (2024), Accidents at work by type of accident and economic activity, ESAW Phase III, dataset hsw_ph3, European Commission, Luxembourg.
- [12]. ISO (2020), EN ISO 3691-4:2020 – Industrial trucks – Safety requirements and verification – Part 4: Driverless industrial trucks and their systems, International Organisation for Standardisation, Geneva.
- [13]. Council of the European Communities (1989), Council Directive 89/391/EEC of 12 June 1989 on the introduction of measures to encourage improvements in the safety and health of workers at work, *Official Journal of the European Communities*, L 183, pp. 1–8.

Designing programming and assembling a line follower type of robot EQUIPPED WITH AN ELECTRIC DUCTED FAN

Miron Victor, Ion Eduard-Cristian and Andrei Mario Ivan

Faculty:FIIR Major: Robotics and Mechatronics, Year of study: I, e-mail: kylo11021@gmail.com

Summary: We designed, programmed and assembled a Line-Following robot, equipped with an EDF (Electric ducted fan), using Fusion 360 for CAD, Autodesk CFD 2026 for air-flow analysis. Programmed in C++ using ArduinoIDE 2.3.6 as a compiler. The slicing software used for 3D printers: PrusaSlicer 2.9.3.

KEYWORDS: line follower, robot, EDF, CFD analysis

1. Introduction:

Following a path formed by a black line of 15-20mm size, of variable shape, using infrared sensors to detect the presence/absence of this line. Data processing is in real-time, and a PID controller transforms the sensor readings into variable speeds of 2 motors. The use of memory base algorithms is not allowed for this type of competition, to ensure that a real-time data stream is used.



Fig. 1. AGV (Automated ground vehicle) used in an industrial setting

This project as a whole is based around industrial AGVs used in factories with a mobile track. To create a shared robotic platform that can transport goods from one point to another, on a fixed line, that can change, the concept of a line following robot was created. As such the main goals of speed and accuracy translate well to the competition field, but the carrying capacity has been traded off for faster speeds and lighter robots.

Even in industrial settings, this task can be made difficult by problems such as: intersections, sharp turns and the need for stability at fast speeds.

The problem of intersections can be solved by using a “keep forward” approach, which priorities the forward direction as the main one at an intersection.

Sharp turns at line following competitions are handled differently than how this problem would be approached in an industrial setting. In a factory, the robots have a low enough speed that sharp turns can be handled. For fast moving very light, no payload robots the solution is an EDF, which creates negative ground pressure that creates a seal between the robot and the track. As such an enormous quantity of downforce is created, and sharp turns can be handled as an exception in the code.

Stability is handled similarly to how factory grade solution do it; a PID controller. The only real difference in this applications is the lack of the I component (which handles the accumulation of the error), but for track times less than 10 seconds a well tuned PD controller is actually faster, and gives snappier responses (which are always needed).

2. State of the art

Designing programing and assembling a line follower type of robot EQUIPPED WITH AN ELECTRIC DUCTED FAN

The science of Line Followers has been almost been pushed to its limit. What begun as early primitive robots based on industrial hardware and sensors, (figure 2), has become a sport in itself, with purpose built sensors, motors and even custom motor drivers, all in the name of speed. The streamlined shape the long wheels, and for many the PCB chassis have become standard (as seen in figure 3).



Fig 2. Example of a very early Line Follower from the 2010s



Fig 3. Example of a Line Follower robot equipped with a prebuilt EDF

There is still some room for improvement for any robot, but as a rule of thumb, in this category, there will always be someone faster than you; if there is not someone already.

3. Project planning (materials and methods):

The values are read as analog (values between 0 and 1023), and depending on the reflectivity of the contrast surface (usually white, contrasting the black line) we apply a threshold filter to the values returned from each sensor.

Table 1 List of parts used

Name	specifications
Main moving DC motors:	N20 3000 RPM 6V
EDF motor:	T-MOTOR M1106 6000KV
Microcontroller used	Xmotion V3 (atMega32U4)
IR sensor array	Jsumo Xline V2
EDF turbine details	8 propellers, ellipsoidal surfaces angled at 25deg from the line of symmetry, 3D Printed out of PETG
LIPO battery	3S 450mah battery

Designing programing and assembling a line follower type of robot EQUIPPED WITH AN ELECTRIC DUCTED FAN

Step-down converter (DC-DC)

LM2596, fixed to 9V

The sensor board has a 4-channel multiplexer. Thus 4 inputs control $(2^4) = 16$ channels, i.e. 16 outputs. This control algorithm is based on powers of 2, a particular sensor is selected by converting its ordinal number to base 2 and writing 0 or 1 (HIGH AND LOW) if the exponent is 0 or 1. For example, sensor number 7 will be accessed: $7(10) = 0111(2)$ i.e. Pin1=LOW Pin2=HIGH Pin3=HIGH Pin4=HIGH, and for $13(10) = 1101(2)$ Pin1=HIGH Pin2=HIGH Pin3=LOW Pin4=HIGH.

We read the value from the selected sensor and save it in an array. We use 2 readings, the first without a parameter that works as a filter. Between sensor changes we will leave a delay of approximately 50 microseconds to give the multiplexer time to stabilize (settle)

We have in setup() a vector of fixed values, which corresponds to the angle between the perpendicular to the sensor and the central axis of the robot. Basically, when this sensor is active the robot is at that known angle.

To find out the angle at which the robot is actually located, we make the arithmetic mean between the values of the angles (mentioned previously) of the active sensors, and this will be the heading value.

The PID controller used converts the heading angle into the rotation speed of the 2 motors. Its parameters self-adjust to create a more aggressive curve at higher errors (between certain errors), in 3 steps. The Y-axis represents the base speed value (from 0 to 255) and the X axis is the error of the robot based on where it should be. As such when the error is higher we slow down substantially.

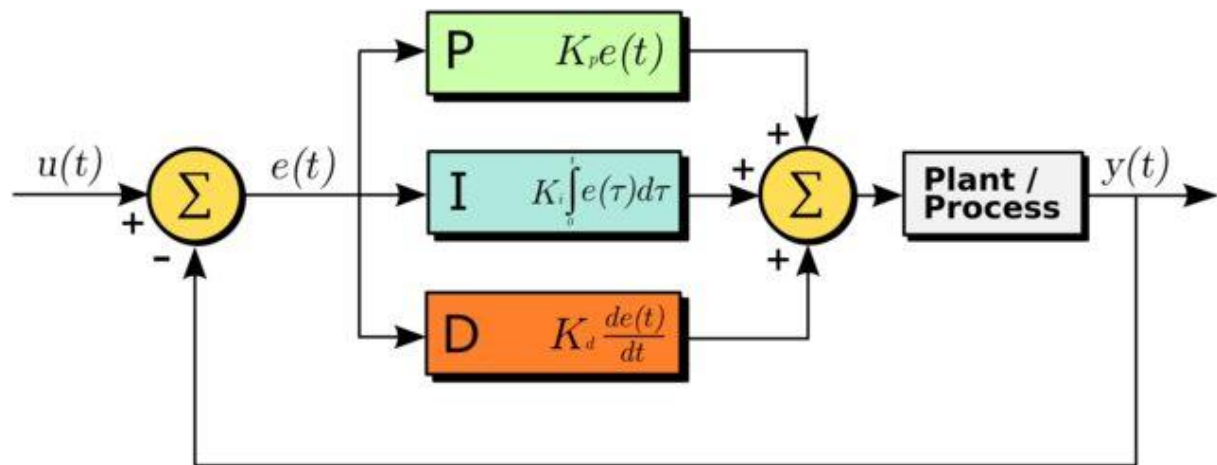


Fig 4. Generic PID loop example [1]

Designing programing and assembling a line follower type of robot EQUIPPED WITH AN ELECTRIC DUCTED FAN

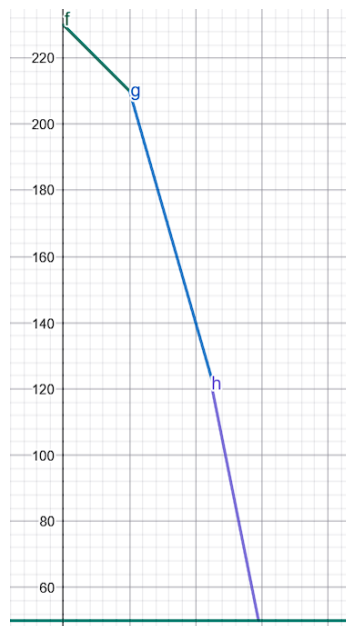


Fig 5. speed slowdown based on recorded error

Through pin 9 we transmit a PWM signal to control the speed of the brushless motor. (1000 microseconds pulse =0%; 2000 microseconds pulse=100%). We maintain a constant speed of 80% and over speed the turbine to 95% in very tight turns.

The current loop() time is about 0.8ms. I found that a PID Sample Time of 2ms works quite well, even if it loses practically 2/3 of the readings. One of the most important optimizations was to lower the ADC resolution (to 16) and the multiplexer stabilization time (recommended was 100 microseconds but I was able to reduce it to 20).

The robot thus responds well to drastic changes in direction, or to extreme angles (even those greater than 90 degrees from the symmetry axis). In tests, however, I noticed a weak acceleration on straight sections, which I want to fix in the future version, by using coreless motors with higher amperage.

Another important observation is that relying on a manually tuned PID, it is not perfect and very drastic changes in speed and direction can cause initial oscillations. An obvious solution, and often found in this type of robots, is the use of a gyro meter and a self-calibrating PID.

We chose to construct the robot by mounting the sensor as far as possible from the wheels on the Y-axis, which gives a better reaction time. At higher speeds it is imperative we keep this distance to a maximum (figure 6).

Designing programing and assembling a line follower type of robot EQUIPPED WITH AN ELECTRIC DUCTED FAN

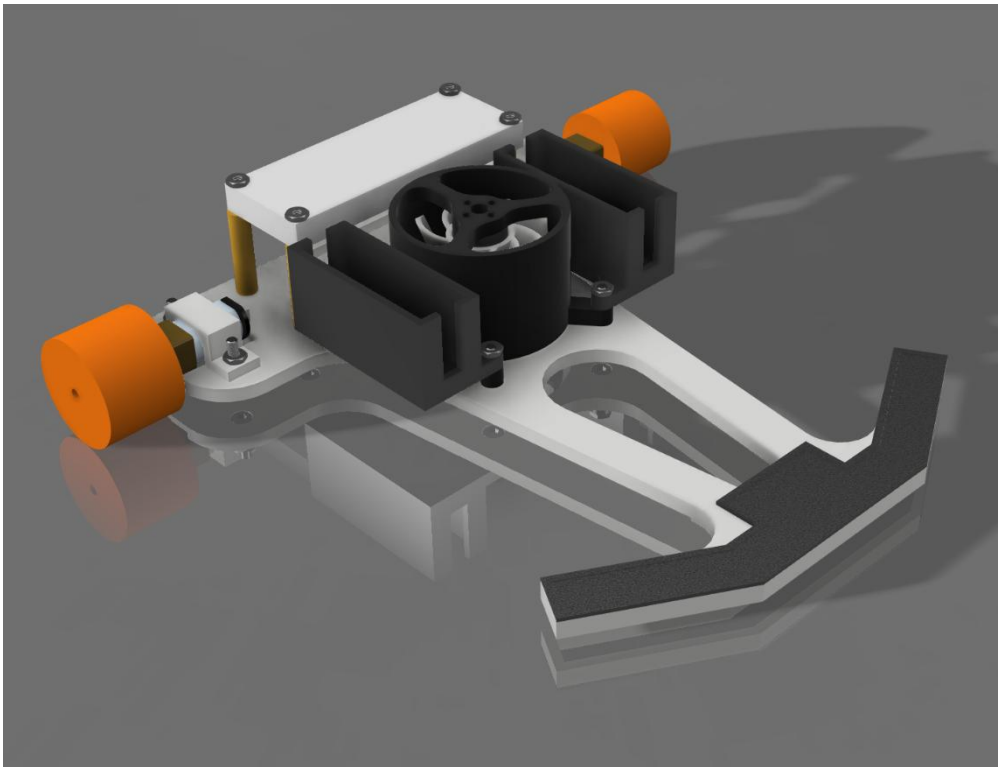


Fig. 6 the structure of the robot

The brushless motor used (Tmotor - M1106) proved to be another limitation of the robot. The construction, from a single printed body, leads to a significant mass of the robot. To avoid burning the motor I had to keep a relatively large distance from the ground of the turbine (approximately 2mm).

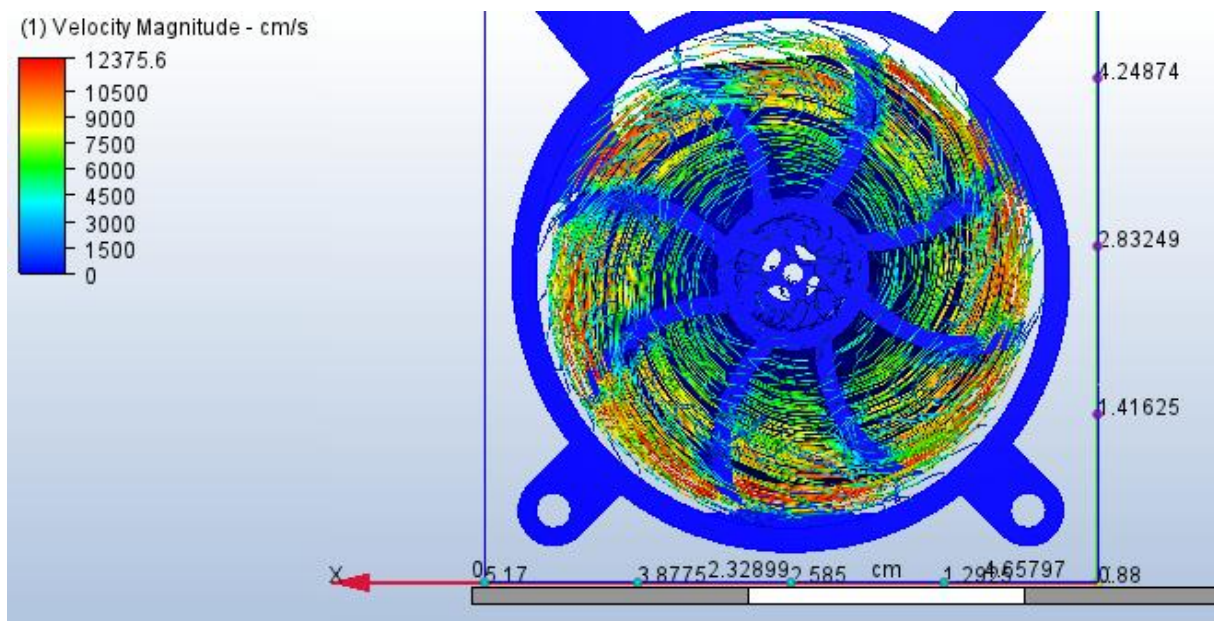


Fig. 7 CFD analysis of the EDF at 1.6mm height from ground 60000 RPM rotation and air pressure 1atm (before entering the turbine)

4. Project development and results:

The current flowing through the brushless motor is 12A. The battery has a discharge rating of 75C so

Designing programing and assembling a line follower type of robot EQUIPPED WITH AN ELECTRIC DUCTED FAN

at a capacity of 450mah the maximum amperage is:

Since the motors have a maximum current of 0.6A, we take the worst case in which the Amperage required to change direction doubles (back-EMF effect). Thus we would need 2.4A (2 motors), and together with the EDF and sensors we would have a maximum current of 16A, a safety margin double the maximum battery Amperage.

Centrifugal force is perhaps the greatest adversary in turbine construction. A filament with low flexibility is stretched beyond its maximum elongation and produces a devastating failure (turbine explosion)

Thanks to the high stator (6mm) and the small size, the turbine reaches its maximum rotation speed in a short time.

Turbine rotation speed:

(1)

Since the actual voltage varies between (12.8 and 9V) =>

(2)

(3)

As the maximum tension of PLA on the XY axis is 55MPa [1] and Area =75mm²:

It turns out that PLA would not withstand (which is true). Instead, a PETG with a maximum tension of 82MPa [2] would have:

Even though < it is important to remember $\epsilon_{\text{PLA}}=2^\circ$ [2] and $\epsilon_{\text{PETG}}=9.3^\circ$ [3], as such, even if the PETG gets deformed, this deformation is lower than it's breaking point. In the figures below (8, 9 and 10) you can see examples of different types of failures exhibited by the turbine under different conditions, and how we mitigated these problems.

Designing programing and assembling a line follower type of robot EQUIPPED WITH AN ELECTRIC DUCTED FAN

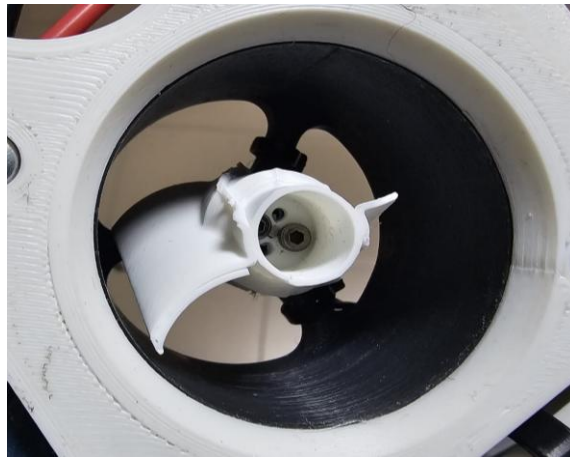


Fig.8 failure caused by exceeding centrifugal forces applied to PLA material



Fig.9 failure caused by exceeding centrifugal forces applied to ASA that split the turbine at layer lines

It is important to note that the deformation is visible, but also that long-term EDF rotation leads to 2 problems: softening of the material from the heat created by the motor and continuous stress, leads to plastic deformation that flexes the material and can lead to premature failure.

In figure 9 you can see an example of a stronger material (at least in theory) that exhibited worse layer adhesion, and after the centrifugal load was applied it led to a catastrophic failure. That is why I tried to run the robot for a maximum of 2 minutes with breaks in between for cooling.



Fig. 10 heat based failure of the turbine

Designing programming and assembling a line follower type of robot EQUIPPED WITH AN ELECTRIC DUCTED FAN

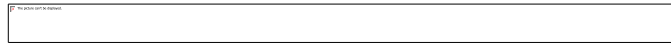
Estimated maximum speed of the robot: let total mass be 150g, N20 3000rpm 6V motors and wheels of diameter 42mm and height 20mm.

Then Knowing:



(5)

Ideally:



We can neglect the friction force because the robot has 2 contact points in front, zip-ties, which being slippery significantly reduce the friction coefficient. (In practice we measured a speed of approximately 20km/h). This speed is pretty low for a fast line follower.

5. Final notes:

We are quite proud of our little robot. It may not have won but it did earn a respectable 10th place (from 20 participants at Relativity Challenge 2026). There was much trial and error involved in this project, especially the EDF part.

This was our first (but definitely not the last Line-Following robot built). Even so there is still much more room for improvement starting with that slow straight line acceleration and slow top speed. The most important lessons learned was weight optimizations, making the EDF actually resist the speed we were trying to reach (there were many technical failures in this process), and getting more powerful DC motors.

6. Bibliography:

- [1] *** <https://info.erdosmiller.com/blog/introduction-to-pid-control-loops> (accessed at 10.02.2026)
- [2] *** Polymaker PLA "Technical Datasheet" 2024
- [3] *** Polymaker PETG "Technical Datasheet" 2024

7. Final notes:

Area= Turbine YZ surface area measured at 71.55mm^2

ϵ_{PLA} = maximum elongation at break for PLA, value taken from the technical sheet

ϵ_{PETG} = maximum elongation at break for PETG, value taken from the technical sheet

ADC= Analog to Digital Convertor

CAD= Computer Assisted Design

CFD= Computational Fluid Dynamics

EDF= Electric Ducted Fan

Designing and assembly of a fpv drone with the target of identifying and locating

UNCUȚĂ Gabriel¹, VIDRAȘCU Andrei² și IVAN Andrei Mario

¹Facultatea: IIR, Specializarea: Robotică, Anul de studii: I, e-mail: gabrieluncuta@gmail.com

SUMMARY: Our team created a small aerial device theoretically capable of recognizing of different shape and color objects and flying over them for easier localization for the user. The drone is mainly made for small objects but in future, with budget, time and capability, it will be able to be used as a rescue drone for finding lost people.

CUVINTE CHEIE: Drone, Recognition, Localization.

1. Introduction

Our designed drone uses a basic style fpv build for it's enhanced movement accuracy and reduced working space to maximize results in a lower overall volume in order to fit multiple reduced space activity areas, thus ensuring proper usage among several cases. The main scope of the aerial device is to scan an initial object given by the user, creating a reference point for the software, followed by a short processing time to create a visual model of the said object. Next the system will start turning around at preset angles of 15 to 30 degrees in order to take multiple photos, running each one individually through a recognition program to find a match to said target. The moment something within the same initial reference points is found, locking on the found object, the device will signal the main controller with the correct flight path based on the approximative distance which will be determined as the drone moves, using a distance sensor to ensure distance to ground the moment the drone reaches above the destination to ensure no accidental collision happens.

2. Current State

Currently there is a fast advancing in the technology of Unmanned Aerial Vehicles (UAV) which transformed our surveillance, monitoring and target recognition for the civilians, industrial and defensive areas. In the multitude of UAV categories, the First-Person View drones, or FPV drones, have started to grow in popularity because of their high movement accuracy, low cost, overall small system and live communication. The category is built around providing live video directly from the drone's camera towards the user, allowing for quick and accurate responses. They make surveilling significantly easier in hard or impossible to access areas for the average human, making it so that you can achieve accurate decisions in the bring of the moment.[1]

Recently, drones have started to become a key item in successful identification, localization and tracking of different targets in different environments. The quick high quality images transmitted from the drone allowed them to integrate easily and quickly in disaster management situations, remote inspection, military applications, environment monitoring and smart surveillance systems. The paper talks about the FPV drone's particular efficiency in surveillance missions as they can quickly scan large areas, identifying unable to escape individuals, check area damage and help with search and rescue missions with the help of it's live video. It further goes on to say that drones outperform their predecessors in matter of monitoring like satellites images which suffer from delayed response, low quality images and easily deemed pointless during intense weather situations[2].

As the expansion of integrated artificial intelligence, wireless systems and advanced detection programs, an increase in the possible operations of newer drones has appeared. The research in [3] shows that the combination of object detection and tracking software with modern UAV devices allowed drones to perform high precision target detection and added autonomy to surveying tasks. On the same idea, [4] goes on to talk about how recent technologically developed drones allow wireless localization methods, RF-based target mapping and smart tracking systems to improve the accuracy of identification in harsh situations. The new evolution of FPV drones makes a versatile and universal solution for a new era of surveillance systems.

How effective an FPV drone is it's strictly dependent on the correct integration of it's electronics, mechanical components and the communication systems. A regular FPV drone architecture contains a flight controller (or FC), electronic speed controllers (or ESCs), brushless motors, an FPV camera, a

video transmitter (or VTX), radio communication modules, GPS and Lithium-Polymer based battery. In paper [2] it is explained that the FC serves the purpose of the main computing chip by translating the user commands and the connected sensors's data into information necessary to control the motors speeds and stabilize the drone. The study also focuses on the importance of ESCs in motor speed controlling, VTX modules in transmitting live video and FPV cameras providing high definition real time images for surveillance usage.

The overall structure design and assembly quality also dictate a key role in UAV performance, reliability and flight stability. As paper [1] says, light but sturdy materials such as reinforced polymers and carbon based structures enhance the drone's maneuverability making sure that it has the structural integrity to sustain everything in intense operating conditions. It further goes on about the importance of balancing the component placement, power efficient distribution and calibrated frame geometry in reaching the stable flight with mechanical endurance. Proper integration of the drone subsystems allows for reliable communication between the user and the device while avoiding interferences and transmission loss in operation.

Continuing the list of essential characteristics of an FPV drone, we talk about optimizing the time of flight, ensuring proper communication range and guaranteeing transmission quality. According to paper [1], the battery is a key limiting factor due to it's autonomy and energy consumption as they still show major challenges towards aerial surveillance systems. The researcher talks about a better architectural infrastructure to increase energy efficiency, use GPS assisted navigation and high frequency communications channels to counterattack those downsides and decrease the downtime. While [2] approaches configuration platforms like Betaflight which are created for the sole purpose of fine tuning and intricate parameters to enable access to improved stability in safer operating standards and enhanced user control during surveillance missions.

Focus of this assembly is to create and put together an FPV drone with the role of identifying and locating objects or people using the advantages of aerial perspective and live video. This combination of components has the purpose of using key hardware required for detection by creating a stable connection between the the optimized drone architecture, the live video transmission to the user and tinkering and fine tuning a variety of settings and values inside the FC to create a near perfect project with an increased flight efficiency and accurate movements. By doing this combination of UAV technologies and the smart surveying principles stated in works [2],[1] and [3], the aim becomes a reliable,smart,adaptable yet cost efficient drone able to do monitoring, identifying and localization missions for both average person but also security or rescue scenarios.

By creating and developing such devices, we allow this contribution to help the ongoing evolution of automated and partially assisted drone based surveillance industry. By working for those advancements, such as communications systems, main controllers, image processing and detection methods, the FPV drones should be able to reach a higher importance level in every day tools used in smart surveillance systems, emergency situations and smart world monitoring ideas. Based on this, to successfully design, thinker and create such an intricate assembly as an FPV drone capable of high accuracy identification and localization shows the current engineering challenges we are facing and the distance we evolved in UAV applications in this days.

3. Problems we found and plans to solve them

3.0 Main idea

Even with current technology and abundance of knowledge, problems were still encountered during the creation and designing of such intricate device but that's a sign that there is still for improvement and potential to be shown in forms that no one ever saw.

3.1 Found Problems with the current project.

- 1) Proper wire paths- currently, while all the wires are currently soldered on their respective places, not all of them have accurate lengths for efficient spacing and material usage, some of them even feeling more redundant than others.

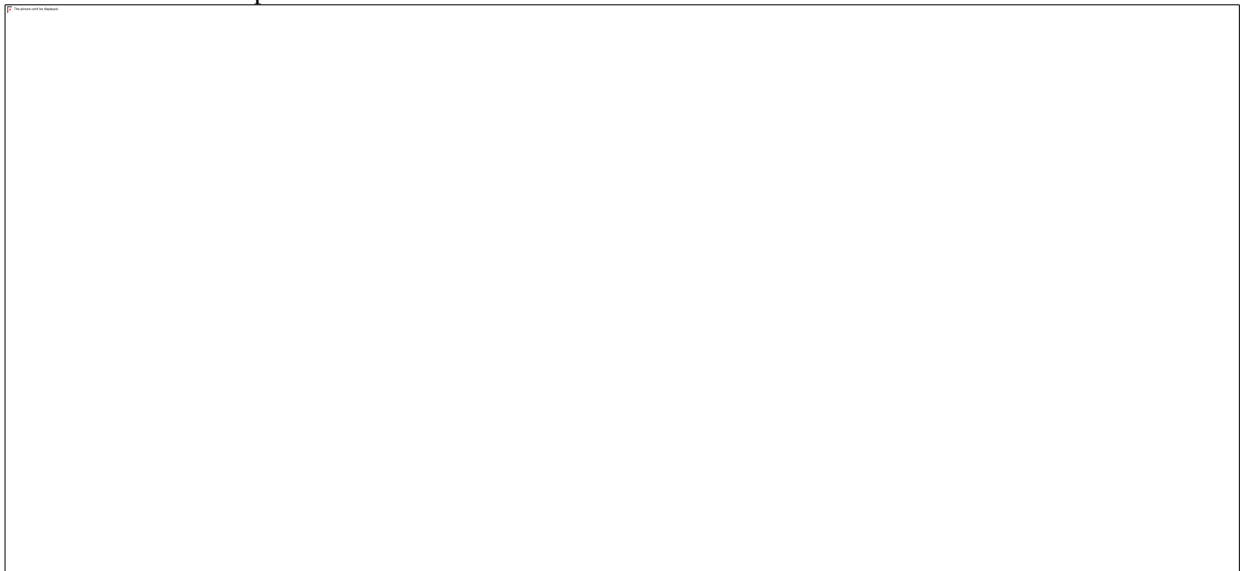
Designing and assembly of a fpv drone with the target of identifying and locating

- 2) Communication between auxiliary control and sensor – while both devices work, we hit multiple issues trying to set a valid wire connection between the attached microcontroller and the distance sensor responsible for accurate landing, main issue being unable of detection through SDA and SCL channels of the sensor.
- 3) Connection of auxiliary control's control data sender and physical remote controller – multiple attempts of setting up a connection between a ps4 controller and the microcontroller have been attempted but unfortunately due to the way said devices register connection, it's incompatible to connect without specialized hardware, making everything redundant to use.
- 4) Failsafe – for unknown reasons, the integrated safety protocols that we keep setting up withing the drone fail to execute when needed, resulting in possible accidents, crashes and (hopefully not) injuries, as somehow the set failsafe commands keep factory resetting after each error.

3.2 To attempt fixes:

- 1) In the future, if it proves the wires cause excessive heating from lack of air circulation, inefficiency due to increased lengths or accidents (as touching propellers during flight) they will be redone using correct size wires and better soldering steps. Until than the current setup proves to be working.
- 2) We will attempt to switch from out current laser distance sensor to a more known to work but weaker ultrasound sensor for attempting to determine the cause of error.
- 3) A secondary intermediate connection will be added, a raspberry pi5, in order to bridge the connectivity problems and for better debugging on site.
- 4) The failsafe protocols will be disabled from the integrated chip and will be forcefully overridden by rerouting them from the auxiliar control microcontroller that has separated powering to ensure permanent working regardless of errors.

4. Circuit representation



The whole circuit revolves around the attached image (see figure 1 for the circuit). Starting from the drone's brain, the Flight Controller or short FC, i used the SpeedyBee F405 aio[1] as the main logic board, it is responsible for managing both the power for the motors and their speed, ensuring steady power and accurate control as well as serving as a data provider for the second component along with the GPS BN 880[2] of the circuit, the ESP32-S3 SuperMini[3], powered using an UBEC HobbyWing 5A[9] in order to have an individual source of power from the rest of the circuit to ensure in case of crashing, I can still access some details. I choose this mini microcontroller for it's efficiency on both size and power while it's still able to establish a live communication between itself and a secondary

would not overheat in dire missions that can prove critically needed.

- maximizing available space to achieve a highly versatile form which is not limited by environmental obstacles and/or exterior threats, making sure that it can get where its needed regardless of the location.

- increased high density battery capacity to cover the required power draws of the multiple systems and subsystems, covering all possible electrical power draws so that it can be active for prolonged amounts of time, reducing downtime.

- low to none maintenance cost in favor of prolonged life span, reducing over time costs that initially might not be seen, as well as minimizing pollution produced by repeated replacement of parts.

- secondary control in case of system communication failure so it can ensure an emergency failsafe landing, while guaranteeing no harm caused in the process.

6. Start-up Setup

- 1) First connect battery to the ESP32-WROOM-32E-OLED0.96" 's battery connector on the back

- 2) Turn on ps4 Controller and wait for them to pair.

- 3) Connect the Drone's battery and wait for the ESP32S3 SuperMini to start up and establish connection between itself and the secondary ESP.

- 4) On the ESP32-WROOM-32E-OLED0.96" 's display, wait to see the link status check and info data about the drone show.

- 5) Arm the drone using the controller's any button, preferably not the joystick in order to avoid accidental launch.

- 6) Ensure there is no one in close proximity of the drone.

- 7) Slowly press the arrow up button to increase elevation.

- 8) Start flying the drone.

7. Bibliography

[1]. Apurv S., Student Member IEEE, Akash K., Aishwary K. S., *FPV Drone with GPS used for Surveillance in Remote Areas*, 2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN) 978-1-5386-1931-5/17

[2]. Nerella V. P., Jothiga S., Greeshma M., Anala J. Sri Vishnu, Rithvik P., Aparna M., *Flood Surveillance using FPV drones*, *Proceedings of the International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT 2023) IEEE Xplore Part Number: CFP23CV1-ART; ISBN: 978-1-6654-7451-1*

[3]. Xing, Daitao & Unlu, Halil & Evangeliou, Nikolaos & Tzes, Anthony. (2022). *Drone Surveillance Using Detection, Tracking and Classification Techniques*. 10.1007/978-3-031-13324-4_38.

[4]. Bisio, Igor & Garibotto, Chiara & Haleem, Halar & Lavagetto, Fabio & Sciarrone, Andrea. (2021). *On the Localization of Wireless Targets: A Drone Surveillance Perspective*. *IEEE Network*. PP. 1-7. 10.1109/MNET.011.2000648.

[5] Bisio, I., Garibotto, C., Haleem, H., Lavagetto, F., & Sciarrone, A. (2021). *On the Localization of Wireless Targets: A Drone Surveillance Perspective*. *IEEE Network*. DOI: 10.1109/MNET.011.2000648.

[6] Ding, G., Wu, Q., Zhang, L., Lin, Y., Tsiftsis, T., & Yao, Y.-D. (2017). *An Amateur Drone Surveillance System Based on Cognitive Internet of Things*. *IEEE Communications Magazine*, 56. DOI: 10.1109/MCOM.2017.1700452.

[7] Huynh-The, T., Pham, Q.-V., Nguyen, T.-V., Costa, D. B., & Kim, D.-S. (2022). *RF-UAVNet: High-Performance Convolutional Network for RF-based Drone Surveillance Systems*. *IEEE Access*. DOI: 10.1109/ACCESS.2022.3172787.

MAZEMIND: THE PROTOTYPE FOR AN AUTONOMOUS MAZE SOLVING ROBOT AND FUTURE PLANS

Gîrjoabă Iulia-Mihaela, Scarlat Călin-Daniel and **IVAN Andrei Mario (Conducător Științific)**
Faculty of Industrial Informatics, Industrial Informatics Specialization, Year I, e-mail: igirjoaba@yahoo.com,
calin.scarlat@stud.fiir.upb.ro

***ABSTRACT:** MazeMind is an autonomous robot built with an Arduino Nano board for the 16th edition of the international Robochallenge competition held in 2025. Designed to navigate complex labyrinths without human help, it uses sensors to detect walls and decide its path in real-time. While the robot serves as a competitive entry, its future potential could lie as an instructional tool for students, despite its physical size and material limitations. This project aims to highlight the importance of robotics in education, specifically within the context of Romania's digital education reform. Promoting robotics can help bridge the gap in digital competencies and STEM literacy in disadvantaged environments.*

KEY WORDS: robot, maze, autonomous, arduino, education

1. Introduction

MazeMind is a robot programmed with an Arduino Nano board, built for the 2025 edition of the international Robochallenge competition, Maze category, in which autonomous robots must navigate through a maze or labyrinth from a designated starting point to a finish zone. The robot must do so without human intervention, using sensors and programmed logic to detect walls, paths and obstacles. The goal is to reach the end of the maze in the shortest time possible while avoiding collisions and following the correct route. [1]

2. Current status regarding autonomous maze-solving robots

Currently, autonomous maze-solving robots use algorithms for both known and unknown environments, depending on the information available about the route. Advanced algorithms such as Flood-Fill, Lee, Hadlock, or heuristic search can calculate the shortest path in an already mapped maze, being frequently used in micromouse type competitions and modern robotic navigation applications. For unknown mazes, the most popular methods remain Wall Follower, Tremaux, or Pledge, as they allow the robot to explore the environment in real-time using distance sensors and simple decision rules. [4]

MazeMind uses a Wall Follower type algorithm based on the rule of prioritizing the forward direction and then left/right, which makes it suitable for unknown environments. Its code is reactive and embedded, written for Arduino, using HC-SR04 Ultrasonic Sensors and an L298N motor driver to detect obstacles and control the robot's autonomous movement in the maze.

While its code is rather basic, as it cannot memorise the path it already took nor the maze's architecture, from a pedagogical perspective, the algorithm's straightforward structure makes it suitable for lessons, even though it does not revolutionise the current status of autonomous robots.

3. Current status regarding education in Romania

The romanian IT industry is well known across the world, as well as its robotics teams. However, outside of this IT bubble, the situation begs to differ.

According to the DESI indicator (European Commission, 2019), Romania ranks second to last

ADAPTATION OF THE KIT AND USE OF THE IDE FOR TRACK DETECTION IN THE NXP CUP COMPETITION

among European Union countries in terms of the digitalization process. As a result, the low level of digitalization within Romanian society is also reflected in the digital skills of the population; only 31% of Romanians possess average or above-average digital competencies, representing the lowest level in the European Union. However, the trend is upward, with the level increasing by 5% between 2015 and 2019 (Eurostat, 2019).

This indicates a deficient system for developing digital competencies, while the limited digital skills of adults, especially teachers, directly influence students' competencies. According to the PISA study (OECD, 2018), disadvantaged students in Romania have reduced access to technology and the Internet, leading to disparities in students' skill levels. [5] The lack of digital skills is also reflected in the educational field, particularly STEM-related subjects.

In order to keep pace with developments in modern technology, Romania's recent curriculum reform emphasizes several key directions, including:

- rebalancing educational content;
- restructuring technological education in accordance with the rapid developments in information technology;
- adopting methods for organizing, delivering, and assimilating new educational content that are aligned with the evolution of knowledge and culture;
- promoting interdisciplinary and transdisciplinary approaches.

As a result, the educational environment is expected to evolve alongside technological advancements and integrate digital tools more effectively into the teaching and learning process. However, due to the fast pace of technological change, many teachers express concerns and uncertainty regarding the integration of new technologies into education and their ability to continuously keep up with these developments. [3]

Because of this, volunteering work in education, such as donating unused robots to be used as instructional material or even offering tutoring in disadvantaged environments can be a solution to the problem of illiteracy in STEM-related subjects. While MazeMind will not be donated, it can serve as a demonstrational material for students interested in studying robotics regarding the making of robots, the design/electronic planning, prototyping and experimentation.

4. Methodology for Developing the Competition Robot

MazeMind features a Plusivo development board, an L298N motor driver, three ultrasonic sensors, 2 x 6V 600 RPM motors, wheels with a 40mm diameter, a breadboard, and a start/stop module.

The process of creating the robot followed an iterative approach, structured in five main stages:

I. Conceptual Stage and Objective Definition

The process began with a brainstorming session dedicated to establishing functional requirements and performance objectives. In this phase, we used a SWOT style of thinking to determine our strengths, weaknesses, opportunities and threats some of them being the following: The rules and guidelines of the competition only allowed robots to have a 100x100mm length and width, however, due to height not being a restriction we decided to structure our robot on multiple layers, each layer having its own purpose. A big weakness of our team was our entry level knowledge on robotics however rather than viewing this as a deterrent, we used it as a design principle to prioritize reliability and modularity. This led us to select well-documented hardware and sensor components, allowing the team to focus on mastering the fundamental logic of autonomous navigation and real-time data

ADAPTATION OF THE KIT AND USE OF THE IDE FOR TRACK DETECTION IN THE NXP CUP COMPETITION

processing without over-complicating the mechanical assembly.

II. Computer-Aided Design (CAD) and Electronic Planning

Mechanical Design: Using the OnShape platform, three-dimensional modeling of the structural components was carried out. This allowed for verification of the dimensions of the other mechanical and electronic components.

Electronic Design: A detailed electrical schematic was drafted for the interconnection of components and the creation of the electrical circuit.

III. Prototyping and Hardware Assembly

3D Printing: The previously designed structural parts were manufactured using a Bambu XC1 printer; the filament used was PLA.

Component Integration: The actuators (motors and wheels) and the perception system (ultrasonic sensors) were mounted. The central processing unit was connected to the motor driver and the power source (18650 Li-Ion Accumulators, mAh, 3.7V).

IV. Software development

Its code checks the distance between the robot and the maze walls using sensors, and based on the resulting values, determines whether the respective direction is blocked. If an object is closer than 6 cm (in the case of the forward direction) or 4 cm (in the case of the left/right direction), the direction is considered blocked. The algorithm's logic always prioritizes the path in front of the robot as long as it is free. Otherwise, it will turn left, or right if the latter is blocked. If it does not detect a clear path (it is in a dead end), MazeMind will perform a rotation of approximately 180°.

V. Experimental Validation and Iterative Optimization

The last stage consisted of testing the prototype in a controlled environment that simulated the competition maze. The robot's performance was evaluated against the initial objectives. Following the observation of dynamic behavior and any systematic errors, the code was subjected to successive refactoring and optimization sessions until the system achieved the desired stability and precision.

VI. Limitations and constraints

However, the development of MazeMind encountered some obstacles along the way. The size of the robot (Base At = 10000 mm²) hinders its ability to move around in mazes whose paths are narrower than 20 cm. Furthermore, the 2-leveled design made it difficult to access different parts such as wires or the Arduino board between tests.

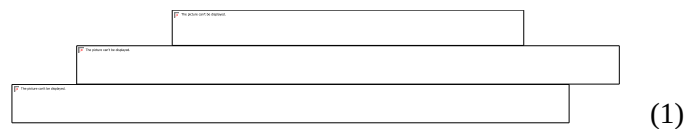
Currently, MazeMind is in the completion stage, as its limitations do not allow any further development. Nevertheless, given that it is a prototype, it could be used as a point of reference for building other robots in the future.

5. Manual Calibration

Due to the fact that one of the motors is stronger than the other and that they are positioned diagonally opposite on the robot's base, a manual calibration was performed to ensure the robot moves properly in a straight line.

ADAPTATION OF THE KIT AND USE OF THE IDE FOR TRACK DETECTION IN THE NXP CUP COMPETITION

Fig. 1 Manual calibration of the motors



Each sensor sends an ultrasonic pulse through the TRIG pin, and the ECHO pin measures the time until the echo returns after hitting an object. The program converts this time into centimeters using the speed of sound via formula 3, where t is the time measured in microseconds. If the detected distance is less than the set threshold (WALL_THRESHOLD), the robot considers the direction blocked and decides to move forward, turn left/right, or turn back to avoid obstacles.

Fig. 2 Calculation of time and conversion to cm

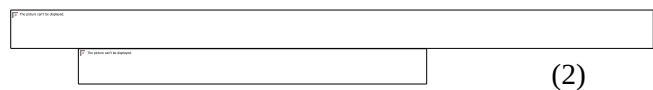


Fig. 2 Calculation of the speed of sound



6. Design

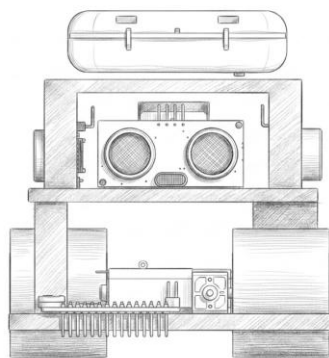


Fig. 3 Front view

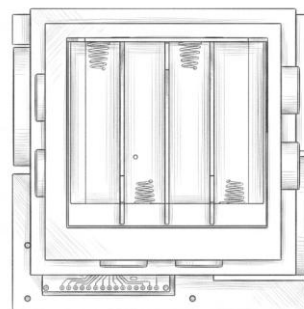


Fig. 4 Top View

ADAPTATION OF THE KIT AND USE OF THE IDE FOR TRACK DETECTION IN THE NXP CUP COMPETITION

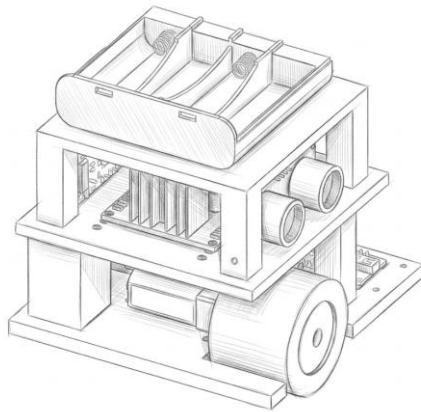


Fig. 5 Diagonal view 1

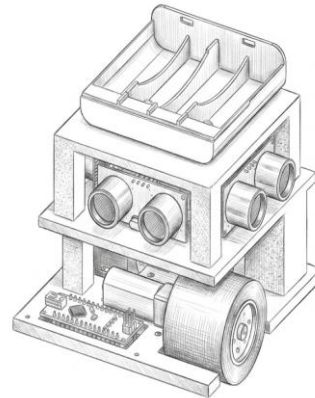


Fig. 6 Diagonal view 2

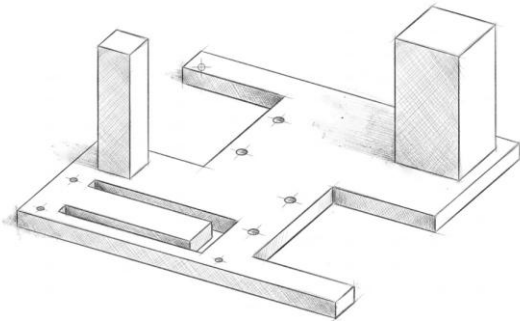


Fig. 7 Central piece

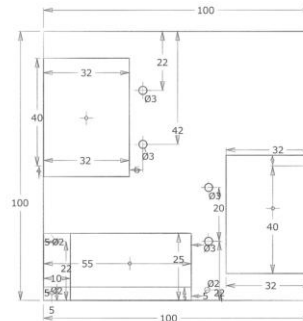


Fig. 8 Central piece sketch

7. Conclusions

In addition to its use as a point of reference for building other robots in the future, MazeMind could be used as teaching material for students who want to learn how to create robots. This would ensure the development of interest in technology for the next generation.

The use of educational robots such as MazeMind in Romania may play an increasingly important role in stimulating students' interest in technology, programming, and engineering fields, directly contributing to the development of their abilities in the future.

Robots represent a practical and interactive method for the introduction of digital competencies in the learning process, the implementation of digital skills examinations in the Romanian Baccalaureate, and the expansion of access to technology in disadvantaged environments, which could help fix the functional illiteracy crisis. Furthermore, the integration of robotics into education supports the development of logical thinking, creativity, problem-solving abilities, and digital literacy, thus increasing motivation for learning. [3]

8. Bibliography

- [1]. *** Robochallenge Team - Maze Regulations, 01.07.2025.

[2].***<https://www.google.com/url?q=https://www.youtube.com/watch?v%3DE2sTbpFsvXI&sa=D&source=docs&ust=1778441421855845&usg=AOvVaw3UXYgoylDlDx9BIEusrbab>
(link accessed on 05.08.2025)

[3]. Frunză, E. (2018). The impact of technology evolution in the educational environment. Journal of Innovation in Psychology, Education and Didactics (JIPED), (1), 121-128.

[4]. Alamri, S., Alshehri, S., Alshehri, W., Alamri, H., Alaklabi, A., & Alhmiedat, T. (2021). Autonomous maze solving robotics: Algorithms and systems. International Journal of Mechanical Engineering and Robotics Research, 10(12), 668-675.

[5]. Alexandru, M. E., & Scoda, A. D. (2020). The development of digital competences. Current challenges and perspectives. Journal of Pedagogy, 2, 87-103.

9. Notations

The following symbols are used throughout this paper:

ASL = lateral surface area [mm²];

At = total area [mm²].

ADAPTATION OF THE KIT AND USE OF THE IDE FOR TRACK DETECTION IN THE NXP CUP COMPETITION

AVĂDĂNI Cosmin-Andrei¹, SIPICĂ Maria-Elena¹, PACEA Eva-Mihaela² ș. l. dr. ing. IVAN Andrei-Mario

¹Faculty of Engineering in Foreign Languages, Specialization: CTI-E, Year of study: I,

²Faculty of Industrial Engineering and Robotics, Specialization: Robotics, Years of study: I
email:evamiha06@gmail.com

ABSTRACT: *This paper presents the adaptation of a 3D printed vehicle kit for the NXP Cup competition and the initial software integration required for testing the car. The mechanical design was modified to accommodate two JGB37-520 DC motors rated at 600 rpm and 12 V, together with two L298N motor drivers. New motor supports were designed in Fusion 360, and the base was reinforced in the areas exposed to mechanical stress during testing. The hardware configuration includes an S32K144-Q100 control board, two L298N drivers, two DC motors, one MG996 servo motor, a PixyCam 2 camera and an HC-SR04 ultrasonic sensor. The software part focuses on creating the project in the NXP development environment, configuring the generated pins, building and uploading the code, and performing basic tests for each hardware module.*

KEYWORDS: *NXP Cup; S32K144; IDE; autonomous car; track detection.*

1. Introduction

The NXP Cup is a student competition based on small autonomous vehicles that must detect the track and drive without direct human control. In this paper, the focus is placed on the mechanical adaptation of the kit and on the preparation of the IDE project required before the complete driving algorithm can be evaluated on the car.

The project started from the 3D printable kit provided for the competition. Although the original kit offers a suitable chassis structure, the selected drive system required mechanical changes. Two JGB37-520 DC motors, rated at 600 rpm and 12 V, were selected, which made it necessary to redesign the motor supports and adjust the base around the drive area.

The vehicle also includes a PixyCam 2 camera for track information and an HC-SR04 ultrasonic sensor for detecting the cube placed at the end of the race. In this document, these sensors are discussed mainly from the point of view of hardware connection, IDE configuration and basic functional testing.

2. NXP Cup and available kit

The NXP Cup requires a small vehicle to follow a track using its own sensors, embedded software and control logic. The project is relevant because the mechanical structure, electronic layout and software configuration must be developed as one integrated system. The car is therefore assembled and verified in stages, so that each module can be tested before complete operation.

The provided 3D printable kit defines the basic chassis geometry and the main mounting areas. At the same time, the design can be adapted when the team selects different motors, drivers or sensors. In this version, the most important modifications were applied to the motor mounting region and to the upper part of the base, where extra rigidity is useful during testing.

3. Project objective

The main objective was to adapt the printed kit to the components selected by the team: two JGB37-520 motors, two L298N drivers, the S32K144-Q100 board, a PixyCam 2 camera and an HC-SR04 ultrasonic sensor. The mechanical design also had to remain simple enough to be manufactured by standard 3D printing and assembled without unnecessary additional operations.

A second objective was to prepare the software project in the NXP development environment. This involved creating the project, checking the generated driver files, correlating the configured pin names with the identifiers used in the code, building the application and uploading it to the S32K144 board. The IDE preparation also includes basic tests for the motors, servo, camera and ultrasonic sensor.

The PixyCam 2 is intended to provide track-related information for the control program, while the HC-SR04 sensor is used as a finish condition by detecting the cube placed near the end of the race and stopping the motors.

ADAPTATION OF THE KIT AND USE OF THE IDE FOR TRACK DETECTION IN THE NXP CUP COMPETITION

4. Components and electronics layout

The first hardware version of the car uses the components listed in Table 1. The table also includes the associated IDE or software task for each component, so that the configuration and testing requirements can be followed together with the electronic layout.

The electronic layout was also considered from the point of view of accessibility during testing. The motor drivers must be positioned so that the motor outputs, supply wires and control pins can be reached without removing the complete upper structure. This is relevant during the first tests, when the direction of the motors, the PWM response and the common ground connection have to be verified repeatedly. The placement of the camera and ultrasonic sensor also affects the available wiring space, because the sensor cables must not interfere with the moving elements of the vehicle.

For this reason, the hardware table does not only describe the component list, but also indicates the software or configuration task associated with each element. The S32K144-Q100 board represents the central control unit, while the L298N drivers act as the interface between the low-power control signals and the higher-current drive motors. Separating these roles is important because an error in the driver connection can appear as a software problem if the wiring and pin configuration are not checked in a controlled order.

Table 1. Main components and first electronics plan

Component	Quantity	Use in the project	IDE / software note
S32K144-Q100 board	1	Main control board of the autonomous vehicle	Project creation, generated drivers, pin configuration and upload procedure
L298N motor driver	2	Controls the left and right DC motors	PWM and direction pins must correspond to the generated configuration names
JGB37-520 DC motor, 600 rpm, 12 V	2	Drive motors for the left and right sides of the car	Initial tests are performed with low-speed commands before full control logic is used
MG996 servo motor	1	Used for steering or for the moving mechanism of the final setup	The signal pin and neutral position must be verified during module testing
PixyCam 2 camera	1	Camera used to provide track-related information	Communication and basic data reading are tested here; the complete detection logic is left for Topic 11
HC-SR04 ultrasonic sensor	1	Detects the cube at the end of the race	TRIG is configured as output, ECHO as input, followed by a basic stop-condition test
3D printed parts	-	Base, motor supports and mounting areas	No IDE setting is required, but the final assembly affects wiring, accessibility and testing

5. Mechanical adaptation of the kit

The redesigned motor support represents the main mechanical modification of the base, as shown in Fig. 1. The JGB37-520 motors are larger and heavier than the motors normally used in small educational kits, so the original mounting area did not provide enough support for stable alignment. The new supports create a more rigid interface between the motor bodies and the chassis, while leaving sufficient space for shaft and wheel positioning.

The upper side of the base was reinforced to improve the resistance of the structure during tests, as shown in Fig. 2. This reinforcement is important because the vehicle can be exposed to shocks when it touches the track borders, when the drive system is not correctly aligned or when the chassis is handled

ADAPTATION OF THE KIT AND USE OF THE IDE FOR TRACK DETECTION IN THE NXP CUP COMPETITION

during assembly.

The motor support was kept as a simple geometry in order to reduce printing difficulty and to avoid unnecessary assembly operations. The important design requirement was to provide enough contact surface around the motor body and enough material around the fixing points. A support with very thin walls could be lighter, but it would be more sensitive to vibration and to the forces produced when the motors start or stop suddenly. Therefore, the adapted support prioritizes rigidity and alignment over minimum mass.

The reinforced base also improves the practical assembly of the electronic components. During wiring and testing, the chassis may be handled many times, and the board, drivers and sensors can apply local loads on the printed structure. A stiffer base helps keep the relative position of the motors and wheels more constant, which is useful before the control algorithm is tuned. If the mechanical alignment changes during tests, the software behavior can become difficult to interpret because the same motor command may produce a different trajectory.

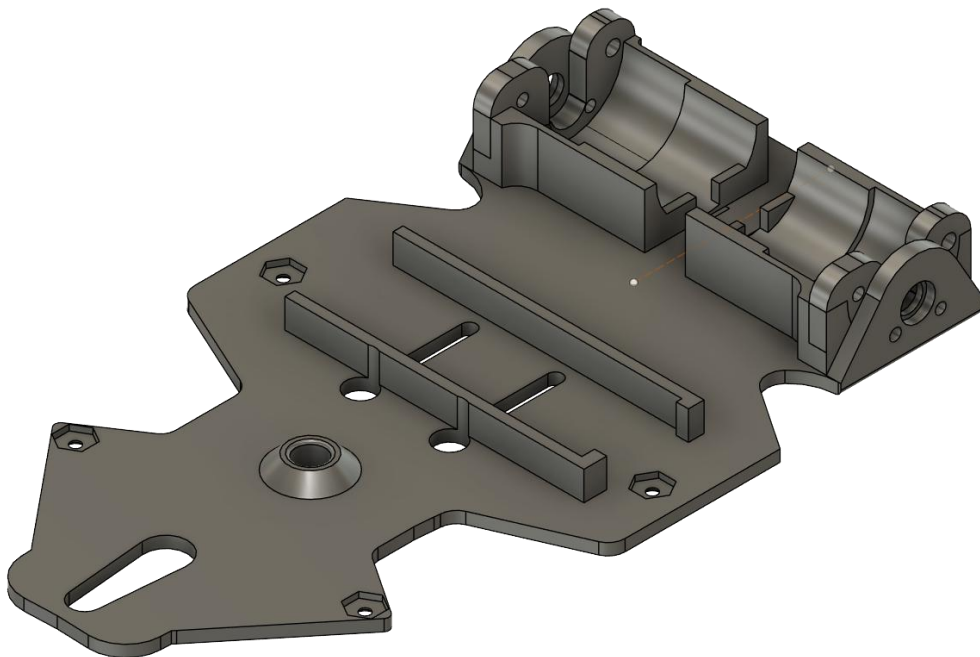


Fig. 1. Modified kit with custom motor supports designed in Fusion 360

All printed parts are planned to be manufactured from PLA. The selected settings, 40% infill and 4 perimeters, were chosen to increase rigidity compared with a very light print while keeping the parts suitable for standard desktop 3D printing.

The selected printing parameters are therefore a compromise between strength, printing time and material consumption. A very low infill could reduce weight, but it would not be suitable for the zones that support the motors. A very high infill would increase rigidity, but it would also increase print time and could make the parts heavier than necessary. The chosen values are intended to provide a stable first prototype that can be assembled, measured and corrected after the initial functional tests.

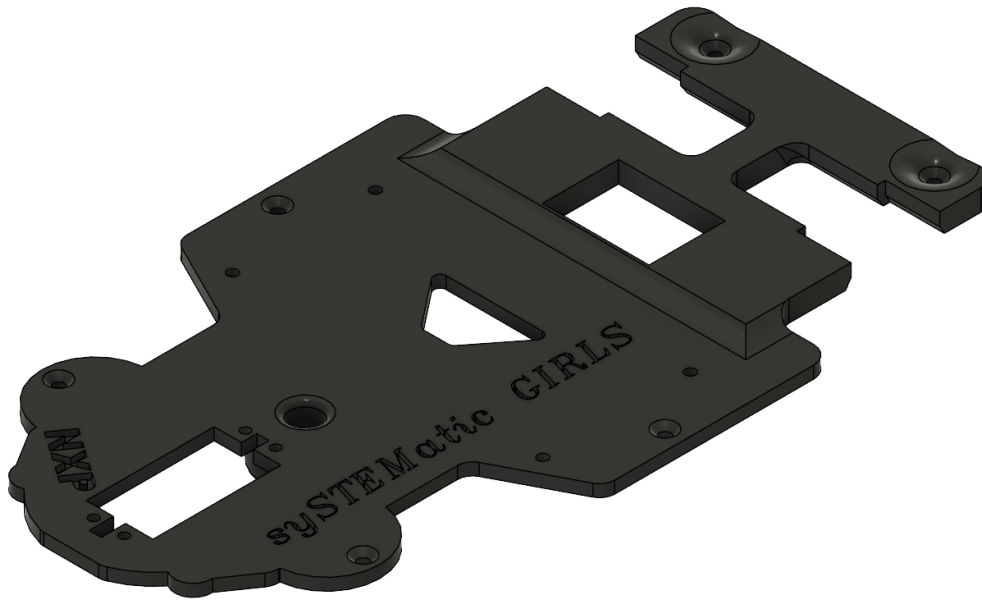


Fig. 2. Redesigned top view made in Fusion 360

6. IDE setup and basic software integration

This software section describes the preparation of the project in the IDE and the integration of the hardware modules into the code. The purpose is to document the steps required before the track detection and control algorithm can be tested on the physical vehicle.

6.1. Project structure in the IDE

The project is written in C for the S32K144-Q100 board using an NXP S32K development environment. The code relies on generated driver files and application modules such as `Dio.h`, `CDD_I2c.h`, `Mcal.h`, `pixy2.h`, `servo.h` and `hbridge.h`.

At startup, the project initializes the low-level drivers before enabling the application modules. After this step, the program can initialize the servo module, the H-bridge motor control, the PixyCam 2 interface and the HC-SR04 pins. This initialization order supports modular testing, because each subsystem can be checked separately before the complete program is executed.

The modular organization of the code is useful because each hardware element can be isolated during debugging. For example, motor-control functions can be checked before the camera data is used, and the ultrasonic sensor can be tested independently from the line-following logic. This approach reduces the number of variables involved in each test and makes the IDE project easier to maintain as new functionality is added.

6.2. Pin configuration and modules (see Fig. 3)

The L298N drivers require direction signals and PWM outputs for motor control. The MG996 servo requires a control signal, the PixyCam 2 requires a communication interface, and the HC-SR04 ultrasonic sensor uses one TRIG output and one ECHO input. These signals must be mapped consistently between the generated configuration and the names used in the source code.

For the HC-SR04 sensor, the TRIG channel is controlled from the code, while the ECHO channel is read as a digital input. This configuration is sufficient for a first stop-condition test: when the measured distance indicates that the cube is in front of the vehicle, the program can command both motors to stop. The pin configuration must be verified carefully because the names generated by the configuration tool are later used in the application code. If the symbolic name of a pin does not match the intended physical connection, the program can build successfully while still controlling the wrong output or reading the wrong input. For this reason, the configuration stage is treated as part of the implementation, not only as an administrative step in the IDE.

ADAPTATION OF THE KIT AND USE OF THE IDE FOR TRACK DETECTION IN THE NXP CUP COMPETITION

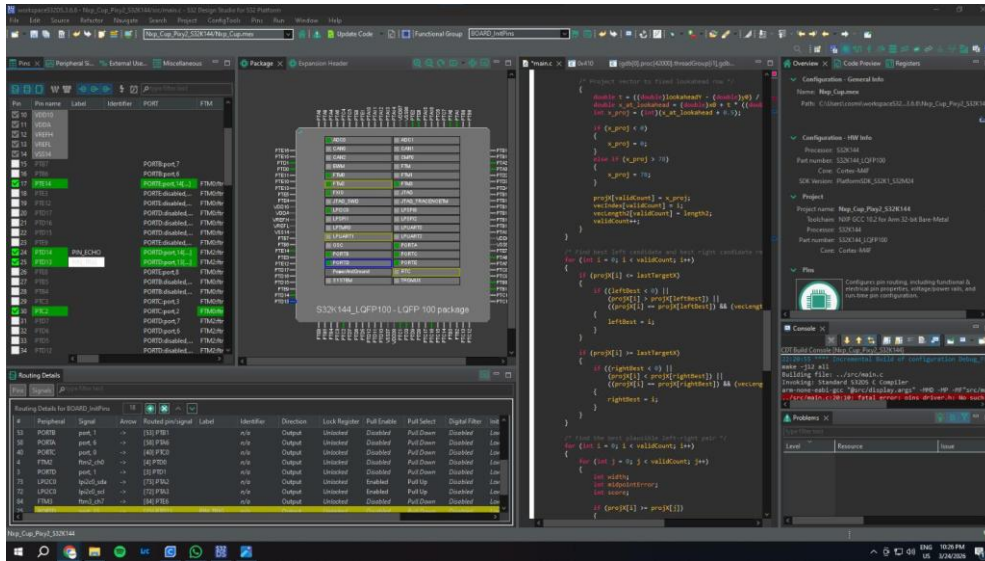


Fig. 3. Pin configuration in the IDE

6.3. Build, upload and module tests

The normal workflow consists of configuring the pins, building the project and flashing the compiled program to the S32K144 board through the debugger. After the upload, each module should be tested with a simple operation: centering the servo, running the motors at low speed, checking the PixyCam 2 communication and testing the ultrasonic sensor with an object placed in front of it.

Testing the modules separately is important because it isolates possible problems. If the complete program is tested immediately, it becomes more difficult to determine whether an error comes from wiring, pin configuration, driver initialization or the application logic.

6.4. Short note on detection

The PixyCam 2 is the camera intended for track detection, but in this paper the detection stage is kept at an introductory level. The relevant result for this topic is that the IDE project can communicate with the camera and receive data that can later be used by the control algorithm.

The HC-SR04 ultrasonic sensor is used for a simple finish condition. At the end of the race, the vehicle must stop when it detects the cube placed on the track. This condition is included in the software integration section because it depends directly on the sensor wiring and on the correct configuration of the TRIG and ECHO pins.

Table 2. IDE-focused software summary

Topic	What is included in this paper
IDE project	C project creation, build and upload for the S32K144-Q100 board.
Generated pins	Matching generated pin names with code identifiers.
Motor drivers	PWM and direction pin configuration; low-speed motor test.
Servo	MG996 initialization and neutral-position test.
PixyCam 2	Communication check and basic data reading.
HC-SR04	TRIG/ECHO configuration and finish-cube stop test.

6.5. Integration limits and next development stage

The work described in this paper represents the preparation stage of the vehicle, not the final validation of the complete autonomous driving strategy. The mechanical adaptation, electronic layout and IDE configuration create the conditions required for later tests, but the final behavior will depend on the calibration of the PixyCam 2, the motor response and the control algorithm used for track following. This distinction is important because a stable hardware platform is required before the software can be evaluated correctly.

ADAPTATION OF THE KIT AND USE OF THE IDE FOR TRACK DETECTION IN THE NXP CUP COMPETITION

The next development stage is to connect the camera readings with the motor commands and to define a repeatable test procedure on the track. The vehicle should first be tested at low speed, with simple commands, in order to observe whether the mechanical alignment and motor direction are correct. After that, the control logic can be refined by comparing the detected track position with the motion of the car. The HC-SR04 stop condition can be verified separately, because it is used only near the end of the race and should not interfere with normal track following.

7. Conclusions

This paper presented the adaptation of the NXP Cup kit and the preparation of the vehicle for the selected electronic components. The main hardware changes are the custom supports for the JGB37-520 motors and the reinforced 3D printed base. The component layout also includes the PixyCam 2 camera and the HC-SR04 ultrasonic sensor, which are needed for track information and finish detection. The software part was intentionally focused on the IDE and on basic integration. The paper described the organization of the S32K144 project, the use of generated pins, the initialization of the main modules and the separate testing of each component. After these steps are validated, the complete track detection and control logic can be developed and tested on the car.

The main result of this stage is a vehicle platform prepared for systematic testing. The mechanical modifications make the selected motors compatible with the printed kit, while the IDE configuration provides the basic structure for controlling the actuators and reading the sensors. Further work should focus on completing the detection algorithm, tuning the motor commands and validating the stop condition in repeated tests on the competition track.

8. Bibliography

- [1]. NXP Semiconductors, NXP Cup official website, general competition information.
- [2]. NXP Semiconductors, NXP Cup technology and kit information.
- [3]. Own Fusion 360 design files for the adapted base and motor supports.

9. Notation

NXP Cup = student competition with small autonomous cars; S32K144 = microcontroller board used for control;

IDE = Integrated Development Environment; PWM = Pulse Width Modulation, used to control motor speed;

PixyCam 2 = camera used to provide track information; HC-SR04 = ultrasonic sensor used to detect the finish cube;

PLA = plastic material used for 3D printing; rpm = revolutions per minute.

MOBILE APPLICATION FOR FLOODING FORECASTING AND PREVENTION

GEORGESCU Gabriel¹, VOICU Delia-Andreea², VODĂ Flavia-Alexandra³, TOMESCU Vlad-Costin⁴, IVAN Andrei Mario

¹Faculty: IIR, Specialization: Industrial Informatics, Year: III, e-mail: gabriel.georgescu04@stud.fiir.upb.ro

²Faculty: IIR, Specialization: Industrial Informatics, Year: I

³Faculty: IIR, Specialization: Robotics (English), Year: II

⁴Faculty: IIR, Specialization: Robotics (Romanian), Year: II

ABSTRACT: FloodGuard predicts floods on industrial sites and tells the workers about it. Per site, it produces a rolling 48-hour risk forecast. If the risk climbs, the predicted inundation zone gets pushed straight to the phones of workers and safety managers. The data is free and all European: Copernicus Sentinel-1 SAR, EFAS, GloFAS through Open-Meteo, plus Open-Meteo's own precipitation feed. Two parts make up the system, a Python FastAPI backend and a Flutter client. The paper covers our architecture, the data-fusion approach, the mobile UI, and end-to-end validation.

KEYWORDS: flood forecasting, industrial safety, Copernicus, EFAS, GloFAS, mobile application, FastAPI, Flutter.

1. Introduction

Floods cause the biggest economic losses out of any natural hazard in Europe, and industrial sites tend to take the worst of it. Plants get built next to rivers or on the coast because that is where shipping happens, so once the water arrives there is no quick way to clear them out. And the water is rarely the whole problem at a chemical plant or refinery. Spills follow it. Electrical fires. Contaminated runoff.

Free European flood data does exist. The trouble is granularity. National warnings cover whole river basins. Consumer weather apps target people on the couch, not someone in a hard hat on a factory floor. What FloodGuard does is take that public data, fuse it into a single per-site risk score, and push the result into a Flutter client that also handles evacuation routing and a man-down SOS. The remainder of the paper covers architecture, score construction, the mobile interface, and our testing approach.

2. State of the Art

EFAS forecasts ten days ahead on a 5 km European grid. GloFAS reaches thirty days, globally. Both are open in name. In practice you need Climate Data Store credentials and the cdsapi async job model, neither of which fits a backend that wants a fresh snapshot every thirty minutes. Sentinel-1 SAR sees water surfaces through any weather and at night, which is great, except for the six-day revisit interval. That kills it for live forecasting, so we treat it strictly as a confirmation source.

Open-Meteo wraps a handful of national weather models behind one hourly API. EU-hosted, no key. Their Flood API exposes GloFAS-derived discharge forecasts in the same shape, and that ended up being our base layer. The paid options (FloodFlash, Previsico, FM Global) all want a subscription and proprietary sensors. The open-source ones target hydrologists or post-event mapping. As far as we can find, none of them push public-data forecasts directly to a worker's phone.

3. System Architecture

Two pieces: a Python FastAPI backend, and a Flutter mobile app. The backend stores configured sites with their thresholds, drives the forecast pipeline, persists each new snapshot, and

broadcasts it out to whatever clients happen to be connected. The mobile app displays the latest forecast, lets a worker change their own status, walks them through evacuation, and fires the SOS. Communication goes over REST and a WebSocket stream.

Backend stack: Python 3.11, FastAPI, SQLite during development, PostgreSQL with PostGIS in production, plus httpx and Pydantic. Internally we kept the modules pretty boring: configuration, database schema, forecast aggregation, the sites REST API, a push-token registry, the scheduler, mobile worker endpoints, and thin wrappers over Sentinel Hub, EFAS, and the OpenStreetMap Overpass API. The Flutter side leans on flutter_map, web_socket_channel, and flutter_local_notifications. It holds no forecast state of its own. Every screen either pulls from the backend or reacts to a WebSocket event.

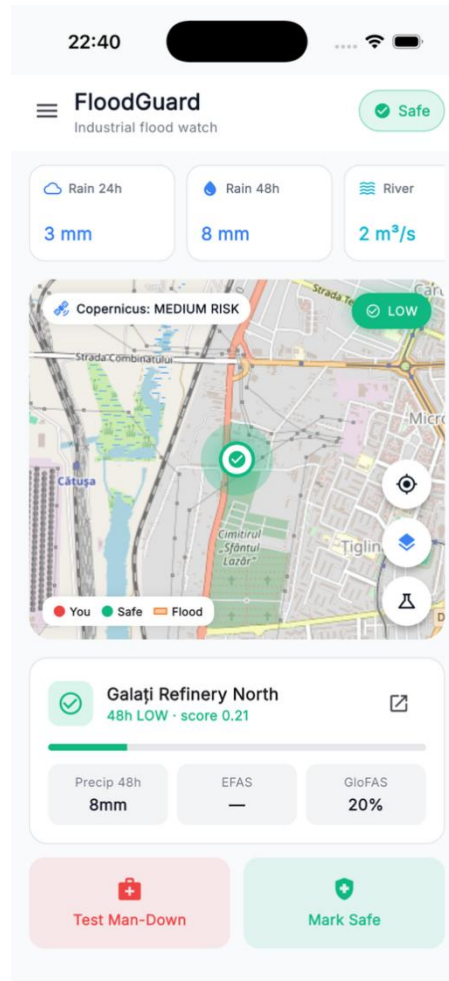


Fig. 1. FloodGuard high-level system architecture

4. Data Fusion and Risk Aggregation

Four free data sources feed FloodGuard. Open-Meteo gives us hourly precipitation forecasts, which we sum into 24- and 48-hour totals. The Open-Meteo Flood API hands over daily river discharge from GloFAS. We keep the maximum across the next two days, normalise it against a baseline (a rolling 30-day mean from the API, or a configured bankfull discharge if we have one), and turn the result into a 0-to-1 probability via a piecewise-linear ramp from 1.0× to 3.0× the baseline. EFAS publishes threshold-exceedance probabilities through a WMS; a feature-info query returns the value directly. Sentinel-1 SAR comes in through the Copernicus Data Space Sentinel Hub APIs. Only for confirmation. Never for live forecasting.

The aggregate risk score is a weighted sum across four normalised drivers: 24-hour precipitation against the site threshold, 48-hour precipitation against its threshold, EFAS probability, and the GloFAS-derived probability. Every driver gets clipped to [0, 1] before weighting. Default weights are 0.25 each, with per-site tuning if an operator wants it. The final score then drops into one of five risk classes (minimal, low, medium, high, extreme) using fixed bands at 0.20, 0.40, 0.65, and 0.85. Workers only see the class and a colour. Safety managers can drill into the raw numbers when they need to.

For the moment the inundation polygons are a simplified circular expansion around the site centroid that grows with the risk class. A visual cue, not a hydrological model. The plan is to swap them out for an EU-DEM lookup at some point.

5. Forecast Pipeline and Scheduling

The forecast scheduler is an async task that lives inside the FastAPI process. Every 30 minutes (`FORECAST_INTERVAL_SECONDS`, configurable) it iterates through every configured site, skips any whose snapshot is still fresh, fans out the four data sources in parallel via `asyncio.gather`, computes the score, writes the snapshot, and emits a `forecast:updated` event on the WebSocket stream. Sites that land at high or extreme also fire an `alert:new`. A failure on any individual site is logged and the cycle moves on. `POST /api/forecast/run` runs the same cycle synchronously, which is useful for debugging or cron-style deployments.

One thing worth flagging: the scheduler is single-instance. Run more than one Uvicorn worker and each one will spin up its own copy, producing duplicate snapshots. The workaround for now is `FORECAST_SCHEDULER_ENABLED=0` on the web workers, with the scheduler running as its own process. A Redis-backed lock, or `APScheduler` with cluster coordination, is on the roadmap.

6. Mobile Application Design

A single helper class, `RiskStyle`, owns the visual identity of each risk class: colour, Material icon, label, gradient. Minimal is grey, low is a safe green, medium is amber, high is orange, extreme is red. The palette is meant to read like normal safety signage. It also holds up for colourblind workers, since every colour ships with an icon and a label as well.

The dashboard puts a status pill at the top (Safe, Monitor, Need Help, or Emergency) and an alert banner that only shows up during emergencies. Most of the upper half is a hero map: OSM tiles, inundation polygons, the evacuation route, the worker's own pulsing position marker, a static safe-point marker, and a small Copernicus risk badge in the corner. Under the map sits a forecast card with the active site, the risk class with a progress bar, and three mini-driver chips. Two action tiles at the bottom trigger Test Man-Down and Mark Safe.

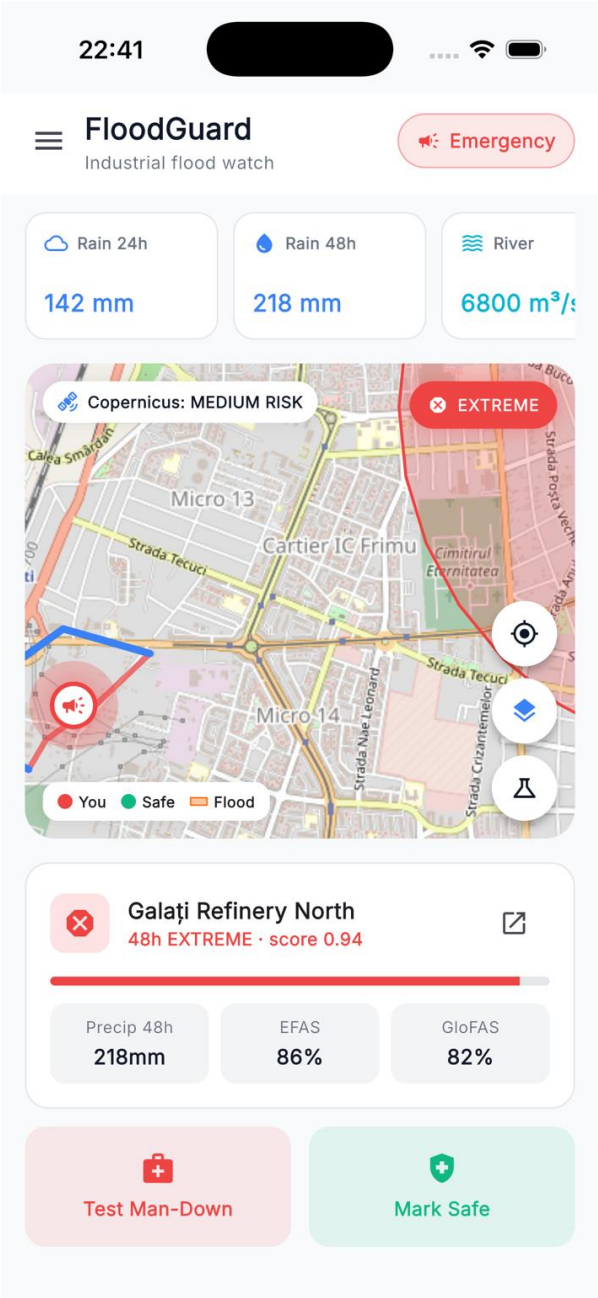


Fig. 2. Dashboard screen during an active high-risk alert

The site list screen carries a portfolio header at the top with Extreme, High, Medium, and Calm counters, followed by a list that refreshes itself on every forecast:updated event. Tap a site and the site forecast screen opens. Its hero card has a gradient background, a circular icon, the risk class spelled out in large bold letters, and a horizontal score bar. The icon pulses with a sine-wave animation when the class is high or extreme. Underneath, a six-tile grid shows the drivers behind the score: 24-hour and 48-hour precipitation, EFAS and GloFAS probabilities, the river-discharge maximum, and the baseline. A map card at the bottom shows the inundation polygon.

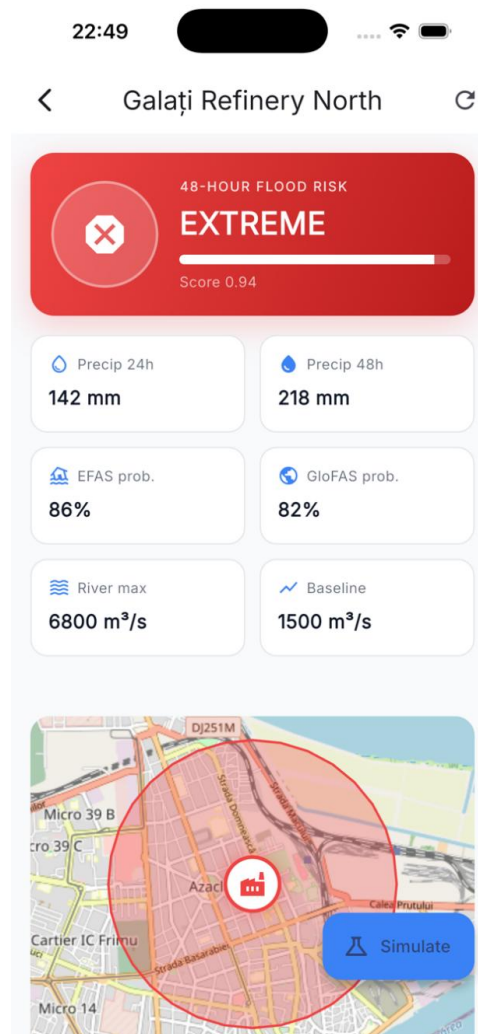


Fig. 3. Site forecast screen with extreme-risk hero card and predicted inundation

The man-down procedure runs entirely on the client. A movement watchdog tracks the GPS during an active evacuation and trips if no movement is detected for sixteen seconds. The screen turns red, a thirty-second countdown starts, and audio plays through `flutter_tts`. Tapping I'M FINE marks the alert as accidental. If the countdown lapses, the SOS is committed with the worker's GPS coordinates and mobility profile, and broadcast as `alert:mobile_emergency`.

For push delivery we picked the WebSocket stream plus `flutter_local_notifications` instead of Firebase Cloud Messaging. When a `forecast:updated` or `alert:new` comes in, the client fires a system tray notification through the local plugin. The downside is obvious: nothing arrives if the OS has killed the app. For an industrial product where the app stays open across a worker's whole shift, that trade-off was acceptable. `ntfy.sh` is on the shortlist for the killed-app case.

7. Demonstration and Validation

Real floods do not show up on demand. So we built a controlled simulation surface that drops synthetic forecast snapshots into the same database table the live ones use, and travels the exact same WebSocket broadcast path. `POST /api/sites/{site_id}/simulate?level=...` accepts low, medium, high, extreme, and reset. The first four each pick a preset combination of risk score, class, precipitation, and probabilities. The synthetic snapshot is committed and the same broadcasts go out as in real life. Reset

clears the synthetic state and re-runs the live pipeline.

On the client this is exposed through a demo controls screen with a site dropdown, a five-tile risk grid, and Trigger Man-Down and Mark Safe buttons. Each tap goes through in a single network round trip, and system tray notifications fire for whichever risk classes are configured to alert. We deliberately did not gate the simulation behind a feature flag. Safety managers use it for worker training, administrators use it to verify push tokens are actually registered, and every simulation leaves auditable rows in the alerts and history tables.

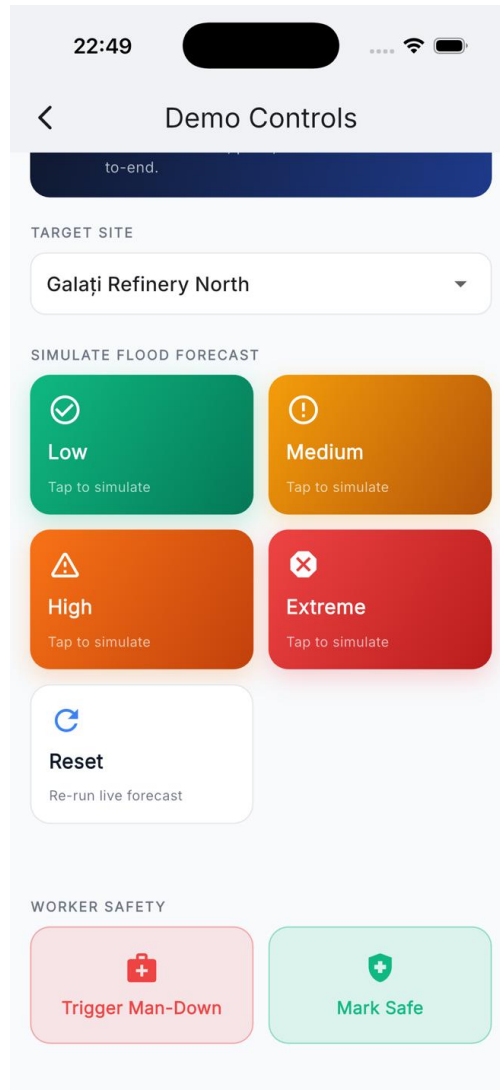


Fig. 4. Demo controls screen used for end-to-end validation

8. Conclusions

FloodGuard takes four free European data sources, mixes them into a per-site 48-hour flood risk forecast, and pushes the result to industrial workers through a Flutter client with a risk dashboard, predicted inundation polygons, and a man-down SOS. A few takeaways from building it. The fusion algorithm shows that precipitation, gridded flood probability, and river discharge can be combined into a tunable per-site score using only public data. Keeping the same risk-class palette across every screen of the Flutter interface made the UI noticeably easier to reason about during development. The man-down procedure sits on top of the forecast rather than next to it, and that is the part we are happiest with. The simulation surface lets us validate the system end to end and doubles as a training tool. And running

notifications through the WebSocket plus local notifications saved us from dragging in Firebase.

Plenty is left to do. The inundation polygon generator should be replaced with a proper EU-DEM lookup. A killed-app push path through ntfy.sh would close the obvious gap in delivery. The scheduler needs a Redis-coordinated cluster setup before it ever sees more than one worker. And the Flutter client should be offline-first rather than assuming the WebSocket is always there. With those four pieces in place, FloodGuard has a credible path from a student project to something industrial safety teams could actually run, without paying anyone for a licence.

9. Bibliography

- [1]. European Environment Agency (2024), Economic losses from weather- and climate-related extremes in Europe, EEA Briefing, Copenhagen, <https://www.eea.europa.eu/>.
- [2]. Thielen, J., Bartholmes, J., Ramos, M.-H., de Roo, A. (2009), "The European Flood Alert System – Part 1: Concept and development", *Hydrology and Earth System Sciences*, 13 (2), 125–140, ISSN 1027-5606.
- [3]. Alfieri, L., Burek, P., Dutra, E., Krzeminski, B., Muraro, D., Thielen, J., Pappenberger, F. (2013), "GloFAS – global ensemble streamflow forecasting and flood early warning", *Hydrology and Earth System Sciences*, 17 (3), 1161–1175, ISSN 1027-5606.
- [4]. European Commission (2024), Copernicus Emergency Management Service – European Flood Awareness System, online documentation, <https://www.efas.eu/>.
- [5]. Schumann, G. J.-P., Moller, D. K. (2015), "Microwave remote sensing of flood inundation", *Physics and Chemistry of the Earth*, 83-84, 84-95, ISSN 1474-7065.
- [6]. Open-Meteo (2024), Free Weather and Flood Forecast API, developer documentation, <https://open-meteo.com/>.
- [7]. ESA Copernicus (2024), Copernicus Data Space Ecosystem and Sentinel Hub APIs, online documentation, <https://dataspace.copernicus.eu/>.
- [8]. Flutter (2024), Flutter SDK Documentation, Google LLC, <https://docs.flutter.dev/>.

10. Notations

Abbreviations used in this paper:

API = Application Programming Interface;
CEMS = Copernicus Emergency Management Service;
DEM = Digital Elevation Model;
EEA = European Environment Agency;
EFAS = European Flood Awareness System;
FCM = Firebase Cloud Messaging;
GloFAS = Global Flood Awareness System;
GPS = Global Positioning System;
REST = REpresentational State Transfer;
SAR = Synthetic Aperture Radar;
SOS = distress signal used by the man-down procedure;
VV = vertical-transmit, vertical-receive radar polarisation;
WMS = Web Map Service;
WebSocket = bidirectional real-time communication protocol.

Suki – Integrating a camera into the operation of a dog-like robot

ACSINTE Ana-Maria¹, ZAMFIROIU Mihai-Alexandru², POSTELNICU Andrei³, TOMESCU Vlad-Costin⁴ and IVAN Andrei Mario (Scientific Supervisor)

¹Faculty of Industrial Engineering and Robotics, Specialization: Robotics (Romanian), Year: III, email: ana_maria.acsinte@stud.fiir.upb.ro

²Faculty of Industrial Engineering and Robotics, Specialization: Robotics (Romanian), Year: III

³Faculty of Industrial Engineering and Robotics, Specialization: Robotics (English), Year: III

⁴Faculty of Industrial Engineering and Robotics, Specialization: Robotics (Romanian), Year: II

ABSTRACT: *This paper presents the architecture, hardware, and software of a real-time computer-vision system designed to be mounted on an autonomous quadruped robot named Suki. The system continuously analyses a live video feed from an onboard camera, detects relevant objects in the environment (pedestrians, vehicles, obstacles, and traffic lights), classifies the state of any detected traffic light (red, yellow, green), and converts these perceptions into motion commands that will eventually be sent to the robot's motor controller. The video feed, annotated with detection boxes, is broadcast over Wi-Fi as an MJPEG stream so that the operator can monitor the robot in real time from a smartphone or laptop browser. The current development phase implements and validates the full perception and streaming pipeline on a Raspberry Pi 4. The detection module is based on a YOLO neural network exported to the NCNN inference framework, while a complementary HSV-based classifier identifies the active traffic-light state. The paper details the design rationale, software architecture, and implementation, demonstrating how a low-cost embedded platform can support reliable robotic perception without dedicated GPU hardware.*

KEYWORDS: *dog robot, camera, object detection, traffic light, YOLO, NCNN, Raspberry Pi, embedded vision*

1. Introduction

The evolution of autonomous robots is one of the most active research fields today, playing a central role in navigating complex environments for both information processing and human interaction. Their real-world usefulness ultimately depends on the ability to receive and transmit information in near real time, and to translate raw sensory data into safe behaviour.

Suki is a dog-like robot designed to recognize external factors such as obstacles, traffic lights, cars and pedestrians, with the purpose of assisting visually impaired individuals during outdoor navigation. To achieve this, we have integrated a YOLO (You Only Look Once) object detector and built a complementary visual-assistance layer that highlights different object categories with colour-coded bounding boxes (red for traffic lights, yellow for pedestrians, blue for vehicles), making the scene easier to interpret on the supervisory screen.

The implementation was developed entirely in Python, using OpenCV for image processing, Flask for the web monitoring interface, PySerial for the future motor link, and Ultralytics YOLO for the neural network. The proposed system demonstrates that reliable real-time robotic perception can be achieved using commercially available low-cost hardware without dedicated GPU accelerators. Furthermore, its modular architecture allows future expansion toward fully autonomous navigation, obstacle avoidance, distance estimation, and intelligent behavioural control.

2. Current State of Art

The current implementation of Suki is based on a lightweight embedded perception architecture optimized for real-time execution on ARM-based hardware. The system uses a Raspberry Pi 4 (8 GB RAM) as the central processing unit, together with a camera acquisition module and a YOLO Nano neural network exported through the NCNN inference framework. NCNN was specifically chosen because it is hand-tuned for ARM NEON SIMD instructions, which makes it substantially faster than a standard PyTorch deployment on the same hardware and reduces RAM consumption to a level compatible with the Raspberry Pi class.

The perception pipeline performs continuous frame acquisition and object inference using single-stage

detection, allowing simultaneous localization and classification of multiple object categories within a single forward pass through the network. Compared with classical two-stage detectors such as Faster R-CNN, this approach significantly reduces inference latency and enables stable real-time operation without dedicated GPU acceleration on the embedded ARM platform.

To improve environmental awareness, the detected objects are additionally processed through a custom visual-assistance layer based on color-coded annotations. Traffic lights, pedestrians, and vehicles are highlighted using distinct bounding-box colors in order to provide intuitive visual feedback for navigation assistance.

The current system architecture also integrates low-latency MJPEG streaming and serial communication interfaces, allowing remote monitoring and future integration with autonomous locomotion and obstacle-avoidance subsystems.

3. System architecture

The proposed system is organized as an embedded vision architecture mounted on a dog-like robotic platform. The main processing unit is a Raspberry Pi 4, which receives visual data from an onboard camera and processes it locally, without relying on an external computer or cloud server. This design reduces communication delay and allows the robot to react faster to elements detected in the surrounding environment.

The architecture follows a classical robotic structure: image acquisition, object detection, visual interpretation, remote monitoring, and future motion control. The camera continuously captures frames from the environment, while the YOLO-based detection module identifies relevant objects such as traffic lights, pedestrians, and vehicles. The detected objects are then displayed using a color-coded bounding-box system, where each object category is visually separated for easier interpretation. The system also includes a web-based monitoring layer implemented through Flask. This layer allows the processed camera stream to be accessed from a smartphone or laptop browser. In the current stage, this interface is useful for testing, validation, and manual supervision. In the future, the same architecture can be connected to the robot's locomotion controller through serial communication.

Table 1. Main hardware and software components

Component	Technical role
Raspberry Pi 4	Main embedded processing unit
Camera module	Real-time image acquisition
YOLO model	Object detection and classification
OpenCV	Frame processing and image annotation
Flask	Remote video streaming interface
PySerial	Communication with future motor controller
Python	Main programming language used for integration

4. Perception and Detection Pipeline

The perception pipeline represents the main functional module of the robot. Its purpose is to transform raw camera frames into useful information for navigation assistance. Each frame captured by the camera is passed through the YOLO detection model, which identifies objects and returns their position in the image, object class, and confidence score.

The YOLO family of detectors was selected because it performs object localization and classification in a single forward pass through the network. The architecture is composed of three logical parts: a convolutional backbone that extracts multi-scale features, a neck (typically a feature pyramid network) that fuses information across scales so that both small and large objects can be detected, and a detection head that emits bounding boxes and class probabilities. A non-maximum suppression step then filters out duplicate detections of the same object. This single-pass design is what allows real-time inference on a Raspberry Pi without GPU support. The model is pre-trained on the COCO dataset (Common Objects in Context, 80 categories), so person, car, traffic light and similar classes relevant to this application are available out of the box; in the COCO class index, the traffic light

corresponds to class 9.

After detection, the system applies visual annotations over the original image. Each detected object is marked using a bounding box and a label containing the object class and confidence value. The color-coding system improves readability: traffic lights are highlighted separately from pedestrians and vehicles, allowing the operator to quickly understand the scene.

For traffic-light interpretation, simple detection of the physical traffic-light object is not sufficient: the robot must also know which lamp is currently lit. A secondary classification step is therefore applied to determine the active state (red, yellow or green). The detected traffic-light region is cropped from the image and converted from BGR to HSV colour space, which separates hue from saturation and brightness and is therefore far more robust to lighting variation than raw RGB. Three pixel masks are then constructed, one for each colour band, using `cv2.inRange`. The red band requires two ranges (0–10 and 170–180) because red wraps around the hue circle in OpenCV. Whichever mask contains the largest number of non-zero pixels determines the reported state, provided it exceeds a minimum confidence threshold; otherwise the state is reported as unknown to avoid acting on spurious reflections.

Table 2. Detection classes used

Detected element	Purpose in the system and visual relevance
Traffic light	Determines possible stop/go conditions and it's critical for navigation
Person	Detects pedestrians in front of the robot and it's important for safety
Car / truck	Detects vehicles in the scene and it's useful for road awareness
Unknown object	Marks uncertain detections and it's used for debugging and improvement

The camera subsystem follows a producer–consumer pattern with two threads. A background producer thread opens the camera with OpenCV at 320×240 and forces the driver buffer to a single frame via `CAP_PROP_BUFFERSIZE = 1`. Without this, V4L2 queues several frames internally and the robot would react to obsolete images; the override guarantees that every read returns the most recent frame. The consumer is provided by Flask: when a browser requests the video feed endpoint, a generator reads the shared frame, encodes it to JPEG at 80% quality, and yields the bytes as parts of a multipart/x-mixed-replace HTTP response — the simplest way to stream live video to a browser without any client-side code.

The decision module converts perception into action through a deliberately simple rule set: a red or yellow light close to the robot triggers a stop; a green light with a clear path triggers a forward command; a person or obstacle near the centre of the frame overrides both and forces a stop. Proximity is estimated from the apparent size of the bounding box, a crude monocular depth proxy justified by the slow operating speed of the platform. The Raspberry Pi will communicate with the future motor controller (an Arduino-compatible microcontroller) over USB at 9600 baud via PySerial, sending one ASCII character per command (F = forward, B = backward, L = left, R = right, S = stop). The same commands are also exposed as HTTP endpoints by the Flask server, so the operator can drive the robot manually from a phone whenever the autonomous mode needs to be overridden. End-to-end latency was measured between the moment a frame is grabbed by the camera and the moment it appears, fully annotated, in the browser of a phone on the local Wi-Fi network. Table 3 summarises the average values measured during indoor tests at 320×240 resolution and JPEG quality 80%. The neural network inference dominates the budget, which confirms that the NCNN export was the correct optimisation target.

Table 3. Average latency contributions of the pipeline

Pipeline stage	Average time (ms)	Share of total
Camera grab (V4L2 + OpenCV)	8	~8%
YOLO inference (NCNN, nano model)	70	~70%
HSV classification of traffic-light crop	2	~2%

JPEG encoding (quality 80%)	6	~6%
Wi-Fi transfer to mobile client	14	~14%
Total (photon to display)	~100	100%

5. Experimental Results and Validation

The system was tested on several urban scenes containing traffic lights, pedestrians, cars, and road infrastructure. The objective of the validation stage was to verify whether the detection pipeline can correctly identify relevant objects in realistic environments, including crowded pedestrian areas and road intersections.

The first tests focused on isolated traffic-light images. These images allowed the validation of the traffic-light detection and state-classification module. The system successfully identified the traffic light region and classified the active signal as green, red, or yellow. This confirms that the combined YOLO and color-analysis approach is suitable for basic traffic-light interpretation.

Additional tests were performed using complex street images containing multiple object categories. In these cases, the system detected several pedestrians, vehicles, and traffic lights within the same frame. This demonstrates that the perception pipeline can operate in more realistic urban scenarios, where multiple objects appear at different distances and scales.



Fig. 1. Traffic lights detection

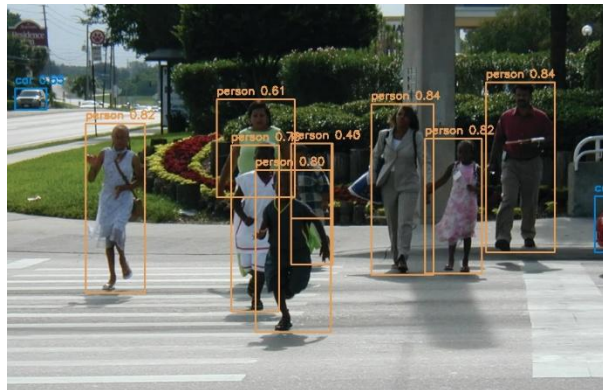


Fig. 2. Pedestrians detection



Fig. 3. Cars detection



Fig. 4. Overall detection

The results show that the system performs better when the target objects are clearly visible and sufficiently large in the image. Smaller or distant objects are still detected in some cases, but with lower confidence values. This behavior is expected for camera-based detection systems, especially when using a lightweight neural model optimized for embedded hardware.

The images containing pedestrians demonstrate that the system can identify several people in the same scene, even when they partially overlap or appear at different distances. This is important for the intended application, because pedestrian awareness is one of the key requirements for a robot designed to assist visually impaired users.

The vehicle-detection tests also show that the model can recognize cars and trucks in road environments. This capability may later be used to improve the robot's situational awareness near streets, crossings, or parking areas.

6. Limitations and Future Development

Although the current implementation provides functional real-time detection, several limitations were observed during testing. The detection accuracy depends on image quality, object size, lighting conditions, and camera angle. Distant traffic lights or partially visible objects may generate lower confidence values or uncertain classifications. The HSV classifier, while robust, can also be fooled by strong sunlight reflections or by the orange tint of high-pressure sodium street lights at night, so the colour bands will need to be re-tuned during outdoor field tests.

Another limitation is related to distance estimation. At the current stage, the robot detects the presence and position of objects in the image, but it does not yet calculate the real physical distance between the robot and the detected object. For future autonomous navigation, this function will be necessary in order to decide when the robot must stop, slow down, or change direction.

Future development will focus on integrating the perception system with the quadruped locomotion controller. The detected objects will be converted into movement decisions, such as stopping at a red traffic light, avoiding pedestrians, or continuing movement when the path is clear. Additional sensors, such as ultrasonic sensors, depth cameras, or LiDAR modules, may also be added to improve distance estimation and obstacle avoidance.

The modular structure of the system allows further improvements without redesigning the entire architecture. The YOLO model can be fine-tuned using custom images, the camera resolution can be improved, and the decision module can be extended with more advanced navigation logic.

7. Conclusions

This paper presented the development of an embedded vision system for Suki, a dog-like robotic platform intended to assist visually impaired individuals. The system uses a Raspberry Pi 4, a camera module, and a YOLO-based detection model to identify important environmental elements such as traffic lights, pedestrians, and vehicles.

The experimental results show that the proposed architecture can detect multiple object categories in real time and display them using clear visual annotations. The traffic-light detection module also demonstrates the ability to classify red, yellow, and green signals, which represents an important function for future assisted navigation.

The project confirms that a low-cost embedded platform can support real-time robotic perception without requiring dedicated GPU hardware. The current implementation represents a functional base for future development, including autonomous motion control, obstacle avoidance, distance estimation, and intelligent navigation behavior.

8. Bibliography

- [1]. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016), “You Only Look Once: Unified, Real-Time Object Detection”, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas;
- [2]. Ultralytics (2024), YOLO11 Documentation, <https://docs.ultralytics.com>;
- [3]. Lin, T.-Y., Maire, M., Belongie, S. et al. (2014), “Microsoft COCO: Common Objects in Context”, Proceedings of European Conference on Computer Vision (ECCV), Zurich;
- [4]. Tencent (2024), NCNN – High-Performance Neural Network Inference Framework Optimized for Mobile Platforms, GitHub repository, <https://github.com/Tencent/ncnn>;
- [5]. OpenCV team (2024), OpenCV-Python Tutorials: Changing Colorspaces (HSV), <https://docs.opencv.org>;
- [6]. Bradski, G. (2000), “The OpenCV Library”, Dr. Dobb’s Journal of Software Tools;
- [7]. Pallets Projects (2024), Flask Documentation, <https://flask.palletsprojects.com>;
- [8]. Mozilla Developer Network (2024), Multipart/x-mixed-replace and HTTP streaming, <https://developer.mozilla.org>;
- [9]. Raspberry Pi Foundation (2023), Raspberry Pi 4 Model B Product Brief.

B.I.L.U.T.A. - Acoustic and visual scanning system for omnidirectional ping-pong ball launching

SAS David-Mihnea, STRUGARIU Eduard-Adrian and Scientific Leader **ș. I. dr. ing. Andrei Mario IVAN**, RSP department

Faculty: FIIR, Specialization: Robotică, Year of study: I, e-mail: davidsasmihnea@gmail.com

SUMMARY: This article explores the development of an automated projectile turret, starting its journey from a conceptual autonomous shooter to a training tool for table tennis players. The turret has two operating modes: Autonomous (using an AI camera) and Manual (using a Joystick).

KEYWORDS: Autonomous, Machine Learning, Shooter, Object Detection

1. Introduction

This initiative began as a PSRE project featuring a turret that used microphones and ultrasound to find targets. We have since upgraded the system, replacing the ultrasound sensor with an ESP32-Cam and Machine Learning for more precise aiming. We also added a manual mode to make testing easier. We call it B.I.L.U.T.A, meaning Binaural Intelligence & Location Unit – Tactical Assassin.

2. Current progress

The project is now fully functional, with room for future improvements. Controlled by an Arduino UNO, the turret rotates using a servo motor and a 3:2 gear transmission. To fire, a micro-servo releases a ball from the chamber in between two DC motors spinning at 2200 RPM, which propel the ball forward. In the autonomous mode, the turret rotates until the camera finds a target to shoot. In Manual Mode we use a joystick to rotate the turret and start the shooter.

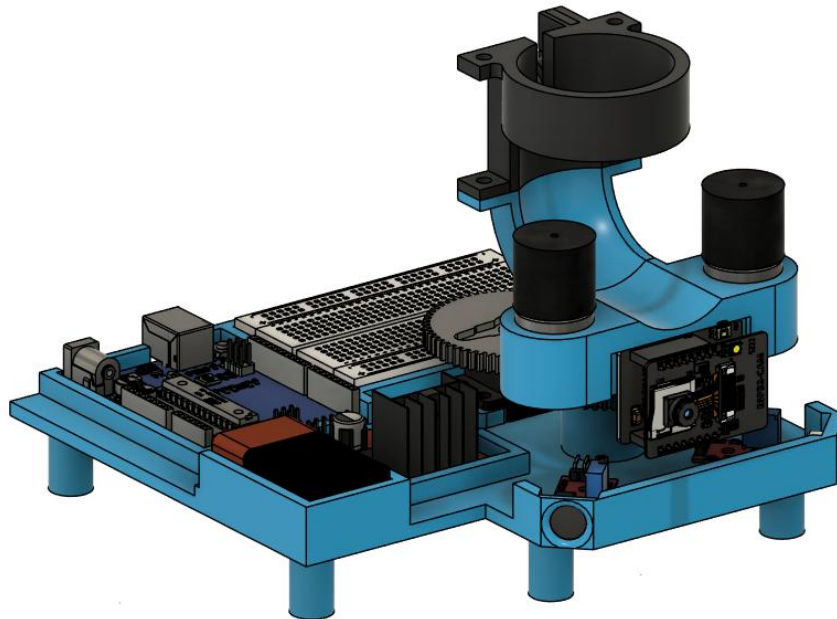


Fig. 1. B.I.L.U.T.A.

3. 3D Design

The device consists of four parts: the base, rotation system, shooter, and cage. The base holds the electronics and supports the assembly. In the front, two microphones are set at 45-degree angles. Each component fits into a specific spot to prevent movement or backlash. The subsystems stay together using screws, friction, and gravity. All the custom parts are 3d printed with PLA for cost efficiency. To ensure stability, the base has 6 legs, in each corner.

4. Turret Rotation System

A MG995 servo handles the turret's rotation via a 3:2 gear transmission (se Fig. 2.). The drive gear is attached to the servo, while the driven gear is designed into the turret. To keep costs low, we used a copper coupling as the primary mounting point for the turret assembly.

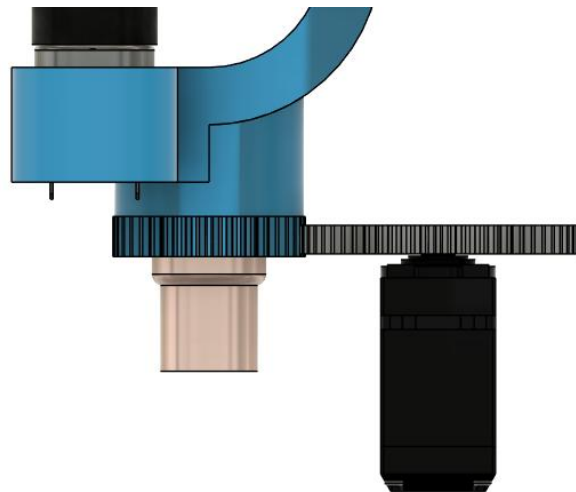


Fig. 2. Rotation System

5. Shooting system

Our shooter uses two DC motors, a micro-servo, and a four-ball chamber. The micro-servo releases one ball at a time into the turret's firing tunnel. As the ball passes between the two DC motors spinning in opposite directions, it is propelled outward (as shown in Fig. 3.). To improve traction and prevent the balls from slipping, we added grip tape to the motors.

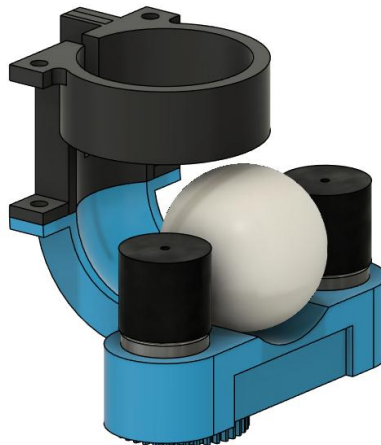


Fig. 3. Shooting system

6. User Interface

The system features a web interface that displays a live camera feed, the active operating mode, and a log of detected targets. Users can toggle between manual and autonomous modes by long-pressing a physical button.



Fig. 4. User Interface

7. Manual Mode

In Manual Mode, the user controls the turret using a KY-023 joystick module. Although the joystick supports dual-axis input, only the X-axis is utilized since the turret rotates on a single plane. The joystick's X-coordinate is mapped directly to the base servo to control rotation. Pressing the joystick's integrated switch initiates a three-second firing cycle.

8. Autonomous Mode

Autonomous mode uses the ESP32-CAM and two KY-037 microphones to find and shoot at targets automatically. The microphones listen for loud noises; when a sound is detected, the turret turns toward the noise to investigate. We used Edge Impulse to train a Machine Learning model to recognize three objects: a hand, a happy face, and an angry face. We programmed the turret to shoot only at the hand and the angry face, while ignoring the happy face. You can watch the camera's live "sight" through a web interface hosted by the ESP32.

9. Electrical Architecture

The electrical system is simple and modular, using an Arduino Uno connected to sensors via a breadboard and jumper wires (as shown in Fig. 5.). Since the sensors have low power requirements, they are powered directly from the board without fluctuations. To ensure the DC motors run at their full 6V capacity, we used a USB power connection instead of a 9V battery. Both the Arduino and the ESP32-CAM stay powered through USB cables connected to a laptop for stability during testing.

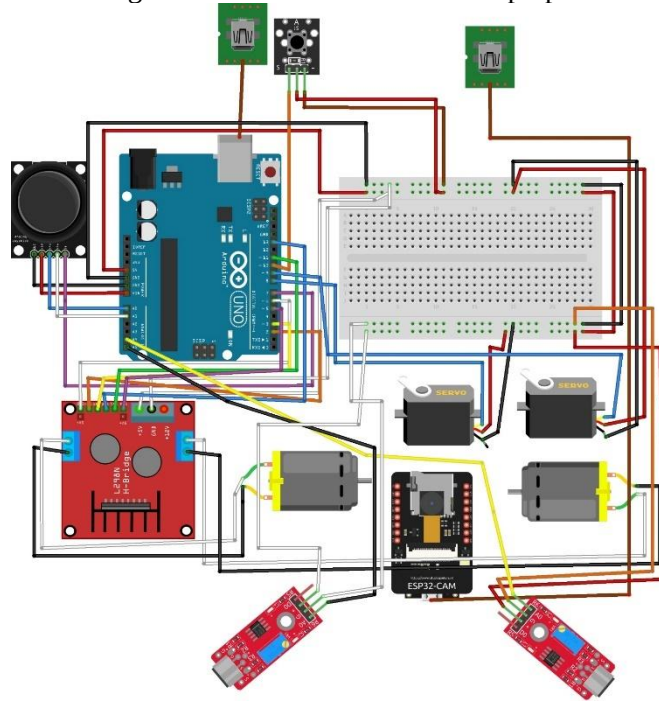


Fig. 5. Electrical Architecture

8. Conclusions

In conclusion, the B.I.L.U.T.A. project demonstrates how simple hardware can be transformed into an intelligent robotic system. By combining 3D printing, acoustic sensors, and Machine Learning, we successfully built a turret capable of both manual operation and autonomous target recognition. The transition from basic sensors to the ESP32-CAM highlights the project's growth into a functional tool for both robotics study and athletic training. While the current prototype is fully operational, it serves as a strong foundation for future improvements in speed, capacity, and precision.

9. Bibliography

[1]. https://docs.espressif.com/projects/arduino-esp32/en/latest/getting_started.html

[2]. <https://docs.edgeimpulse.com/>

[3]. <https://docs.arduino.cc/hardware/uno-rev3/>

BARTLEY – MOBILE ROBOT FOR DISABILITY ASSISTANCE BASED ON COMPUTER VISION

STAN Ioana -Teodora, POPA Andrei – Marius and Scientific Leader ș.l. dr. ing. **Andrei Mario IVAN**, RSP department

Faculty of Industrial Engineering and Robotics, Specialization: Robotics, Years of study: I, e-mail: ioana.teodora2017@yahoo.com

ABSTRACT: *Bartley is an innovative mobile robot designed for personal assistance, equipped with a 4WD propulsion system consisting of four motors with reducers and encoders, ensuring stable and precise movement. The platform integrates an advanced computer vision system using a Pi camera module and a Raspberry Pi 5 processing unit, which coordinates the robot's autonomous behavior. Upon detecting the user, Bartley activates its tracking function, maintaining an optimal distance for assistance; simultaneously, the system can interpret voice commands via a USB microphone. Designed with a dedicated compartment for object storage, the robot can improve daily activities such as shopping.*

KEYWORDS: *mobile robotics, computer vision, Raspberry Pi 5, disability assistance, autonomous navigation*

1. Introduction

Bartley is a four-wheeled robotic prototype developed with the goal of assisting people with disabilities in daily tasks, while incorporating intelligent behavior through voice recognition. This project combines mechanical design, control systems, and basic artificial intelligence to create a robot capable of both mobility and human interaction. The primary aim of Bartley is to explore how robotic systems can respond autonomously to visual and auditory stimuli, specifically by identifying the user, tracking them, and adjusting the movement accordingly.

2. Current state

In recent years, technological progress has turned robotics into an essential tool for social assistance, bringing new solutions for people with disabilities. For these users, simple activities such as shopping or lifting weights can be major barriers to the smooth running of their daily lives. Therefore, the integration of robots into everyday life is fundamental to restoring their independence. However, obstacles such as high costs, data protection, and the customization of solutions still exist.

Currently, personal assistance robots have evolved from simple teleoperated platforms to complex autonomous systems. Most solutions existing on the market focus either on automatic control or on limited social interaction. Recent research highlights the efficiency of using Raspberry Pi platforms for processing computer vision algorithms in assistive robots. Furthermore, the integration of Robot Operating Systems (ROS) facilitates the development of complex modules for autonomous navigation and human-robot interaction. Specifically, recent studies such as those by Rakibul Hasan and his colleagues illustrate how monocular vision systems can be optimized to identify and follow human targets in real-time, ensuring a

constant and safe distance. This approach proves that advanced tracking can be achieved with accessible hardware [1].

Additionally, the use of Robot Operating Systems (ROS) enables the creation of high-level software layers, which are essential for achieving both reliable autonomy and intuitive interaction between the robot and its user. In this context, the development of specialized interfaces is crucial; for instance, research on robotic assistants for senior citizens shows that voice-controlled systems powered by Raspberry Pi significantly improve accessibility and user experience for people with limited mobility [2]. By merging these two key functionalities, visual recognition and voice commands, prototypes like Bartley aim to transform complex robotic systems into practical tools that enhance the independence of users in their own homes.

3. Development status:

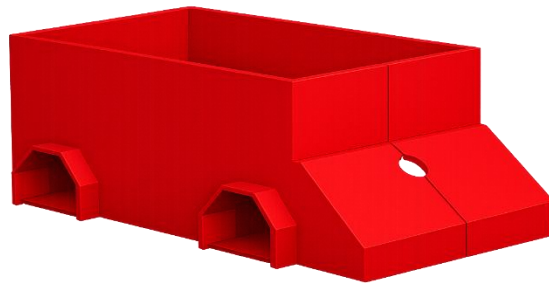
The development of the Bartley prototype involved following clear stages, from digital modeling to writing the control algorithms. The objective was to create an integrated system where the mechanical design efficiently supports the electronic components:

- **Computer-Aided Design (CAD):** To construct the chassis, we used the Onshape platform. This allowed us to simulate the positioning of the components ensuring sufficient space for wiring and the necessary airflow for cooling the processor. Additionally, the design features an upper compartment sized to allow for the easy loading of household objects. (see fig.1 and fig.2)
- **Software Architecture and Control Logic:** The operating system used is a Linux distribution optimized for Raspberry Pi 5; the robot's entire code is written in Python 3.15.3 and C++. The control logic integrates computer vision algorithms for user recognition and tracking, correlated with the processing of signals received from the USB microphone for voice command interpretation. This software architecture allows the robot to coordinate the movement of the four motors autonomously, adapting its trajectory and behavior based on the visual and auditory stimuli received from the surrounding environment
- **Used parts:** The central element of the robot is the Raspberry Pi OS 13 (Trixie) development board (8GB RAM), which acts as the central processing unit. This model was chosen for its ability to simultaneously run computer vision and voice processing algorithms. For the 'senses,' a Raspberry Pi Camera Module V3 was used for visual detection, alongside a USB microphone for capturing voice commands. The locomotion system consists of four DC motors equipped with reducers and magnetic encoders. These are controlled via an BTS7960 power driver (H-Bridge), which allows for speed and direction adjustment for each wheel. The power supply for the entire assembly is provided by a Samsung SDI 2L27 battery pack, capable of delivering the necessary current for both the motors and the logic components.

4. Project development and results:

Following assembly and calibration, the robot features the following technical specifications:

- **Dimensions:** $500 \times 310 \times 165$ mm;
- **Maximum Speed:** 0.7 m/s, optimized for safety in indoor environments;
- **Reaction Time:** The system processes the video stream and adjusts motor movement with an average latency of 45ms, enabled by the high computing power of the Raspberry Pi 5.



The program logic is structured on a multi-threading model, allowing the vision modules and motor control modules to run simultaneously:

As you can see in fig. 4., the code acts as a smart filter for the robot's hearing. Since the name 'Bartley' can be mispronounced, we added similar-sounding names like 'Barkley' or 'Barclay' so the robot doesn't get confused. We also gave it a short list of commands (stop, follow, follow me) to make it react faster and avoid mistakes from background noise.

The code shown below, in fig. 5., is designed to detect the user. It uses a camera that



runs at 30 frames per second, so the robot always knows exactly where the person is. On top of that, the camera can even track specific body parts, like the nose, eyes, ears, and joints.

6. Conclusion

The Bartley project successfully demonstrates that a high-performance assistive robot can be built using accessible components. By integrating Raspberry Pi 5 with multi-threading logic, we achieved a stable system capable of simultaneous person-following and voice command processing with a medium latency of 45ms.

Fig. 5. The code for the detection

The current prototype proves that the mechanical design effectively supports the electronics, allowing for the safe transport of household objects at a speed of 0.7 m/s. However, the testing phase highlighted areas for improvement, particularly regarding navigation in cluttered spaces and the precision of sensor fusion.

Based on the experience gained during this project, the next stage of development will involve:

- Building a second-generation robot with an optimized C++ architecture to further reduce reaction times and improve fluidity.
- Integrating LiDAR or ultrasonic sensors for active obstacle avoidance.
- Developing custom machine learning models for gesture recognition to complement the existing voice control.

7. Bibliography

[1]. Rakibul Hasan, Jahedul Islam, Rashed Md. Murad Hasan and Adhri Nandini Paul (2016), “Design of a Vision Based Person Following Robot”, *Proceedings, 2019 5th International Conference on Artificial Intelligence and Communication Technology (ICAICT), Chittagong university of Engineering and technology, Bangladesh, 16-17 September.*

```
# Profilele modelelor predefinite, cu setari recomandate pentru detectie rapida pe Hailo 8L.
MODEL_PROFILES = {
    "pose": {
        "title": "YOLOv8s pose - schelet, nas, ochi, urechi, articulatii",
        "lores": 640,
        "section": "hailo_yolo_pose",
        "hef_file_8L": MODELS_DIR / "yolov8s_pose_h8l_pi.hef",
    },
    "personface": {
        "title": "YOLOv5s person/face - recomandat pentru 30 FPS",
        "lores": 640,
        "section": "hailo_yolo_inference",
        "hef_file": MODELS_DIR / "yolov5s_personface_h8l.hef",
        "hailopp_config_file": MODELS_DIR / "yolov5_personface.json",
        "max_detections": 8,
        "threshold": 0.50,
    },
    "yolov6n": {
        "title": "YOLOv6n COCO - rapid, detecteaza mai multe clase",
        "lores": 640,
        "section": "hailo_yolo_inference",
        "hef_file_8L": MODELS_DIR / "yolov6n_h8l.hef",
        "max_detections": 20,
        "threshold": 0.40,
    },
    "yolov8s": {
        "title": "YOLOv8s COCO - mai bun, dar mai greu",
        "lores": 640,
        "section": "hailo_yolo_inference",
        "hef_file_8L": MODELS_DIR / "yolov8s_h8l.hef",
        "max_detections": 20,
        "threshold": 0.40,
    },
}
```

[2]. Prakash Kanade, Sunay Kanade (2020), “Raspberry Pi Project - Voice Controlled Robotic Assistant for Senior Citizens”, *International Research Journal of Engineering and Technology (IRJET), pages 1044-1049, ISSN: 2395-0056.*

SYNTHETIC DATA GENERATION ALGORITHM FOR OBJECT DETECTION IN THE ARM CHALLENGE COMPETITION

ROȘIORU Gabriel-Sebastian and ș. l. dr. ing. IVAN Andrei-Mario

Faculty: Industrial Engineering and Robotics, Specialization: Mechatronics and Robotics, Year of study: II, email: sebirosioru12@gmail.com

Abstract: Deep learning-based object detection models require massive, accurately annotated datasets to perform reliably in physical environments. For specialized competitions such as the ARM Challenge, acquiring and manually labeling thousands of images is a severe bottleneck. This paper presents a fully automated Synthetic Data Generation (SDG) pipeline utilizing NVIDIA Omniverse Replicator. By importing USD models of competition-specific objects, the algorithm utilizes extensive Domain Randomization—varying lighting, background textures, camera poses, and object scattering—to bridge the Sim-to-Real visual gap. Furthermore, a custom post-processing algorithm is introduced to extract raw 2D bounding boxes and semantic labels, autonomously converting them into YOLO and MATLAB training formats.

KEYWORDS: *Synthetic Data Generation, Omniverse Replicator, Object Detection, YOLO, Domain Randomization, ARM Challenge.*

1. Introduction

The ARM Challenge requires robots to visually identify, classify, and manipulate specific objects (colored cans, bottles, markers, scales, etc.) within a dynamic workspace. To achieve robust visual perception, modern robotic systems rely on Convolutional Neural Networks (CNNs) like YOLO (You Only Look Once).

The primary obstacle in training these networks is the "data hunger" inherent to deep learning. A highly accurate model requires thousands of diverse images, each manually annotated with bounding boxes and class labels. This manual process is notoriously slow, expensive, and prone to human error (imprecise bounding box edges). The objective of this research is to completely bypass manual annotation by developing an autonomous algorithmic pipeline. By leveraging a physics-based rendering engine, we can generate photo-realistic datasets where the ground-truth annotations are calculated mathematically by the engine, ensuring pixel-perfect accuracy at infinite scale.

2. Current State

Traditionally, robotic vision datasets are collected by photographing real objects from multiple angles and under various lighting conditions, followed by weeks of manual drawing of bounding boxes using tools like CVAT or Labellmg.

To accelerate this, the robotics industry has shifted towards Synthetic Data Generation (SDG). However, neural networks trained purely on pristine 3D renders often fail when deployed in the real world—a phenomenon known as the visual "Sim-to-Real gap" [3]. To combat this, modern algorithms employ Domain Randomization (DR) [2]. Instead of trying to perfectly simulate reality, DR forces the neural network to learn the essential geometry of the objects by exposing it to a massive, chaotic spectrum of randomized visual parameters (bizarre lighting, random backgrounds, and occlusions). If the network can detect a bottle under extreme, randomized conditions, it will easily detect it in a standard laboratory setting.

3. Simulation Environment and Pipeline Architecture

The SDG pipeline was built using Python and the NVIDIA Omniverse Replicator [1] extension within Isaac Sim. The process begins with importing exact 3D digital twins of the

SYNTHETIC DATA GENERATION ALGORITHM FOR OBJECT DETECTION IN THE ARM CHALLENGE COMPETITION

ARM Challenge objects, saved in the Universal Scene Description (USD) format.

To maximize the robustness of the dataset, several randomization graphs were registered within the Replicator orchestrator:

- **Lighting:** Both a global Dome light and a localized Sphere headlight are randomized in spatial orientation and intensity (ranging from 1,000 to 25,000 lumens).
- **Camera Pose:** The virtual camera's position is uniformly randomized within a defined 3D hemispherical boundary around the target zone.
- **Background Textures:** The background plane's material is dynamically altered using an omnipbr material. An array of heavily weighted color palettes (ranging from dark earthy tones to vibrant neon colors) and varying surface roughness parameters (0.7 to 1.0) are randomly applied to prevent the neural network from memorizing the background.

4. Data Generation Methodology

To ensure the neural network learns both to identify objects in isolation and to handle complex occlusions, the generator algorithm is split into two distinct rendering phases.

4.1. Phase 1: Scatter and Clutter Generation:

In the first phase, the algorithm generates 500 frames of highly cluttered environments. Using `rep.randomizer.scatter_2d`, a random assortment of 40 to 80 objects is dropped onto a simulated plane. The physics engine automatically calculates collisions, allowing objects to pile up, roll, and occlude one another naturally.

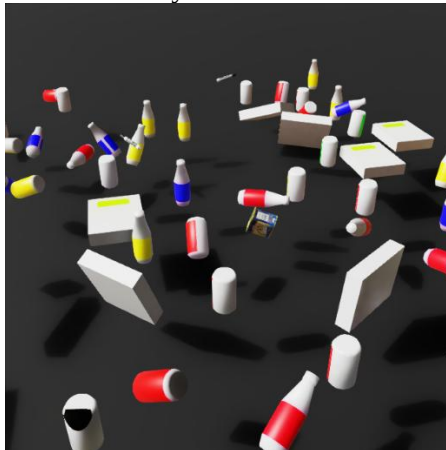


Fig. 1. Generated clutter



Fig. 2. Generated clutter

As seen in Figure 1 and Figure 2, the heavy occlusion forces the detection model to

SYNTHETIC DATA GENERATION ALGORITHM FOR OBJECT DETECTION IN THE ARM CHALLENGE COMPETITION

identify objects based on partial geometric features rather than relying on the entire silhouette.

4.2. Phase 2: Individual Close-Up Generation

To reinforce the core features of each specific class, the second phase isolates each USD object. For each of the competition assets, 100 individual frames are generated. The camera focuses directly on the object while the orchestrator rapidly cycles through extreme randomizations of scale, 3D rotation, and illumination as seen in the images below.



Fig. 3. Collage with individual close-up

SYNTHETIC DATA GENERATION ALGORITHM FOR OBJECT DETECTION IN THE ARM CHALLENGE COMPETITION

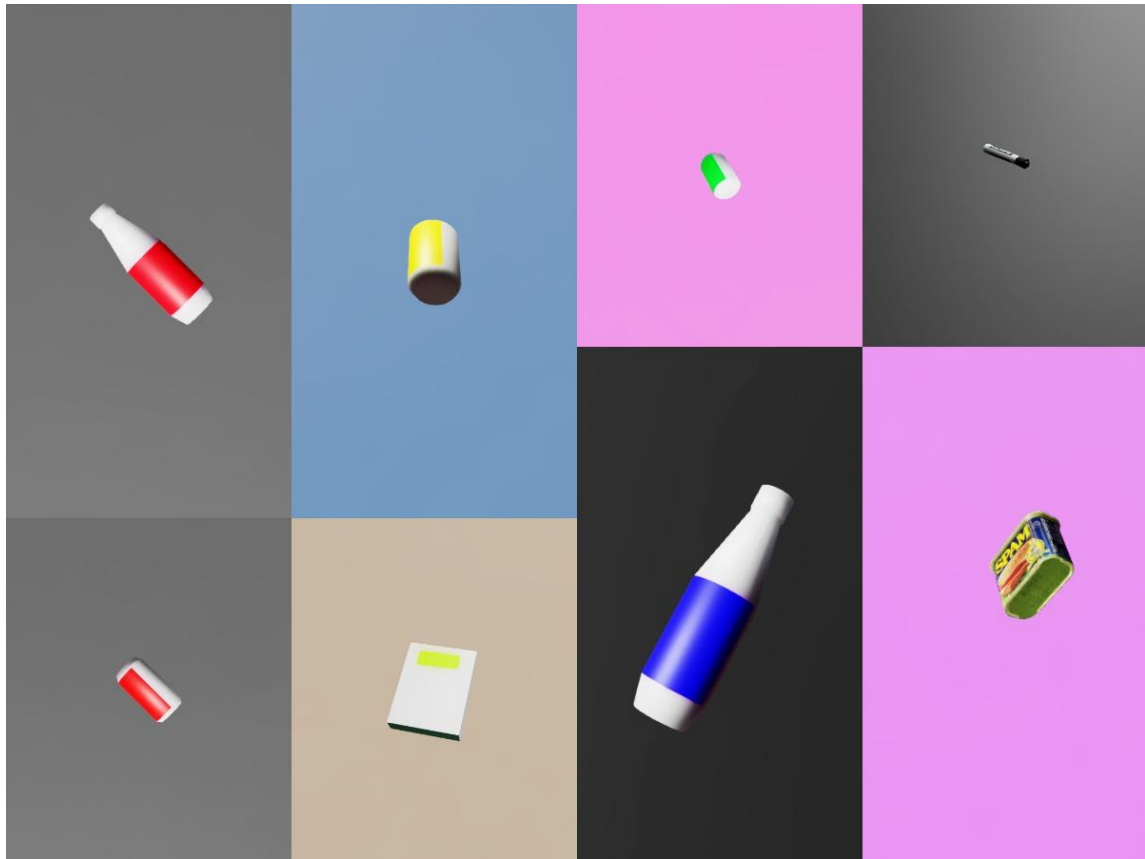


Fig. 4. Collage with individual close-up

Following these two phases, a consolidation function aggregates the generated files, prefixing filenames to prevent data collisions, resulting in a cohesive raw dataset containing thousands of images.

5. Automated Annotation and Formatting Pipeline

Omniverse Replicator outputs raw images alongside NumPy arrays (.npy) containing tight 2D bounding boxes and JSON files containing semantic ID mappings. Because standard AI training frameworks cannot read this raw output, a custom parser (Dataset_gen.py) was developed to autonomously construct training-ready datasets.

The parser scans the output directory and cross-references the bounding box coordinates with the semantic JSON file to filter out background elements. Depending on the selected OUTPUT_FORMAT, the algorithm executes specific mathematical transformations:

5.1. YOLO Format Conversion

For YOLO networks, the absolute pixel coordinates must be normalized relative to the image dimensions into a $[0, 1]$ range. The script calculates the normalized centre points and dimensions:



SYNTHETIC DATA GENERATION ALGORITHM FOR OBJECT DETECTION IN THE ARM CHALLENGE COMPETITION

The algorithm then generates individual .txt files for each image and autonomously writes the dataset.yaml metadata file, mapping the string class names (e.g., "redCan", "blueBottle") to integer IDs required by the YOLO framework.

5.2. MATLAB Format Conversion

For MATLAB compatibility, the script calculates the absolute bounding box width and height and structures the output into a single unified labels.csv file, formatted as [image_filename, class_name, x, y, width, height].

6. Future Improvements

While the current Synthetic Data Generation pipeline is highly automated and effective, several enhancements are planned for future iterations to further streamline the workflow and improve model performance:

- **Post-Processing Data Augmentation:** Beyond the 3D Domain Randomization applied in the rendering engine, integrating 2D data augmentation techniques (e.g., motion blur, Gaussian noise, and perspective distortion) directly into the Dataset_gen.py script will further harden the neural network against real-world camera artifacts and sensor noise.
- **Graphical User Interface (GUI):** To increase the accessibility of the tool for team members. This interface will allow users to intuitively configure generation parameters—such as the number of frames, background color palettes, and lighting extremes—without manually editing the underlying scripts.
- **Extended 3D Format Compatibility:** The current Replicator orchestrator relies exclusively on Universal Scene Description (USD) files. Future versions of the pipeline will incorporate an automated conversion module to natively ingest other widely used 3D CAD formats, such as OBJ, STL, and glTF, thereby broadening the pipeline's applicability to other robotic challenges.

7. Conclusion

The developed algorithmic pipeline successfully eliminates the most resource-intensive phase of robotic vision training: manual data annotation. By utilizing NVIDIA Omniverse Replicator, the pipeline generates thousands of pixel-perfect, highly randomized training images in a matter of minutes. The integration of heavy Domain Randomization ensures that the simulated data is varied enough to bridge the Sim-to-Real gap. Furthermore, the custom formatting script seamlessly bridges the gap between raw simulation data and modern deep learning frameworks.

8. References

- [1] NVIDIA Corporation (2023), Omniverse Replicator Documentation, developer.nvidia.com.
- [2] Tremblay, J., et al. (2018), "Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
- [3] Tobin, J., et al. (2017), "Domain randomization for transferring deep neural networks from simulation to the real world," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

9. Notations

The following variables and formulas are utilized within the automated annotation parsing algorithm:



= The absolute pixel coordinates of the top-left corner of the bounding box

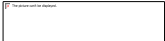
SYNTHETIC DATA GENERATION ALGORITHM FOR OBJECT DETECTION IN THE ARM CHALLENGE COMPETITION

 = The absolute pixel coordinates of the bottom-right corner of the bounding box

box

,  = The absolute dimensions of the generated image (e.g., 640x640 pixels)

pixels)

 = The normalized center coordinates of the bounding box (YOLO format)

INTEGRATION OF AN ARTICULATED NECK MECHANISM IN A HUMANOID ROBOTIC HEAD

POSTELNICU Andrei, MITICA Octavian-Alexandru, GANEA Victor and Scientific Leader:

IVAN Andrei Mario

Faculty: Industrial Engineering and Robotics, Specialization: Mechatronics and Robotics, Year of study: III, e-mail: OriginalAndreiposte@yahoo.com

***ABSTRACT** This paper presents the design and integration of an articulated neck mechanism within the PersonaAI humanoid robotic head. The proposed solution provides three-axis articulation — pitch, yaw and roll — by means of a cardan joint coupled to four high-torque digital servomotors. The mechanical assembly combines a 3D-printed PETG/PLA structure with a dual-bearing rotational base, ensuring low-friction operation and stable load distribution. The actuation chain is driven by a PCA9685 servo controller interfaced with a Raspberry Pi 5, allowing precise pulse-width modulation across all channels. The resulting system delivers a natural, expressive range of motion, suitable for use in therapeutic, educational and commercial human-robot interaction scenarios.*

KEY WORDS: Robotics, Mechanical ,Servos ,Humanoid, Neck

1. Introduction

PersonaAI is a humanoid robotic head (as seen in fig. 1.) designed to provide natural and expressive human-machine interaction, integrating real-time speech recognition, conversational artificial intelligence and coordinated facial motion. A defining element of any humanoid platform that aims to convey emotion and attention is the ability to articulate the head along multiple axes. In the absence of expressive head motion, gestures such as nodding, glancing or turning toward the interlocutor (fundamental components of nonverbal communication) cannot be reproduced credibly.

This paper focuses on the design and integration of the articulated neck mechanism of PersonaAI. The objectives addressed are: to provide three-axis articulation (pitch, yaw, roll) capable of supporting the mass of the head assembly; to ensure smooth, low-friction motion; to integrate the mechanism with the existing servo-driven control architecture of the robot; to maintain a modular construction that can be replicated or adapted to similar humanoid platforms.



Fig. 1. Robot's Current Appearance

2. Current state of the field

Recent developments in humanoid robotics have focused on combining conversational AI with expressive robotic platforms. Projects such as the InMoov initiative [1][2] and TonyGPT [3] demonstrate how language models can be integrated into humanoid systems capable of speech interaction and emotional expression.

Modern robotic assistants increasingly include features such as speech recognition, facial tracking, and computer vision to improve human-robot communication. PersonaAI follows this direction by combining conversational AI with realistic movement and visual interaction capabilities.

3. Proposed solution

3.1. Mechanical architecture

The neck assembly connects the head to the head of the robot through a cardan joint (as seen in fig. 2.), which allows independent rotation along two perpendicular horizontal axes. The central element is a 3D-printed PETG block, traversed by a threaded steel rod that runs through its middle. The rod ensures rigidity and provides a continuous load path between the cardan joint and the actuation system.

The base of the neck is mounted on a 3D-printed PLA platform fitted with a dual bearing system. This configuration ensures low-friction, smooth rotation around the vertical axis and prevents off-centre loads from inducing wobble in the head during dynamic movements. The choice of PETG for the highly loaded central element and PLA for the lower-stress platform reflects the trade-off between mechanical strength and ease of manufacturing characteristic of additive-manufactured components.



Fig. 2. Exploded view of the robot

3.2. Actuation system

Head articulation is produced by four high-torque DS1560 digital servomotors [4], each with a stall torque of approximately 60 kg*cm. They are arranged in two functional pairs. The first pair, attached via threaded rods, controls the up-and-down (pitch) and side-to-side (roll) motion of the head. The mechanical leverage provided by the rods amplifies the linear travel produced by the servomotors and distributes the load symmetrically between them, enabling

natural nodding and tilting gestures while ensuring stable support of the head mass. The second pair, integrated at the level of the rotational platform, controls the left-right (yaw). Working in coordinated motion, they reproduce the expressive turning gesture characteristic of human interaction.

Through this arrangement, the head can be moved reliably along all three rotational axes, achieving a full range of motion that enhances the perceived presence of the robot during interaction.

3.3. Design and manufacturing considerations

All structural elements of the neck assembly were produced through fused-deposition-modelling 3D printing, which allowed iterative refinement of the geometry throughout the development of the prototype. The decision to combine two distinct materials in the same sub-assembly is motivated by the local mechanical demands: the PETG block surrounding the cardan joint is subjected to bending loads transmitted by the threaded rod and therefore benefits from the higher impact resistance and inter-layer adhesion of PETG, while the lower bearing platform, which mainly distributes static loads, can be produced in the more rigid and easier-to-print PLA.

The PETG block is imbedded with threaded inserts, ensuring central positioning of the threaded rod. The head is also connected with ball joints extended by another pair of threaded rods that ensure the right height of the head when motionless. The choice of using threaded rods in this instance has been made by the fact that they can be adjusted ensuring the right head position. The same assembly can be found on the base of the robot, motors are connected to the pushing points of the neck with ball joints and threaded rods.

4. Control and power supply system

All four servomotors are controlled through a PCA9685 16-channel servo driver [5], which communicates with the Raspberry Pi 5 [6] over the I2C bus using the SDA and SCL lines. The PCA9685 generates the pulse-width modulation (PWM) signals required by each servomotor, providing angular resolution suitable for smooth, coordinated motion across all axes.

The driver is powered by a regulated 6 V rail produced by a step-down DC-DC converter [7] connected to an 11.1 V lithium-polymer battery. This separation prevents voltage drops on the digital electronics during simultaneous servomotor movement. The Raspberry Pi 5 itself is powered independently from a dedicated AC adapter, which guarantees stable operation of the high-level control software during demanding processing tasks.

At the software level, the four servomotor channels are addressed in parallel through the PCA9685 I2C interface, with target positions sent at a refresh rate of 50 Hz. Coordinated motion across the three axes is obtained by applying linear interpolation between consecutive set-points and by enforcing matched accelerations on the paired pitch and yaw / roll servomotors. This strategy avoids the small mechanical conflicts that can arise when the two servomotors of a pair receive set-points with slightly different timing, and it produces visibly smoother trajectories during composite gestures such as combined nod-and-turn movements.

5. Control and power supply system

The integration of the articulated neck within the Persona AI humanoid head provides three-axis motion through a compact cardan-based mechanism actuated by four high-torque DS1560 digital servomotors. The combined PETG/PLA mechanical structure with a dual-bearing rotational base ensures smooth, stable, low-friction articulation, while the PCA9685 driver enables coordinated PWM control from the Raspberry Pi 5.

Preliminary testing of the assembled prototype confirmed the suitability of the proposed configuration. The head reaches angular displacements of approximately $\pm 35^\circ$ on the pitch axis and $\pm 60^\circ$ on the yaw axis with no audible servomotor distress, and the dual-bearing platform showed no measurable backlash after several hours of cumulative operation.

The proposed solution supports the natural nodding, turning and tilting motions essential for expressive human-robot interaction, while keeping the actuation chain modular, low-cost and easy to reproduce. Future development will focus on enclosing the neck assembly and developing a smoother motion of the head when turning and nodding. Integration with the vision module currently under development is also intended, with the goal of enabling gaze-following behaviour, further increasing the naturalness of the interaction.

6. References

- [1]. EZ-inMoov Robot Head, <https://www.thingiverse.com/thing:6030262>
- [2]. InMoov project wikipedia page, https://en.wikipedia.org/wiki/InMoov?utm_source=chatgpt.com
- [3]. TonyGPT, https://www.roboticgizmos.com/tonygpt-humanoid-robot-chatgpt/?utm_source=chatgpt.com
- [4]. DS1560 Datasheet, <https://m.media-amazon.com/images/I/81EFGw8qkhL.pdf>
- [5]. PCA9685 Datasheet, <https://www.alldatasheet.com/datasheet-pdf/pdf/293576/NXP/PCA9685.html>
- [6]. Raspberry Pi 5 Datasheet, <https://datasheets.raspberrypi.com/rpi5/raspberry-pi-5-product-brief.pdf>
- [7]. LM2596S Datasheet, <https://www.alldatasheet.com/datasheet-pdf/pdf/543794/TI1/LM2596S.html>

USING AI ALGORITHMS TO SOLVE INVERSE KINEMATICS FOR XARM ARTICULATED ROBOTIC ARMS

ROȘIORU Gabriel-Sebastian and ș. l. dr. ing. IVAN Andrei-Mario

Faculty: Industrial Engineering and Robotics, Specialization: Mechatronics and Robotics, Year of study: II, email: sebirosioru12@gmail.com

Abstract: This paper explores the usage of Deep Reinforcement Learning algorithms, such as PPO, in solving the inverse kinematics and trajectory planning problem for the XARM articulated robotic arm. NVIDIA Isaac Lab was used to train a control policy in a highly randomized physics simulation. The resulting policy is deployed on the physical hardware via the ONNX runtime, evaluating both relative and absolute joint positioning.

KEYWORDS: Inverse Kinematics, XARM, Deep Reinforcement Learning, Isaac Lab, Sim-to-Real, ONNX.

1. Introduction

The primary purpose of this research is to demonstrate the advanced capabilities of Reinforcement Learning (RL) in robotics and to showcase what can be practically achieved using readily available commercial hardware. Traditionally, controlling articulated arms like the XARM series [1] requires solving the Inverse Kinematics (IK) problem to position the end-effector in 3D space. By replacing these rigid mathematical solvers with a Deep Reinforcement Learning (DRL) policy trained in NVIDIA Isaac Lab, this paper illustrates how modern AI can map Cartesian poses directly to optimal joint commands. Ultimately, this work highlights the accessibility of bridging the gap between simulated RL environments and real-world hardware, proving that complex, real-time trajectory planning is entirely feasible.

2. Current State

The conventional approaches to solving Inverse Kinematics are broadly divided into analytical and numerical methods. Analytical methods rely on rigid geometric equations specific to a robot's exact anatomy, while numerical methods iteratively converge on a solution. Both suffer catastrophic failures when approaching "singularities", physical alignments where the mathematical matrix becomes non-invertible, causing the algorithm to crash or output infinite joint velocities.

The Reinforcement Learning Paradigm: Unlike supervised learning, which requires millions of pre-calculated IK datasets, DRL allows the robot to learn kinematically viable paths through trial and error in a physics engine. Once the neural network policy is trained, inference takes only a fraction of a millisecond ($O(1)$ time complexity). Because the AI maps spatial coordinates directly to motor actions, it is completely immune to Jacobian singularities.

3. Simulation Environment and DRL Framework

The AI policy was trained using NVIDIA Isaac Lab [2], utilizing a Markov Decision Process (MDP) framework. The physical configuration of the XARM was imported via a USD (Universal Scene Description) file generated from a URDF file provided by Ufactory, intentionally locking Joint 4 to emulate a 5-DOF configuration matching that of the physical model.

3.1. Observation and Action Space

The policy's observation space (fig. 1) includes the current joint positions, joint velocities, the previous action taken, and the target end-effector pose command. To encourage robust learning, additive uniform noise is injected into the simulated joint position sensors (0.01 rad) and velocity sensors (0.03 rad/s).

USING AI ALGORITHMS TO SOLVE INVERSE KINEMATICS FOR XARM ARTICULATED ROBOTIC ARMS

```
class XArm5ObservationsCfg:
    """Observation specifications for the MDP."""

    @configClass
    class PolicyCfg(ObsGroup):
        """Observations for policy group."""

        joint_pos = ObsTerm(
            func=mdp.joint_pos,
            noise=Unoise(n_min=-0.01, n_max=0.01),
            params={"asset_cfg": SceneEntityCfg("robot", joint_names=["joint1", "joint2", "joint3", "joint5", "joint6"],
                                                preserve_order=True)}
        )
        joint_vel = ObsTerm(
            func=mdp.joint_vel,
            noise=Unoise(n_min=-0.03, n_max=0.03),
            params={"asset_cfg": SceneEntityCfg("robot", joint_names=["joint1", "joint2", "joint3", "joint5", "joint6"],
                                                preserve_order=True)}
        )
        pose_command = ObsTerm(func=mdp.generated_commands, params={"command_name": "ee_pose"})
        actions = ObsTerm(func=mdp.last_action)

        def __post_init__(self):
            self.enable_corruption = True
            self.concatenate_terms = True

# observation groups
policy: PolicyCfg = PolicyCfg()
```

Fig. 1. Observation space

For the action space (fig. 2), a `RelativeJointPositionActionCfg` was implemented. Instead of predicting absolute angles, the network outputs a normalized delta value within $[-1, 1]$. This is scaled by a factor of 0.0625 (approximately 3.6° per step), ensuring a highly stable and constrained movement profile during the physical deployment.

```
self.actions.arm_action = mdp.RelativeJointPositionActionCfg(
    asset_name="robot",
    joint_names=["joint1", "joint2", "joint3", "joint5", "joint6"],
    preserve_order=True,
    scale=0.0625,
    use_zero_offset=True,
)
```

Fig. 2. Action space

3.2. Reward Function Architecture

The reward architecture is designed to balance precision with smooth kinematics. The primary positive rewards are dense tracking errors comparing the end-effector's actual position and orientation to the target pose. To prevent erratic movements, strict penalty terms (negative weights) are applied for:

- Excessive action rates (weight = -0.01)
- High joint velocities (weight = -0.01)
- Violating hardware joint limits (weight = -0.02)
- A severe termination penalty (weight = -20.0) triggered by self-collision, monitored via an integrated simulated contact sensor.

4. Domain Randomization and Sim-to-Real Transfer

The most significant challenge in robotic AI is the "Sim-to-Real" gap, the discrepancy between ideal simulated physics and real-world mechanics. To ensure the policy trained in Isaac Lab could be deployed on the physical XARM without catastrophic failure, extensive Domain Randomization was employed [3].

During the simulation startup phase, the environment applies uniform probability distributions to alter the robot's physical properties:

- **Rigid Body Mass:** Randomized by 20% to simulate varying payloads and manufacturing

USING AI ALGORITHMS TO SOLVE INVERSE KINEMATICS FOR XARM ARTICULATED ROBOTIC ARMS

tolerances.

- **Actuator Gains:** Stiffness is scaled between 95% - 105% and damping between 90% - 110%.
- **Joint Friction:** An additive uniform friction ranging from 0.0 to 0.05 Nm is injected to replicate real-world mechanical wear and gear resistance.
- **Starting position:** Randomised joint position when the simulation starts.

By training the policy to succeed across a massive spectrum of randomized physical parameters, the resulting neural network becomes highly generalized and resilient to the unmodeled dynamics of the physical hardware.

5. Hardware Deployment and Safety Logic

The trained policy was exported as an ONNX (Open Neural Network Exchange) model and deployed on the physical XARM using the `xarm.wrapper.XArmAPI` in Python.

5.1. Control Modes and Absolute vs. Relative Positioning

Two modes of execution were evaluated, namely the Position Control Mode (Mode 0) and Servo Motion Mode (Mode 1). When moving relatively, the output of the network is limited within the range of $[-1.0, 1.0]$, which is then multiplied by the training parameter (0.0625) and added to the joint positions obtained from the robot's sensors.

5.2. Jitter Mitigation: Velocity Scaling and Stability Analysis

When deploying the relative positioning policy via the continuous streaming API, real-world sensor noise and rapid control loop execution can induce high-frequency jitter in the servos. To ensure safety and mechanical stability, a global velocity scaling parameter (`SPEED_FACTOR`) was implemented to modulate the raw delta actions outputted by the neural network before they are added to the target joint angles.

To quantify the effectiveness of this scaling factor on mechanical stability, joint velocity telemetry was recorded during physical spatial navigation under two different speed configurations.

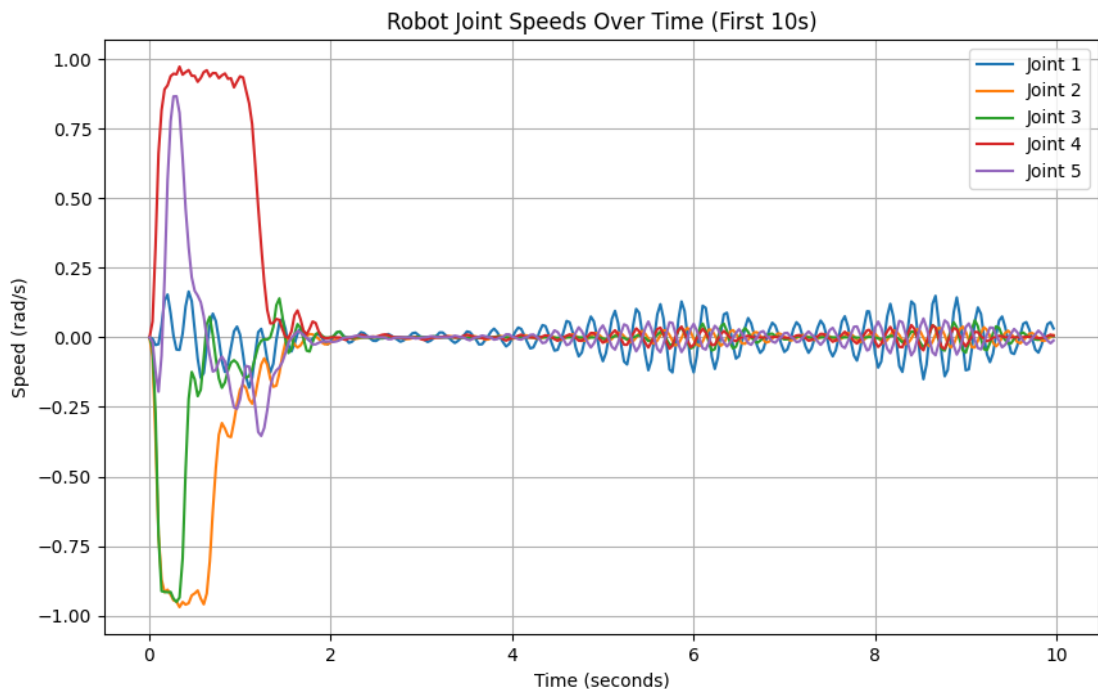


Fig. 3. Robot joint speeds over time using an aggressive speed multiplier (0.5)

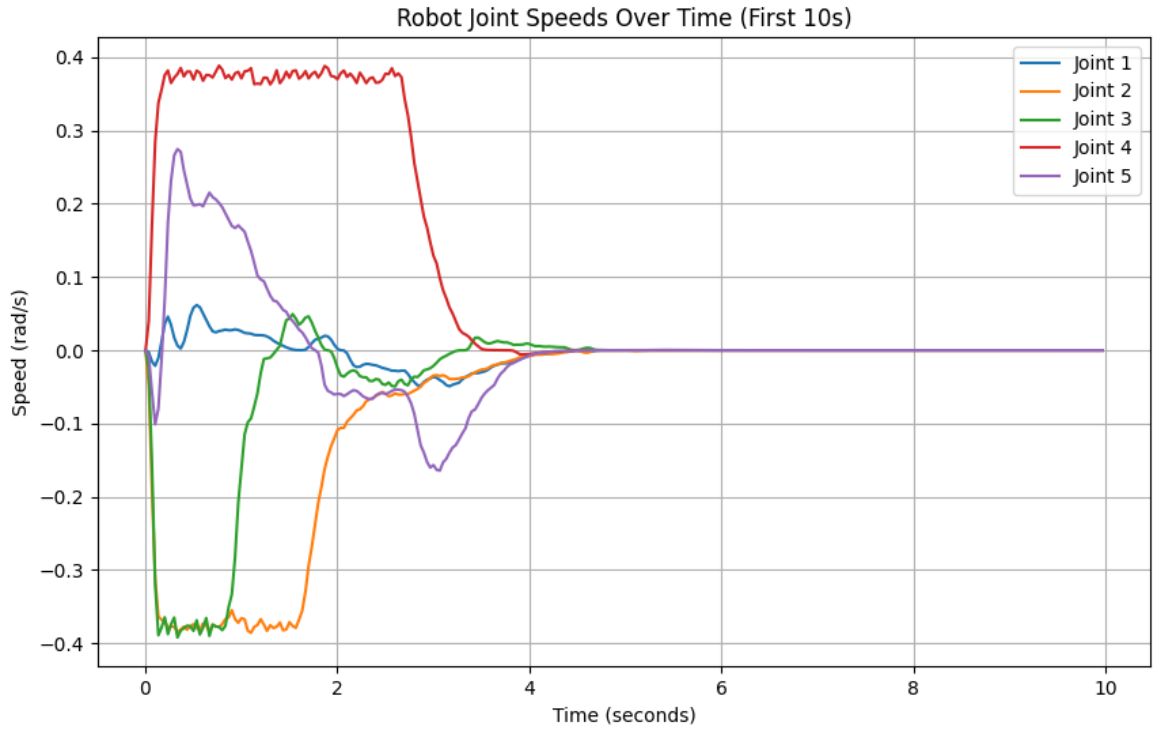


Fig. 4. Robot joint speeds over time utilizing an optimized speed multiplier (0.2)

As illustrated in the figures above, commanding the robot with a higher speed factor (Fig. 3) generated abrupt, over-corrected torque spikes that induced unacceptable mechanical vibrations once the robot reached the target vicinity. In contrast, the optimally scaled commands (Fig. 4) yielded a smooth, continuous trajectory suitable for real-world servo limits. This optimization allows direct integration via `set_servo_angle_j` without relying on the robot's internal, slower trajectory planners.

5.3. Sim-to-Real Trajectory Validation and Visual Comparison

To definitively validate the success of the Domain Randomization and the Sim-to-Real transfer pipeline, a static waypoint test was conducted targeting an exact point in the 3D workspace. The objective was to command the end-effector to the Cartesian coordinates:

- $X = 0.2535 \text{ m}$
- $Y = 0 \text{ m}$
- $Z = 0.195 \text{ m}$
- $\text{Roll} = 0.0$
- $\text{Pitch} = \pi$
- $\text{Yaw} = \pi$

The trained policy was executed targeting these exact coordinates, first within the NVIDIA Isaac Lab digital twin, and subsequently on the physical xArm hardware.

USING AI ALGORITHMS TO SOLVE INVERSE KINEMATICS FOR XARM ARTICULATED ROBOTIC ARMS

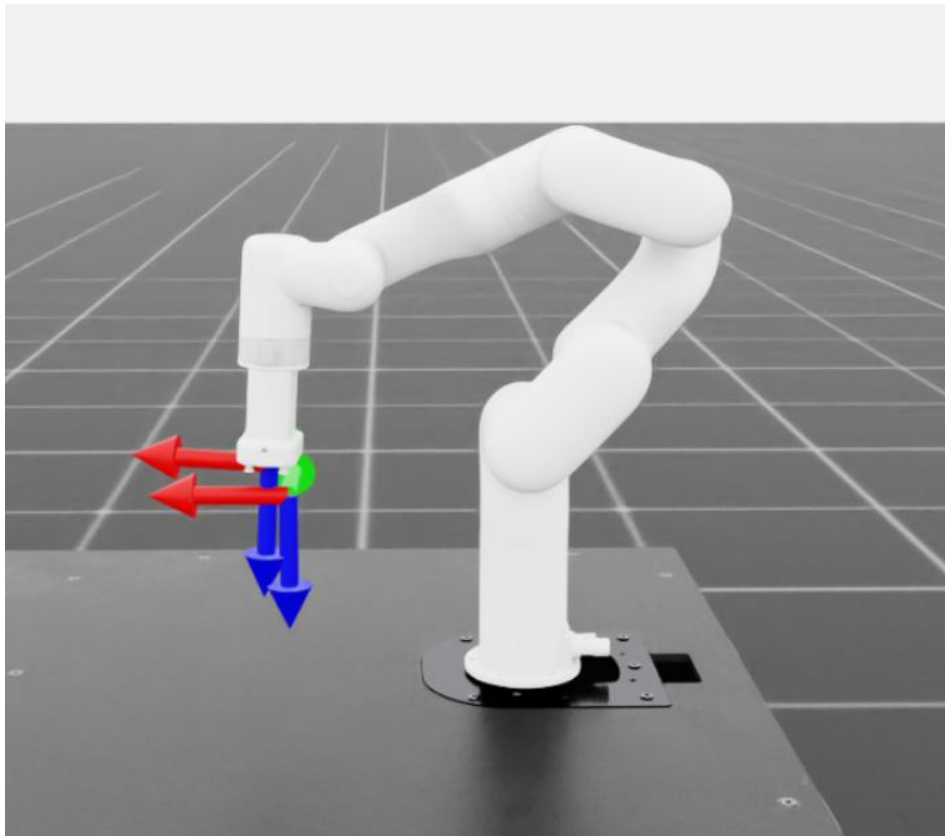


Fig. 5. The simulated xArm robotic arm reaching the target pose within the NVIDIA Isaac Lab environment



Fig. 6. The physical xArm robotic arm successfully replicating the simulated trajectory

As visually demonstrated in the comparison above, the macroscopic joint configurations generated by the RL agent are practically identical across both domains. This visual confirmation proves that the simulated Domain Randomization effectively accounted for the physical gravity, actuator stiffness, and inertia that impact real-world mechanics, allowing the ONNX policy to function seamlessly outside the simulation.

6. Conclusion

This research successfully demonstrates the end-to-end pipeline of training a Deep Reinforcement Learning policy for Inverse Kinematics and deploying it onto physical hardware. By utilizing NVIDIA Isaac Lab's Domain Randomization, the simulated policy robustly bridged the Sim-to-Real gap, effectively managing physical friction and actuator discrepancies. The integration of optimized velocity scaling and strict rate-limiting in the Python deployment script ensured smooth, singularity-free Cartesian navigation by actively suppressing terminal mechanical oscillations.

7. References

- [1] UFactory (2023), xArm User Manual and API Specifications, Shenzhen.
- [2] Viktor Makoviychuk, et al. (2021), "Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning," *arXiv preprint arXiv:2108.10470*.
- [3] Peng, X. B., et al. (2018), "Sim-to-Real Transfer of Robotic Control with Dynamics Randomization," 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 3803-3810.

Adaptive laser engraving application through vision and digital twin data integration for flexible manufacturing

TUDORICĂ Alexandru-Gabriel, GUȚESCU Robert-Andrei and Scientific Leader: IVAN Andrei Mario

Faculty: Industrial Engineering and Robotics, Specialization: Digital Production Systems, Year of study: III, e-mail: tudoricaalexandru1907@gmail.com

Abstract: Modern flexible manufacturing requires the seamless synchronization of physical assets and virtual models, yet complex calibration processes often limit accessibility for non-experts users. The developed adaptive laser engraving application addresses this by converging high-precision computer vision with Digital Twin technology within a beginner-friendly interface. Utilizing a 3MP optical sensor and ArUco fiducial markers, the system establishes an autonomous coordinate mapping with a measured spatial deviation of only 0.2mm to 0.3mm. Detected workpiece geometries are instantly mirrored in a 3D virtual environment, allowing users to intuitively position and independently scale SVG patterns on the projected objects.

KEY WORDS: Laser, Engraving, Vision, DigitalTwin, Interface

1. Introduction

Industrial laser engraving traditionally relies on complex manual metrology and iterative coordinate adjustments, creating a steep learning curve for non-expert users. Such workflows often result in setup delays and material waste. The developed adaptive engraving application dismantles these technical barriers through an intuitive, beginner friendly interface powered by Digital Twin synchronization. By abstracting the intricacies of machine control, the architecture mirrors the physical workspace into a real-time 3D environment. To further enhance accessibility across diverse operational environments, the graphical interface integrates multi-language support, natively accommodating Romanian, English, German, French, Russian, Chinese, Japanese and Korean.

The metrological foundation of this system operates on a 3MP optical sensor and ArUco fiducial markers, establishing an autonomous coordinate frame with sub-millimeter precision. Operators can then import SVG files, retaining full creative freedom to independently scale, rotate, and position their designs within these detected safe zones. Designed with a modular architecture that translates digital layouts into standard machine instructions, ongoing development efforts are focused on expanding the application's hardware compatibility. The future objective is to make the application fully compatible with the majority of laser engraving setups without requiring brand-specific modifications.

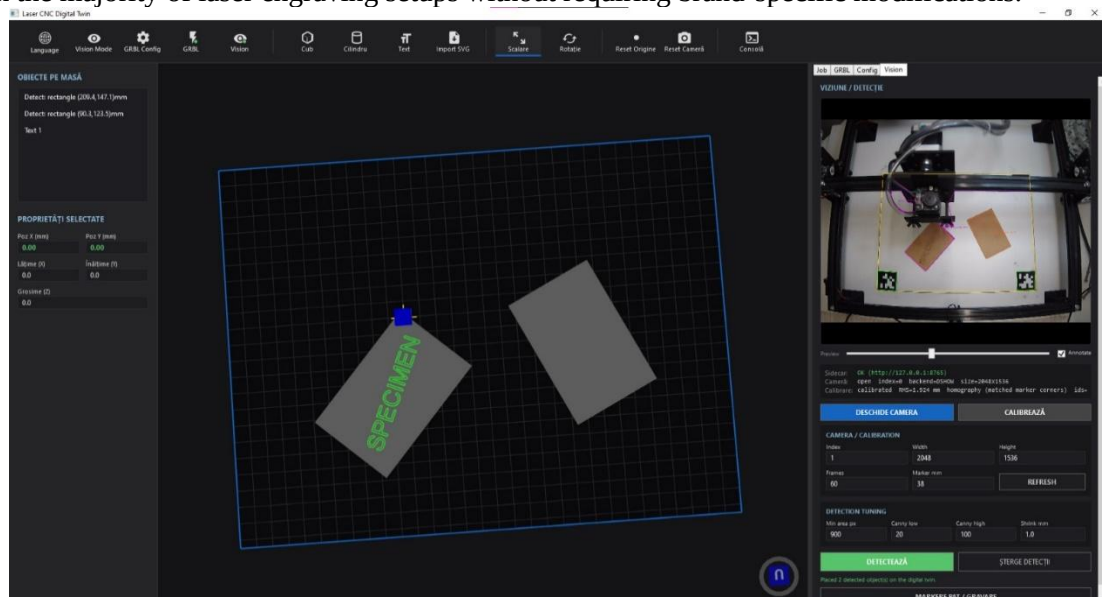


Fig. 1. Users Interface

2. Current state of the field

Industrial laser manufacturing currently relies on fragmented software ecosystems. Computer vision systems for workpiece detection and CNC control interfaces typically operate in isolated environments [1]. Consequently, operators must perform manual calibrations, establish physical origin points, and utilize mechanical jigs to align materials with digital toolpaths. While Digital Twin technology has gained traction in modern production lines, its primary application remains focused on passive machine telemetry or complex CAD-based simulations, both of which demand specialized engineering knowledge [2].

The existing market is heavily dominated by proprietary, brand-specific control software, limiting hardware flexibility and requiring extensive operator training [3]. Recent industry advancements highlight the necessity for integrated cyber-physical systems that simplify user-machine interaction. Despite this progress, a significant gap remains in developing accessible platforms that merge visual metrology directly with interactive 3D environments. Existing solutions rarely offer an intuitive bridge between raw camera data and direct vector-based design manipulation, leaving a hard-time for non-expert users.

3. Simultaneous Vision and Digital Twin Synchronization

The application architecture utilizes parallel processing to maintain real-time synchronization between the physical workspace and virtual environment. To prevent high-computational image processing from causing interface latency, the vision pipeline and the Digital Twin module operate as independent, concurrent threads. Data exchange between these components is handled through a localized messaging layer that streams coordinate updates at regular intervals.

As the 3MP optical sensor captures the workspace, the vision module performs continuous ArUco marker detection and planar homography to establish the metric frame. The resulting spatial data including the exact dimensions and orientation of the workpiece is instantly pushed to the Digital Twin. This allows the 3D interface to update the virtual model's position and boundaries simultaneously with any physical movement on the laser bed.

When an operator positions or scales an SVG design on the virtual object, the system performs a real-time coordinate transformation. The software maps the SVG's local coordinates to the machine's global coordinate system based on the latest visual data. This bi-directional synchronization ensures that the digital layout remains locked to the physical material, allowing the user to visualize the final result before the engraving process begins. By automating this data loop, the system eliminates the need for manual origin point calibration or physical alignment checks.

4. Vector File Format Optimization and Design Mapping

This application utilizes SVGs as the standardized input format for all engraving designs. While initial implementation efforts tested DXF files, a common standard in industrial CNC environments, it presented critical limitations within the real-time Digital Twin environment. As presented in the second image, importing DXF files resulted in geometric fragmentation. Instead of maintaining smooth, mathematical curves, the DXF geometry was reconstructed as a tessellated array of disconnected line segments, leading to a poor visual interpretation and increased complexity for real-time manipulation.

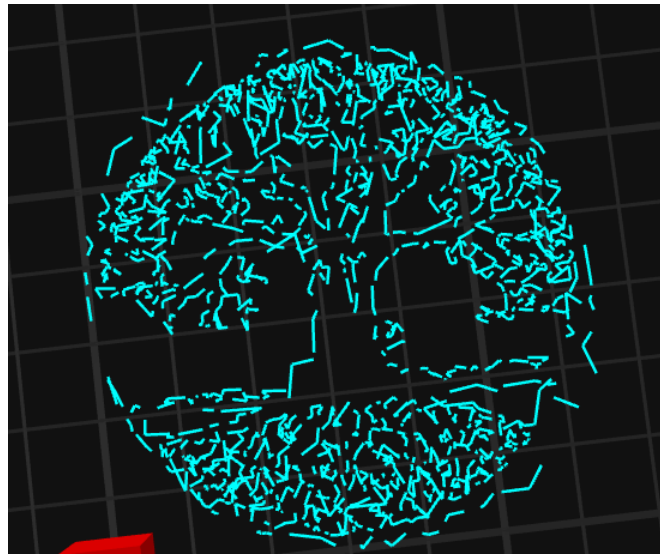


Fig. 2. DXF Attempt

Additionally, the transition to the SVG standard resolved these rendering inefficiencies. The application's core logic leverages the native XML structure of SVGs to maintain mathematically accurate path definitions. This is illustrated in the first image, where the imported text design "SPECIMEN" maintains a sharp, continuous contour within the 3D viewport. Maintaining design integrity is essential for the system's beginner-friendly approach, ensuring that what the user visualizes is exactly what the laser will execute.

5. UI Ergonomics and Global Accessibility

The graphical interface incorporates a localization module supporting eight languages: Romanian, English, German, French, Russian, Chinese, Japanese and Korean. This feature facilitates deployment in international production environments by removing linguistic barriers. For immediate operational support, a dedicated help menu provides technical descriptions for every function within the application, centralizing the documentation and reducing the need for external manuals.

To minimize eye strain during extended use, the software includes a toggleable Night Vision and Light Vision switching mode. These visual presets adjust interface brightness and colors, adapting the Digital Twin's visibility to varying ambient lighting conditions in the workspace. This focus on ergonomic design and multi-language support supports the application's objective of being an accessible, beginner-friendly tool for diverse user groups.

6. Machine Control and Operational Safety

The final stage of the workflow involves translating the virtual design into physical motion through a dedicated communication bridge between the Digital Twin and the machine's firmware. The application converts the SVG vector paths, already transformed within the virtual workspace into standardized G-code instructions. This process utilizes the GRBL serial communication protocol, allowing the software to operate independently of the laser manufacturer's proprietary ecosystem [4]. By generating toolpaths directly from the synchronized 3D model, the system maintains a high fidelity between the user's digital placement and the actual engraving trajectory.

To mitigate human error, we added a must-have software-level boundary-checking mechanism into the control loop. Before the G-code stream is initiated, the application performs a spatial intersection check between the toolpath coordinates and the workpiece boundaries previously identified by the vision pipeline. If any part of the design falls outside the detected material limits, the software triggers an execution lock to prevent damage to the machine bed or wasted material.

7. Conclusions

The development of a vision-guided application synchronized with a Digital Twin environment provides a functional method for simplifying laser engraving workflows. By utilizing SVG-based design mapping and real-time coordinate synchronization, the system maintains a positioning

accuracy between 0.2mm and 0.3mm, reducing the need for manual alignment. The integration of automated boundary-checking and standardized communication protocols ensures operational safety and basic compatibility across different hardware configurations.

The project focused on making the software intuitive through a multi-language interface, an easy to work with 3D workspace and adaptable vision modes. By merging machine control with visual feedback, the application simplifies the transition from design to physical execution, allowing for a more direct and efficient workflow.

8. References

[1]. Radonix, 2025, *Types of Controller in CNC Machines Explained*, <https://radonix.com/types-of-controller-in-cnc-machine/>

[2]. Eduardo Freitas, Assis T. De Oliveira Filho, Pedro R. X. Carmo, Djamol Sadok, Judith Kelner, 2026, *Digital Twin Synchronization: towards a data-centric architecture*, <https://arxiv.org/html/2601.23051v1>

[3]. Amit M. Mankar, R. L. Shrivastava, Jayant P. Giri, 2018, *Open Architecture Control for CNC System – An Approach*, <https://www.sciencepubco.com/index.php/IJET/article/view/20034>

[4]. Said Zergane, Salah Amroune, Mohamed Slamani, Khalissa Saada, Moussa Zaoui, Chawki Farsi, Abdenasser Bakhti, 2024, *Investigation for the Automated Generation of Toolpaths on a 3 Axis CNC Machine*, https://www.ajme.ro/PDF_AJME_2024_1/L6.pdf

Robot conveyor belt for color-based object sorting

Toma Cristian¹, Alex Popescu², și 1. dr. Ing. IVAN Andrei-Mario (Conducător Științific)

Faculty of Engineering and Robotics, Specialization: Robotics, Year of study: I,
e-mail:cristiantomacristi@gmail.com

ABSTRACT:

The main goal of this project is to build an automated robotic system that can sort objects based on their color. We wanted to simulate a real industrial conveyor belt line, but at a smaller scale suitable for a college project. The system uses a GA12-N20 micro motor to move the conveyor belt. When an object passes in front of an Infrared (IR) Line Sensor, the system detects it and stops the belt exactly under a TCS230 color sensor. The color sensor reads the Red, Green, and Blue (RGB) values. To make sure the system works well even if the room lighting changes, we calculate a normalized color ratio. Power is managed by an LM2596 step-down module to keep the voltage safe at 5V. Finally, an SG90 90° micro servomotor acts as a mechanical arm to push the object into the correct sorting bin (Red, Green, or Blue). The programming is done without blocking the Arduino, using a Finite State Machine (FSM) approach.

KEYWORDS: Conveyor, Sorting, Automation, FSM, Arduino.

1. Introduction

Automation and conveyor belt systems are fundamental pillars of modern industry. Automated tasks, such as color-based sorting, are widely used in sectors like recycling and food processing to replace repetitive manual labor, therefore increasing production speed, safety, and accuracy.

The main goal of this project is to design and build a functional, small-scale robotic prototype capable of autonomously detecting, identifying, and sorting objects by their color (Red, Green, Blue). A key objective was to ensure the system remains reliable under varying ambient lighting conditions and operates without software freezing.

To achieve these goals, a combined hardware and software approach was used. The hardware relies on an Arduino UNO controlling a GA12-N20 gear motor via an L298N driver to move the belt. An IR Line Sensor detects the object's arrival, a TCS230 sensor scans its RGB values, and an SG90 micro servomotor physically pushes the item into the correct bin. Power is stabilized using an LM2596 5V step-down module. On the software side, the system is programmed using a Finite State Machine (FSM) for non-blocking, real-time control, combined with mathematical normalization formulas to eliminate optical noise caused by room lighting.

2. State of the art

In the modern industrial sector, automated sorting systems rely on high-speed computer vision, complex Programmable Logic Controllers (PLCs), and precise pneumatic actuators. These large-scale systems can process thousands of units per minute with near-perfect accuracy, dynamically adapting to variations in the production line.

To address the analyzed problem effectively, this project bridges the gap between basic prototypes and industrial standards. Rather than relying on simple sequential logic, the system utilizes a dedicated TCS230 color sensor and implements a Finite State Machine (FSM) architecture. This non-blocking software approach, combined with a mathematical color normalization algorithm, allows the microcontroller to process sensor data in real-time without freezing. This closely emulates the deterministic behavior and reliability required in professional automated sorting lines.

3. Description of the system and components

We selected affordable, reliable components to effectively simulate an industrial conveyor belt. The system's hardware architecture includes:

Table 1. Components

Robot conveyor belt for color-based object sorting

No	Component Name	Primary Function
1	Arduino UNO	Main microcontroller
2	Micro Motor GA12-N20 (1:200)	Conveyor belt mechanical drive
3	Dual Motor Driver L298N	DC motor power control
4	LM2596 5V Step-Down Module	Voltage regulation and stabilization
5	Infrared (IR) Line Sensor	Object presence detection
6	TCS230 Color Sensor Module	RGB color recognition
7	Micro Servomotor SG90 90°	Physical sorting mechanism

3.1. Arduino UNO: The main microcontroller that acts as the "brain," processing sensor data and executing the project's logic.

3.2. Micro Motor GA12-N20 (1:200 Gearbox): Drives the conveyor belt. The 1:200 reduction gearbox lowers the speed while multiplying torque, ensuring smooth movement so objects do not fall off.

3.3. Dual Motor Driver L298N: Since the Arduino cannot output enough current for a DC motor, this driver takes a PWM signal (pin 6) and channels battery power to the GA12-N20 motor.

3.4. LM2596 5V Step-Down Module: Regulates our 9V battery down to a stable 5V. This is critical for preventing motor-induced voltage drops that could restart the Arduino.

3.5. Infrared (IR) Line Sensor: Placed alongside the belt, it uses an IR LED and receiver to detect arriving objects, sending a digital signal (pin 7) when the beam reflects off a block.

3.6. TCS230 Color Sensor Module: Features a grid of light detectors with RGB filters. It measures the reflected color of the block and sends a frequency-based square wave to the Arduino (pin 2).

3.7. Micro Servomotor SG90 90°: Equipped with a plastic arm, it serves as the physical sorting mechanism. It is connected to pin 5 to avoid timer conflicts with the belt motor.

4. Mechanical Design and 3D Modeling

To ensure precise alignment and structural integrity, the mechanical components of the robotic system were custom-designed using AutoDesk Fusion360. The core of the mechanical design is the conveyor belt assembly itself. The 3D CAD model features a rigid main chassis designed to support the lateral tension of the belt without warping.

We modeled custom housing slots for both the drive and idle rollers. The drive roller mount was specifically engineered to perfectly mate with the 10 mm shaft of the GA12-N20 gear motor, ensuring zero slip during torque transfer. Furthermore, the lateral side rails were designed with tight tolerances to guide the objects in a straight line and prevent the colored blocks from falling off the belt prematurely before reaching the sorting mechanism.

Designing the entire assembly in a virtual 3D environment allowed for rigorous interference checks. We verified the spatial clearances between the moving belt, the optical sensors, and the sweeping radius of the SG90 servomotor arm long before any physical parts were manufactured. A digital snapshot of the resulting mechanical assembly is presented below.(Fig. 1.)

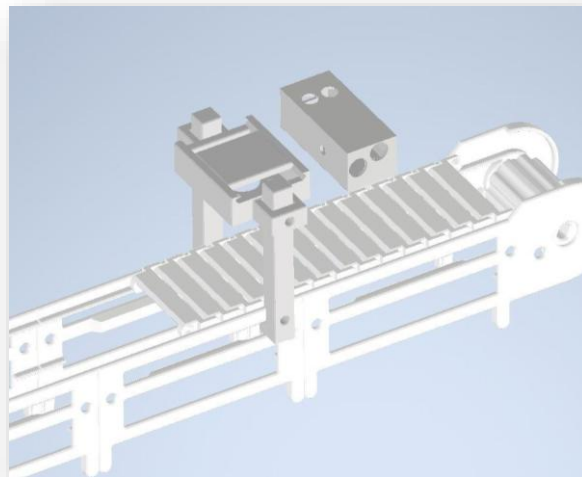


Fig. 1. Mechanical assembly of the conveyor belt with the supports in Fusion360

5. Equations and Calculations

5.1 Belt Speed Calculation

We needed to calculate the speed of our conveyor belt so we could tell the Arduino exactly when to stop the motor. The basic physics equation for speed is:



where:

v = speed of the conveyor belt [m/s]

d = distance between the IR sensor and the color sensor [m]

t = the time delay programmed in the Arduino [s]

On our physical build, the distance from the IR sensor to the color sensor is about 10 centimeters (0.1 meters). By testing, we found that a delay of 0.89 seconds (890 ms) places the object perfectly under the scanner. If we do the math: $0.1 / 0.89 = 0.112$ m/s. This means our belt moves at 11.2 centimeters per second, which is a slow, safe speed thanks to the 1:200 gearbox.

5.2 Color Normalization Math

When we first tested the robot, we had a huge problem. If we tested it near a window, the sunlight made all the numbers change. If we tested it under a desk lamp, it failed again.

To fix this, we used a math trick called normalization. The TCS230 sensor can read Red (R), Green (G), Blue (B), and Clear (C). "Clear" is just the total amount of light hitting the sensor. We changed our code to divide the individual color value by the total light value.



Robot conveyor belt for color-based object sorting

Because of how the Arduino reads the sensor using the pulseIn() function, a stronger color returns a smaller number (a shorter pulse of time). Therefore, whichever calculated ratio is the lowest tells us the real color of the object, regardless of the room lighting. A flowchart of the sorting algorithm is present below. (Fig. 2.)

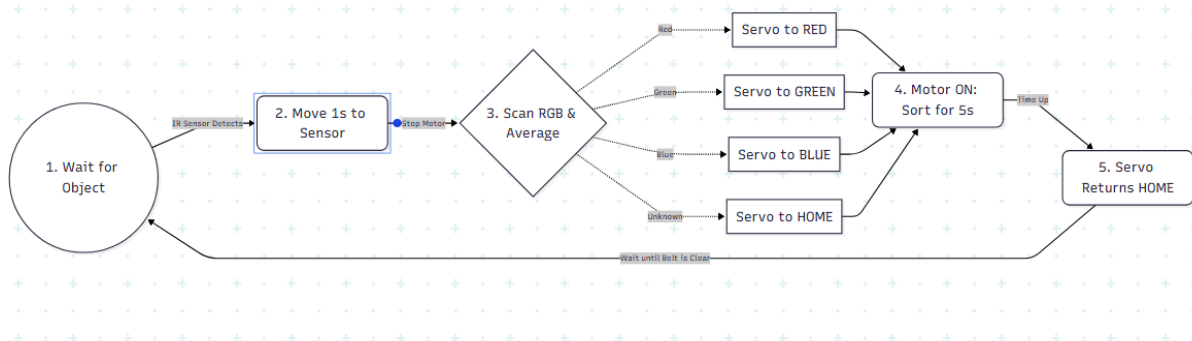


Fig. 2. Flowchart of the sorting algorithm

6. Robot Programming

The software architecture completely avoids the severe limitations of sequential, blocking programming. Standard functions like delay() halt the microcontroller entirely, preventing it from polling sensors or reacting to real-time changes.

To achieve continuous operation, the logic utilizes a Finite State Machine (FSM). A dedicated variable tracks the active operational phase, continuously evaluated via a switch-case block. Time-dependent actions are managed non-blockingly using the integrated millis() function acting as a hardware timer. This approach ensures the main execution thread remains constantly unblocked for simultaneous sensor monitoring.

Here is a simplified piece of our C++ code showing how we sort the colors(Fig. 3):

```
1- case STOPPED_SCANNING_COLOR: {
2   // We read the color 10 times and get the average
3   RGBClear p = readColorAvg();
4
5   // We do the math to normalize the lighting
6   ColorDetected col = decideColorNormalized(p);
7
8   // Figure out where the servo should go (20 degrees is home)
9   int target = 20;
10  if (col == COL_RED) target = 50;
11  else if (col == COL_GREEN) target = 80;
12  else if (col == COL_BLUE) target = 20;
13
14  // Move the servo arm and turn the belt back on
15  sorter.write(target);
16  motorForward(MOTOR_SPEED);
17
18  // Move to the next step: wait 3 seconds for the object to fall
19  goState(SORTING_HOLD_5S);
20 } break;|
```

Fig. 3. Simplified Code of the Color Sorting Algorithm

Another important thing we added is an "anti-retrigger" loop. After the object is pushed off, the code checks the IR sensor one more time. If an object is stuck and still blocking the sensor, the code just waits there. This stops the robot from scanning the exact same block twice.

7. Advantages and Disadvantages

The implementation of this robotic sorting system presents several technical and economic trade-offs.

Advantages: The primary advantage is cost-efficiency, utilizing accessible components to achieve a fully functional conveyor belt. The mechanical architecture demonstrates stability and high torque at low speeds due to the GA12-N20 gear motor. Furthermore, the project effectively integrates critical

Robot conveyor belt for color-based object sorting

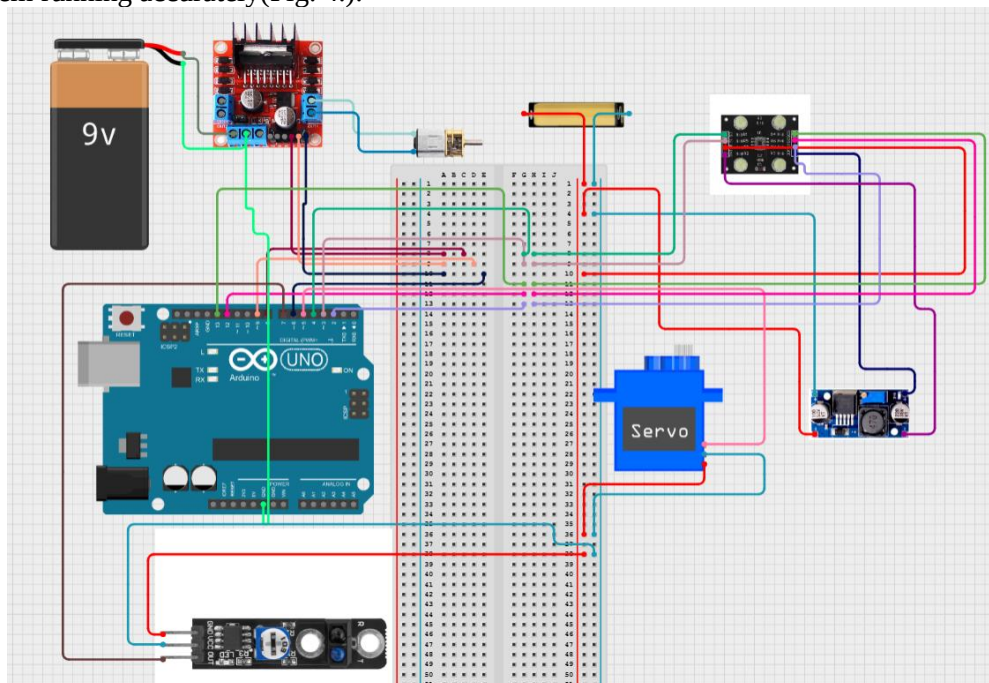
engineering principles, including LM2596 voltage regulation, PWM motor control, and non-blocking Finite State Machine (FSM) programming, resulting in a highly reliable logic execution.

Disadvantages: The main limitation is the overall throughput speed. Unlike industrial systems that utilize advanced vision algorithms for continuous scanning, this prototype requires the conveyor belt to stop momentarily to acquire accurate sensor readings. Additionally, the TCS230 optical sensor remains highly sensitive to focal distance. Variations in the height or surface geometry of the sorted objects can alter light reflection angles, occasionally causing wrong calculations despite the implemented color normalization.

8. Electrical Connections

The wiring follows a simple plan where all components share a common ground to prevent errors. The Arduino UNO acts as the center, connecting to the L298N motor driver using three pins (D6, D8, and D9) to control the belt's speed and direction. The TCRT sensor uses pin D7 to send a signal when an object is detected. The TCS3200 color sensor uses five pins (D2, D3, D4, D12, and D13) to manage its light filters and send color data back to the brain. Finally, the SG90 servo receives its movement commands through pin D5. All sensors and the servo are powered by a steady 5V line to keep them running accurately(Fig. 4.).

Fig.



4.

Wiring

diagram showing how the sensors and motors connect to the Arduino board.

9. Conclusions

In conclusion, this project successfully shows how to build a working automated system by combining mechanical parts, electronics, and code. By using the Arduino UNO to coordinate the GA12-N20 motor and various sensors, we created a machine that can sort objects by color with high reliability.

We solved several real-world problems during the build: the LM2596 module kept the power steady, our math formulas fixed issues with room lighting, and the Finite State Machine (FSM) code kept the robot running without freezing.

10. Bibliography

- [1]. Niku, S. B., *Introduction to Robotics*, John Wiley & Sons, 2020.
- [2]. Groover, M. P., *Automation, Production Systems, and Computer-Integrated Manufacturing*, Pearson, 2018.
- [3]. Own CAD 3D Models and design

11. Notation

GA12-N20 Motor = Small gear motor providing high torque.

TCS230 Sensor = Light sensor measuring RGB color frequencies.

Arduino UNO = Microcontroller acting as the system brain.

L298N Driver = Module controlling motor speed and direction.

LM2596 Module = Converter providing 5V power supply.

SG90 Servo = Small motor moving the sorting arm.

IR Sensor = Detector used to sense arriving objects.

FSM = Logic moving through specific operating steps.

PWM = Signal used to control motor power.

MINOTAUR – Robot for maze solving

Flavia-Alexandra VODA¹, Andrei Mario IVAN

¹ Faculty: Faculty of Industrial Engineering and Robotics, Specialization: Robotics (English), Year of study: II,
e-mail: flavia.voda@stud.fiir.upb.ro

ABSTRACT: Minotaur is a small micromouse robot that solves mazes using three distance sensors for wall detection, wheel encoders for tracking movement and a left-hand rule navigation strategy. It corrects its path using feedback from the sensors. The project applies sensing and decision-making.

KEYWORDS: micromouse robot, left-hand rule, maze navigation, wall detection

1. Introduction

Minotaur is a robot designed to navigate and solve a maze autonomously. The robot is programmed using an Arduino ESP 32, equipped with three Sharp GP2Y0A21YK0F sensors that help it detect and map its environment. The objective is to solve the maze in Micromouse competition, by finding the shortest path from the starting point to the end of the maze. To move inside the maze, Minotaur combines these methods: it uses a gyroscope to make accurate turns, and wheel encoders to count steps and keep track of how far it has moved. For navigation, it follows the left-hand rule, keeping the wall on its left side and prioritizing left turns at each intersection until it reaches the end of the maze.

2. Current state

Micromouse robots already exist, but they aren't as common as other competition robots like sumo robots or line followers. Because of that, the amount of available online documentation and tutorials for competition-ready builds is pretty limited. Most examples show basic prototypes, not robots that are actually meant to run in real contests.

I decided to build Minotaur mainly because I wanted to explore and test different navigation algorithms, seeing how each one behaves in a real maze.

3. Problem Statement

- **Size** – The robot must be small enough to fit within a maze cell that is up to 20 cm x 20 cm. Some competitions have a height or weight restriction;
- **Time** – The robot is limited to a period of 3 minutes, in which it must solve the maze;
- **Communication** – The robot must be fully autonomous, with no outside communication allowed;
- **Maze Integrity** – The robot cannot cause damage to the maze nor can it leave debris behind its path;

3. 3D Printed Parts and Robot Design

Designing Minotaur came with its own constraints, since micromouse robots used in competitions must fit within a 10 × 10 cm footprint, and some events also impose limits on height and weight. This makes the mechanical layout challenging, as all components need to be packed tightly without compromising accessibility or stability.

MINOTAUR – Robot for maze solving

The chassis holds all of the other hardware components together and provides a stable framework. It is fully 3D-printed with PLA filament and built in two levels to make efficient use of the available space. The lower level holds the drive motors, battery, and the three Sharp distance sensors used for wall detection. The upper level contains the step-down converter, the ESP 32 board, the gyroscope, and the motor driver. This stacked layout allowed the robot to remain compact while still accommodating all required electronics.

For future versions, the plan is to replace the current wiring setup with a custom PCB. This would reduce cable clutter, improve reliability, and potentially eliminate the need for a two-level structure. A cleaner, single-layer design would also free up space for additional features or performance upgrades.

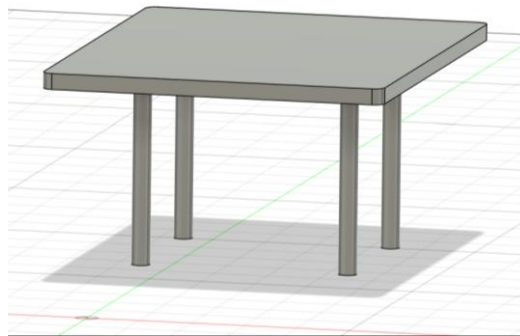


Fig. 1. 3D Model Upper Floor

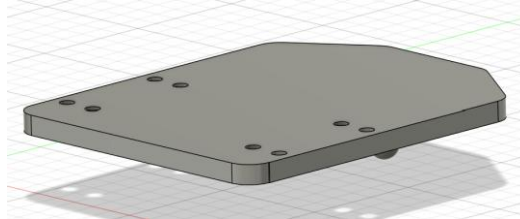


Fig. 2. 3D Model Main Chassis

4. Hardware specifications

The robot combines a capable microcontroller, regulated power delivery, precise motors and infrared sensing to handle real-time maze exploration. The hardware includes:

- ESP 32-WROOM – the main microcontroller handling all processing, communication and control loops;
- 11.1V Li-Po battery – provides the primary power source for the entire system;
- GA12-N20 DC micro motors (6V, 1000 RPM) – two motors, each equipped with encoders and gear reduction, used for avoiding drift over time;
- Sharp GP2Y0A21YK0F sensors – three infrared distance sensors used for wall detection and navigation. They consist of an infrared emitter that sends an invisible beam of light into the environment and an infrared receiver that absorbs the reflected light beam back;
- LM2586 step-down converter – regulates the battery voltage down to the levels required by the electronics;
- TB6612FNG motor driver – controls both DC motors with efficient dual-channel output;

MINOTAUR – Robot for maze solving

- XL6009 step-up converter – boosts voltage where needed for specific components;
- IR receiver and remote — allows remote start and stop through an infrared remote.

5. Code and logic

The code running on Minotaur's ESP 32 microcontroller is written in C++ using Arduin IDE. At each step, the robot reads the three Sharp GP2Y0A21YK0F infrared distance sensors to detect walls on the left, front and right. Based on these readings, it decides whether to move forward, turn left or turn right according to the left-hand rule: always prefer the left openings. Principles of the left-hand rule navigation

The left-hand rule is one of the methods used for a micromouse robot to explore a maze. The idea is that the robot always tries to keep a wall on its left side and chooses its next move based on a fixed priority order. The priority is:

- Turn left
- Go forward
- Turn right
- Turn around (180°)

The use of encoders and a gyroscope in the code:

To make sure the robot doesn't drift while moving forward, Minotaur uses the wheel encoders together with the side sensors. The encoders count the steps on each motor and the side sensors check the distance to the walls. By combining both, the robot can correct its movement and stay centered in the corridor instead of slowly drifting into one of the side walls. PID is implemented to balance the speed of the two motors. This helps the robot keep a straight line, because it constantly adjusts the motor power if one wheel starts moving faster than the other. PID is also used for the robot to stay centered between the walls by using the readings from the two side sensors.

The gyroscope is used for all turning actions: right turn, left turn and 180° turn. It ensures the robot rotates the correct angle instead of guessing based on motor steps alone.

PID Control

PID stands for Proportional-Integral-Derivative and each part contributes to how the robot corrects its motion:

- Proportional term (P)** – reacts to the current error.

If the robot drifts to one side, the proportional term increases motor correction based on how far off-center it is.

- Integral term (I)** – reacts to accumulated error over time.

This helps eliminate small biases, like one motor being slightly stronger than the other.

- Derivative term (D)** – reacts to the rate of change of the error.

It predicts how the robot is moving and smooths out sudden corrections, preventing oscillation.

Together, these three terms allow the robot to adjust its motor speeds smoothly and

continuously. Instead of jerky, on/off corrections, the PID controller produces a stable, balanced response that keeps the robot:

- moving in a straight line
- centered between walls
- turning precisely to 90° or 180° using gyro feedback

This is why PID is essential for micromouse robots: it transforms the sensor readings into smooth and controlled motion.

6. Results and performance

Minotaur is already fully built and has participated in two micromouse competitions, proving that the design and navigation system work reliably in real maze conditions. The most notable achievement was at the Relativity Challenge, where Minotaur secured 1st place.

7. Future improvements

Even though the robot is functional, there is still room for improvement. One of the main goals moving forward is to replace the simple left-hand rule with a more efficient navigation method that uses a full maze map stored in a matrix. This approach would allow Minotaur to analyze the entire layout from the beginning, choose the shortest routes and significantly reduce the time needed to reach the goal.

Additional upgrades are also being considered. Once the improved algorithm is stable, a small turbine, similar to the ones used in line followers, could be added to increase speed and helping the robot maintain better control during fast runs.

8. Conclusions

This project has been an invaluable learning experience, but with many challenges.

Looking back, a few design mistakes stood out as the main factors limiting the robot's performance. The first was relying on infrared sensors with a minimum detection range of 10 centimeters. The Sharp GP2Y0A21YK0F simply couldn't read anything closer than this threshold, which meant the robot lost awareness exactly when it needed it most. Considering most mazes are at maximum 20 centimeters in width and the robot itself is 10 centimeters, that blind zone became a critical flaw. The takeaway is clear: future designs must use sensors without a close-range limitation.

The second lesson was that encoders and a gyroscope aren't optional. Without them, the robot had no reliable sense of how far it had moved or how accurately it was turning. Everything became guesswork and the drift accumulated fast. Adding proper motion feedback it's a requirement for any robot that needs consistent, repeatable behavior.

9. Bibliography

- [1]. https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf
- [2]. https://en.wikipedia.org/wiki/PID_controller
- [3]. https://www.physics.unlv.edu/~bill/ecg497/Drew_Tondra_report.pdf
- [4]. <https://www.instructables.com/Micro-Mouse-for-Beginnersth/>

SHIFTED ONE

AGHIANĂ Sebastian Daniel, COSTEA Dimitris-Mihai și IVAN Mario Andrei

Faculty: IIR, Specialization: Robotics, Year: II, e-mail: aghiana.sebastian.daniel@gmail.com

ABSTRACT: *We have decided to build a robot that has the purpose of participating in Mini Sumo competition, and for this, our skills combined to emerge SO – Shifted One. SO despite its bulky structure is made for a specific scope, to outlast the other robots in endurance. The structure encompasses a 750mAh LIPO battery which feeds the XMOTION V3 controller with a higher voltage for longer compared to the usual 300mAh LIPO. The controller drives the 2 NOVAMAX 400RPM motor with the full 12V of the battery, while the 5 JSUMO JS40F sensors acquire data for enemy tracking. Added to this 2 JSUMO ML1 line detection sensors keep the robot inside the dohyo. For the physical structure, SO is soft on the inside to dampen shocks, but has a sturdy shield on the outside to block the blunt force from the enemy robot.*

KEY WORDS: PLA+, NOVAMAX, XMOTION V3, 3DPRINTING, JS40F

1. Introduction

The engineering behind a Mini Sumo robot might look simple, but some parts require really fine tuning. For this reason, we have decided to make our robot modular, all soft parts being generated by FDM 3D printing and the hard parts being cut out of sheet metal, thus allowing us to change components as required, to modify parameters, mounting positions or adapt new parts to an already built chassis. An important aspect that requires the aforementioned technique of modular structures is weight balance, finely tuned and the robot can compete, fail to achieve good weight balance, the enemy will win.

2. State of the Art and Comparative Analysis

2.1 Research: As of today, Mini Sumo robots are not new in their defining structure, they include a chassis, means to detect an enemy, a controller to process data and motor wheel combination to make the whole thing mobile. In the studies that we have read [6], [7], [8], the structure defined before is the starting point, from which each team adds on their own ideas. For example some might prefer IR sensors as seen in one of the study cases [8], and in other cases some might prefer ultrasonic sensors as seen in another case study [7]. Same for the motor wheel combination, control units, fight logic, inner/outer chassis structure, electrical/battery parameters which may be different across each team depending on strategies and mechanical parameters to be achieved.

2.2 Approach: Our robot SO is fully ready for fighting, the chassis is complete with weights allowing it to be competitive, shields to protect the soft PLA+ chassis and the sensitive electronics inside. One change that has been made to the robot from the basic common choice [1] JSUMO CORE 6V 400RPM to [2] NOVAMAX 6V 400RPM not being that popular but bringing higher torque to the equation giving our robot more fighting power. With the new motors, the fight logic was updated completely to make use of the enhanced parameters. [3] The integrated XMOTION V3 having the microcontroller atmega32u4 is fast for reading 7 IO pins at the same time, IO pins to which the [9] JS40F IR digital sensors are connected sending signals constantly searching for “IR visible” targets, and the [4] ML1 line sensors that scan the ground for the white dohyo border. The added weights make it balance towards a 40% front, 60% back weight distribution which is not completely optimal but good for higher grip. [10] The battery, a 11.1V TATTU 3S 750mAh 95C LIPO is a bit oversized compared to the 7.4V 2S 700mAh which we saw in a case study [8]. To prove that the battery doesn't just eat up space the next chapters will show why we choose to combine it with the parameters of the NOVAMAX motors. Finally the 3D printed PLA+ chassis together with a honeycomb infill ensures adequate dampening while maintaining structural rigidity, compared to the metal structures that we have documented [8] it is soft, but the added shields protect it.

3. Equations

Fig. 1 Discharge/Power/Torque equations

SHIFTED ONE



(1)



(2)

4. Tables

**Table 1. Main Running Parameters
For NOVAMAX 6V 400RPM Scenario 1**

Lock Scenario	
VCC [V]	12.6
Capp[mAh]	750
CurrentDrawMotor[A]	3.65
Runtime [min] Theoretical	12.32
Runtime [min] Aprox - Real	14.32
Assuming Safe 10.4V lower limit	Average discharge per 0.2V drop
Motor winding res [ohm]	2
Resistance Loss [V]	7.3
Stop Torque [kg/cm]	6.5
Step [V]	0.2
Assumed ratio of torque to current	1.780821918

The table above defines the first running scenario, the full lock, the scenario in which the robot has to stress the motors to the maximum capabilities while draining the battery the fastest. VCC is the battery input that the XMOTION V3 controller gets, Capp is the battery capacity, CurrentDrawMotor is the max current drawn by the motor in this scenario. Continuing on Runtime Theoretical refers to the runtime calculated for a continuous max current draw, Runtime Aprox Real is the runtime calculated by evaluating the dynamic draw of the motor. For defining the dynamic draw some constraints are defined, safe lower limit for battery voltage is 10.4V, and the evaluation is done in steps of 0.2V. The resistance of the motor winding is 2 Ohm, thus generating by ohms law a Resistance Loss of 7.3V. For this scenario the max torque is evaluated starting from the rated torque as to see its decrease with battery discharge. To do so, a current to torque ratio is estimated, and the next table is generated.

Table 2. Dynamic Discharge Parameters for NOVAMAX 6V 400RPM Scenario 1

i	VCC [V]	Capacity	Discharge rate (assumed)
1	12.6	690	
2	12.4	630	-8%
3	12.2	570	-8%
4	12	510	-8%
5	11.8	450	-8%
6	11.6	390	-8%

SHIFTED ONE

7	11.4	330	-8%
8	11.2	270	-8%
9	11	210	-8%
10	10.8	150	-8%
11	10.6	90	-8%
12	10.4	30	

The table above defines the voltage for each step, the capacity for each step that decreases by an assumed 8%.

Table 3. Dynamic Current To Torque for NOVAMAX 6V 400RPM Scenario 1

i	Discharge Time [min]	I [A] (per motor)	Torque [kg/cm]
1	1	3.65	6.5
2	1.028571429	3.55	6.321917808
3	1.058823529	3.45	6.143835616
4	1.090909091	3.35	5.965753425
5	1.125	3.25	5.787671233
6	1.161290323	3.15	5.609589041
7	1.2	3.05	5.431506849
8	1.24137931	2.95	5.253424658
9	1.285714286	2.85	5.075342466
10	1.333333333	2.75	4.897260274
11	1.384615385	2.65	4.719178082
12	1.411764706	2.55	4.54109589

The table above defines the discharge time for each step which is defined above as 60mAh, the max current that the battery can current at each step, and the maximum torque that the motor can generate with the given parameters.

Table 4. Main Running Parameters For NOVAMAX 6V 400RPM Scenario 2

Fight Scenario	
VCC [V]	12.6
Capp[mAh]	750
CurrentDrawMotor[A]	2
Runtime [min] Theoretical	22.5
Runtime [min] Aprox - Real	32.73
Resistance Loss [V]	4
Step [V]	0.2
Stop Torque [kg/cm]	4.4
Assumed ratio of torque to current	2.2

The characteristics are the same for the JSUMO motor, only their values are different.

Table 5. Dynamic Discharge Parameters for NOVAMAX 6V 400RPM Scenario 2

i	VCC [V]	Cappacity	Discharge rate (assumed)
1	12.6	690	
2	12.4	630	-8%

SHIFTED ONE

3	12.2	570	-8%
4	12	510	-8%
5	11.8	450	-8%
6	11.6	390	-8%
7	11.4	330	-8%
8	11.2	270	-8%
9	11	210	-8%
10	10.8	150	-8%
11	10.6	90	-8%
12	10.4	30	

Table 6. Dynamic Current To Torque for NOVAMAX 6V 400RPM Scenario 2

i	Discharge Time [min]	I [A] (per motor)	Torque [kg/cm]
1	1.846153846	2	4.4
2	1.945945946	1.9	4.18
3	2.057142857	1.8	3.96
4	2.181818182	1.7	3.74
5	2.322580645	1.6	3.52
6	2.482758621	1.5	3.3
7	2.666666667	1.4	3.08
8	2.88	1.3	2.86
9	3.130434783	1.2	2.64
10	3.428571429	1.1	2.42
11	3.789473684	1	2.2
12	4	0.9	1.98

Table 7. Main Running Parameters For JSUMO CORE 6V 400RPM Scenario 1

Lock Scenario	
VCC [V]	12.6
Capp[mAh]	750
CurrentDrawMotor[A]	3.2
Runtime [min] Theoretical	14.0625
Runtime [min] Aprox - Real	13.13086227
Assuming Safe 10.4V lower limit	Average discharge per 0.2V drop
motor winding res in [ohm]	4
Resistance Loss [V]	12.8
Stop Torque [kg/cm]	3.9
Step [V]	0.2
Assumed ratio of torque to current	1.21875

Table 8. Dynamic Discharge Parameters for JSUMO 6V 400RPM Scenario 1

i	VCC [V]	Capacity	Discharge rate (assumed)
1	12.6	690	
2	12.4	630	-8%

SHIFTED ONE

3	12.2	570	-8%
4	12	510	-8%
5	11.8	450	-8%
6	11.6	390	-8%
7	11.4	330	-8%
8	11.2	270	-8%
9	11	210	-8%
10	10.8	150	-8%
11	10.6	90	-8%
12	10.4	30	

Table 9. Dynamic Current To Torque for JSUMO CORE 6V 400RPM Scenario 1

i	Discharge Time [min]	I [A] (per motor)	Torque [kg/cm]
1	0.566929134	6.4	7.8
2	0.576	6.3	7.678125
3	0.753926702	6.2	7.55625
4	1.090909091	3.35	4.0828125
5	1.125	3.25	3.9609375
6	1.161290323	3.15	3.8390625
7	1.2	3.05	3.7171875
8	1.24137931	2.95	3.5953125
9	1.285714286	2.85	3.4734375
10	1.333333333	2.75	3.3515625
11	1.384615385	2.65	3.2296875
12	1.411764706	2.55	3.1078125

Something interesting occurs in this case, the motor is stressed beyond the max defined parameters (red boxes), thus the current draw skyrockets, the torque is just an estimate, and the windings heat up very fast. In this case the motor fully locks and it can break.

**Table 10. Main Running Parameters
For JSUMO CORE 6V 400RPM Scenario 2**

Fight Scenario	
VCC [V]	12.6
Capp[mAh]	750
CurrentDrawMotor[A]	2
Runtime [min] Theoretical	22.5
Runtime [min] Aprox - Real	12.81754396
Resistance Loss [V]	8
Step [V]	0.2
Stop Torque [kg/cm]	1.2
Assumed ratio of torque to current	0.6

Table 11. Dynamic Discharge Parameters for JSUMO 6V 400RPM Scenario 2

i	VCC [V]	Capacity	Discharge rate (assumed)
1	12.6	690	
2	12.4	630	-8%
3	12.2	570	-8%
4	12	510	-8%
5	11.8	450	-8%
6	11.6	390	-8%
7	11.4	330	-8%
8	11.2	270	-8%
9	11	210	-8%
10	10.8	150	-8%
11	10.6	90	-8%
12	10.4	30	

Table 9. Dynamic Current To Torque for JSUMO CORE 6V 400RPM Scenario 2

i	Discharge Time [min]	I [A] (per motor)	Torque [kg/cm]
1	0.911392405	4	2.4
2	0.935064935	3.9	2.34
3	0.96	3.8	2.28
4	0.98630137	3.7	2.22
5	1.014084507	3.6	2.16
6	1.043478261	3.5	2.1
7	1.074626866	3.4	2.04
8	1.107692308	3.3	1.98
9	1.142857143	3.2	1.92
10	1.180327869	3.1	1.86
11	1.220338983	3	1.8
12	1.24137931	2.9	1.74

5. Figures

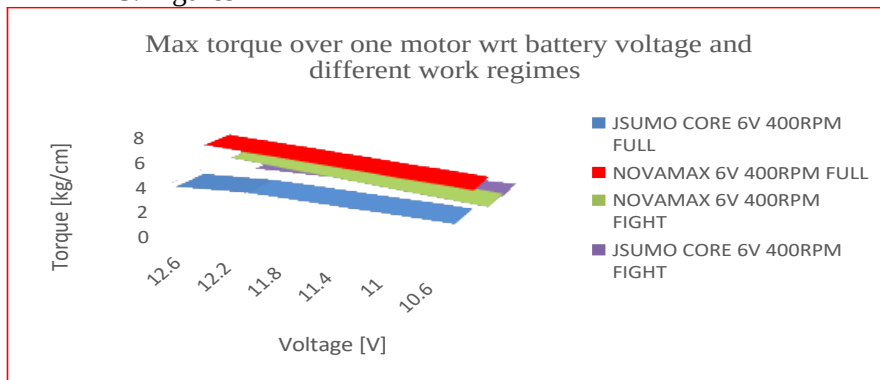


Fig. 1. Motor Comparison with respect to current draw and working regime (lower is better)

$$V = 240 \cdot 300 \cdot 300 = 21,6 \cdot 10^4$$

(1)

SHIFTED ONE

Fig. 2. Motor Comparison with respect to torque and working regime (higher is better)

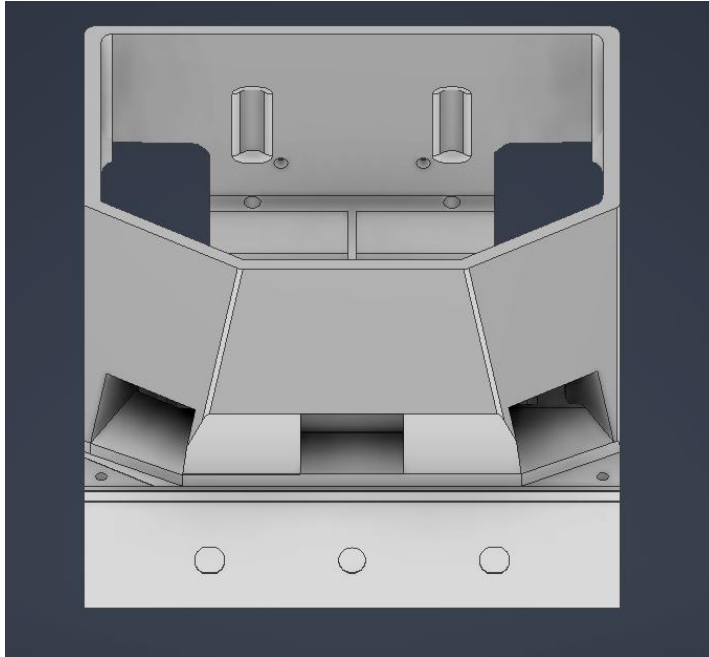


Fig. 3. SO Chassis to be 3D printed Front+Top

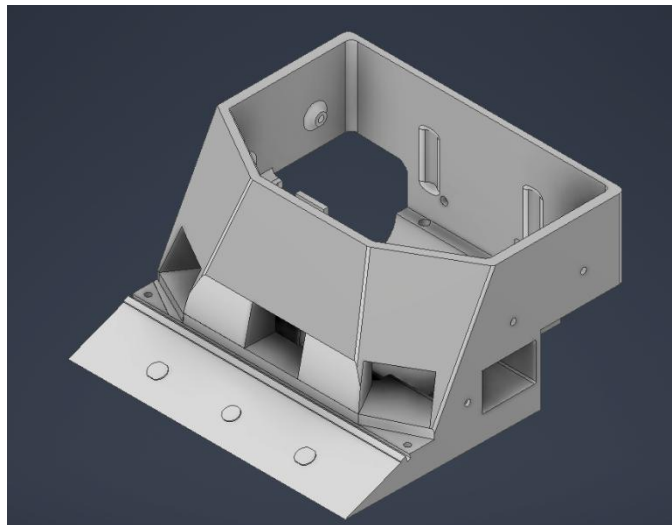


Fig. 4. SO Chassis to be 3D printed Front+Side

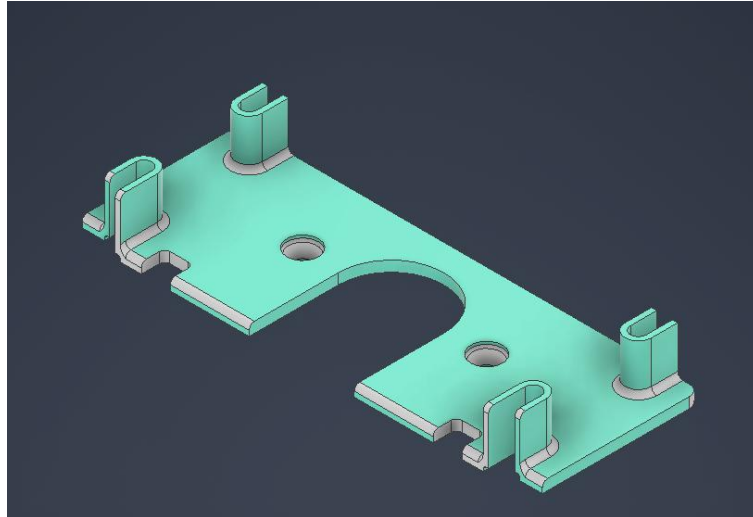


Fig. 4. XMOTION V3 Holder

6. Conclusions

As for conclusions, taking into account the presented data with respect to the research that we have done, the standard structure is improved by increasing the runtime and fighting power of the robot while mostly keeping the same envelope as other robots. That said, our robot is built with endurance in mind and will outlast a robot with a smaller capacity battery. At the same time, the same battery will outperform other batteries with lesser output parameters. The motors picked having better internal characteristics will outperform other motors in endurance, by generating less heat and using less power, while generating more torque. Finally, referring to the custom built chassis structure, the easy altering of components in CAD software allows us to endlessly tweak running parameters to push the most out of the components used.

7. References

- [1]. <https://www.jsumo.com/core-dc-motor-6v-400-rpm>
- [2]. <https://sumozade.com/novamax-6v-400rpm-dc-motor>
- [3]. <https://www.jsumo.com/xmotion-micro-all-in-one-controller>
- [4]. <https://www.jsumo.com/micro-line-sensor-ml1>
- [5]. <https://www.jsumo.com/js40f-digital-infrared-ir-distance-sensor-min-40-cm-range>
- [6]. [https://www.utgjiu.ro/rev_mec/mecanica/pdf/2024-01/35_Gheorghe%20GÎLCĂ,%20Alina-Elena,%20LUPULESCU%20\(MACHIDON\)%20-%20A%20STUDY%20ABOUT%20MICRO%20AND%20MINI%20SUMO%20ROBOTS.pdf](https://www.utgjiu.ro/rev_mec/mecanica/pdf/2024-01/35_Gheorghe%20GÎLCĂ,%20Alina-Elena,%20LUPULESCU%20(MACHIDON)%20-%20A%20STUDY%20ABOUT%20MICRO%20AND%20MINI%20SUMO%20ROBOTS.pdf)
- [7]. <https://ocw.cs.pub.ro/courses/pm/prj2021/alazar/mini-sumo>
- [8]. <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKewjJmY-Sj6-UAxV2BNsEHXOiPKoQFnoECBwQAQ&url=https%3A%2F%2Fbibliotekanauki.pl%2Farticles%2F2135011.pdf&usq=AOvVaw2NDwSwPciuL-ABsZ9W0Ra4&opi=89978449>
- [9]. <https://www.jsumo.com/js40f-digital-infrared-ir-distance-sensor-min-40-cm-range>
- [10]. https://genstattu.com/ta-rl-95c-750-4s1p-xt30.html?srsId=AfmBOopuqj7pPK9_tVar05qb8j5xGT1L7v9gKlsLZn-6lnVFFCX5ZJ2y

Off-line programming and simulation of a palletizing system using the ABB RobotStudio environment. CAD synthesis and sensor mapping

BALABAN Andrei-Claudius

Research Advisor: DOBRESCU Tiberiu Gabriel

Faculty: IIR, Specialization: Robotics, Academic Year: 1st Year Master's,

e-mail: andrei.balaban2707@stud.fiir.upb.ro

Summary:

This project aims to design and off-line simulate a high-capacity hybrid palletizing cell using the ABB Robot Studio environment. The methodology involves using the CAD library with already configured models of the ABB industrial robots and mapping the application with sensory elements to orchestrate a multi-robot system. The cell integrates four ABB IRB 460 robots working in parallel: two manage the orientation and displacement of cardboard boxes, while the other two handle water bottle packs to pre-form structured layers. A thin cardboard layer pad is precisely applied to the water bottle stacks to ensure structural stability. These fully formed layers are then handled by a heavy-payload ABB IRB 8700 robot equipped with a full-layer end-effector, before reaching the final wrapping stage. In conclusion, the cell design proves highly efficient, confirming that the robotic configuration eliminates mechanical interference and successfully handles the complex material flow.

1. Introduction:

The present project falls within the field of end-of-line packaging automation, addressing a complex hybrid palletizing application. Palletizing represents a critical stage in industrial logistics, with the role of taking finished products from production lines and organizing them into stable stacks that are safe for transport. The analyzed system is a high-capacity one and simultaneously processes two types of products with different geometric and dynamic properties: cardboard boxes and water packs.

The particularity of this application lies in its multi-robot architecture, where material flows are managed in parallel by four fast palletizing robots (ABB IRB 460) and consolidated by a heavy-duty robot (ABB IRB 8700), which handles complete product layers and integrates cardboard layer pads for load stabilization prior to the wrapping phase.

2. Current state:

2.1. Analysis of similar solutions in the literature

The use of virtual prototyping and offline simulation represents a critical standard in the design of automated packaging and palletizing lines. To establish the technical foundation for this project, several recent scientific papers in the field were analyzed. Paper [1] presents a simulation model of a palletizing end-effector that accounts for jaw kinematics and stack mass. The analysis is conducted using Abaqus software and aims to evaluate clearance gaps and structural deformations under load. Similarly, paper [2] presents a 3D simulation model of an end-effector for pick-and-place operations, focusing on the geometric optimization of the gripper at high accelerations.

Moving beyond component design, paper [3] analyzes the use of the Process Simulate environment for validating complex cells, demonstrating how the correct mapping of sensor elements enables the elimination of logic errors. Additionally, the issue of stability for unstable products (such as water packs) is addressed in paper [4], which demonstrates the necessity of using cardboard layer pads to stabilize the stacks.

2.2. Operating logic of the palletizing cell

Before analyzing the processes and events within the studied application, it is necessary to delineate the most critical zones that collectively configure the final formed and wrapped stack.

The following steps will then present the flowchart for each individual key process. In the figure below (Fig. 1.37), these boundaries are marked both on the flowchart and in the top-down view of the application. Each major process is enclosed in a color-coded frame to precisely isolate the zone within the cell where those specific events take place.

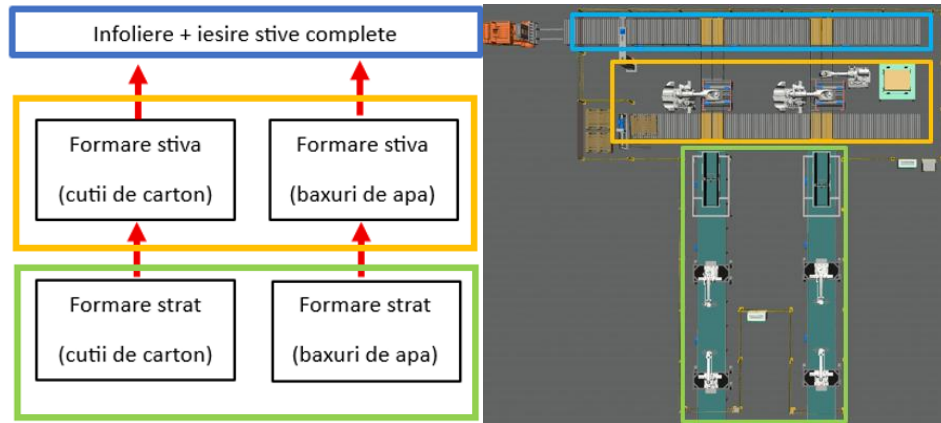


Fig.2.2.1. Simplified flowchart (left) of the palletizing process within the application (right)

In the chronological sequence of process activities (Fig. 1.38), we find the layer-forming zone, where objects of the same type are manipulated and oriented. It can be observed that the primary condition that must be respected within the application is its immediate cessation the moment a human operator enters the workspace. In the flowchart below, a "temporary stop" instruction is highlighted in the event that the condition "Are there persons in the work zone?" yields a positive response. However, we are not strictly limited to this specific command, as it is also possible to implement functions that trigger an audible alarm or return a light signal to indicate human intervention.

Furthermore, the maximum number of objects is verified, given that within the studied application, the quantity of items per layer varies by type; specifically, box-type elements require 18 units to form a complete layer, whereas water pack elements require 20 units.

A final critical aspect within this layer-forming process involves the method by which elements must be picked to guarantee the seamless operation of the picking and handling sequence executed by the two robots. To achieve this, it was established that the elements are indexed (counted) via sensory systems, whereby even-indexed elements are picked by one robot, and odd-indexed elements by the other. For a more optimized startup sequence, it is preferable for the first element (and consequently, any odd-indexed element) to be picked by the second robot.

**Off-line programming and simulation of a palletizing system using the ABB RobotStudio environment.
CAD synthesis and sensor mapping**

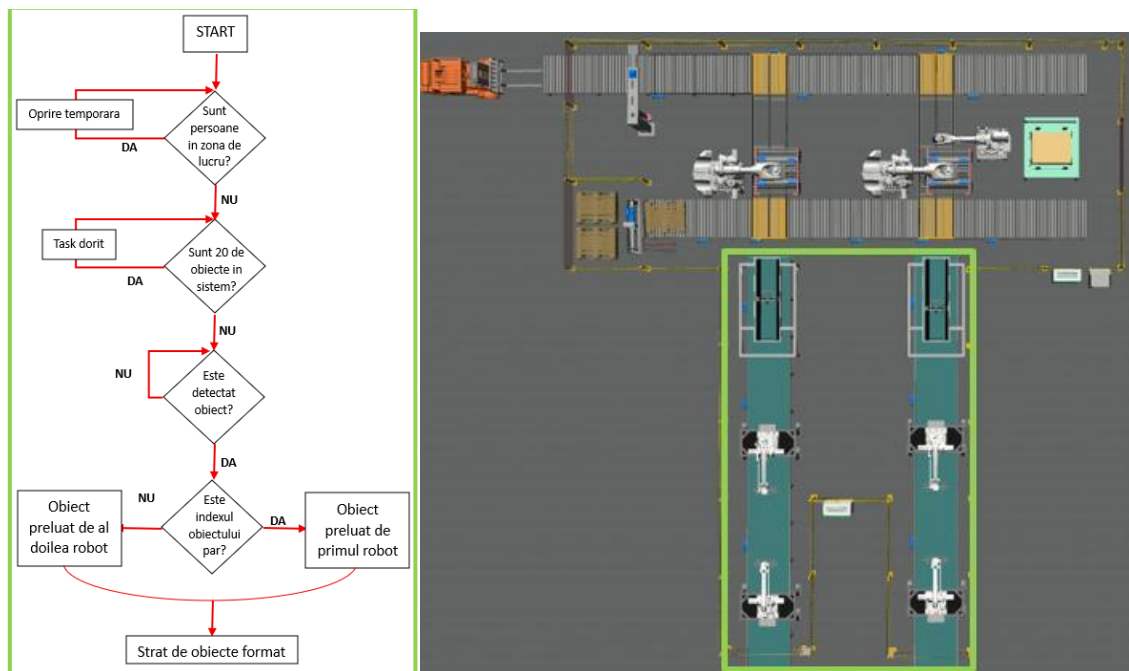


Fig.2.2.2. Layer-forming process flowchart alongside its corresponding zone within the application

Following the completion of the layer-forming process, the sequence advances to the stack-forming stage, which consists of a fixed number of layers depending on the object type. Thus, in accordance with the reference video, it is established that box-type elements require 12 layers for the stack to be considered complete, whereas water pack elements require 10 layers. In both cases, the completed stacks reach nearly the same total height—the water pack stack being slightly shorter due to the dimensional differences of the items and because adding an extra layer would likely exceed the maximum limit required for structural stability.

A slight variation can also be observed between the two flowcharts presented in Fig. 1.39: for the water pack elements, an additional cardboard slip sheet is placed between layers to stabilize them. This is necessary because, compared to the flat surfaces of box-type elements, the packs have a more irregular surface geometry, which introduces instability into the stack structure.

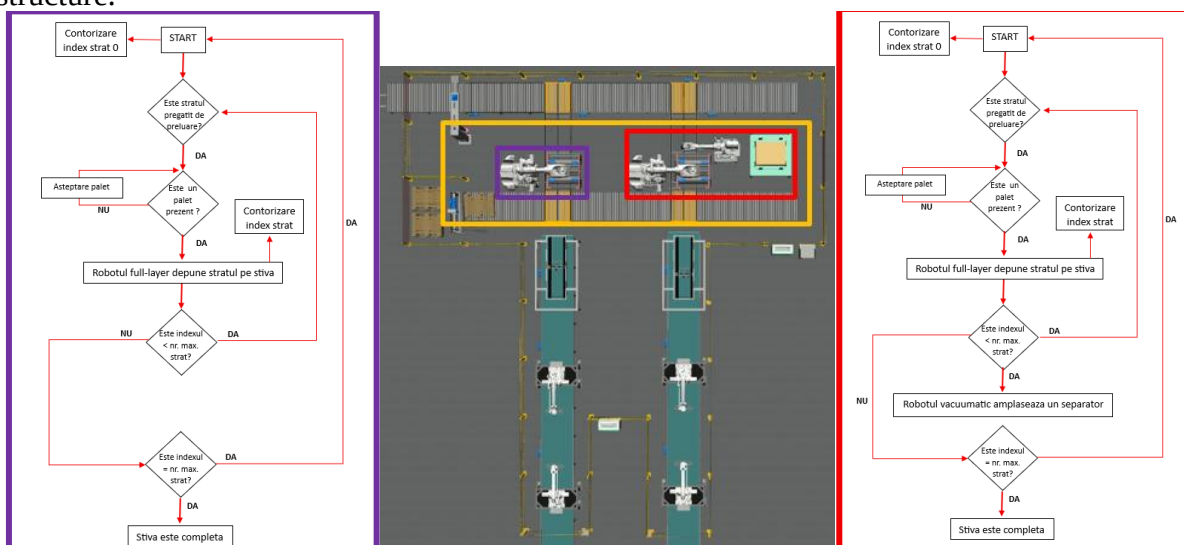


Fig.2.2.3. Flowchart proces de formare stiva cu cutii (stanga) sau baxuri apa (dreapta)

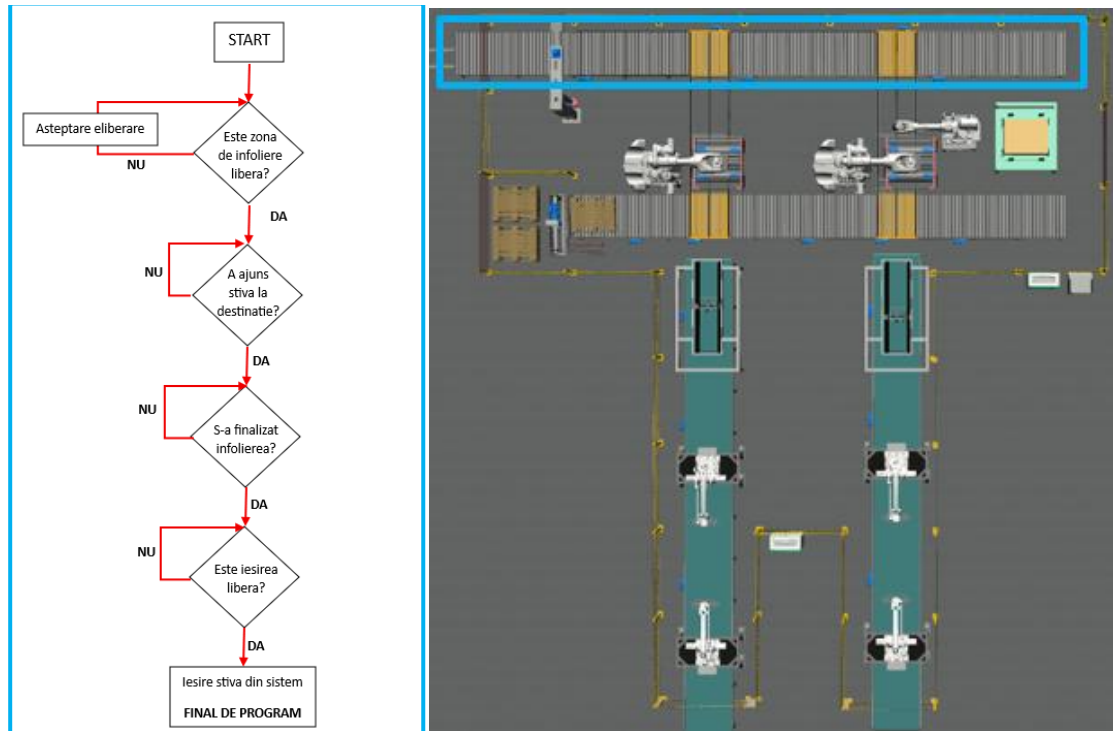


Fig.2.2.4. Stack-forming process flowchart for boxes (left) and water packs (right)

2.3. Conclusions on the current state

The analysis of the literature shows that most studies focus either strictly on the structural analysis of grippers (in Abaqus) or on simple logistic flows. The complexity of the present project lies in integrating these directions into a single hybrid cell. The paper proposes a complex architecture of 5 robots (four ABB IRB 460 and one ABB IRB 8700 with a full-layer effector) coordinated in Process Simulate, simultaneously solving kinematics issues, the handling of different products, and the insertion of layer pads.

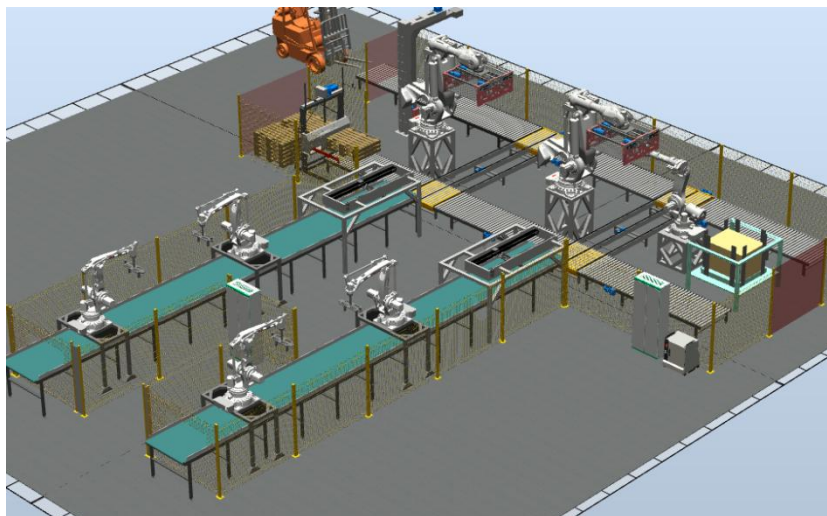


Fig.2.3.1. Isometric view of the application

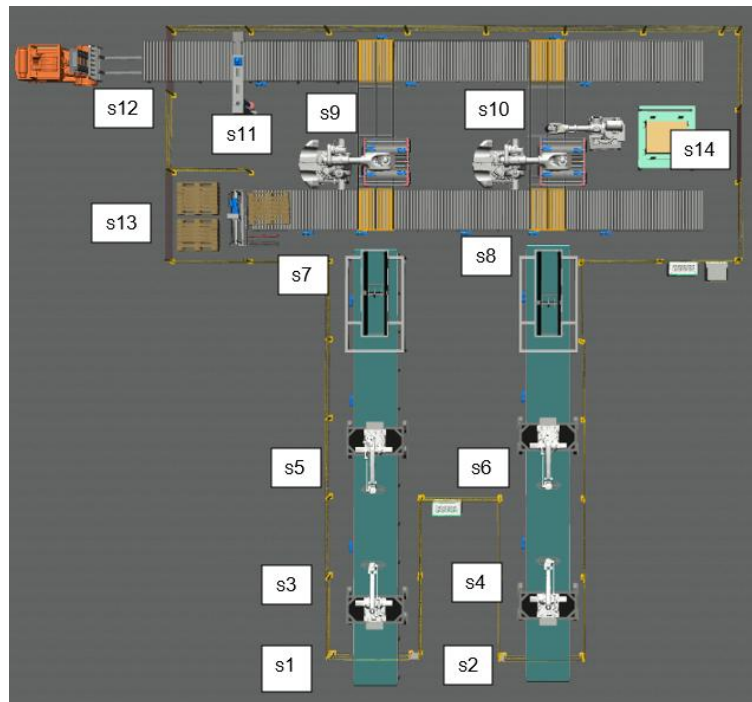


Fig.2.3.2. Sensor position mapping within the studied application

3.1. Calculation model. Materials used in the model:

The analysis begins with defining the materials to be used. To access the materials, we enter Engineering Data, where the software's library is located. In case the software does not possess the desired material, a new material can be created, provided that the necessary characteristics for the type of analysis being performed are defined.

For this analysis, we must select and define a series of materials. From the software's library, we select steel and/or aluminum in both cases for the entire general assembly of the robot riser support. Given that our riser structure is fabricated as a single piece from a single material, we only need to define the steel based on its Young's modulus value and Poisson's ratio.

Table 1

Material's characteristic	Value
Density	7850 kg/m ³
Young module	2*10 ⁵ Mpa
Poisson's ratio	0,3

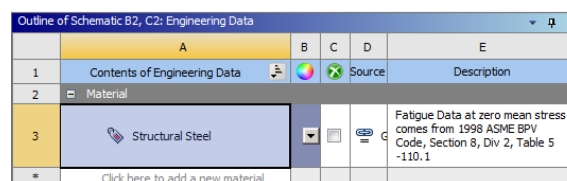


Fig.3.1.1. Table with list of materials

Off-line programming and simulation of a palletizing system using the ABB RobotStudio environment. CAD synthesis and sensor mapping

Properties of Outline Row 3: Structural Steel				
	A	B	C	D E
1	Property	Value	Unit	
2	Material Field Variables	Table		
3	Density	7850	kg m ⁻³	
4	Isotropic Secant Coefficient of Thermal Expansion			
6	Isotropic Elasticity			
7	Derive from	Young's ...		
8	Young's Modulus	2E+11	Pa	
9	Poisson's Ratio	0,3		
10	Bulk Modulus	1,6667E+11	Pa	
11	Shear Modulus	7,6923E+10	Pa	
12	Alternating Stress Mean Stress	Tabular		
16	Strain-Life Parameters			
24	Tensile Yield Strength	2,5E+08	Pa	
25	Compressive Yield Strength	2,5E+08	Pa	
26	Tensile Ultimate Strength	4,6E+08	Pa	
27	Compressive Ultimate Strength	0	Pa	

Fig. 3.1.2. Table of material properties (Structural Steel)

3.2. Workflow steps up to discretization:

The geometry used for the transient analysis was created in the CATIA V5 design software and constitutes a riser element for an industrial robot. From CATIA, we export the model in STEP format (AP 203) so it can be loaded into ANSYS Workbench and later edited in DesignModeler. A STEP file is a 3D model file formatted in the Standard for the Exchange of Product Data, an ISO standard exchange format. It contains three-dimensional data in a format that can be recognized by multiple programs, such as Catia, Ansys, Dassault Systèmes SolidWorks, IMSI TurboCAD, Autodesk Fusion 360, etc. STEP can be used for data exchange between CAD, computer-aided manufacturing, computer-aided engineering, product data management, and other CAX systems.

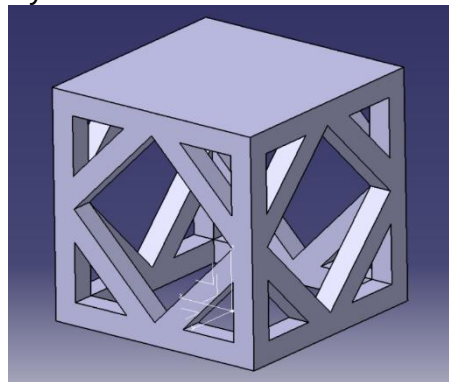


Fig.3.2.1. Initial geometry of the riser support

In static analysis, structures are calculated at a specific moment of loading, considering the forces to be applied statically.

Transient structural analysis (Transient Structural) determines the dynamic response of the structure to time-varying forces (time response). The obtained results include displacements, strains, stresses, etc., all of which vary over time. Through this analysis, the evolution of the response quantities can be monitored throughout the entire operation of the assembly or for a limited duration over which the simulation is performed. The equations solved in the transient analysis are written based on D'Alembert's principle.

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{F(t)\} \quad (1)$$

Unde:

- $[M]\{\ddot{u}\}$ - represents the inertia forces,
- $[C]\{\dot{u}\}$ - the damping forces, which are proportional to the velocities,

- $[K]\{u\}$ - the elastic forces,
- $\{F(t)\}$ - the time-varying external forces

The time step for the simulation to be performed on the model is set over a one-second interval, during which a force acts upon our riser support. The point of application is directed toward one of its edges, thereby simulating the scenario where our robot leans in the same direction as the compressive force.

It is recommended to verify the accuracy of the results. The solver varies the time steps between a minimum and maximum value; these must be small enough to capture the response frequency or rhythmicity, the wave propagation effects, and the nonlinear response of the structure (plastic deformations, buckling, etc.). Smaller time steps lead to a faster convergence of the results. In the Workbench interface, these steps have default values, but they can be adjusted based on loads, experience, or reference examples.

Transient analysis utilizes the Finite Element Method (FEM), the most widely used numerical method for structural analysis. Initially conceived for structural calculations in the linear-elastic mechanical field, the method has been expanded and generalized.

Structural analysis programs simulate model behavior using FEM. In FEM calculations, the continuous structure is replaced by a discrete model. Discretization represents the approximation of the geometric model through a mesh with a large but finite number of elements featuring a simple geometric configuration. The intersections between the mesh elements form nodes. The discretization must adapt as closely as possible to the geometry of the structure and approximate it as accurately as possible. The elements used to approximate the geometry are called finite elements. The core concept of FEM is that an approximate solution regarding structural behavior across a large but finite number of points can be obtained more easily than an exact analytical solution for a continuous structure across an infinite number of points.

Programs utilizing FEM calculate the structural behavior at the nodes of the discretization mesh and, based on these results, express the behavior of the entire structure. In principle, the larger the number of nodes, the finer the discretization mesh and the more precise the geometric approximation; however, the computation time also increases. Practically, beyond a certain number of nodes, further refinement of the mesh no longer significantly improves the accuracy of the results.

As observed in Figure 5.8, the results of the FEM calculation are typically below the exact solution, meaning the structure "appears" stiffer than it is in reality. The Mechanical module contains discretization options within the Mesh menu.

Note: The quality of the discretization is directly reflected in the accuracy of the results. There is no single method for discretization, but there are FEM modeling rules that must be followed for correct mesh generation.

Solving Steps:

The analysis will be solved in the Workbench interface of the Ansys 19.0 program, a software dedicated to computer-aided engineering analysis. To begin the analysis, we first imported the geometry of the industrial robot riser support, and then selected the type of analysis to be executed from the project tree: **Static Structural**.

Off-line programming and simulation of a palletizing system using the ABB RobotStudio environment. CAD synthesis and sensor mapping

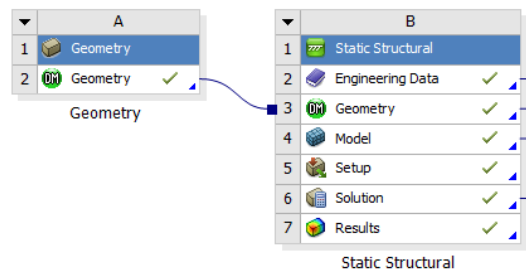


Fig.3.2.2.Static analysis

After **Static Structural**, we have our simplified object. However, for the simulation to be accurate, we need to bring this simplified form as close as possible to the initial, complex shape of the object (with the difference that the geometry will be constructed from simpler geometric shapes; in our case, however, the structure consists mostly of rectangular geometries, so extensive geometry simplification is not required). Because we need to optimize our riser support from this perspective, we want our object to take a more convenient form that requires less material while maintaining the same performance parameters as the previous shape—the only difference being that the general form of the assembly will no longer be symmetrical and rectangular.

Therefore, we introduce the "**Topology Optimization**" module into our application setup to optimize the geometry. It works as follows: the simplified geometry is subjected to forces similar to those exerted by an industrial robot equipped with a vacuum gripper that is leaning toward one of the faces of the support. Following this, we introduce a constraint/objective that eliminates the regions that are not subjected to the stress generated by those forces, allowing the support to take a more economical shape as a result of the simulation. Now, what remains to be done is to adjust the parameters to achieve the most geometrically appropriate configuration.

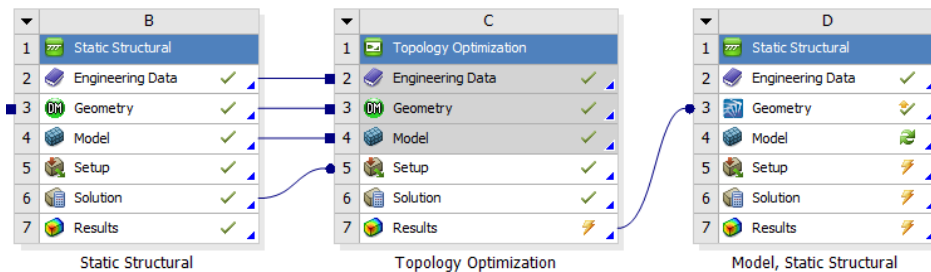


Fig.3.2.3. Static analysis of the modeled support

3.3. Discretization of the element:

The discretization of the riser support is achieved using a "**Face Meshing**" control, where we utilize simple rectangular elements (4-sided/quadrilateral elements). When we run another static structural analysis (on the new model – the image on the right), the discretization of the object is made up of triangular elements (3-sided) in order to capture and approximate the geometric shape as accurately as possible around the "cutouts" resulting from the optimization process..

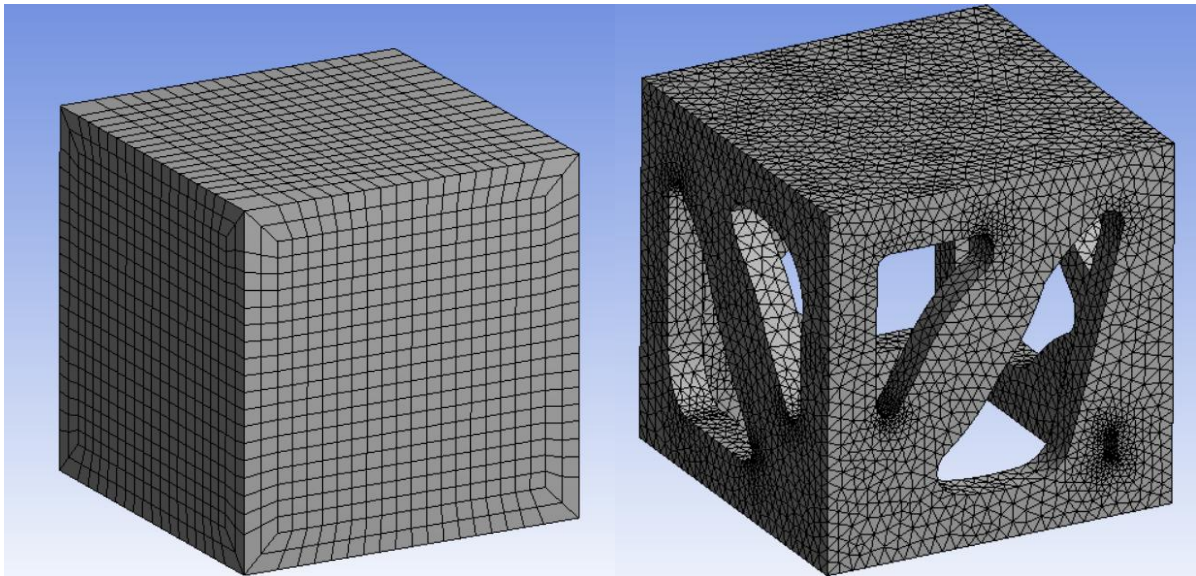


Fig.3.3.1. Initial discretization of the simplified geometry (left) and discretization of the geometry obtained from the optimization (right)

3.4. Simulations:

As previously mentioned, after the **Static Structural** analysis, we have our simplified object. However, for the simulation to be accurate, we need to bring this simplified form as close as possible to the initial, complex shape of the object (with the difference that the geometry will be constructed from simpler geometric shapes; in our case, however, the structure consists mostly of rectangular geometries, so extensive geometry simplification is not required). Because we need to optimize our riser support from this perspective, we want our object to take a more convenient form that requires less material while maintaining the same performance parameters as the previous shape—the only difference being that the general form of the assembly will no longer be symmetrical and rectangular.

Therefore, we introduce the "**Topology Optimization**" module into our application setup to optimize the geometry. It works as follows: the simplified geometry is subjected to forces similar to those exerted by an industrial robot equipped with a vacuum gripper that is leaning toward one of the faces of the support. Following this, we introduce a constraint/objective that eliminates the regions that are not subjected to the stress generated by those forces, allowing the support to take a more economical shape as a result of the simulation. Now, what remains to be done is to adjust the parameters to achieve the most geometrically appropriate configuration.

The input data for the simulation consists of the dimensions and the 3D assembly of the support, while for the forces, we account for the weight of the robot. The required output data from our simulation includes:

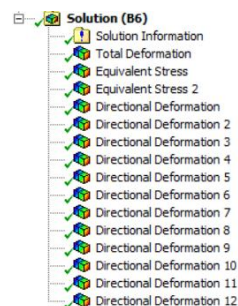


Fig.3.4.1. Output data

Off-line programming and simulation of a palletizing system using the ABB RobotStudio environment. CAD synthesis and sensor mapping

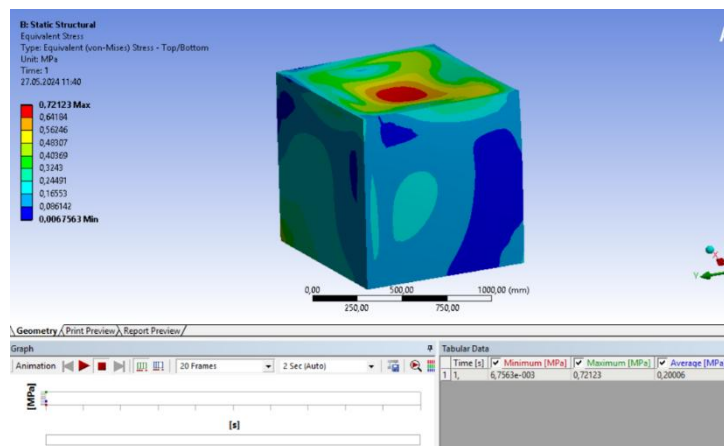


Fig.3.4.2.Reprezentarea actionarii fortelor de deformare totala

3.5. Interpretarea rezultatelor obținute

In cadrul rezultatelor ne intereseaza tensiunile maxime (Total Deformation) si tensiunile echivalente maxime (Equivalent Stress).

Deplasarile maxime in cadrul analizei mele sunt de 0.72123 MPa, aceasta reprezentand deplasarea in urma fortei de greutate a robotului care apasa pe suprot.

Deplasarea maxima in cadrul deformatiilor totale este de 0.0054439 mm, ceea ce implica faptul ca structura este stabila si putem vedea din imaginea de mai jos ca zona marcata cu albastru reprezinta zona deloc solicitata, reprezentand si zone ce urmeaza sa fie scoasa (geometria scoasa in urma procesului de optimizare va reconfigura geometria pentru a se obtine o forma asemenatoare cu a unui fagure).

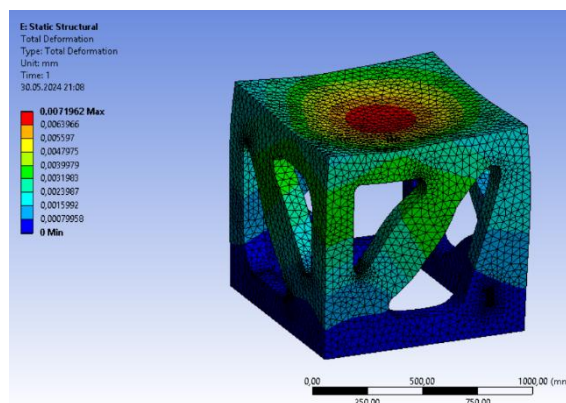


Fig.3.5.1.Reprezentarea fortelor de deplasare totala (geometrie 100%)

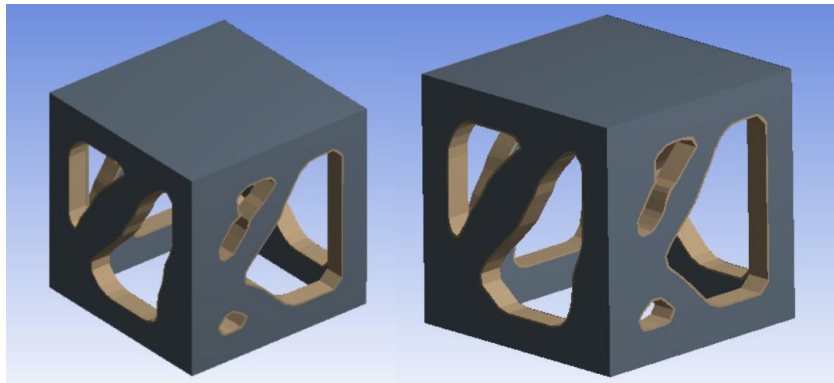


Fig. 3.5.2. Optimization geometry (100% left, 110% right)

Within our application, which features only this riser support as a component, we are interested in determining the values to be entered for the surfaces undergoing optimization. Thus, in our case, for each surface, we take the system coordinates from the surface of each chosen plane (selected in purple); to this coordinate, we add the initial coordinate value of that respective plane multiplied by the efficiency (100% in our case) and divide by 100, resulting in the following formula:

Tabel 2 $\text{Valoare_noua} = \text{Valoare_initiala} + (\text{Valoare_initiala} * \text{randament}) / 100 * \text{Valori initiale}$

Zone 1	Zone 2	Zone 3	Zone 4
0.0011071	0.001124	0.0010624	0.0010052
0.00089191	0.0011937	0.0010937	0.00095299
-0.00029956	-0.00078815	0.00025165	-0.0002338

Tabel 3 Valori 80%

Zone 1	Zone 2	Zone 3	Zone 4
0.00199278	0.0020232	0.00191232	0.00180936
0.001605438	0.00214866	0.00196866	0.001715382
-0.000539208	-0.00141867	0.00045297	-0.00042084

Tabel 4 Valori 90%

Zone 1	Zone 2	Zone 3	Zone 4
0.00210349	0.0021356	0.00201856	0.00190988
0.001694629	0.00226803	0.00207803	0.001810681
-0.000569164	-0.001497485	0.000478135	-0.00044422

Tabel 5 Valori
100%

Zone 1	Zone 2	Zone 3	Zone 4
0.0022142	0.002248	0.0021248	0.0020104
0.00178382	0.0023874	0.0021874	0.00190598
-0.00059912	-0.0015763	0.0005033	-0.0004676

Tabel 6 Valori
110%

Zone 1	Zone 2	Zone 3	Zone 4
0.00232491	0.0023604	0.00223104	0.00211092

**Off-line programming and simulation of a palletizing system using the ABB RobotStudio environment.
CAD synthesis and sensor mapping**

0.001873011	0.00250677	0.00229677	0.002001279
-0.000629076	-0.001655115	0.000528465	-0.00049098

Tabel 7 Valori
120%

Zone 1	Zone 2	Zone 3	Zone 4
0.00243562	0.0024728	0.00233728	0.00221144
0.001962202	0.00262614	0.00240614	0.002096578
-0.000659032	-0.00173393	0.00055363	-0.00051436

From a volumetric standpoint, it was found that the initial volume is 0.234 m³ and the final optimized volume is 0.3 m³. The difference shows that the optimized support has a volume that is 0.057 m³ larger than the initial structure of the support; therefore, unfortunately, we do not benefit from an economic optimization regarding the amount of material it is made of.

3.6. Conclusions drawn from the analyses

In conclusion, within this application, we can state that both the initial structure of the riser support and its optimized version are capable of withstanding the static forces represented by the weight of our robot.

In my opinion, this type of analysis is not very common in CAE applications, as it provides a distinct and specialized set of results compared to typical scenarios involving stress and motion.

To achieve a more accurate optimization, it is necessary to select a geometric efficiency below 100% in order to reach an economic optimization regarding the material used.

The approach was straightforward, progressing from an initial structure to a final one (we imported the initial model, optimized it, modeled the assembly similarly to the initial model but with simplified geometric shapes, and finally verified the results).

4. Conclusions:

The present work validated the kinematic and logical functionality of the hybrid palletizing cell through digital prototyping within the Process Simulate environment, demonstrating the efficiency of the multi-robot architecture composed of four ABB IRB 460 units and the ABB IRB 8700 heavy-duty manipulator equipped with a full-layer end effector. Running the offline simulation enabled the early identification and elimination of mechanical interferences within the shared workspace of the feeding lines, with collisions being resolved exclusively by optimizing trajectories, reconfiguring motion vectors, and implementing logical interlocking routines based on sensory elements. Notably, no geometric modifications were required for the initial spatial layout, structural components, or robot positioning, as the working ranges and cycle times proved to be fully compliant with the throughput requirements. Concurrently, the integration of the cardboard slip sheet placement operation confirmed that the macrostructural stability of the pack stacks is secured prior to the wrapping process. In conclusion, the project demonstrates that utilizing advanced virtual programming techniques eliminates the risks of physical collision and considerably reduces commissioning time, with the cell completely fulfilling all imposed kinematic performance and industrial productivity criteria.

5. Bibliography:

- [1]. Groover, M. P. (2019), Automation, Production Systems, and Computer-Integrated Manufacturing, Pearson, Boston, ISBN 978-0133499612.
- [2]. Spahiu, A. și Popescu, M. (2022), "Optimization of Multi-Robot Palletizing Cells using

Digital Twins”, International Journal of Production Research, 60 (14), 4112-4128, ISSN 0020-7543.

[3]. Dumitru, I. și Ionescu, C. (2024), “Trajectory Optimization for Industrial Manipulators along Conveyor Belts”, în: A. Smith and B. Jones (eds.), Advanced Robotics and Manufacturing Systems, Springer, Berlin, 145-162, ISBN 978-3-031-12345-6.

[4]. Vasilescu, G. și Popa, F. (2023), “Collision Avoidance Strategies in Shared Multi-Robot Workspaces using Process Simulate”, Proceedings, International Conference on Manufacturing Systems (ICMS 2023), Politehnica University of Bucharest, București, 12-14 Octombrie;

[5]. Drăghici, G. (1999), Ingineria integrată a produselor, Editura Eurobit, Timișoara, ISBN 973-96065-7-1.

[6]. *** Siemens Tecnomatix Process Simulate – User Manual and System Guide, 2024.

OFF-LINE PROGRAMMING AND SIMULATION OF A ROBOTIC PALLETIZING CELL USING PROCESS SIMULATE

Student: Bundă Cristian-Ionuț, Supervisor: Ș.l. dr. ing. Tudor ALEXANDRU

Faculty: IIR, Specialization: Robotics, Year of study: I, E-mail: bundacristianionut@gmail.com

ABSTRACT: The purpose of this project was the off-line programming and simulation of a robotic palletizing cell for cardboard boxes containing sauce products using the Process Simulate environment. The project focused on developing a virtual model of the robotic application, including the industrial robot, conveyors, rotary wrapping tables, sensors, and the multifunctional end-effector. The simulation included robot kinematics, TCP definition, coordinate systems, and sensor mapping. Photoelectric sensors, mechanical stoppers, pushers, and safety barriers were integrated into the simulation logic. The generated flow-chart enabled synchronization between palletizing and wrapping operations. The final virtual model reproduced the operation of a real industrial palletizing cell, validated the proposed automated solution, and demonstrated the functional integration of all simulated subsystems.

KEYWORDS: Robotic palletizing, Process Simulate, industrial robot, off-line simulation, sensor integration

1. Introduction

The presented project falls within the field of robotic palletizing and wrapping applications used in the food industry for the automated handling and transportation of packaged products. The analyzed application is designed to perform the automatic palletizing of cardboard boxes containing various types of sauces, using an articulated-arm industrial robot integrated into a complex robotic cell.

The purpose of the project was to carry out the off-line programming and simulation of the entire robotic application in Process Simulate, with particular emphasis on the integration of signals, sensors, and the operational logic specific to a real industrial application. The project aimed to develop a complete functional simulation capable of reproducing the sequence of technological operations and the interaction between all subsystems of the robotic cell [1].

Several important stages were completed during the project. In the first phase, a 3D CAD synthesis of the entire robotic application was developed, along with the definition of the industrial robot's kinematic structure, including the definition of the end-effector TCP and the coordinate systems used in the simulation. Subsequently, the conveyors, rotary tables, wrapping systems, cardboard separators, and safety elements were integrated. In the next stage, the application was developed through the use of signals and photoelectric sensors required for synchronizing the transportation, positioning, and palletizing processes of the boxes. In addition, complete sensor and control-element mapping of the application was performed to simulate the automatic operation of the entire system.

Compared to the initial version of the application, modifications were required at the simulation level to integrate signal-based logic and synchronize the operations between the robot, conveyors, and rotary tables. These modifications enabled the development of a simulation that more closely resembles the real operation of an automated industrial cell.

2. Current state

Following the analysis of previous works in the field of industrial robotic applications, it can be observed that off-line simulation is frequently used for the development and validation of complex automated cells. Most applications focus on integrating industrial robots, conveyors, sensors, and auxiliary systems into an automated and synchronized technological flow.

Paper [2] presents a robotic palletizing application for milk carton bundles using an ABB IRB 660 industrial robot. The application includes multiple conveyors, pusher and stopper mechanisms, as well as the simultaneous construction of two pallet stacks. Compared to this work, the developed project

uses a similar simultaneous palletizing principle but additionally integrates rotary wrapping tables, cardboard separators, and a more complex logic based on signals and sensors integrated within the Process Simulate environment.

Paper [3] presents the simulation of a robotic application designed for servicing an injection molding machine using the ABB RobotStudio environment. The robotic cell includes industrial robots, conveyors, optical sensors, and automated handling systems for plastic semi-finished products. Similar to this work, the developed project focuses on creating a complete off-line simulation and synchronizing integrated subsystems through the use of signals and sensing elements. The main difference lies in the fact that the proposed application is oriented toward palletizing and automatic wrapping processes, featuring a different technological flow and a structure specific to the food industry.

By analyzing these two studies, it can be observed that the developed project adopts modern automation and virtual simulation principles, creating a complex robotic application based on the complete integration of the industrial robot, conveyors, positioning systems, and operational logic specific to a real industrial cell.

3.1. 3D CAD Synthesis of the Robotic Application in Process Simulate

The CAD synthesis of the studied assembly is presented in Figure 1.

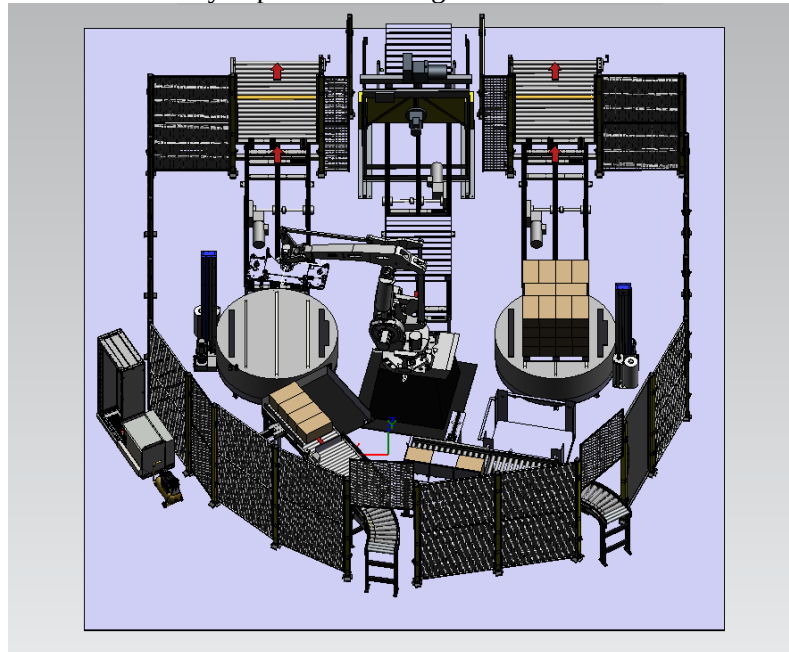


Fig. 1. 3D synthesis imported into Process Simulate

Siemens Process Simulate was used to simulate the off-line robotic application, a software platform widely used in industry for the development and validation of automated processes. This software environment enables the creation of complex robotic simulations, the integration of signal-based logic, and the verification of synchronized operation among all subsystems involved in the technological process.

Unlike other simulation environments primarily focused on defining robot trajectories, Process Simulate offers the possibility of developing virtual applications that closely resemble the real operation of an industrial cell. Thus, not only the movements of industrial robots can be simulated, but also their interaction with sensors, conveyors, pneumatic systems, and auxiliary equipment.

During this stage, the complete 3D CAD synthesis of the robotic cell was developed by importing and positioning the main application components: the industrial robot, transport conveyors, rotary tables, separator stands, and auxiliary equipment. Their placement was carried out in such a way as to comply with the real operating and accessibility conditions of

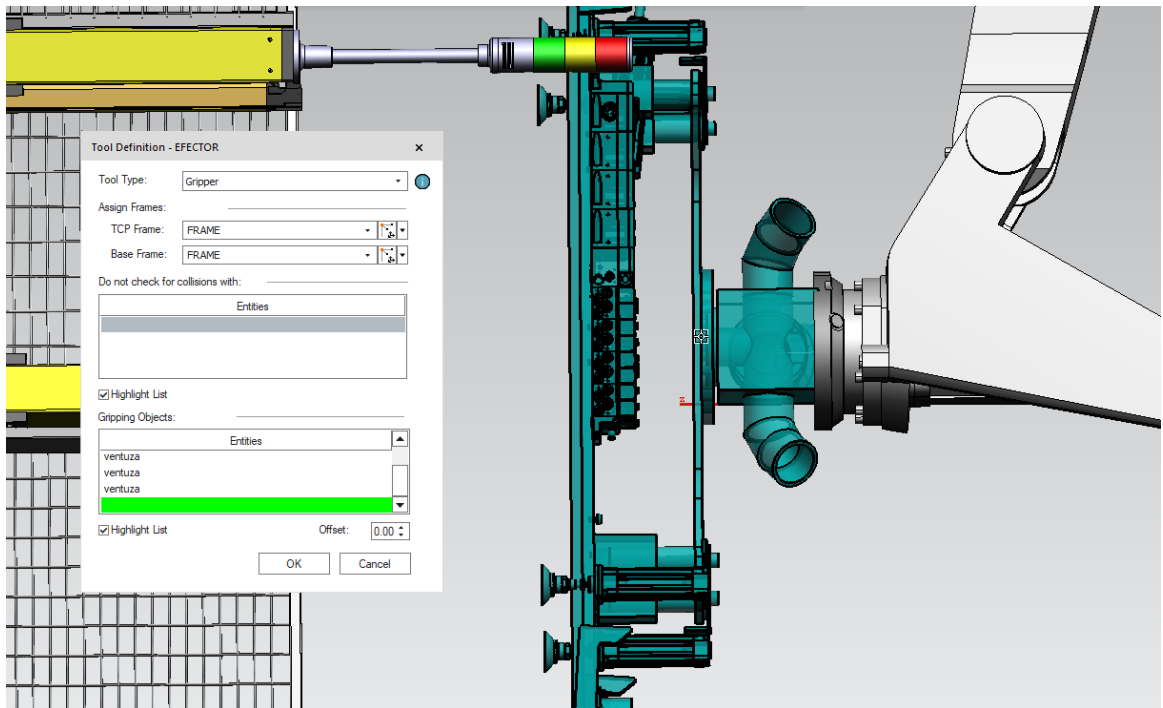


Fig. 3. Defining the robot end-effector TCP.

3.2. Development of the FLOWCHART Including All Processes, Events, and Programmed Logic Elements (Signals) Specific to the Operation of the Robotic Application

The FLOWCHART represents the main logical diagram of the application and highlights the way industrial robots, conveyors, sensors, and auxiliary equipment interact throughout the technological cycle. Through its use, the sequence of operations and the transmission of signals between the various system components can be monitored.

Within the developed application, the operational logic was structured into several main stages:

- Conveyor feeding;
- Product presence detection;
- Object positioning in the pick-up area;
- Object handling and manipulation by the robot;
- Product transfer to the next stage of the technological flow;

For each process stage, logical start and stop conditions were defined based on signals received from virtual sensors and the subsystems integrated into the simulation. Thus, the execution of an operation depends on confirmation that the previous operation has been completed, ensuring the correct synchronization of the entire robotic application [6].

OFF-LINE PROGRAMMING AND SIMULATION OF A ROBOTIC PALLETIZING CELL USING PROCESS SIMULATE

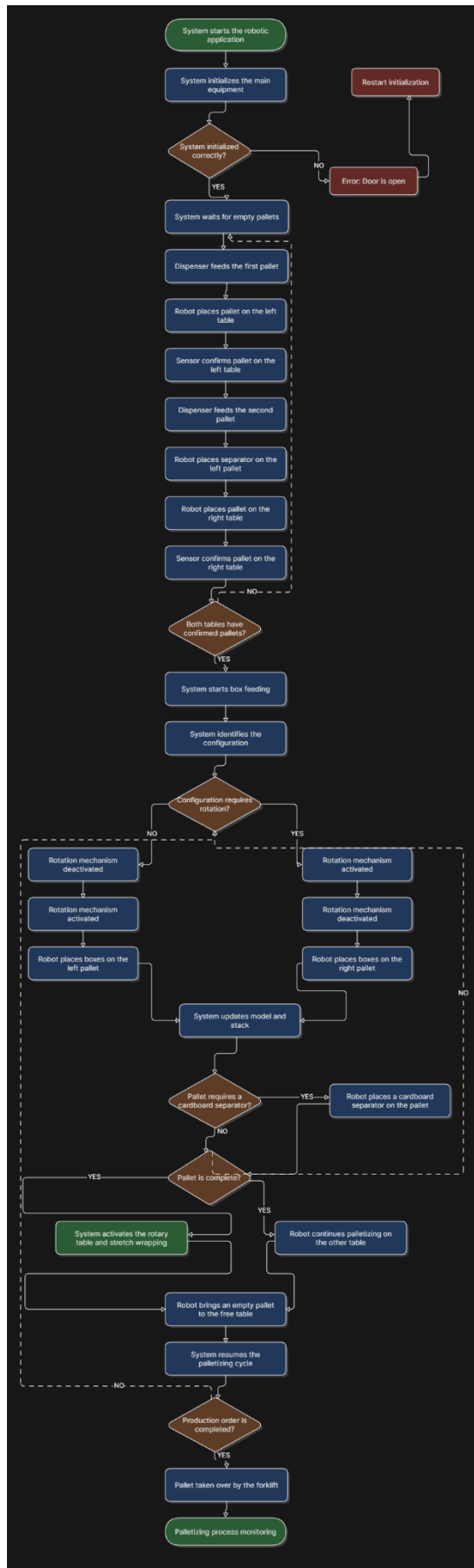
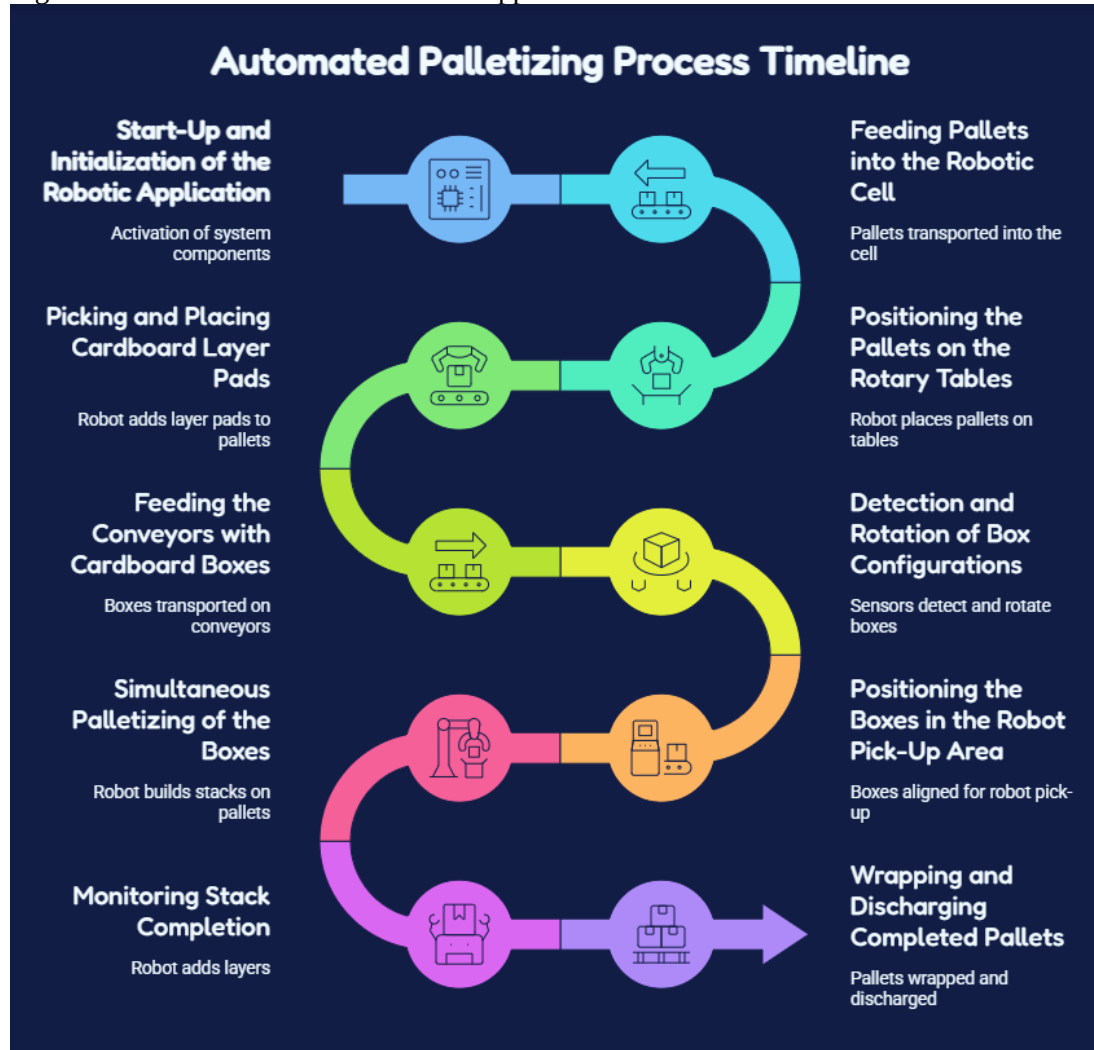


Fig. 4. Detailed flowchart of the robotic application.



F Fig. 5. Simplified flowchart of the robotic application.

3.3. Sensor Mapping of the Entire Robotic Application

A complete sensor mapping was carried out within the robotic application to monitor and synchronize all stages of the technological process. The virtual sensors integrated into Process Simulate provide the signals necessary for coordinating the operation of the industrial robot and auxiliary subsystems.

Table 1. Sensors used in the application

Marking Color	Sensor Type	Sensor Function
Green	C/T slot magnetic sensors	Detection of piston position in pneumatic cylinders
Blue	Safety Light Curtain	Detection of fully loaded pallet evacuation
Yellow	Ultrasonic distance sensors	Detection of separator stack level
Red	Photoelectric sensors	Detection of cardboard boxes position

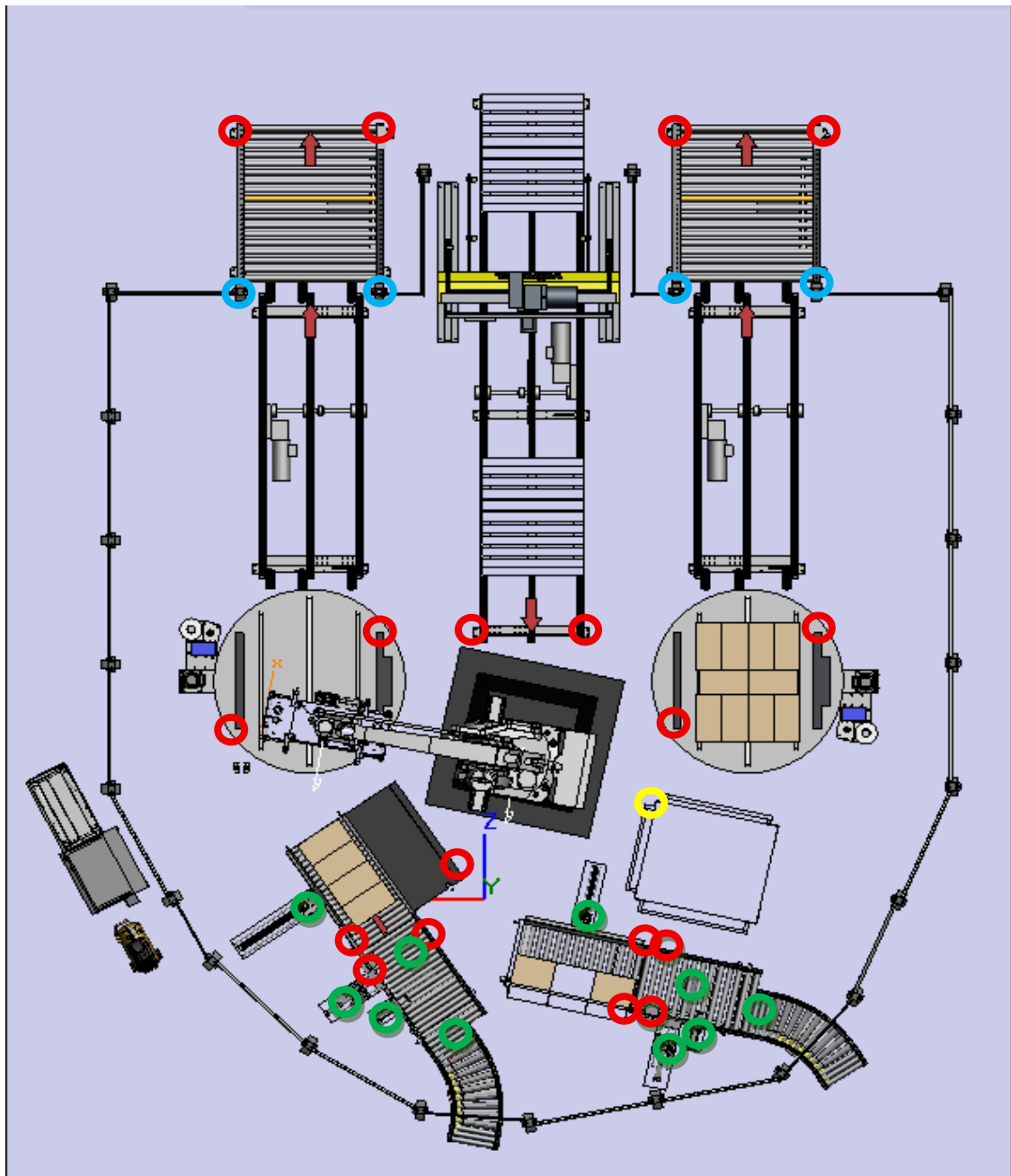


Fig. 6. Sensor mapping of the robotic cell.

4. Conclusions

Within this project, the 3D CAD synthesis and off-line simulation of a robotic palletizing cell were carried out using Process Simulate. The industrial robot kinematics, the end-effector TCP, and the complete structure of the robotic application were defined.

Through the integration of sensors and digital signals, synchronized operation between the robot, conveyors, rotary tables, and auxiliary systems was achieved. In addition, the application flowchart was developed to illustrate the sequence of operations and the operational logic.

The developed simulation reproduces the complete operation of the palletizing and wrapping process, allowing the analysis of the technological flow and the validation of the operational sequences of the robotic application.

5. Bibliography

[1]. *** BW Integrated - Stact Robotic Case Palletizer, 2023, valabil la:
<https://www.bwintegratedsystems.com/products/palletizing/stact-robotic-case-palletizer>

[2]. Barb, I.A., „Celulă robotizată pentru paletizarea baxurilor de cutii de carton cu lapte integrând un robot industrial de tip braț articulat”, *Lucrare de diplomă, București, 2025*

[3]. Chen, S., & Wen, J. T. (2019). Industrial robot trajectory tracking using multi-layer neural networks trained by iterative learning control. *arXiv preprint arXiv:1903.00082*. [4]. Nicolescu, A., *Roboti Industriali*, ISBN 973 – 30 – 1244 – 0, Editura Didactica si Pedagogica RA, cod CNCIS 165, 2005, Bucuresti

[5]. Srasrisom, K., Srinoi, P., Chaijit, S., & Wiwatwongwana, F. (2020). Improvement of an automated CAN packaging system based on modeling and analysis approach through robot simulation tools. *Int. J. Robot. Autom*, 9, 178-189.

[6]. Nicolescu, A. – *Implementarea Robotilor Industriali in Sistemele de Productie*, note de curs si metodologii de proiectare, UPB, 2024

OFF-LINE SIMULATION OF A MULTI-I/O ROBOTIC CELL FOR PALLETIZING AND WRAPPING NESTLÉ PRODUCTS

NEGOITA Adrian-Vasile and Tiberiu Dobrescu (Research Advisor)

¹Faculty: IIR, Major: Robotics, Year of study: I Master's program, e-mail:
adrian.negoita1603@stud.fiir.upb.ro

This paper presents the off-line programming and simulation of a multiple input and output robotic cell for the transport, palletizing, and wrapping of prismatic boxes containing Nestlé products, integrating an ABB IRB 660 four-axis industrial robot. The study begins with a comparative analysis of similar robotic palletizing applications and robot configurations to establish design guidelines. A detailed 3D CAD model of the cell was developed, featuring conveyors, lifts, pallet storages, a rotary table, a wrapping machine, an automated guided vehicle (AGV), and a custom multifunctional end-effector. The material flow logic was mapped through comprehensive flowcharts covering all process decisions and events within the line. Sensor placement was carefully defined, specifically integrating photoelectric and inductive sensors, to enable signal-based simulation and accurate collision detection. The work lays the foundation for virtual commissioning by establishing the architecture needed for PLC integration within the Siemens Process Simulate environment. Overall conclusions highlight the potential of offline simulation to validate the sensing system and optimize motion trajectories prior to physical deployment.

1. Introduction

In this paper, we will address the development of a flexible palletizing line that integrates an IRB 660-180/3.15 four-axis CNC robot manufactured by ABB. The purpose of this paper is to optimize the application's performance by creating a signal-based simulation to eliminate downtime between stages and to verify the sensor system within the cell. The steps completed so far include establishing the system's performance objectives, evaluating and comparing the system with other research in the field, and selecting the simulation system best suited to the requirements imposed by the material flow.

2. Current status

To create a simulation by implementing the virtual commissioning method, we must design—using existing documentation and by comparing with other industrial palletizing applications in the field—a flexible production cell that meets the performance requirements. For this step, we must create comprehensive flowcharts of the material flow to help us select and map the necessary sensor components, such as photoelectric and inductive sensors, for connection to a PLC and to perform a robust signal-based simulation.

3.1 Comparative Analysis of Similar Robotic Applications

To establish a robust design foundation, a comparative analysis of existing industrial palletizing setups was conducted.

The first reference application features a dual-robot setup utilizing two 4-axis Fanuc industrial robots. These robots are equipped with custom vacuum end-effectors comprising three adjustable vacuum islands, which dynamically adapt to the specific dimensions of the handled parts. The handled items consist of parallelepiped-shaped boxes containing diverse food products. The material flow is managed through four input conveyors, from which the robots pick the items and systematically arrange them onto pallets. A continuous supply of wooden pallets is maintained by automated pallet storage units, with internal transport facilitated by a rail-guided vehicle (RGV) and a network of roller conveyors. Upon completion of the palletizing cycle, the fully stacked pallet is transferred via the roller conveyors and the RGV to the designated outfeed conveyor.



Fig.1 Robotic Application

The second analyzed application integrates a 4-axis CNC-controlled ABB robot, which closely resembles the architecture targeted in this study. This system is equipped with a highly versatile multifunctional end-effector designed to handle cardboard boxes, layer separators, and wooden pallets. This versatility is achieved by combining mechanical clamping mechanisms with vacuum-based gripping systems. The continuous flow of rectangular cardboard boxes is managed by belt conveyors. Empty pallets are retrieved from storage areas and transported by an RGV to the infeed conveyor within the active palletizing zone. During the palletizing sequence, after a complete layer of boxes is formed, the robot utilizes the vacuum cups on its end-effector to pick and place a layer separator. Once the full stack is completed, it is discharged via the roller conveyors onto the RGV, which subsequently transfers the finalized pallet to the cell's outfeed conveyor system.

3.2 3D CAD model of the robotic application in the work environment where it will be implemented, and its offline programming and simulation

The 3D modeling of the flexible manufacturing line was meticulously executed using CATIA V5. The layout of the subsystems specific to the flexible palletizing line reveals several key architectural features. Primarily, the line is characterized by the presence of industrial robots, multiple pallet storage units intended to ensure a continuous flow of pallets of varying types and dimensions, and a rail-guided vehicle (RGV) dedicated to transporting pallets across the conveyor network.

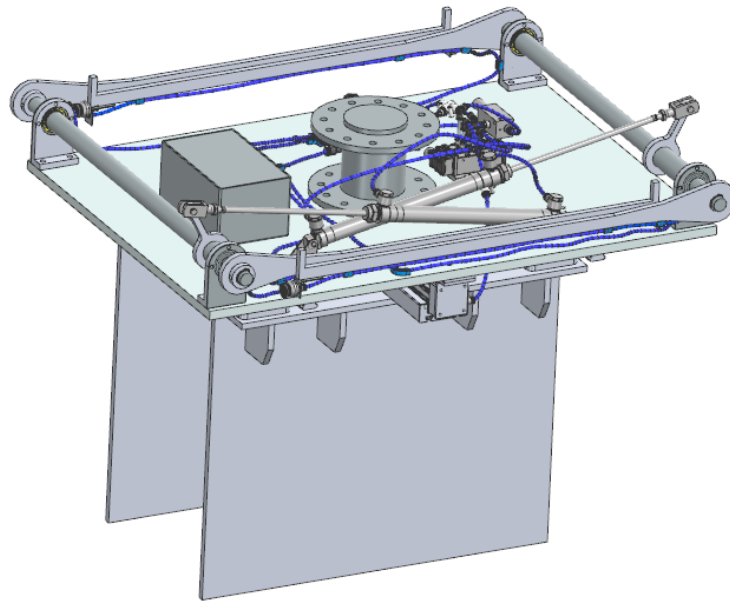


Fig. 2

The operational flow begins at a main feed conveyor. Here, an integrated vision sensor scans the barcode on incoming boxes. Based on the specific type and dimensions extracted from the barcode data, each box is systematically directed onto a designated branch of the central conveyor. The categorization of the box strictly dictates its routing path. A rotating pusher mechanism then transfers the sorted boxes onto the appropriate conveyor, guiding them toward the robot's designated pickup zone.

Simultaneously, the pallet feeding process is sustained by automated storage units, which are periodically replenished by human operators. Pallets dispensed from these units are conveyed to the industrial robot's primary workspace via an integrated system of roller conveyors and the RGV. This transport system is highly adaptable, designed to handle various pallet footprints seamlessly.

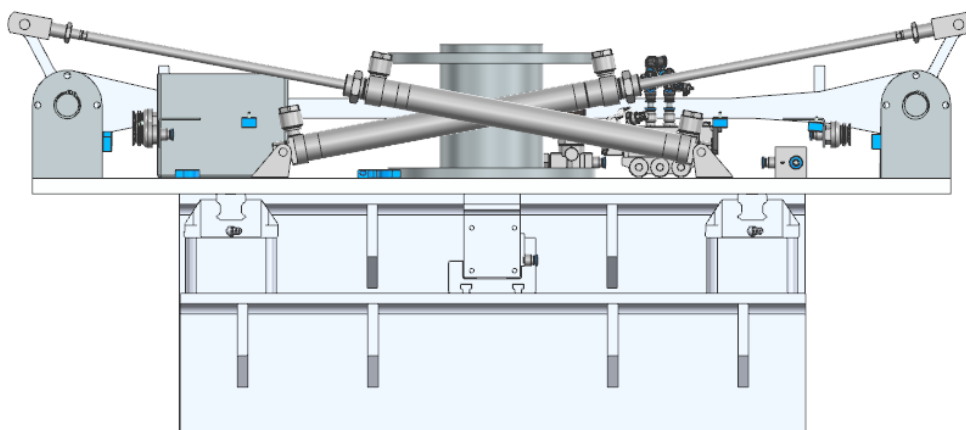


Fig. 3

Upon the completion of a full stack by the robot, the loaded pallet is advanced toward the transfer conveyors. These specific conveyors are engineered with a dual-section mechanism. The primary fixed section consists of a roller conveyor that acts as a bridge between the robot's outfeed rollers and a secondary chain conveyor system. To execute the transfer, the chain mechanism elevates above the roller level to lift the pallet, transports it forward, and then retracts below the roller surface. This

sequence effectively moves the finalized stack toward a connecting conveyor that feeds directly into an arm-type wrapping machine.

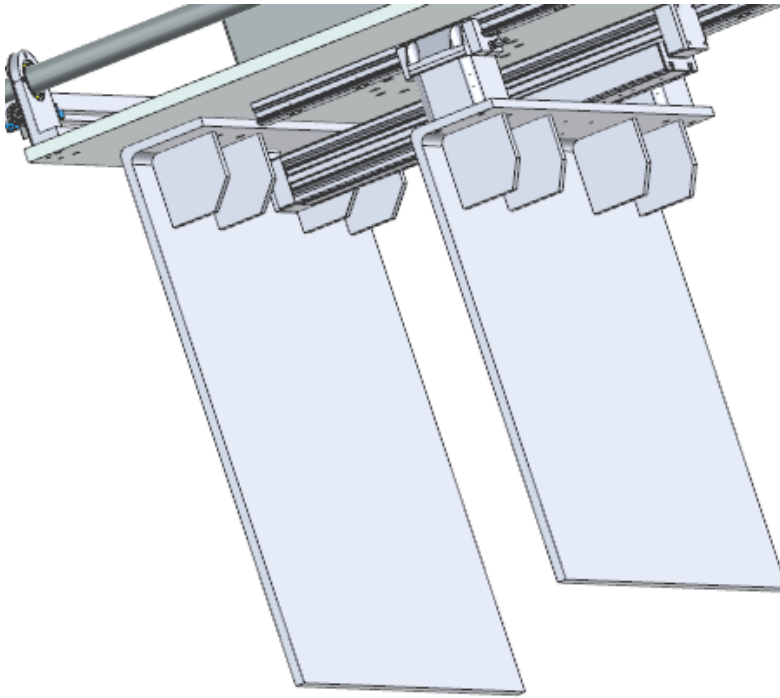


Fig. 4

To achieve accurate simulation within the Process Simulate environment, comprehensive kinematic definitions were established for all dynamic components. This involved defining constraint graphs and specific positional states for the multifunctional end-effector, the pneumatic pushers, and the transport conveyors, thereby enabling a realistic sequence of operations during the virtual validation phase.

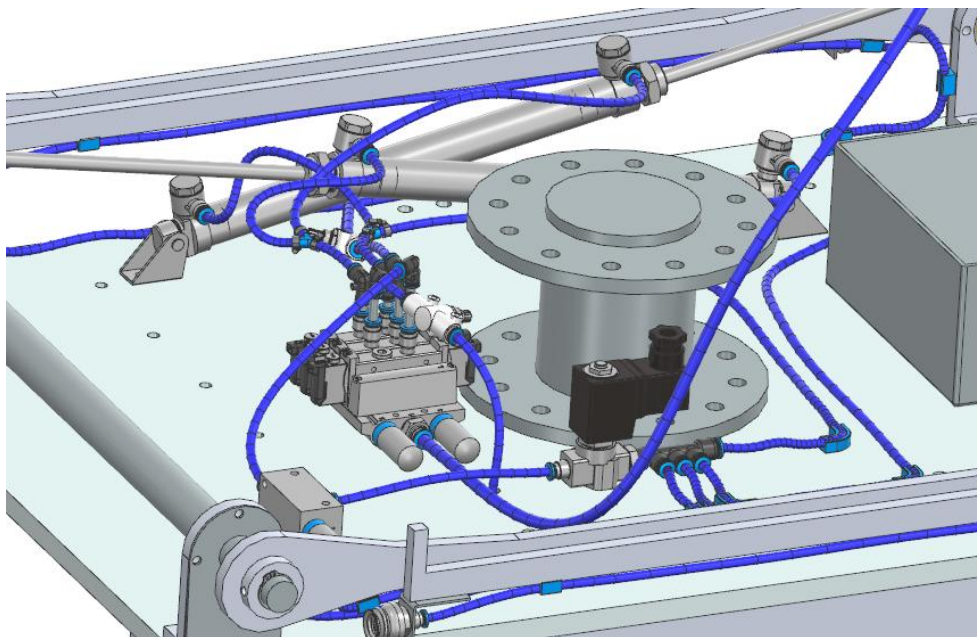


Fig. 5

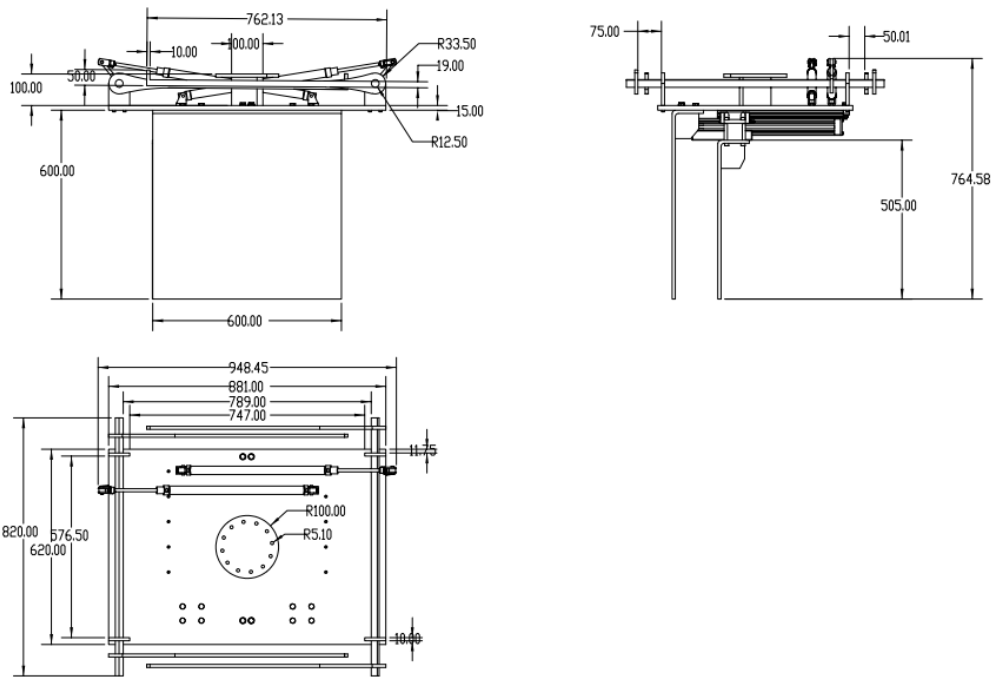


Fig. 6

3.3 Development of the flowchart detailing all processes and events + sensor mapping of the entire robotic application

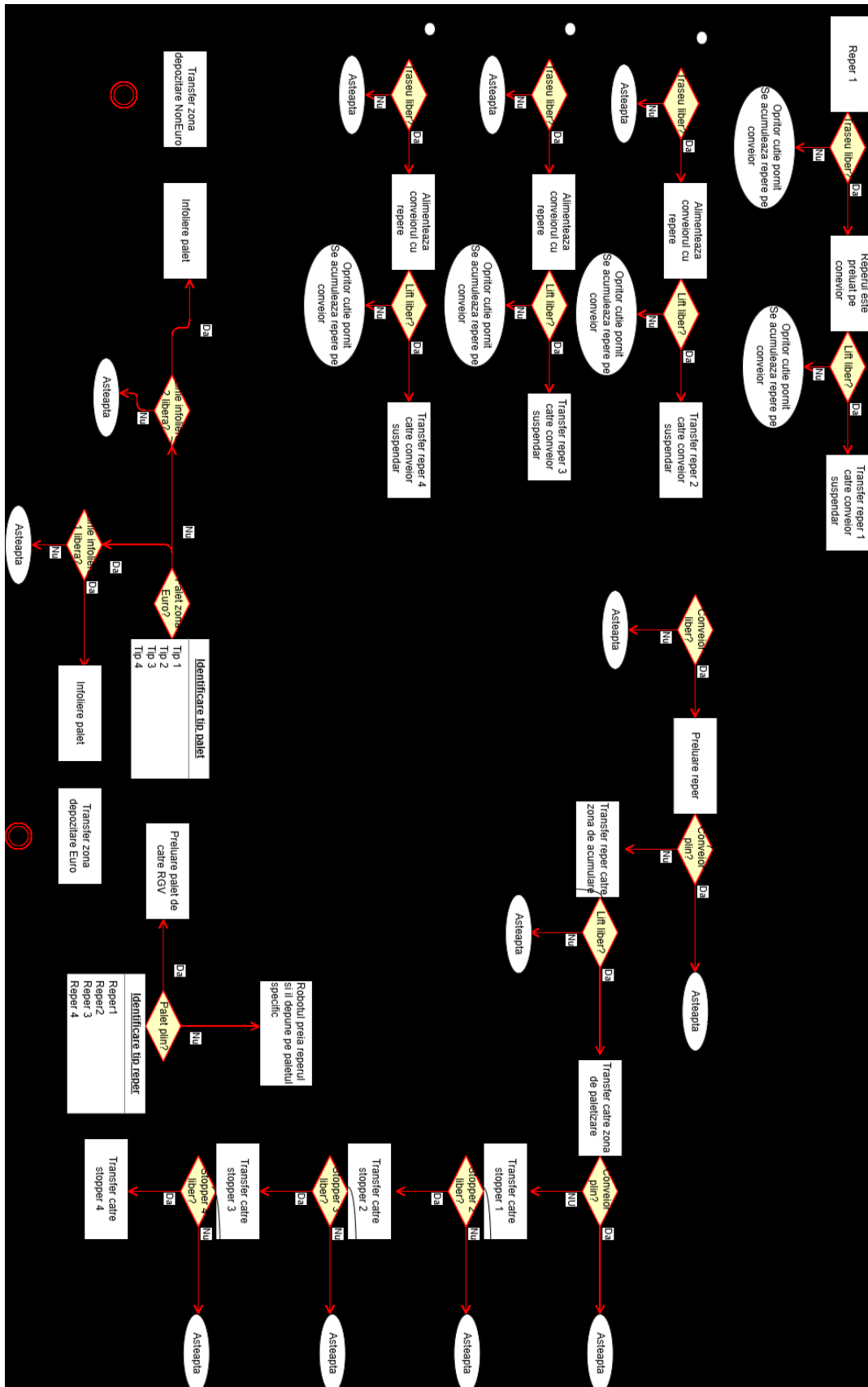


Fig. 7

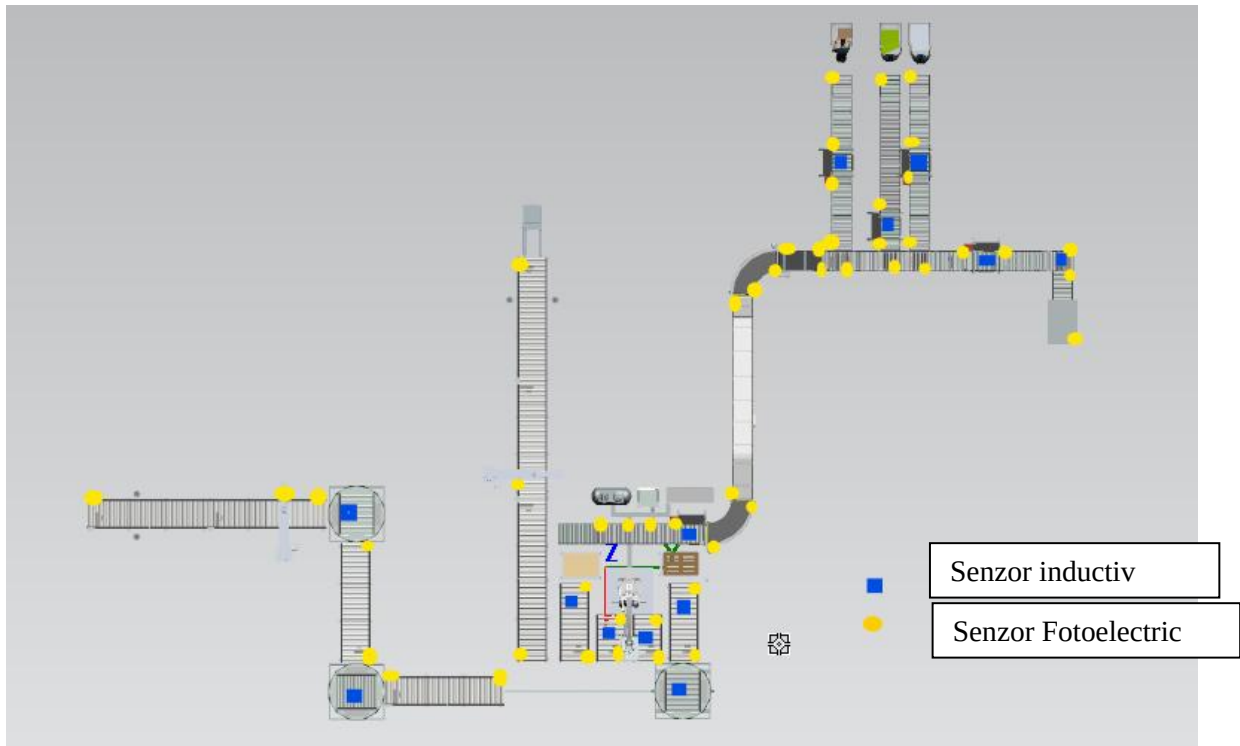
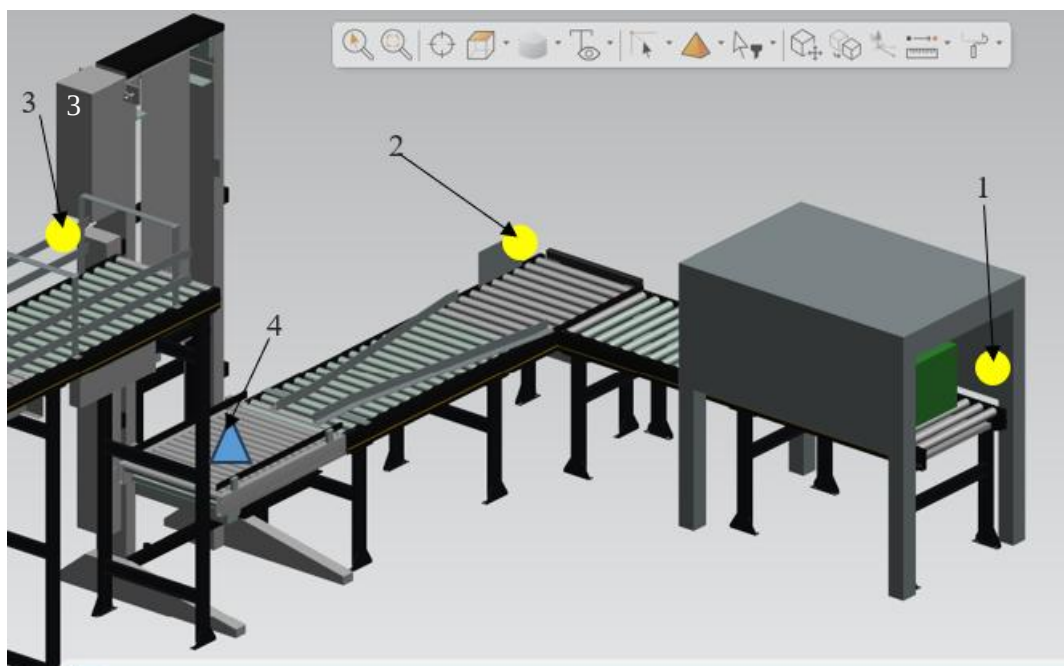
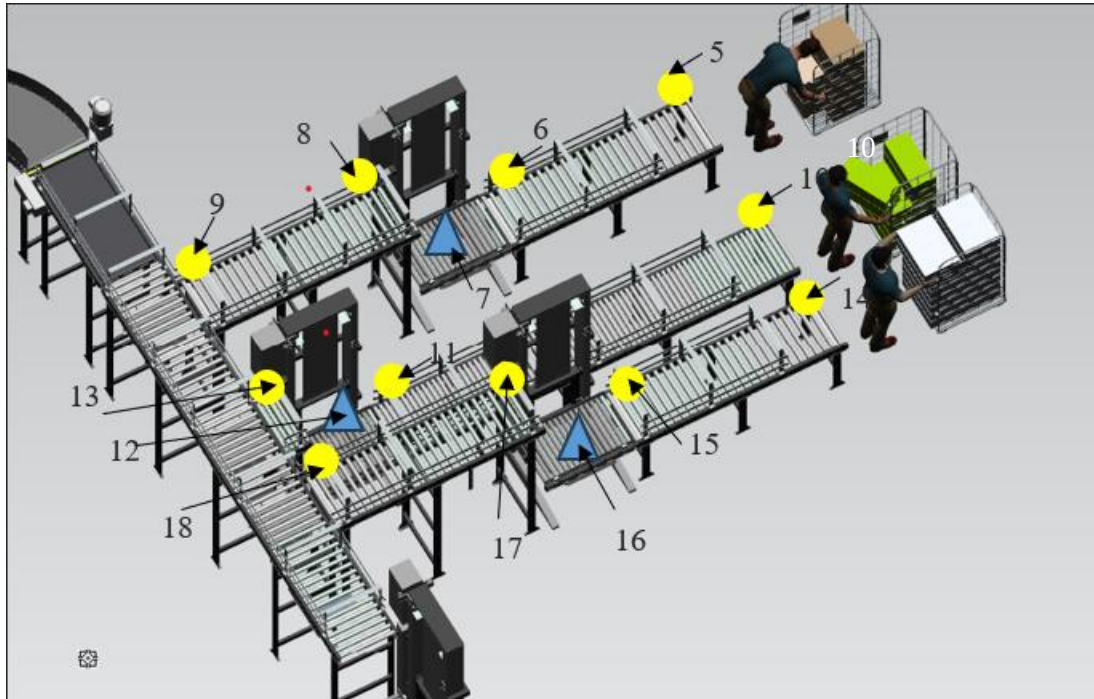


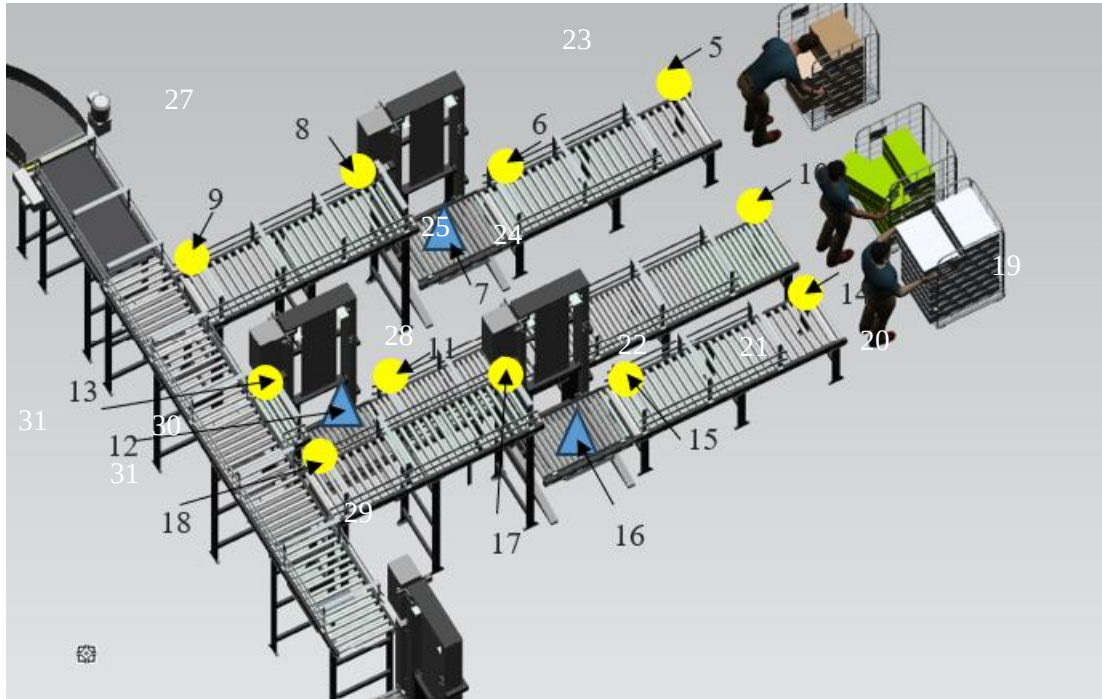
Fig. 8



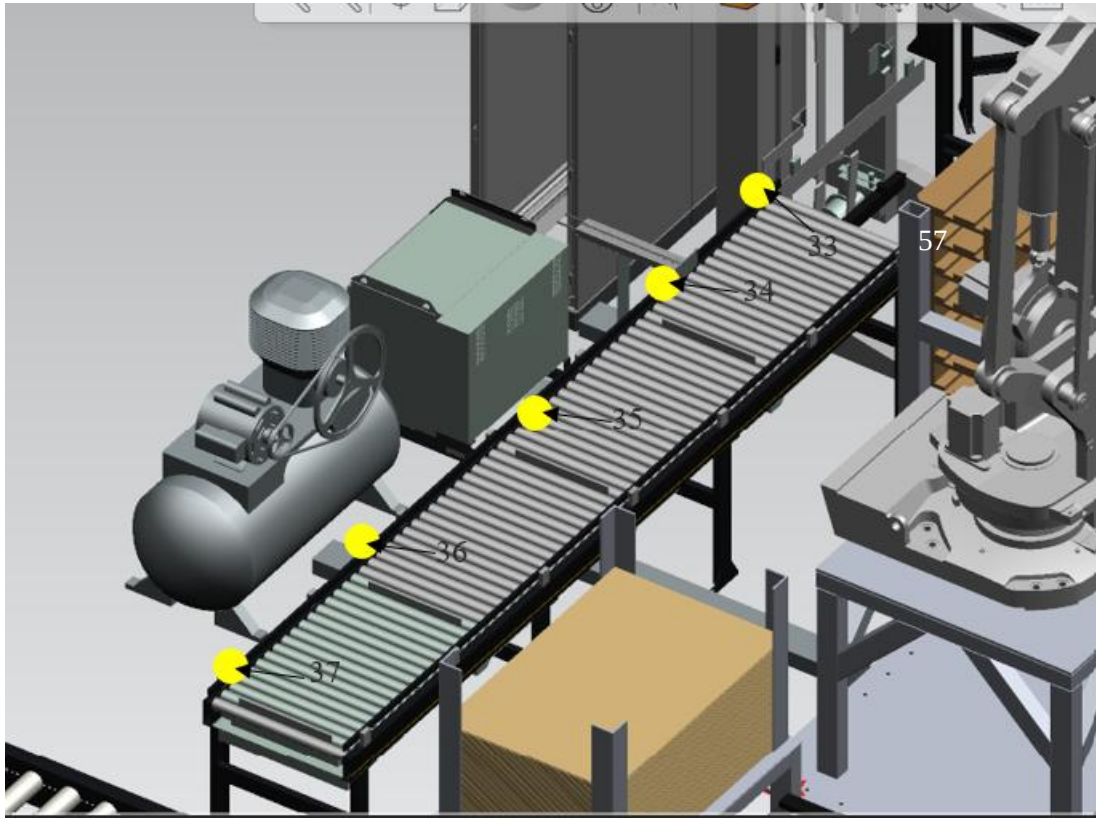
1. Photoelectric sensor for detecting box entry
2. Photoelectric sensor for detecting boxes at the end of the conveyor
3. Inductive sensor for detecting presence on the lift
4. Photoelectric sensor for detecting boxes at the beginning of the conveyor



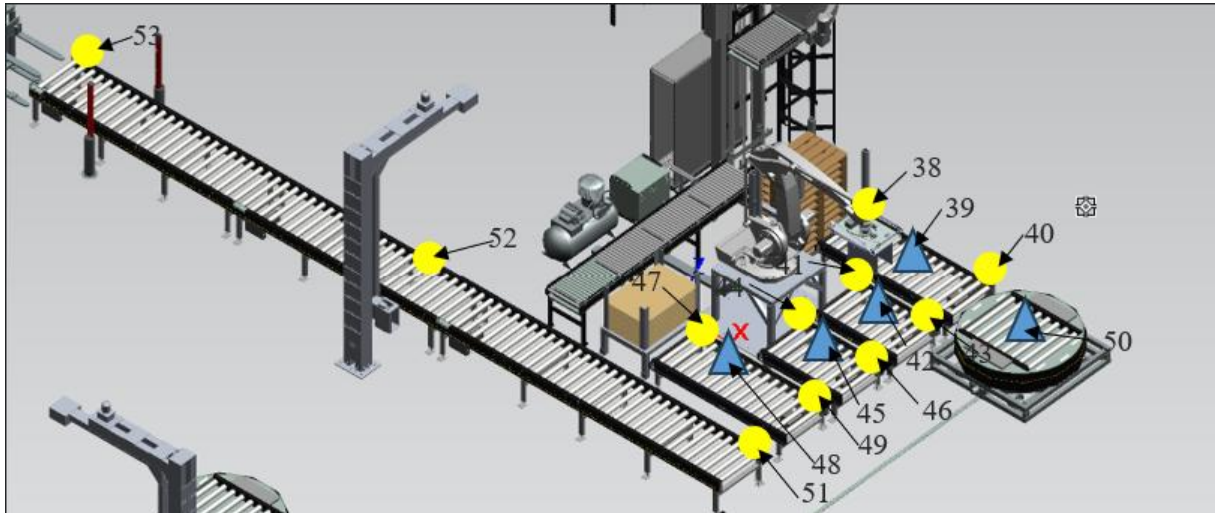
- 5. Photoelectric sensor for detecting box entry
- 6. Photoelectric sensor for detecting boxes at the end of the conveyor
- 7. Inductive sensor for detecting presence on the lift
- 8. Photoelectric sensor for detecting boxes at the beginning of the conveyor
- 9. Photoelectric sensor for detecting boxes at the end of the conveyor
- 10. Photoelectric sensor for detecting box entry
- 11. Photoelectric sensor for detecting boxes at the end of the conveyor
- 12. Inductive sensor for detecting presence on the lift
- 13. Photoelectric sensor for detecting boxes on the conveyor
- 14. Photoelectric sensor for detecting box entry
- 15. Photoelectric sensor for detecting boxes at the end of the conveyor
- 16. Inductive sensor for detecting presence on the lift
- 17. Photoelectric sensor for detecting boxes at the beginning of the conveyor
- 18. Photoelectric sensor for detecting boxes at the end of the conveyor



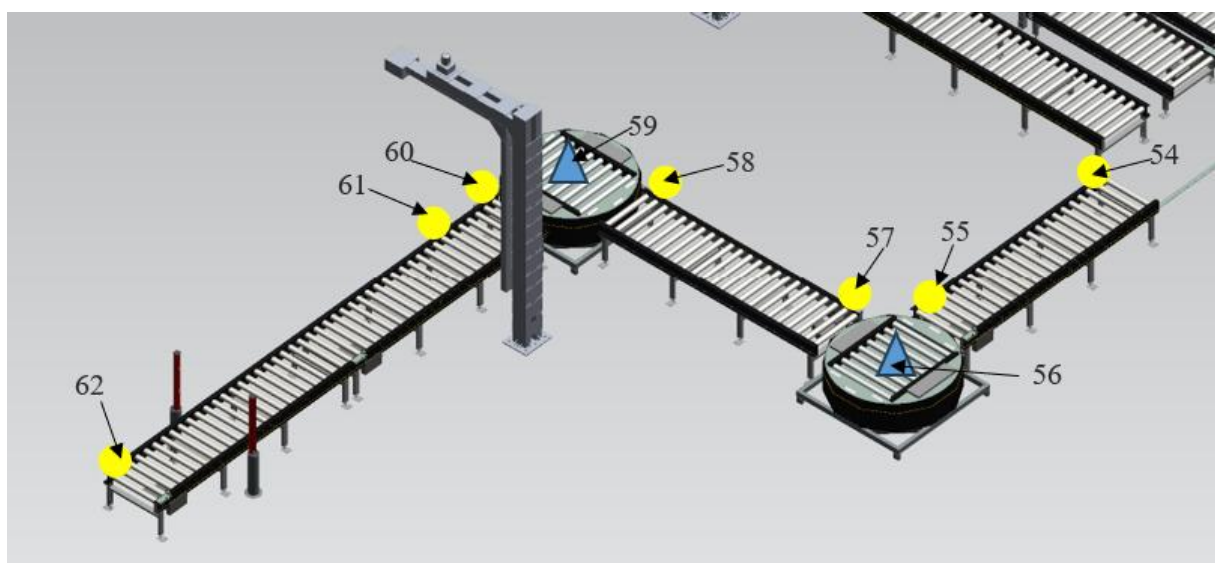
- 19. Photoelectric sensor for detecting boxes at the beginning of the conveyor
- 20. Photoelectric sensor for detecting boxes on transfer system 1
- 21. Photoelectric sensor for detecting boxes on transfer system 2
- 22. Photoelectric sensor for detecting boxes on transfer system 3
- 23. Photoelectric sensor for detecting boxes at the end of the conveyor
- 24. Photoelectric sensor for detecting boxes at the beginning of the conveyor
- 25. Photoelectric sensor for detecting boxes at the end of the conveyor
- 26. Photoelectric sensor for detecting boxes at the beginning of the conveyor
- 27. Photoelectric sensor for detecting boxes at the end of the conveyor
- 28. Photoelectric sensor for detecting boxes at the beginning of the conveyor
- 29. Photoelectric sensor for detecting boxes at the end of the conveyor
- 30. Photoelectric sensor for detecting boxes at the beginning of the conveyor
- 31. Photoelectric sensor for detecting boxes at the end of the conveyor
- 32. Inductive sensor for detecting presence on the lift



- 33. Photoelectric sensor for detecting boxes at the beginning of the conveyor
- 34. Photoelectric sensor for detecting boxes at accumulation point 1
- 35. Photoelectric sensor for detecting boxes at accumulation point 2
- 36. Photoelectric sensor for detecting boxes at accumulation point 3
- 37. Photoelectric sensor for detecting boxes at the end of the conveyor



- 38. Photoelectric sensor for detecting pallets at the beginning of the conveyor
- 39. Inductive sensor for detecting conveyor presence
- 40. Photoelectric sensor for detecting pallets at the end of the conveyor
- 41. Photoelectric sensor for detecting pallets at the beginning of the conveyor
- 42. Inductive sensor for detecting conveyor presence
- 43. Photoelectric sensor for detecting pallets at the end of the conveyor
- 44. Photoelectric sensor for detecting pallets at the beginning of the conveyor
- 45. Inductive sensor for detecting conveyor presence
- 46. Photoelectric sensor for detecting pallets at the end of the conveyor
- 47. Photoelectric sensor for detecting pallets at the beginning of the conveyor
- 48. Inductive sensor for detecting conveyor presence
- 49. Photoelectric sensor for detecting pallets at the end of the conveyor
- 50. Inductive sensor for detecting RGV presence
- 51. Photoelectric sensor for detecting pallets at the beginning of the conveyor
- 52. Photoelectric sensor for detecting pallets at the wrapping point
- 53. Photoelectric sensor for detecting pallets at the end of the conveyor



- 54. Photoelectric sensor for detecting pallets at the beginning of the conveyor
- 55. Photoelectric sensor for detecting pallets at the end of the conveyor
- 56. Inductive sensor for detecting presence on the rotary table
- 57. Photoelectric sensor for detecting pallets at the beginning of the conveyor
- 58. Photoelectric sensor for detecting pallets at the end of the conveyor
- 59. Inductive sensor for detecting presence on the rotary table
- 60. Photoelectric sensor for detecting pallets at the beginning of the conveyor
- 61. Photoelectric sensor for detecting pallets at the wrapping point
- 62. Photoelectric sensor for detecting pallets at the end of the conveyor

4. Conclusions

The offline programming and simulation of the flexible palletizing cell using Siemens Process Simulate successfully validated the system's design and operational logic. By integrating an ABB IRB 660 robot with a custom multifunctional end-effector and mapping an extensive network of photoelectric and inductive sensors, the study achieved a highly accurate, signal-based digital twin.

Ultimately, this study demonstrates that offline simulation significantly minimizes implementation risks, eliminates production downtime during the testing and programming phases, and ensures a highly efficient transition from virtual design to physical deployment.

5. Bibliography

- 1. Anania D. – Fabricatie asistata, note de curs UPB, 2024
- 2. Bucuresteanu A. – Actionarea Pneumatica a Robotilor Industriali, note de curs UPB, 2023
- 3. Bucuresteanu A. – Elemente si sisteme pneumatice pentru actionarea robotilor industriali, Editura Printech, ISBN 978-606-23-0081-4, Bucuresti 2013.
- 4. Cristoiu C. – Proiectare asistata de calculator 3 - NX CAD, note de curs si suport aplicatii, UPB, 2023
- 5. Cristoiu C. A., Nicolescu A. F. - NX CAD-Basic - Proiectare asistată de calculator, 96 pag., Ed. Politehnica Press, 2020, Cod CNCSIS 19, ISBN 978-606-515-914-3, CIP BNR 2020-13437
- 6. Nicolescu, A. – Implementarea Roboților Industriali în Sistemele de Producție, note de curs și metodologii de proiectare, UPB, 2024;
- 7. Enciu G., Popescu A. – Senzori Industriali, note de curs, UPB, 2022
- 8. Nicolescu A., – Componente mecanice tipizate, note de curs si metodologii de proiectare, UPB, 2021

PROGRAMMING AND SIMULATING A FLEXIBLE PALLETIZING LINE USING THE PROCESS SIMULATE ENVIRONMENT

TOITAN Laurențiu-Florentin and ANANIA Florea Dorel(Research Advisor)

¹Faculty: IIR, Major: Robotics, Year of study: I Master's program, e-mail:laurentiuflorentin2018@gmail.com

This paper presents the programming and simulation of a flexible palletizing line integrating two ABB IRB 660 four-axis industrial robots, continued to be developed with the Siemens Process Simulate environment. The study begins with an analysis of similar robotic palletizing applications to establish performance benchmarks and guide the design process. A detailed 3D CAD model of the cell was created in CATIA V5, with conveyors, pallet storages, a rail-guided vehicle, and a multifunctional end-effector. The material flow logic was mapped through comprehensive flowcharts covering all process decisions and events within the line. Sensor placement was defined, including photoelectric, capacitive, barcode reader, and safety barrier sensors, to enable signal-based simulation. The work lays the foundation for virtual commissioning by establishing the signal architecture needed for PLC integration in Process Simulate. Overall conclusions highlight the potential of offline simulation to eliminate idle times and validate the sensing system before physical deployment.

1. Introduction

In this paper, we will address the issue of a flexible palletizing line that integrates two IRB 660/-180/3.15 four-axis CNC robots manufactured by ABB. The purpose of this paper is to optimize the application's performance by creating a signal-based simulation to eliminate downtime between stages and to verify the sensor system within the cell. The steps completed so far include establishing the system's performance objectives, evaluating and comparing the system with other research in the field, and selecting the simulation system best suited to the requirements imposed by the material flow.

2. Current status

To create a simulation by implementing the virtual commissioning method, we must design—using existing documentation and by comparing with other industrial applications in the field—a flexible production line that meets the performance requirements. For this step, we must create flowcharts to help us select the necessary sensor components for connection to a PLC and to perform signal-based simulation.

3.1 Comparative analysis of robotic applications similar to the one being designed

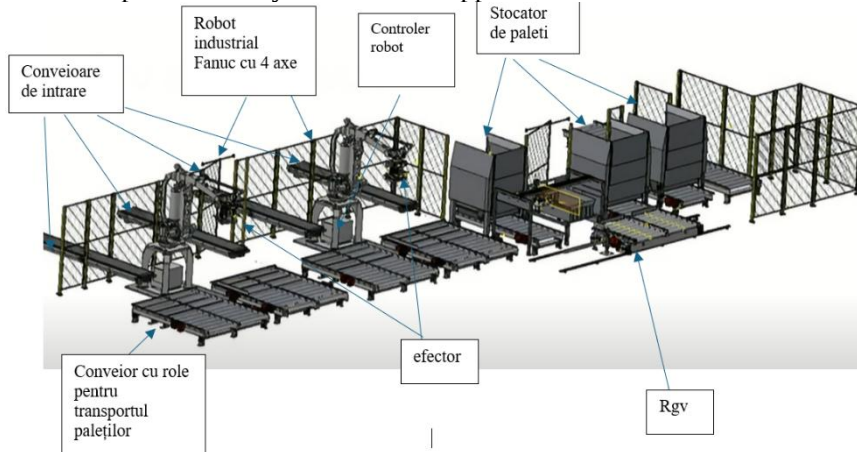


Fig.1 Similar Robotic Application

In this palletizing application, there are two 4-axis Fanuc robots equipped with a vacuum gripper consisting of three vacuum islands that can adjust their position based on the dimensions of the part being handled. The parts are parallelepiped-shaped boxes containing various food products.

The material flow is handled by the four input conveyors. The robot and the end-effector pick up the items and arrange them in an orderly fashion on a pallet. A constant flow of wooden pallets is provided by the pallet storage units, and their transport to the work area is handled by a rail-guided vehicle (RGV) as well as roller conveyors. Once the stack has been formed, the pallet is transported by the same roller conveyors and the RGV to the roller conveyor at the cell's exit.

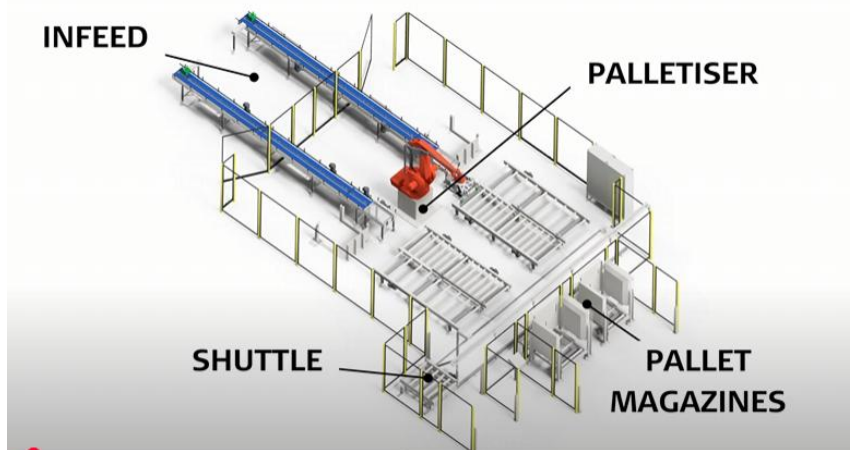


Fig.2 Overview of the cell's essential components

In this application, we have a 4-axis CNC-controlled ABB robot equipped with a multifunctional gripper effector capable of handling box-type parts, dividers, and wooden pallets using both bucket-type and vacuum-type gripping systems. The flow of rectangular cardboard boxes is handled by belt conveyors. Pallets are transported from the pallet storage areas via a rail-guided vehicle to the pallet feed conveyor in the box palletizing zone. Once the robot has finished filling a row of boxes, a layer separator must be inserted. This is picked up by the vacuum system of the end-effector using vacuum cups. Once the stack is complete, it is transported by the pallet roller conveyors to the RGV (rail-guided vehicle), which picks up the stack and then positions it on the conveyor leading out of the cell.

3.2 3D CAD model of the robotic application in the work environment where it will be implemented, and its offline programming and simulation

Figures 3, 4, and 5 show the layout of the subsystems specific to the flexible palletizing line, as well

PROGRAMMING AND SIMULATING A FLEXIBLE PALLETIZING LINE USING THE PROCESS SIMULATE ENVIRONMENT

as its floor plan. From Figure 3, we can identify several key features of the flexible palletizing line. These include: the presence of two industrial robots; the presence of three pallet storage units to ensure a continuous flow of pallets of various types and sizes; and the presence of a rail-guided vehicle to transport pallets on the pallet conveyors, etc.

The flexible manufacturing line was modeled in 3D using CATIA V5.

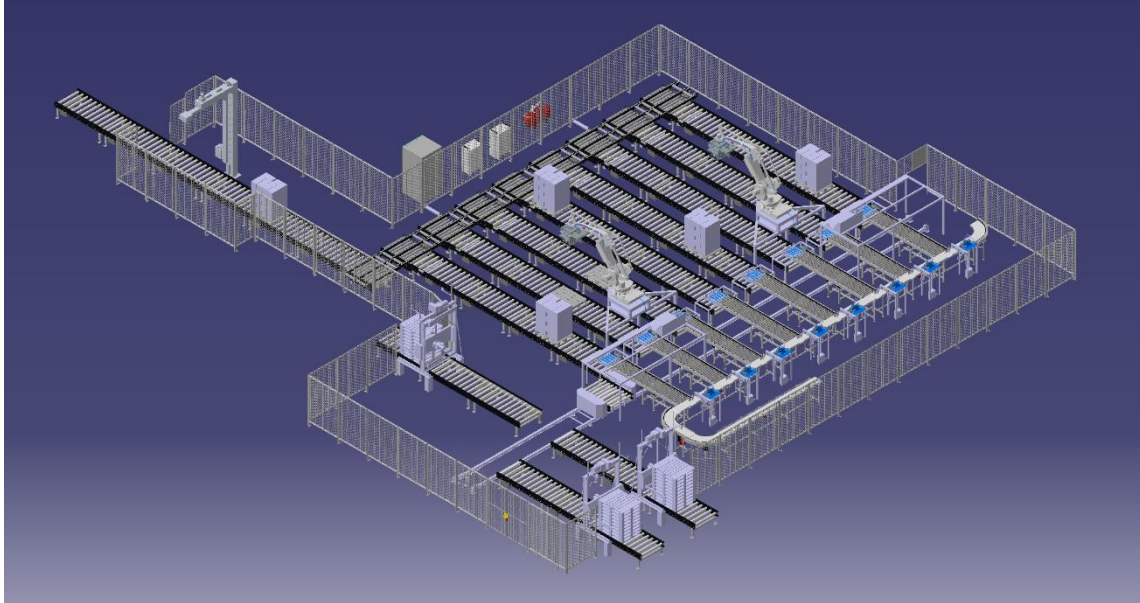


Fig.3 Isometric view of the flexible manufacturing line

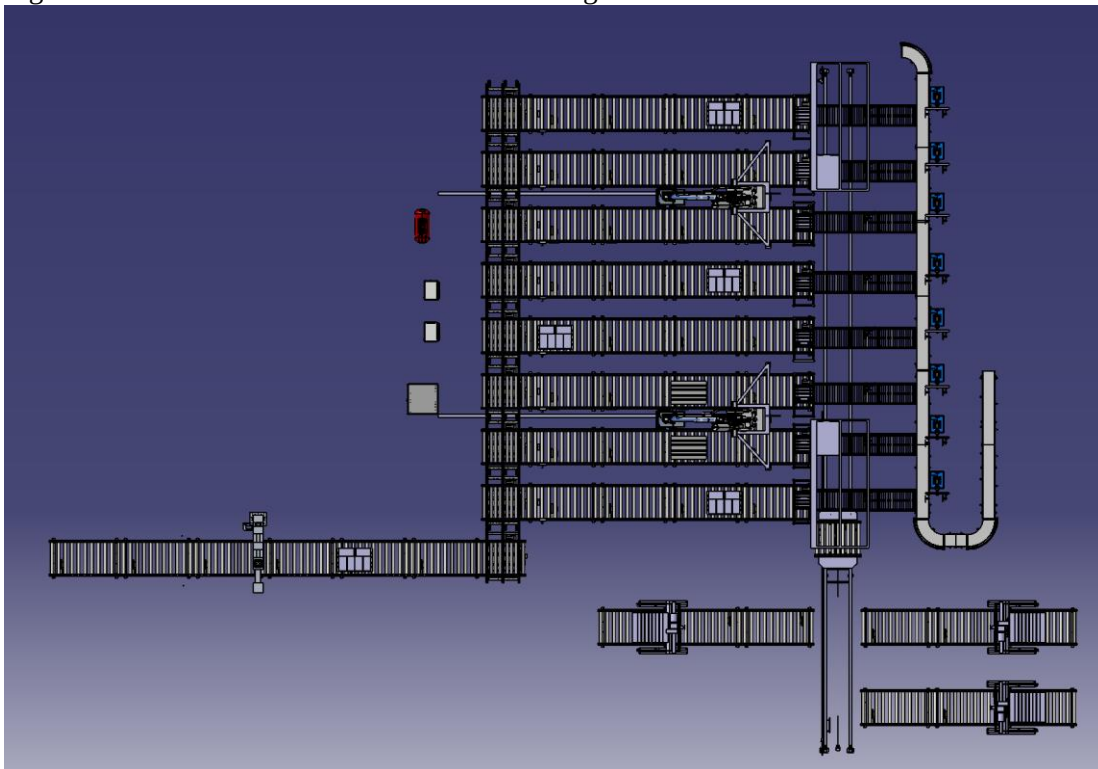


Fig.4 Top view

The cell has a main feed conveyor; after a barcode on the boxes is scanned using a vision sensor, each box is directed onto a branch of the central conveyor based on its type and dimensions. The type of box determines which conveyor it will be transported on. The boxes proceed to the robot's pickup

area after being pushed onto the appropriate conveyor by a rotating box pusher.

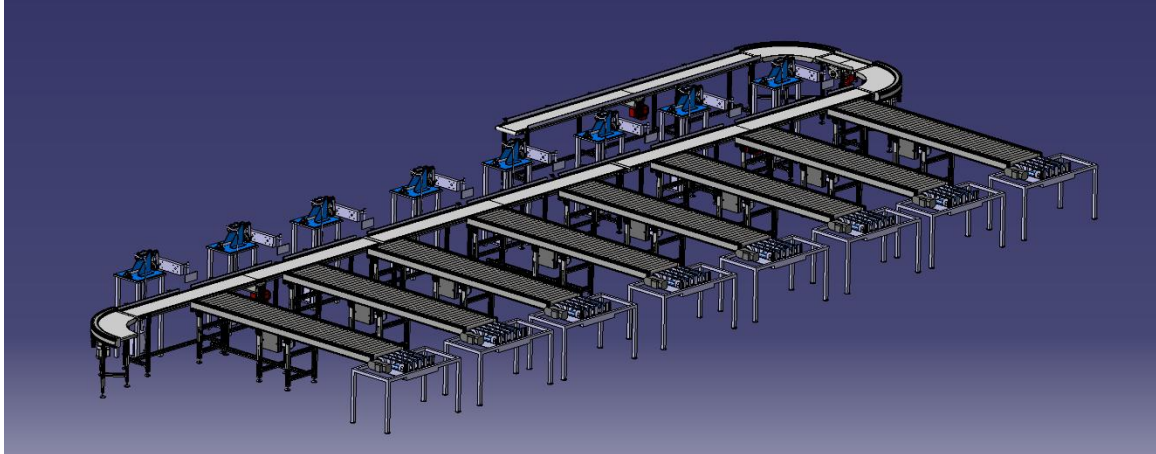


Fig.5 Isometric view of the box entry system into the workspace

Pallet feeding is performed by pallet storage units that are periodically replenished by operators. The pallets released from the storage units are transported to the industrial robot's work area by a roller conveyor system and a rail-guided vehicle that is perfectly adapted to different types of pallets and compatible with the roller conveyors present in the cell.

Once the stack is complete, it is transported to the transfer conveyors (Fig.6). These conveyors consist of two sections. The first part is the roller conveyor (fixed), which serves as the link between the roller conveyors coming from the robot and the chain conveyors (which move to pick up the pallet by lowering below the level of the rollers and then rising above them to transport it on the chain). These conveyors transfer the stack to the conveyor that connects to the arm-type wrapper.

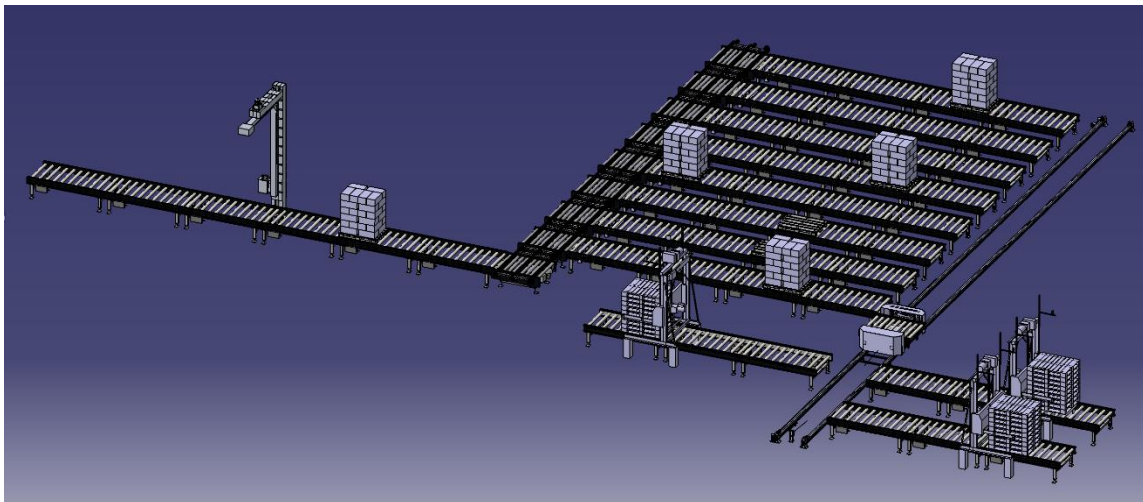


Fig.6 Isometric view of the system for feeding pallets into the workspace and discharging stacks outside the palletizing line

**PROGRAMMING AND SIMULATING A FLEXIBLE PALLETIZING LINE USING THE PROCESS
SIMULATE ENVIRONMENT**

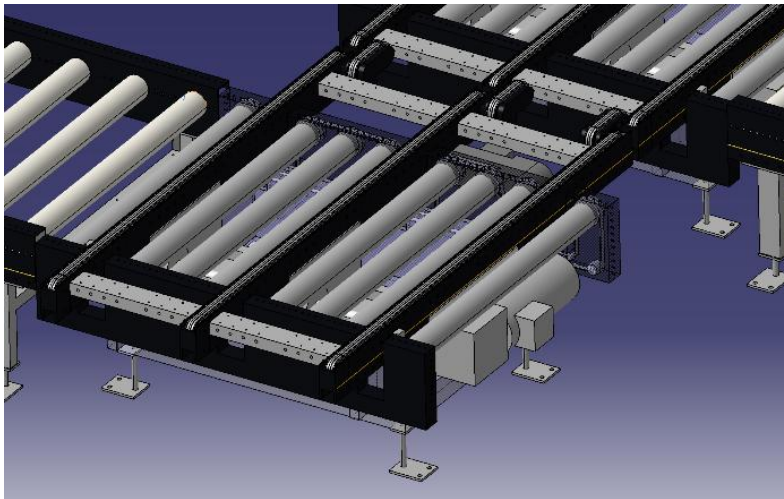


Fig.7 Roller-and-chain transfer conveyors

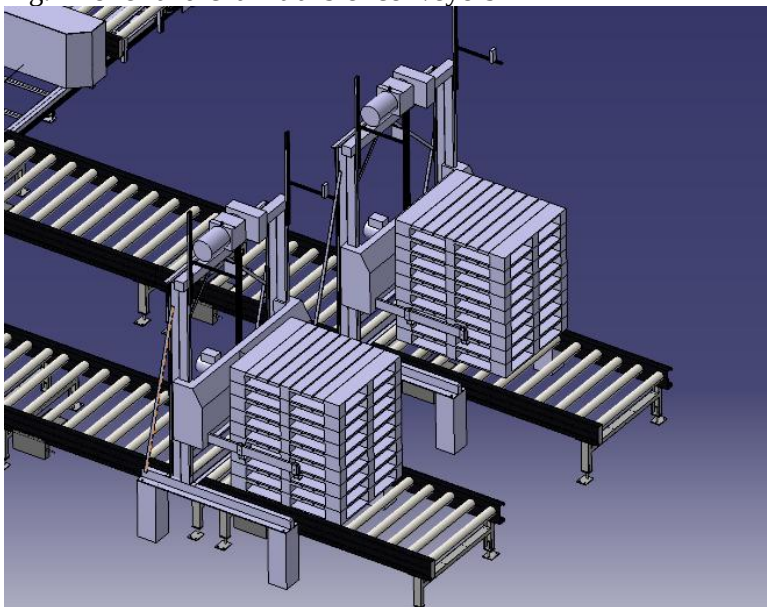
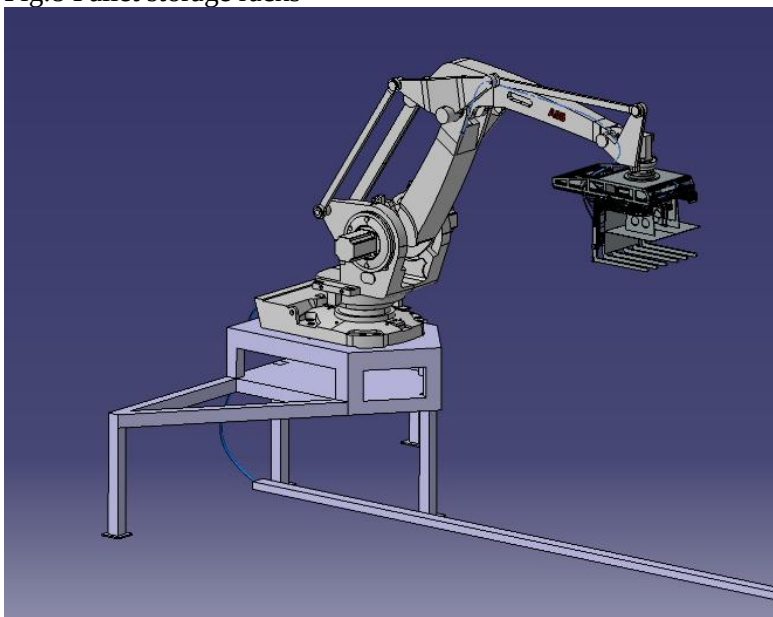


Fig.8 Pallet storage racks



PROGRAMMING AND SIMULATING A FLEXIBLE PALLETIZING LINE USING THE PROCESS SIMULATE ENVIRONMENT

Fig.9 Lifting system + industrial palletizing robot equipped with a multifunctional end-effector

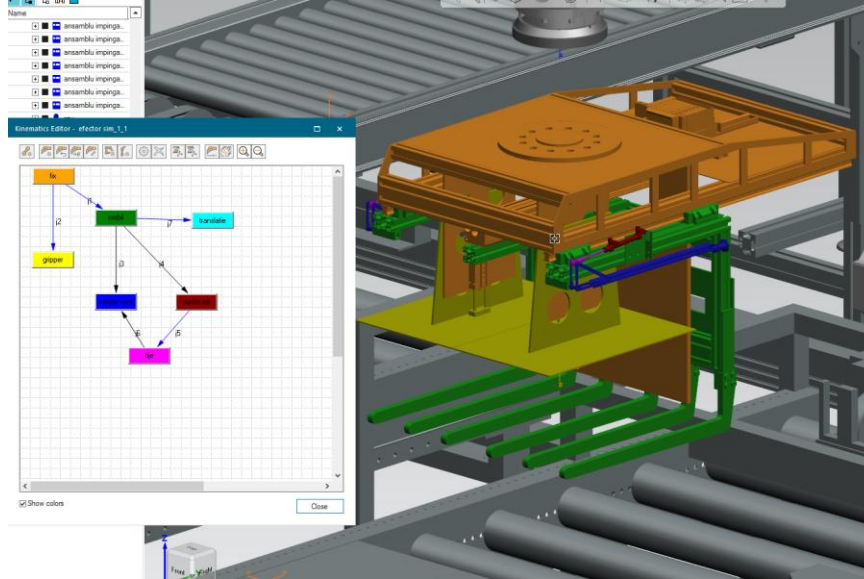


Fig 10 Connections between components to achieve the end-effector's kinematics in the simulated process environment

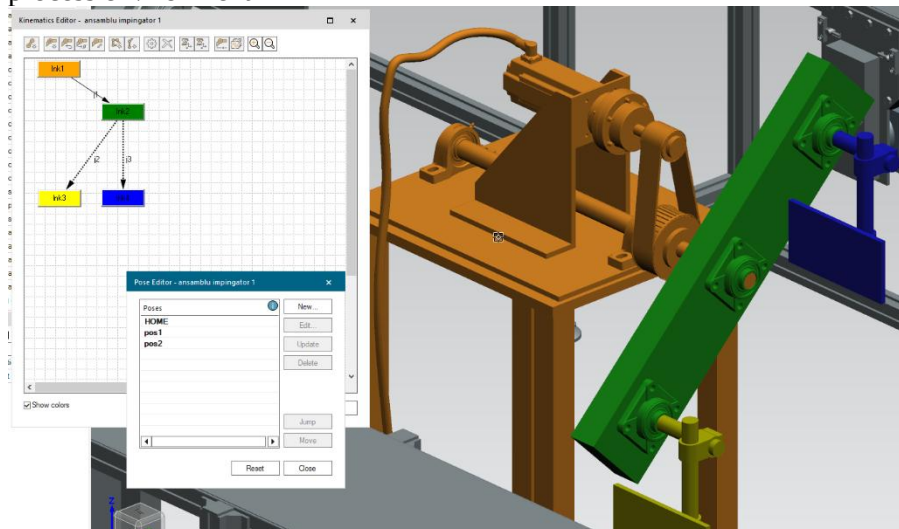
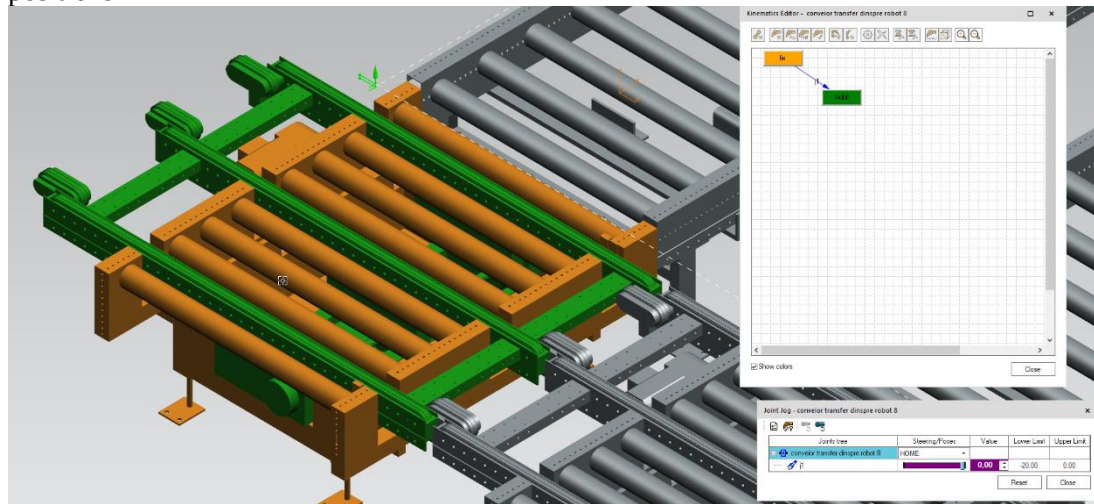


Fig 11 Image showing the constraint graph for the pusher's kinematics and the definition of its positions



PROGRAMMING AND SIMULATING A FLEXIBLE PALLETIZING LINE USING THE PROCESS SIMULATE ENVIRONMENT

Fig 12 Image showing the constraint graph for the transport conveyor's kinematics and the definition of its positions



Fig 13 Example view of operations in the simulation



Fig 14 Sequence from the simulation of the flexible line created in Process Simulate

PROGRAMMING AND SIMULATING A FLEXIBLE PALLETIZING LINE USING THE PROCESS SIMULATE ENVIRONMENT

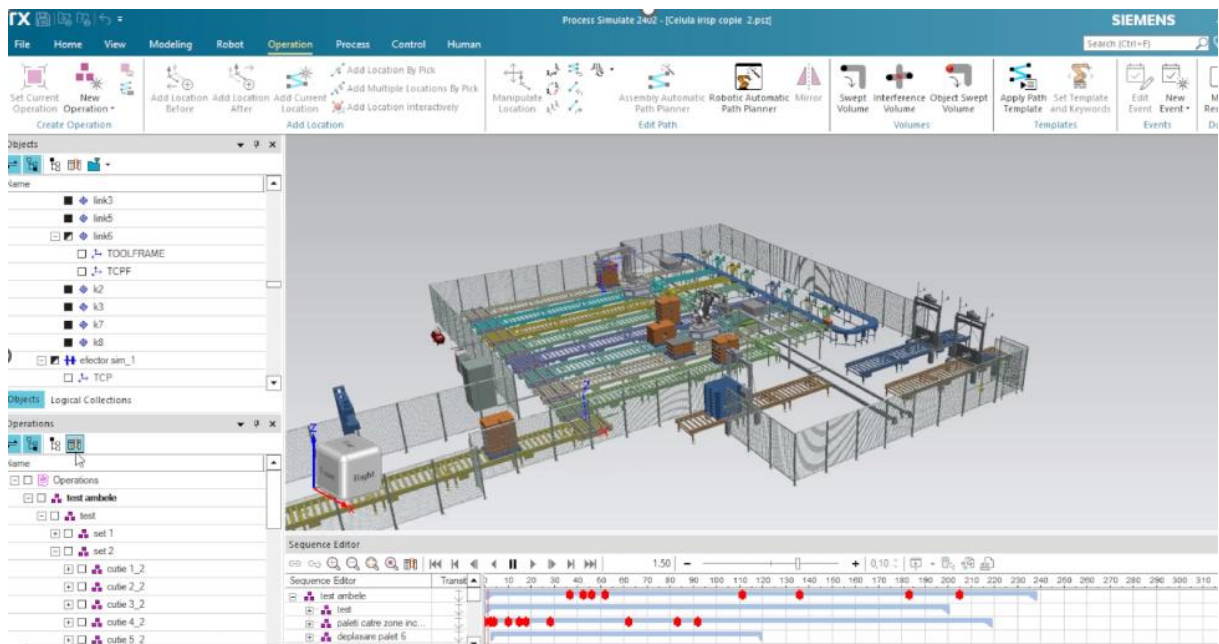


Fig 15 Image taken from the Process Simulate software showing the implemented application

3.3 Development of the flowchart detailing all processes and events + sensor mapping of the entire robotic application

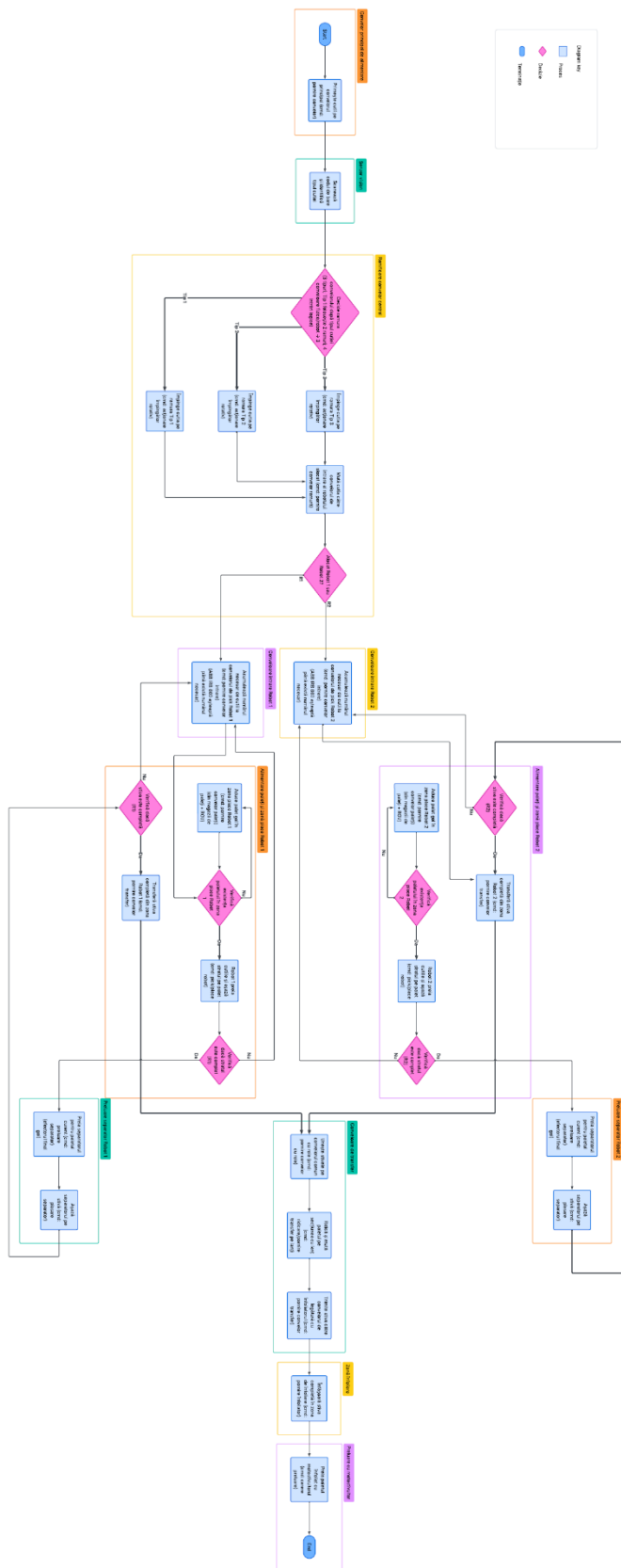


Fig.16 Flowchart describing decisions and events in the flexible palletizing line

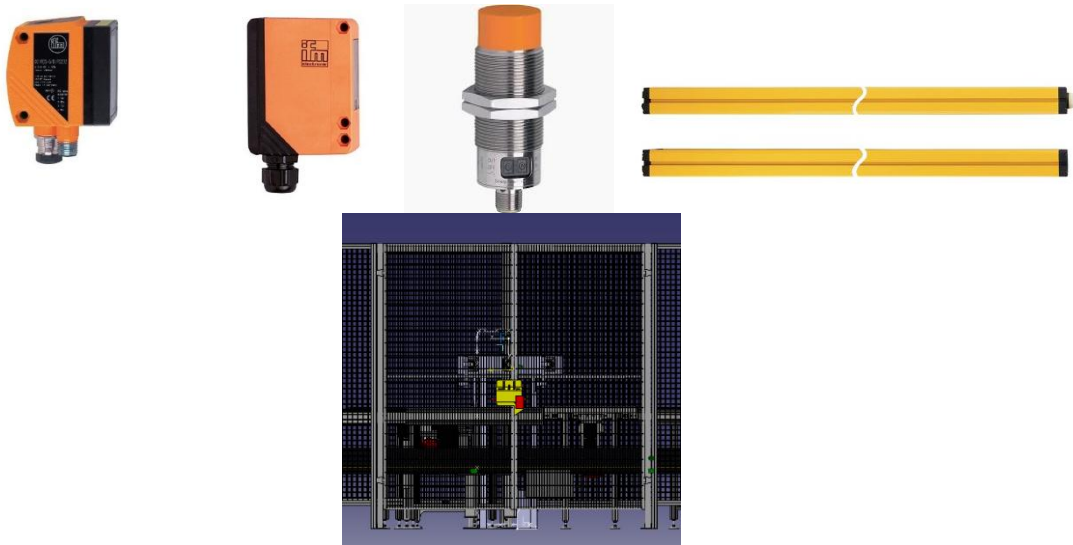


Fig.17 Examples of sensor components present in the first iteration of selecting the components necessary to perform the signal-based simulation



Fig.18 Sensor layout in the cell

The following types of sensors are arranged in the cell:

- 1. Photoelectric barrier sensors
- 2. Capacitive sensors
- 3. 1D/2D barcode reader
- 4. Light barrier sensor
- 5. Gate-opening sensor

**PROGRAMMING AND SIMULATING A FLEXIBLE PALLETIZING LINE USING THE PROCESS
SIMULATE ENVIRONMENT**

Tabel 4.3.3

Types of sensors

Sensor type	Function	Operating principle
Photoelectric sensors	Check for the presence of objects in the cell.	Consisting of a separate transmitter and receiver, they detect when an object interrupts the light beam. They offer a very long operating distance and provide excellent penetration through dust or dirt.
Capacitive sensors	Check for the presence of objects in transfer zones or waiting areas	They can detect the presence of any type of material (plastic, liquids, metals, wood) by detecting changes in the electrostatic field. They have adjustable sensitivity, making them ideal for checking the contents directly through the wall of a non-metallic box. Their purchase price is much lower than that of photoelectric sensors, though this comes at the cost of slower object detection time.
1D/2D Barcode Reader	Detects the barcode that identifies the type of box as it enters the sorting area and helps index it.	High-speed optical devices that scan standard or complex codes (QR, Data Matrix) directly from conveyor belts. They feature industrial communication interfaces and built-in lighting to ensure product traceability.
Light Curtain Sensor	Monitors areas where an operator might enter the working zone of the palletizing line	Also known as light curtains, these create a dense grid of light beams to protect operators. Interruption of this grid instantly stops hazardous machinery, with extremely short response times.
Gate-opening sensor	Detects whether the access gate is open	Stops the palletizing line when the gate opens, ensuring operator safety upon entering the work area

4. Conclusions

This paper lays the groundwork for signal-based simulation of a flexible palletizing line by creating a flowchart and presenting the layout of the work systems and sensors, which will help us define the signals and program the robotic application in Siemens Process Simulate.

5. Bibliography

9. Anania D. – Fabricatie asistata, note de curs UPB, 2024
10. Bucuresteanu A. – Actionarea Pneumatica a Robotilor Industriali, note de curs UPB, 2023
11. Bucuresteanu A. – Elemente si sisteme pneumatice pentru actionarea robotilor industriali, Editura Printech, ISBN 978-606-23-0081-4, Bucuresti 2013.
12. Cristoiu C. – Proiectare asistata de calculator 3 - NX CAD, note de curs si suport aplicatii, UPB, 2023
13. Cristoiu C. A., Nicolescu A. F. - NX CAD-Basic - Proiectare asistată de calculator, 96 pag., Ed. Politehnica Press, 2020, Cod CNC SIS 19, ISBN 978-606-515-914-3, CIP BNR 2020-13437
14. Nicolescu, A. – Implementarea Roboților Industriali în Sistemele de Producție, note de curs și metodologii de proiectare, UPB, 2024;

Offline programming and simulation of a robotic cell for loading and unloading tools from CNC-machine.

IAMANDEIU Andrei-Sebastian¹, IVAN Andrei Mario

¹Facultatea: IIR, Specializarea: Robotică, Anul de studii: II Master, e-mail:
iamandeuandreisebastian@gmail.com

Conducător științific: S.I. dr. ing. Andrei Mario IVAN

Summary: *This project presents the development of a digital twin for an industrial robotic cell used for automated tool handling in a CNC machining environment. The objective is to evaluate how simulation improves validation efficiency and reduces project turnaround time through offline testing and early detection of design issues before physical implementation.*

KEY WORDS: *Robotics, Digital Twin, Simulation*

1. Introduction

This work focuses on automating the loading and unloading of tools in a CNC machining environment through the development and validation of a digital twin of the robotic cell. The virtual model is used to optimize the workflow and improve the implementation process from the early design stages. Particular attention is given to offline simulation, virtual validation, identification of potential issues before on-site deployment, reduction of project turnaround time, and enhancement of system productivity, flexibility, and future scalability.

2. Current state

Several scientific publications describe robotic cells similar to the system developed in this project, in which an articulated robot arm mounted on an Automated Guided Vehicle (AGV) services Computer Numerical Control (CNC) machines by removing worn tools and installing replacements. This configuration combines the dexterity of the robot arm with the mobility of the AGV, allowing one system to serve multiple CNC machines. Studies such as Toward a Digital Twin of a Robot Work cell: Standards and Methods by Deogratias Kibira and related research from the National Institute of Standards and Technology demonstrate how digital twins can model robotic work cells that integrate robots, CNC machines, and mobile platforms. In this project, the digital twin is used to simulate AGV travel times, robot motions, tool exchange durations, and total cycle times, generating detailed data sheets for performance evaluation. This enables offline validation, collision detection, and early identification of design issues, reducing commissioning effort and project turnaround time while improving productivity and system flexibility.

3. Designing the Digital Twin

The design of the digital twin involves creating a virtual representation of the physical system and its operational logic within a simulation environment. Each component is modelled using geometric, kinematic, and dynamic data obtained from manufacturer datasheets, including dimensions, payload, speeds, and acceleration limits. The model reproduces the sequence of operations and enables the simulation of movement times, process durations, and complete cycle times. This virtual representation is used for offline validation, collision detection, and performance analysis, allowing potential issues to be identified and corrected before physical implementation.[1]

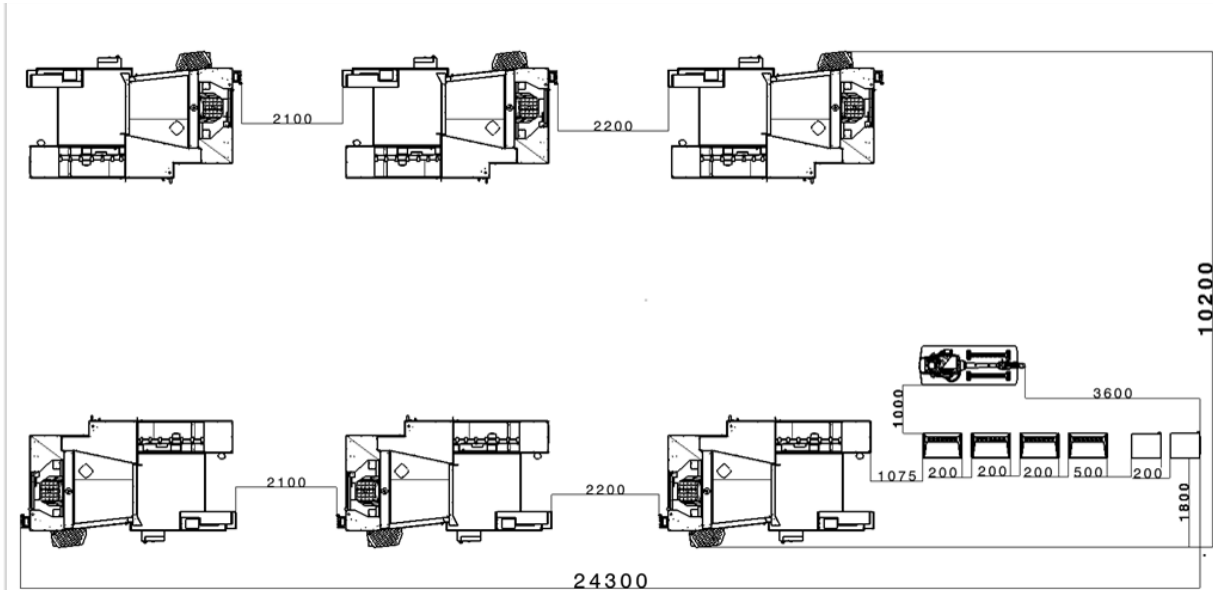


Fig. 1 Solution layout with relative distances between components

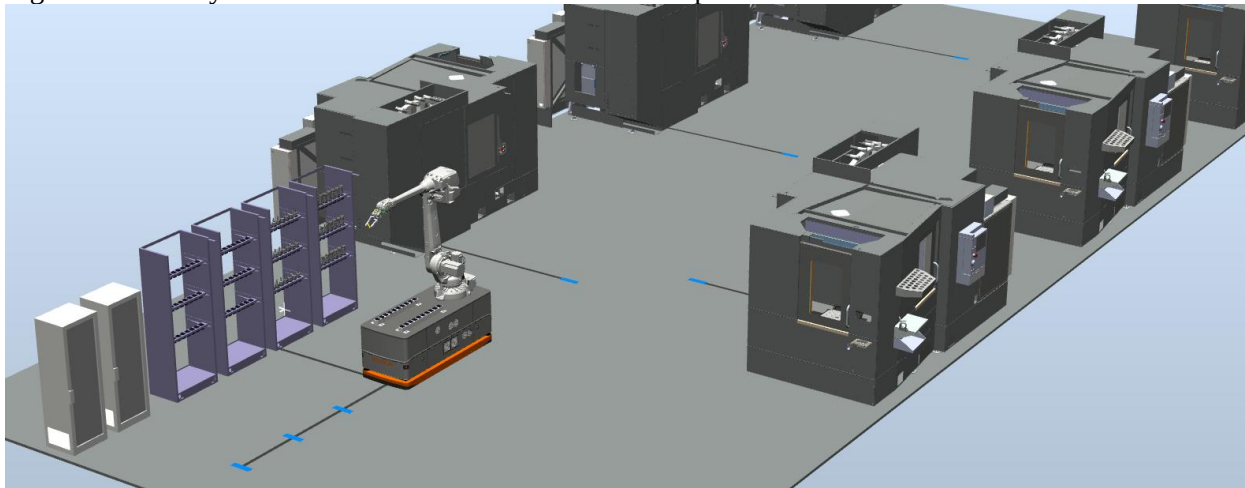


Fig. 2 Digital Twin

4. Defining Mechanism and project characteristics

Defining the mechanism and cell characteristics is a fundamental step in developing the digital twin, as it establishes the physical and functional parameters of the system. This stage involves collecting and organizing data from manufacturer datasheets and technical documentation, including dimensions, masses, payload capacities, working envelopes, joint limits, maximum speeds, and acceleration values. In addition, the spatial layout of the equipment, interaction points, and operational constraints are specified. These characteristics form the basis for accurately modeling the kinematics and behavior of the system and for obtaining realistic simulation results regarding motion, cycle times, and overall performance.[2]

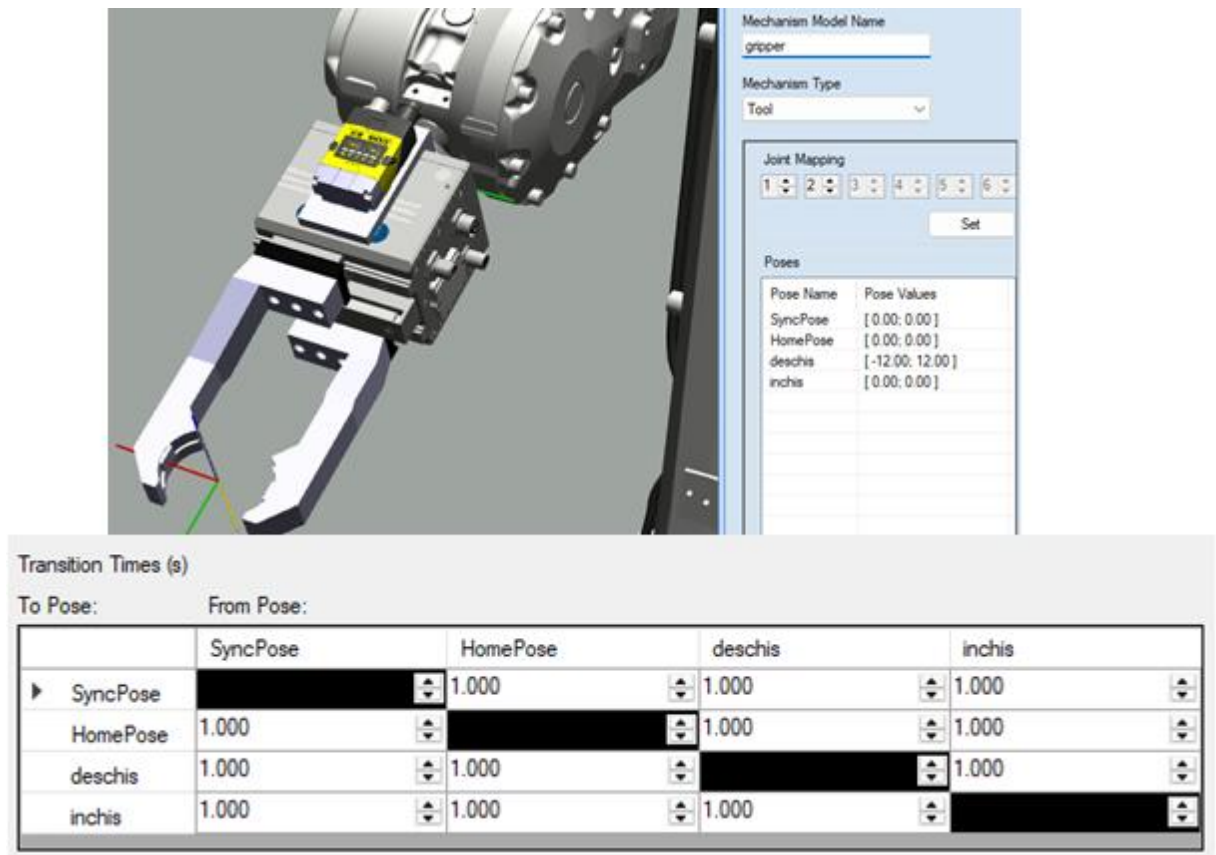


Fig. 3 Gripper Mechanism

The gripper mechanism is designed to securely grasp and release tools during the automated handling process. Its characteristics are defined based on the geometry, weight, and contact surfaces of the tools, as well as the required gripping force and positioning accuracy. Parameters such as jaw stroke, payload capacity, opening and closing times, and mounting dimensions are obtained from manufacturer datasheets and incorporated into the digital twin to ensure realistic simulation of tool pickup and placement operations.

The position of the gripper's center of gravity and its corresponding moments of inertia are important for ensuring robot stability and selecting an optimal robotic configuration during operation. These parameters define how the payload is distributed relative to the robot flange and directly influence joint loads, dynamic behavior, and motion accuracy. By incorporating these values into the digital twin, the simulation can identify stable and efficient robot configurations while reducing excessive loads and improving overall system performance.

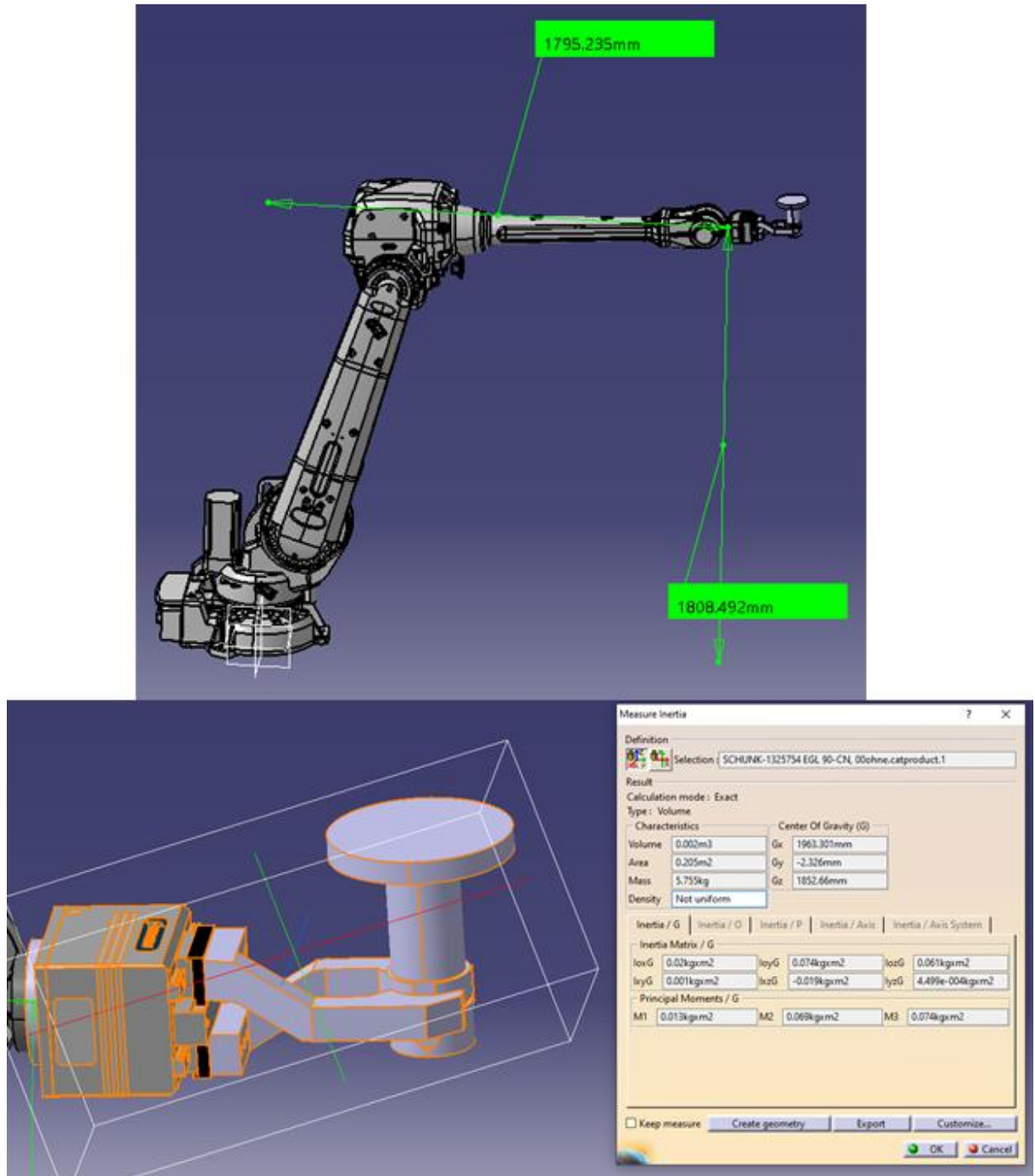


Fig. 4 Gripper center of gravity and resulting inertia moments

After this the rest of the equipment needs to be added and defined within the Digital Twin so the motions, working zones, configurations and clearances can be tested for potential issues.

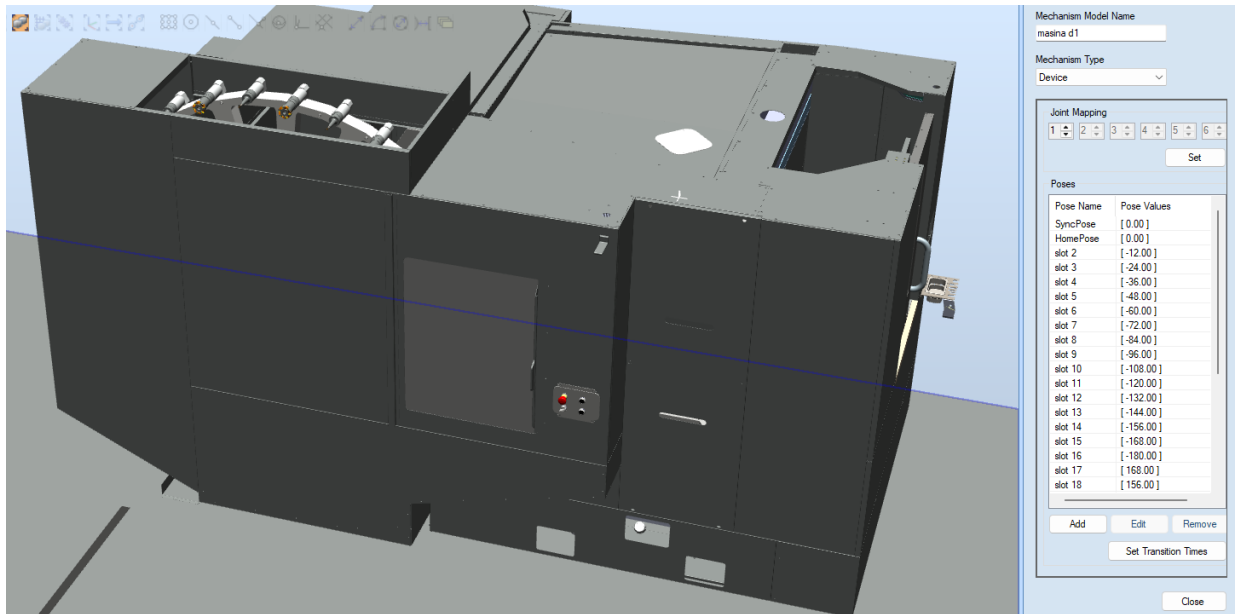


Fig. 5 CNC Machine and position definition for tools slots

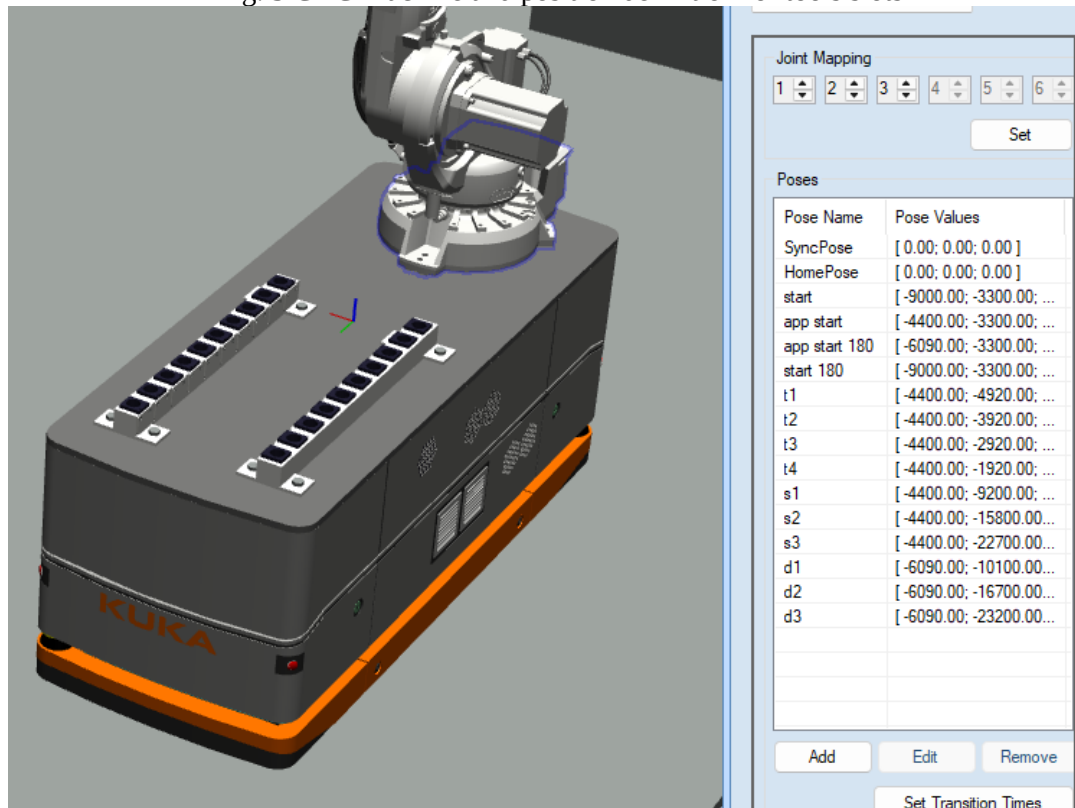


Fig. 6 AGV definition and position definition

5. Events and Signal mapping

Events and signal mapping define the communication between the system components and coordinate the sequence of operations. This step involves assigning digital inputs and outputs, sensors, and control signals to specific actions such as tool detection, gripper actuation, AGV positioning, and CNC machine status. By integrating these signals into the digital twin, the simulation accurately reproduces the control logic and interaction between components,

enabling reliable validation of the operational sequence and synchronization.[3]

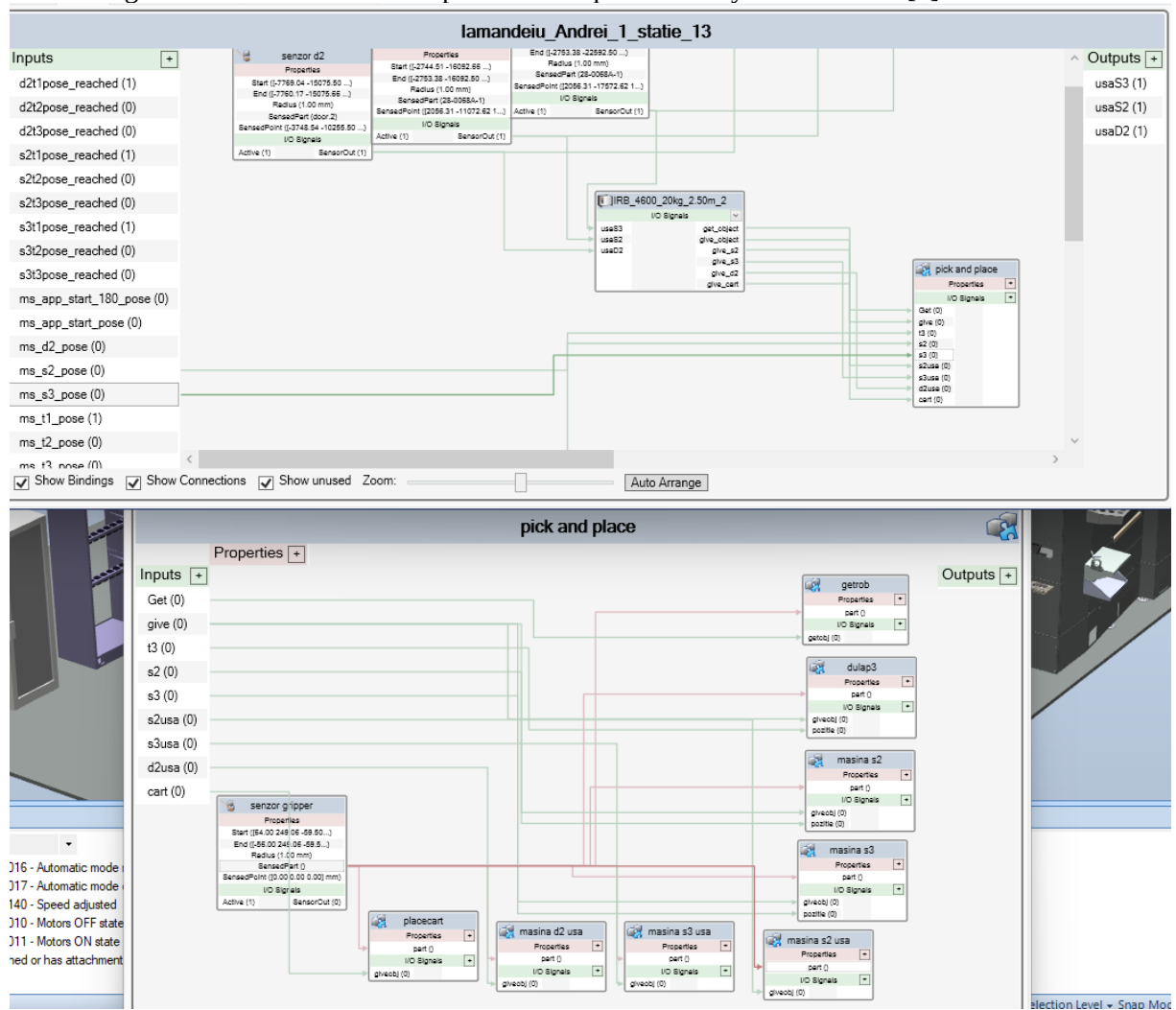


Fig. 7 Station logic scheme

Event Manager										
Events	Activation	Trigger Type	Trigger System	Trigger Name	Trigger Parameter	Action Type	Action System	Action Name	Action Parameter	Time (s)
Add...	On	I/O	IRB_4600_20k...	d21	1	Multiple			Multiple	
Delete	On	I/O	IRB_4600_20k...	d22	1	Multiple			Multiple	
Copy	On	I/O	IRB_4600_20k...	d23	1	Multiple			Multiple	
Refresh	On	I/O	IRB_4600_20k...	s21	1	Multiple			Multiple	
Export...	On	I/O	IRB_4600_20k...	s22	1	Multiple			Multiple	
Import...	On	I/O	IRB_4600_20k...	s23	1	Multiple			Multiple	
	On	I/O	IRB_4600_20k...	s31	1	Multiple			Multiple	
	On	I/O	IRB_4600_20k...	s32	1	Multiple			Multiple	
	On	I/O	IRB_4600_20k...	s33	1	Multiple			Multiple	
	On	I/O	IRB_4600_20k...	reset_sg	1	Multiple			Multiple	9:03:02 PM
	On	Connection			d23pose_reached	Change I/O	IRB_4600_20k...	d23pr		
	On	Connection			d22pose_reached	Change I/O	IRB_4600_20k...	d22pr		
	On	Connection			d21pose_reached	Change I/O	IRB_4600_20k...	d21pr		
	On	Connection			s21pose_reached	Change I/O	IRB_4600_20k...	s21pr		
	On	Connection			s22pose_reached	Change I/O	IRB_4600_20k...	s22pr		
	On	Connection			s23pose_reached	Change I/O	IRB_4600_20k...	s23pr		
	On	Connection			s31pose_reached	Change I/O	IRB_4600_20k...	s31pr		
	On	Connection			s32pose_reached	Change I/O	IRB_4600_20k...	s32pr		
	On	Connection			s33pose_reached	Change I/O	IRB_4600_20k...	s33pr		
	On	I/O	Station Signals	reset_cart_signal	1	Multiple			Multiple	
	On	I/O	IRB_4600_20k...	mv_start	1	Multiple			Multiple	
	On	I/O	IRB_4600_20k...	mv_app_start_180	1	Multiple			Multiple	
	On	I/O	IRB_4600_20k...	mv_app_start	1	Multiple			Multiple	
	On	I/O	IRB_4600_20k...	mv_d2	1	Multiple			Multiple	
	On	I/O	IRB_4600_20k...	mv_s2	1	Multiple			Multiple	
	On	I/O	IRB_4600_20k...	mv_s3	1	Multiple			Multiple	
	On	I/O	IRB_4600_20k...	mv_start_180	1	Multiple			Multiple	
	On	I/O	IRB_4600_20k...	mv_11	1	Multiple			Multiple	
	On	I/O	IRB_4600_20k...	mv_12	1	Multiple			Multiple	
	On	I/O	IRB_4600_20k...	mv_13	1	Multiple			Multiple	
	On	Connection			ms_app_start_180_pose	Change I/O	IRB_4600_20k...	mv_app_start_180_pr		
	On	Connection			ms_app_start_pose	Change I/O	IRB_4600_20k...	mv_app_start_pr		
	On	Connection			ms_d2_pose	Change I/O	IRB_4600_20k...	mv_d2_pr		
	On	Connection			ms_s2_pose	Change I/O	IRB_4600_20k...	mv_s2_pr		
	On	Connection			ms_s3_pose	Change I/O	IRB_4600_20k...	mv_s3_pr		
	On	Connection			ms_start_180_pose	Change I/O	IRB_4600_20k...	mv_start_180_pr		
	On	Connection			ms_start_pose	Change I/O	IRB_4600_20k...	mv_start_pr		
	On	Connection			ms_11_pose	Change I/O	IRB_4600_20k...	mv_11_pr		
	On	Connection			ms_12_pose	Change I/O	IRB_4600_20k...	mv_12_pr		
	On	Connection			ms_13_pose	Change I/O	IRB_4600_20k...	mv_13_pr		
	On	I/O	IRB_4600_20k...	close_grip	1	Multiple			Multiple	9:03:16 PM
	On	I/O	IRB_4600_20k...	open_grip	1	Multiple			Multiple	9:03:11 PM
	On	Connection			deschis	Change I/O	IRB_4600_20k...	open_grip_pr		9:03:16 PM
	On	Connection			inchis	Change I/O	IRB_4600_20k...	close_grip_pr		9:03:17 PM

Fig. 8 Event manager

6. Results analysis

Results analysis involves evaluating the simulation outputs to assess the general performance of the system and its ability to execute scheduled interventions, such as tool replacement operations, within the required time constraints. Key indicators including AGV travel times, robot motion durations, gripper actuation times, tool exchange durations, waiting times, and total cycle times are analyzed to determine whether the system can complete interventions without disrupting production. The results are used to identify bottlenecks, inefficient motions, and synchronization delays between the robot, mobile platform, and CNC machines. This analysis verifies that the system meets operational and productivity requirements while providing quantitative data to support optimization of trajectories, process sequences, and equipment layout before physical implementation.[4]

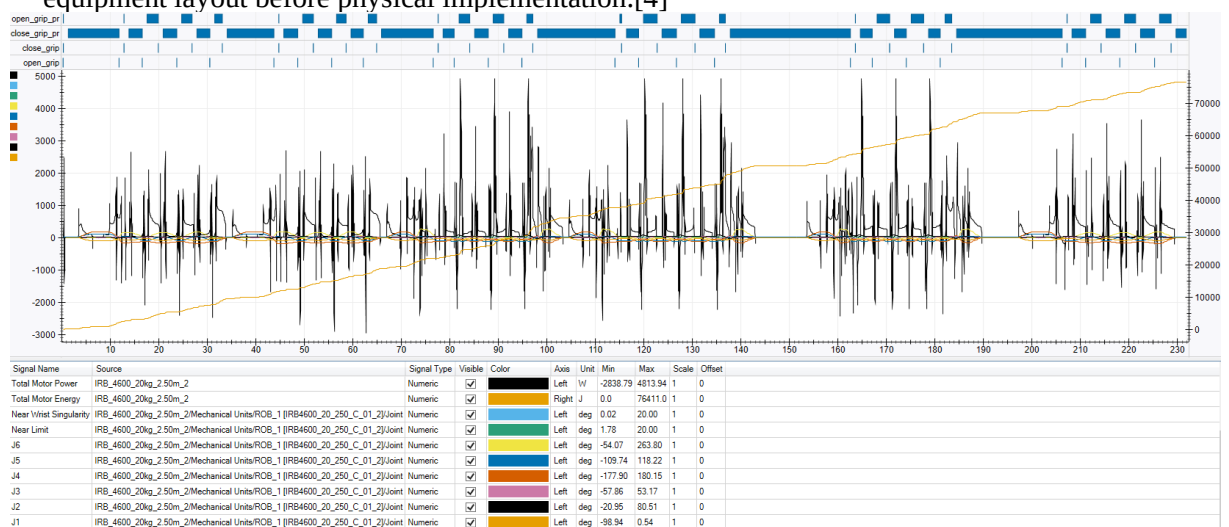


Fig. 9 Signal and robot parameters analysis with graphic legend

7. Conclusions

In conclusion, this project demonstrated the development and validation of a digital twin for an industrial robotic system designed to automate tool replacement in CNC machining environments. By integrating mechanical characteristics, control logic, and operational sequences into a virtual model, the system was able to simulate travel times, robot motions, and complete tool exchange cycles. The results showed that scheduled interventions can be performed within the required time constraints while maintaining efficient operation. In addition, the digital twin enabled early identification of design and synchronization issues, reducing commissioning effort and project turnaround time. Overall, the project confirms that digital twin technology is an effective approach for analyzing, validating, and optimizing robotic automation systems in modern manufacturing.

8. Bibliography

[1]. Nicolescu, A. – *Implementarea Robotilor Industriale in Sistemele de Productie, note de curs si metodologii de proiectare*, UPB, 2023

[2]. Nicolescu, A., Stanciu, M.D., Popescu D. – *Conceptia si Exploatarea Robotilor Industriale - Vol.1 Tendinte actuale in conceptia si exploatarea RI. Precizia de lucru si precizia volumetrica. Componente organologice specifice. Tehnici si metode de studiu al comportarii elastice si performantelor robotilor industriale*. 156 pag, Editura Printech 2004, Bucuresti, cod CNC SIS 54, ISBN 973-718-007-0, Cota BNR III 263217

[3]. Nicolescu, A. – *Implementarea Robotilor Industriale in Sistemele de Productie, note de curs si metodologii de proiectare*, UPB, 2023

[4]. Nicolescu A., Dobrescu T., Ivan M., Avram C., Brad S., Doroftei I., Grigorescu S. – *Roboti Industriale, Tehnologii si Sisteme de Productie Robotizate*, 190 pag., Ed Academiei Oamenilor de Stiinta din Romania, 2011, ISBN 978 – 606 – 8371 – 48 – 1, CIP BNR 2011-21962, Cota BNR III 307660

Robotic palletizing cell integrating articulated arm industrial robots and pallet transport systems using autonomous guided vehicles.

MOTONEA Andrei-Daniel, DOBRESU Tiberiu Gabriel

Facultatea: IIR, Specializarea: Robotică, Anul de studii: II Master, e-mail: motoneaandrei@gmail.com

Conducător științific: Prof. Dr. Ing. DOBRESU Tiberiu Gabriel

***Summary:** The purpose of this dissertation project was the offline programming and simulation of a robotic palletizing cell integrating ABB industrial robots, conveyors, sensors, and AGV systems for automated pick-and-place operations. The methods used included 3D CAD modeling, creation of a virtual prototype, offline simulation in Process Simulate, sensor mapping, and signal-based programming. The application integrated ABB IRB 660 robots, conveyors, barcode scanners, and AGV transport systems. The obtained results confirmed the successful simulation of box sorting, palletizing, barcode scanning, AGV transport, and storage processes. The simulations validated the synchronization between all subsystems and ensured correct robot trajectories and process flow. The conclusions showed that the proposed robotic cell operated efficiently and reliably in the virtual environment, without major collisions or functional errors.*

1. Introduction

This paper aims to program and offline simulate a robotic palletizing cell that integrates ABB industrial robots, conveyor systems, sensors, and AGV-type autonomous vehicles for the automation of palletizing operations, specifically pick-and-place operations. The virtual model created allows the analysis and optimization of the technological flow from the design stage, as well as the verification of the operation of the entire application before physical implementation. The paper focuses on the virtual validation of the process, the coordination of subsystems based on signals and sensors, the identification of possible collisions or functional errors, the reduction of implementation time, and the increase of the efficiency, flexibility, and reliability of the automated system.

2. Current state

In the specialized literature, numerous robotic applications similar to the system developed in this project are presented, based on the use of industrial robots for palletizing, handling, and automated product transport processes. These applications frequently integrate articulated-arm industrial robots, conveyor systems, sensors, and AGV autonomous vehicles to automate logistics flows and increase the efficiency of industrial processes. Modern studies and offline simulation platforms, such as Process Simulate, allow the creation of virtual prototypes of robotic cells and the validation of their operation before physical implementation. In this project, the virtual prototype is used to simulate robot movements, synchronize conveyors and AGVs, validate signals from sensors, as well as analyze cycle times and the technological flow. Offline simulation allows the identification of possible collisions or functional errors from the design stage, reducing implementation time and optimizing the productivity, flexibility, and reliability of the entire automated system.

3. Modeling and Simulation of the Robotic Application

The robotic application developed within the project represents an automatic palletizing cell intended for handling and transporting boxes within an industrial logistics flow. The system consists of ABB IRB 660 industrial robots, conveyor systems for transporting boxes, AGV-type autonomous vehicles for handling pallets, sensors for detecting and identifying products, a pneumatic end effector intended for pick-and-place operations, as well as auxiliary equipment such as the electrical panel, compressor, and safety systems.

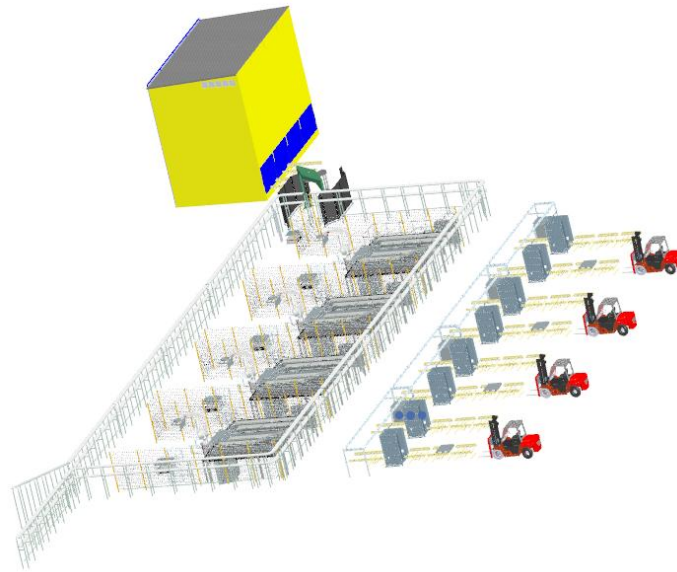
The technological flow begins with the transport of boxes on the feeding conveyors, where they are detected and directed using the sensors integrated into the application. The boxes are picked up by the industrial robot and palletized according to the established configuration. After the stack is formed, the pallets are automatically transported by the AGVs to the wrapping and label scanning area, followed by their storage. The entire process is coordinated through signals and sensors that ensure the

synchronization of the components and the correct operation of the automated system.

The following figure presents the overall view of the robotic cell, highlighting the positioning of the main components and the path of the technological flow.

The robot application is 660 250/3.15 articulated-arm with 4 degrees specifically high-speed handling. The robot has a payload of 250 maximum reach providing high repeatability of ± 0.05 mm.

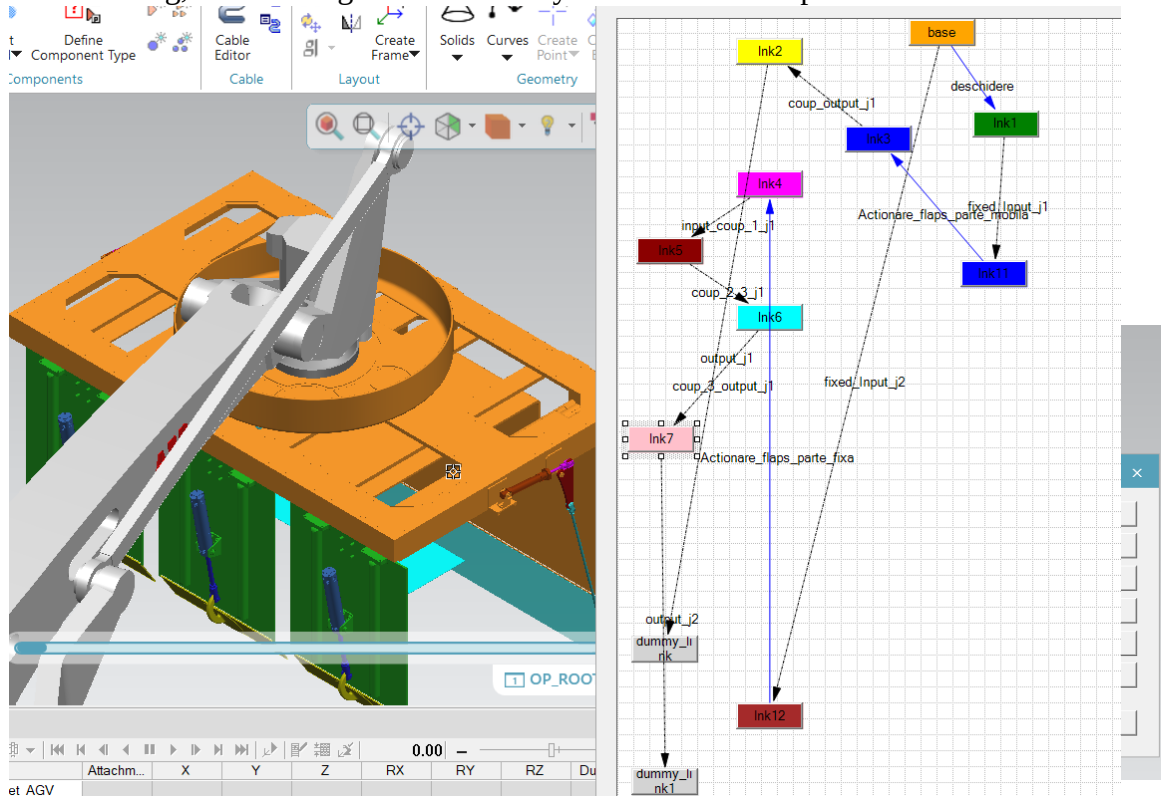
The end designed for application is a system intended for the simultaneous handling of boxes. It consists of a main structure, side pneumatic pistons for securing the products, and vertical mechanisms for lifting and releasing the boxes during the palletizing process. The end effector configuration allows safe and fast product handling, contributing to the efficiency of the automated process.



Img.1. Isometric view of the industrial robotic cell.

used in the the ABB IRB model, an industrial robot of freedom, designed for palletizing and applications. maximum kg and a of 3150 mm, precision and a approximately

effector the pneumatic



Img. 2. End Effector Kinematics and Positions.

4. Offline programming and Signal-Based Simulation

The operation of the robotic application is carried out through a programming system based on events and signals, used to synchronize all components within the cell. The industrial robots, conveyors, AGV system, and sensors communicate continuously through digital signals, so that each operation is executed in the correct order.

Within the simulation performed in Process Simulate, the signals were mapped in order to reproduce the real behavior of the industrial application. Each sensor and each actuation mechanism were assigned input and output signals, used to control the work sequences and to verify the status of the equipment.

The photoelectric sensors detect the presence of boxes on the conveyors and transmit signals to the control system. Based on these signals, the pneumatic mechanisms for pushing and positioning the boxes are activated. The industrial robot performs the pick-and-place operations only after confirming that the product is in the correct position.

The application also uses signals to coordinate the AGV system and to validate the palletizing process. After the stack of boxes is formed, the AGV receives the transport signal and moves the pallet to the scanning and storage area.

The integration of signal-based logic in the virtual environment allows the synchronization between equipment to be verified and possible errors to be identified before the physical implementation of the application. Thus, the simulation contributes to reducing commissioning time and increasing the safety and reliability of the robotic system.

	Name	Value	Access	Type	Originator	Destination	Initial Value *	Auto Reset	Signal List	Comment
1	Box_Present	False	Input	Bool	Sensor_Box	SPS	False	False	<None>	Detectie cutie pe conveyor
2	Conv1_Start	False	Output	Bool	SPS	Conv_1	False	False	<None>	Pornire conveyor 1
3	Conv1_Stop	False	Output	Bool	SPS	Conv_1	False	False	<None>	Opre conveyor 1
4	Pusher_Extend	False	Output	Bool	SPS	Pusher	False	False	<None>	Extindere pusher
5	Pusher_Retract	False	Output	Bool	SPS	Pusher	False	False	<None>	Retragere pusher
6	Robot_Ready	False	Input	Bool	ABB IRB 660_250_...	SPS	False	False	<None>	Robot in pozitie initiala
7	Robot_Busy	False	Input	Bool	ABB IRB 660_250_...	SPS	False	False	<None>	Robot executa ciclu
8	Gripper_Close	False	Output	Bool	SPS	Gripper	False	False	<None>	Inchidere efector
9	Gripper_Open	False	Output	Bool	SPS	Gripper	False	False	<None>	Deschidere efector
10	Pallet_Present	False	Input	Bool	Sensor_Pallet	SPS	False	False	<None>	Detectie palet
11	AGV_Available	False	Input	Bool	AGV_1	SPS	False	False	<None>	AGV disponibil
12	AGV_Arrived	False	Input	Bool	AGV_1	SPS	False	False	<None>	AGV in pozitie la preluare
13	AGV_Load	False	Output	Bool	SPS	AGV_1	False	False	<None>	Comanda incarcare palet
14	AGV_Unload	False	Output	Bool	SPS	AGV_1	False	False	<None>	Comanda descarcare palet
15	Stack_Complete	False	Output	Bool	SPS	AGV_1	False	False	<None>	Stiva completa
16	Scanner_OK	False	Input	Bool	Scanner	SPS	False	False	<None>	Scanare cod OK
17	System_Reset	False	Output	Bool	SPS	SPS	False	False	<None>	Reset sistem
18	Emergency_Stop	False	Input	Bool	E_Stop	SPS	False	False	<None>	Buton de urgenta
19	Cycle_Start	False	Input	Bool	SPS	SPS	False	False	<None>	Start ciclu automat
20	Cycle_End	False	Output	Bool	SPS	SPS	False	False	<None>	Final ciclu automat

Img. 3. Signals

5. Results and analysis

The analysis of the simulation results aimed to verify the correct operation and synchronization of the components within the robotic application. In the Process Simulate environment, the movements of the industrial robot, the operation of the pneumatic end effector, the movement of the AGV, and the signals generated by the sensors were analyzed.

The obtained results showed that the application operates properly, without collisions or synchronization errors between the equipment. The robot correctly performs the pick-and-place operations, while the sensors detect and direct the boxes according to the programmed logic.

The simulation allowed the identification of possible waiting times and the optimization of the work sequences before the physical implementation of the application. Thus, the process can be validated in a virtual environment, reducing commissioning time and the risk of errors during operation.

6. Conclusions

Within this paper, the modeling and simulation of a robotic palletizing application was carried out

using the Process Simulate environment. By integrating industrial robots, conveyor systems, sensors, and AGV vehicles into a virtual prototype, it was possible to validate the operation of the entire process before physical implementation.

The simulation allowed the verification of the signal-based operating logic and the correct synchronization of the equipment within the robotic cell. The obtained results demonstrated that the application can perform pick-and-place, palletizing, and transport operations efficiently and without functional conflicts.

The use of the virtual prototype also contributed to identifying and correcting possible synchronization or process organization issues, reducing the time required for commissioning and the risk of errors during real operation. In conclusion, offline simulation and the use of the digital twin concept represent effective solutions for the analysis, validation, and optimization of modern robotic applications.

7. Bibliografy

- [1] ABB – ABB Robotics – *Product Manual IRB 660 Industrial Robot*, ABB Documentation.
- [2] *Process Simulate – Siemens Tecnomatix Process Simulate Documentation and User Guide*, Siemens.
- [3] Siemens – *Tecnomatix Process Simulate – Digital Manufacturing and Simulation Solutions*, Siemens Documentation.
- [4] Nicolescu, A. – *Implementarea Robotilor Industriale in Sistemele de Productie*, note de curs si metodologii de proiectare, UPB, 2023
- [5] Nicolescu A., Dobrescu T., Ivan M., Avram C., Brad S., Doroftei I., Grigorescu S. – *Roboti Industriale, Tehnologii si Sisteme de Productie Robotizate*, 190 pag., Ed Academiei Oamenilor de Stiinta din Romania, 2011, ISBN 978 – 606 – 8371 – 48 – 1, CIP BNR 2011-21962, Cota BNR III 307660

Off-line programming and simulation of a robotic cell for machining petroleum pumps for the oil industry

MUREȘAN Ștefan-Claudiu¹, NICOLESCU Florin-Adrian

¹Facultatea: IIR, Specializarea: Robotică, Anul de studii: I Master, e-mail: stefanclaudiu2812@gmail.com

Conducător științific: Prof. dr. ing. **Florin-Adrian Nicolescu**

SUMMARY: The project focuses on developing the digital twin of a industrial robotics cell for machining petroleum pumps for the oil industry, focusing on the impact of digital twin simulation on the project turn around and validation.

KEY WORDS: Simulation, Programming, Robotics, Digital Twin

1. Introduction

The project "Off-line programming and simulation of a robotic cell for machining petroleum pumps for the oil industry" delves into the critical machining process of orthogonal turning, which is fundamental for producing precise and durable pump components used in the oil industry. By focusing on understanding and optimizing this process, the project aims to develop the workflow and the development of the automation from the early stages of implementation, focusing on off-line optimizations, possible problems that could appear in the on-site installation stage, system productivity increase and flexibility enhancement for further upgrades.

2. Current state

In recent years, significant technological advancements have been made in the field of software solutions for simulating different industrial manufacturing processes. With the fast growth of Industry 4.0 and the necessity of pre-installation validation of complex automation solutions more and more companies decided to develop and implement the digital twin segment in their factory acceptance test phase. To obtain a digital twin as close as possible to what the final solution will be the CAD databases from the original equipment manufacturers are essential to be available, accompanied by all the datasheets and product manuals from where essential parameters and specifications are extracted to be utilized. Compared with the last two decades, the project restrictions and demands has been in a constant increase, which led to a necessity of tool that permitted the end user to visualize the process and results before the actual implementation of it, reducing the costs and the duration of project in terms of troubleshooting and improvements.

3. Solution design and validation

In order to develop a solution it is necessary to obtain a significant quantity of demands from the customer and information for all the equipments that will have to be integrated in the application, such as cycle times, productivity demands, time between service interventions, physical performances and even more. After all the information needed has been obtained, the layout design stage can begin, where the application first starts to obtain a more clear overall image, with all the components placed relative with each other, the working envelopes are displayed in order to assure proper reachability of different equipments, the self-build equipments are tested and validated etc [1].

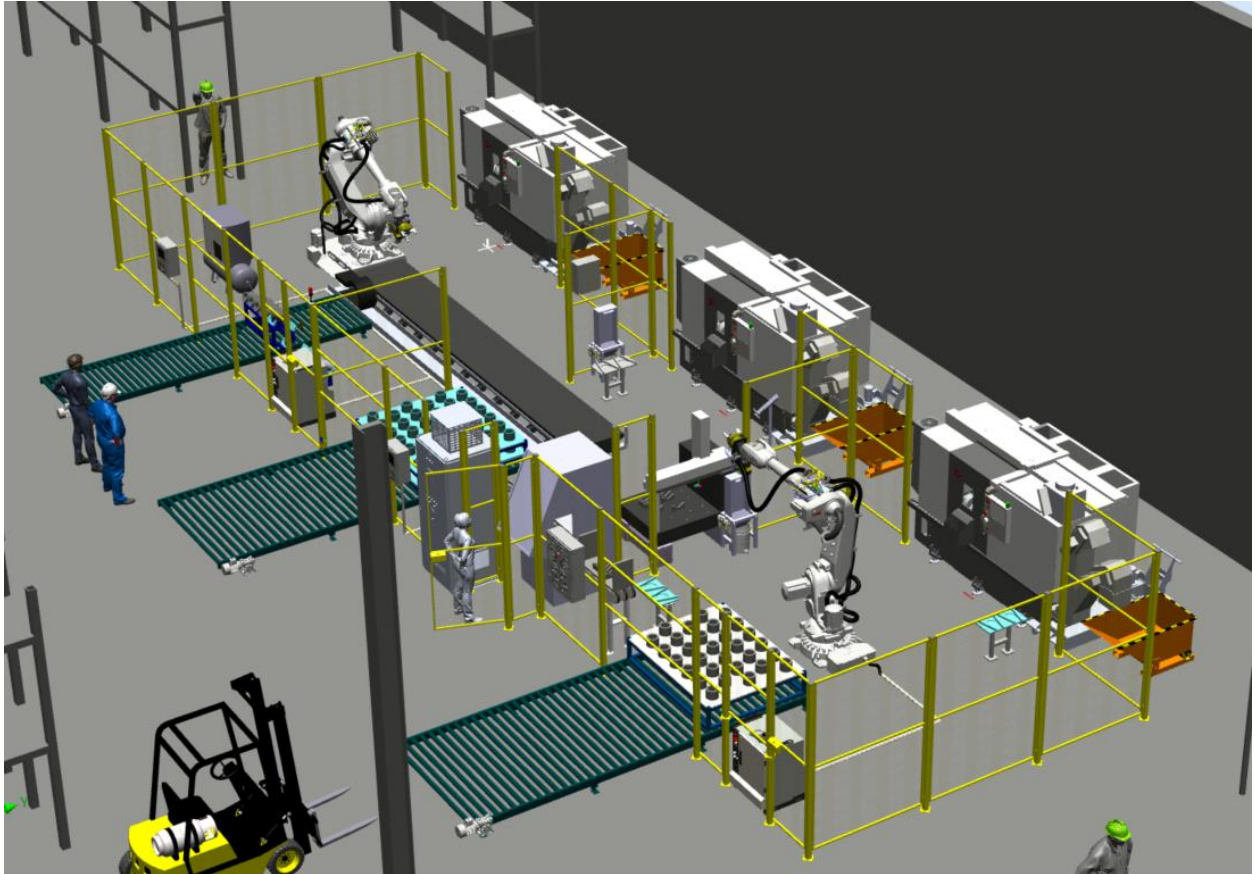


Fig. 1 Digital twin of solution

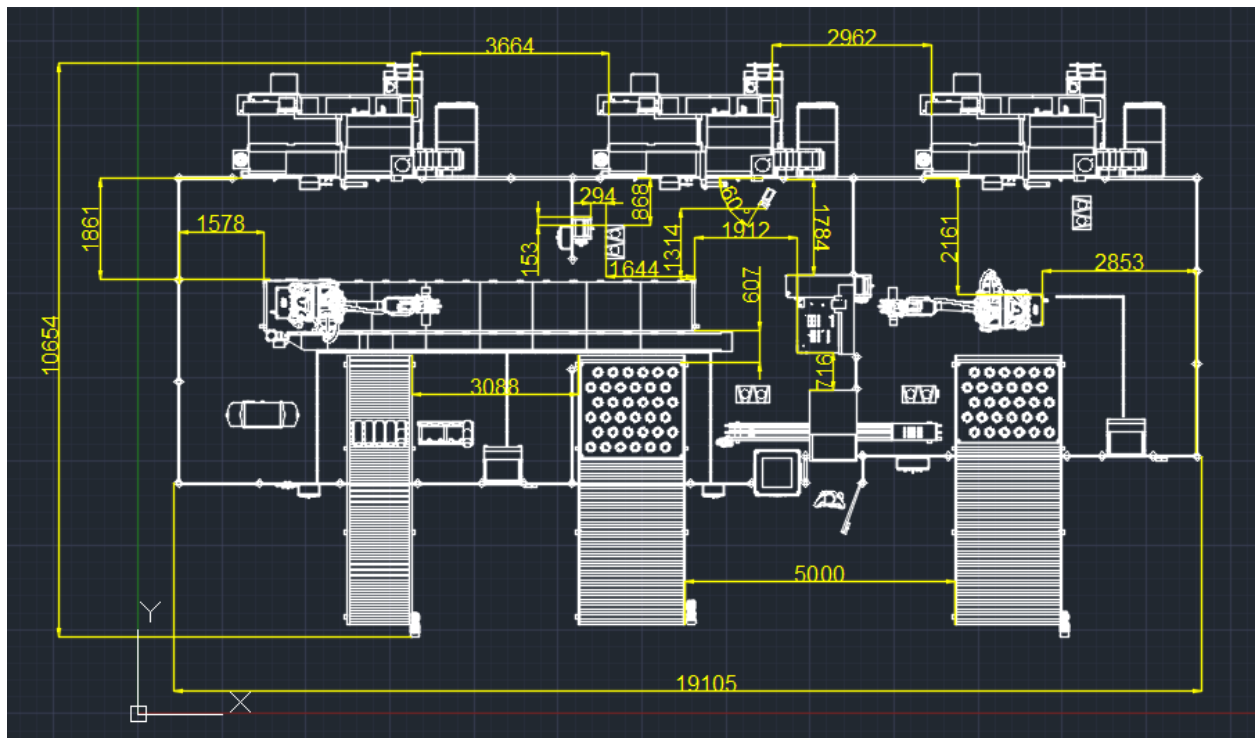


Fig. 2 Solution layout with relative position of components

4. Mechanisms defining and real-world characteristics insertion

After the layout development stage has been concluded, in order to obtain a behavior as close as possible to reality it is necessary to define all the mechanisms integrate in the solution, with their

specific specifications, such as range of motion, working parameters such as speed, acceleration and deceleration times, kinematic joints etc. As an example, for the dual gripper attached to the robotic arm there is necessary to define the mechanical joints of each finger individually, the working range of them, the closing and opening times and the specific positions for all the parts that have to be handled.

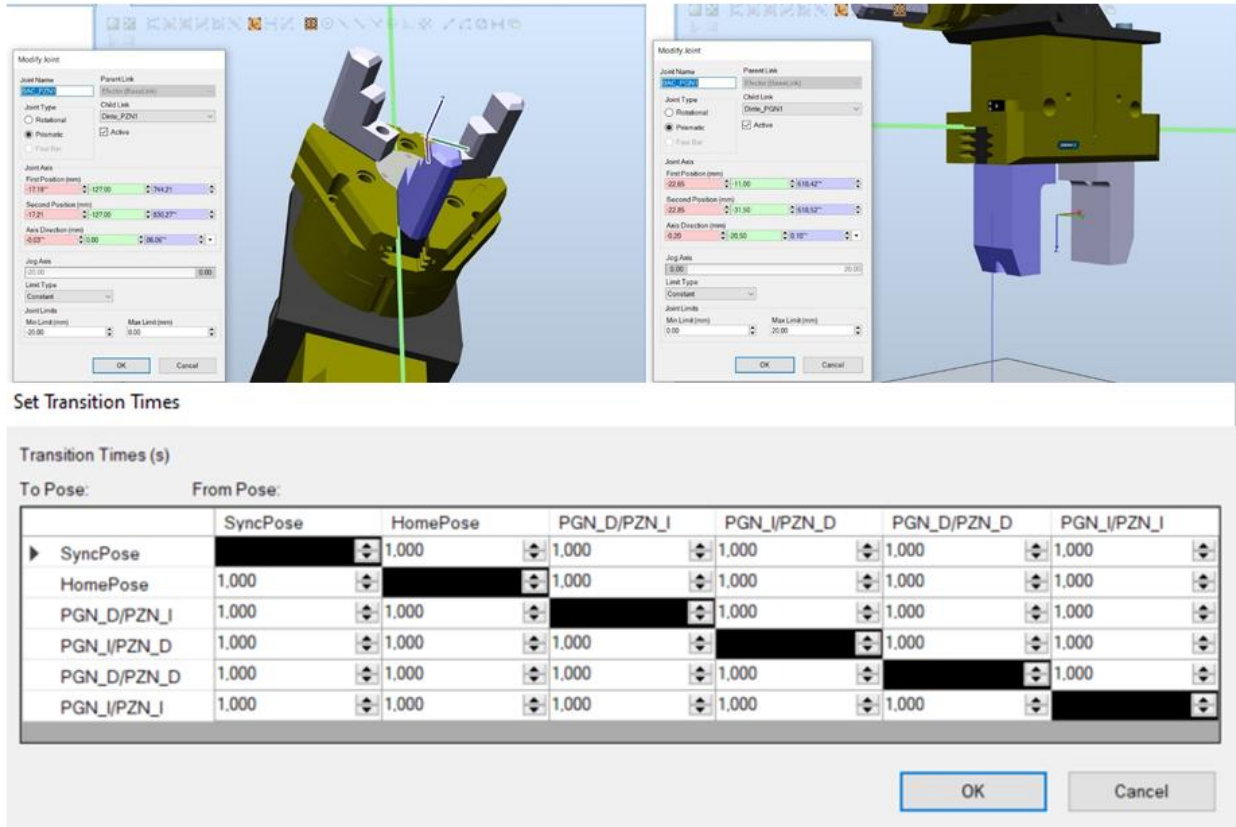


Fig. 3 Gripper defining

Doing so we can validate the reachability and the accesibility of the robotic arm along side all the crucial positions that have to be reached and also assuring the avoidance of any possible collisions with different obstacles. Also, applying the correct materials and densities of components we can check the fitting of the end of arm tool in the manufacturer specification regarding the payload and its impact in the cycle times [2].

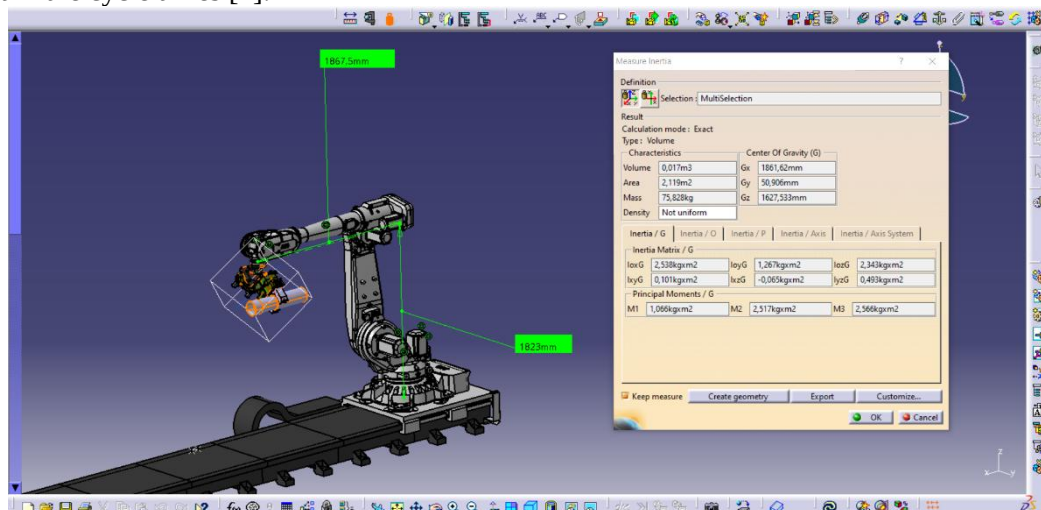


Fig. 4 Gripper center of gravity and inertia moments

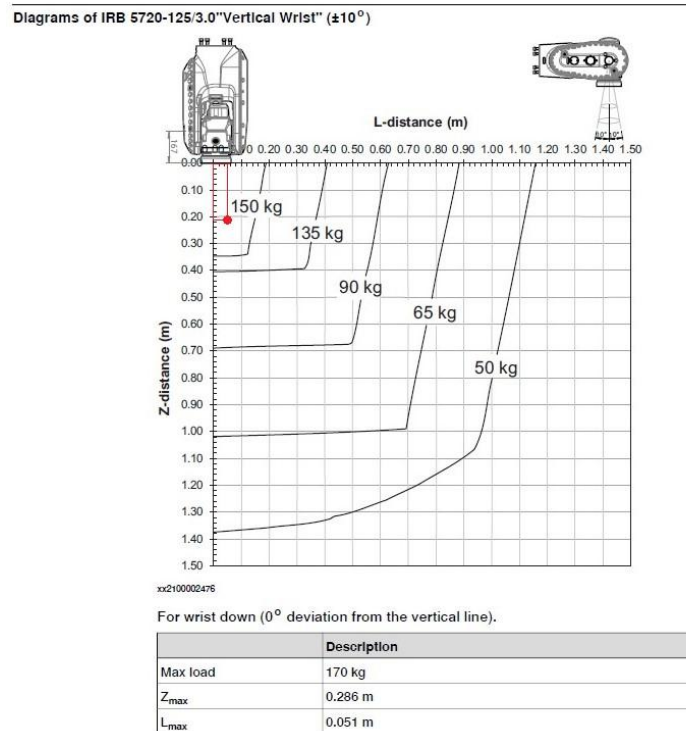


Fig. 5 Gripper validation in terms of payload restrictions

Going further, the complementary equipments must also be physically defined to test the robot motion routines, accessibility inside different working zones and establishing the proper configurations in these circumstances.

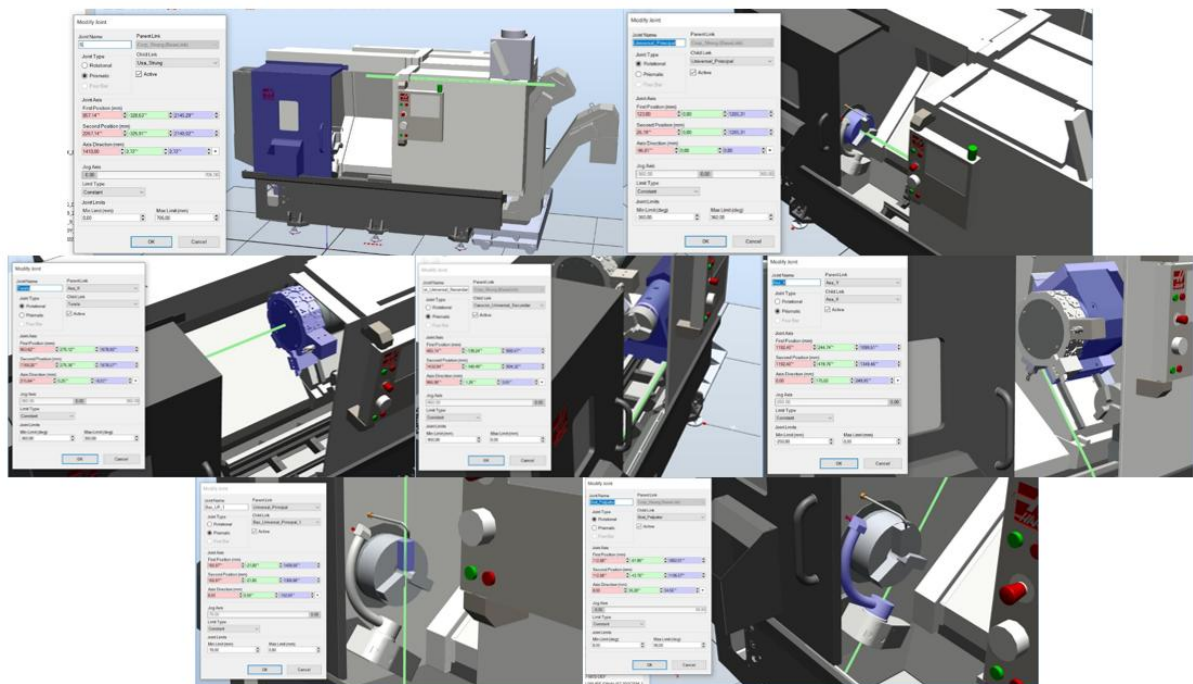


Fig. 6 CNC Lathe definition and characteristics declaration

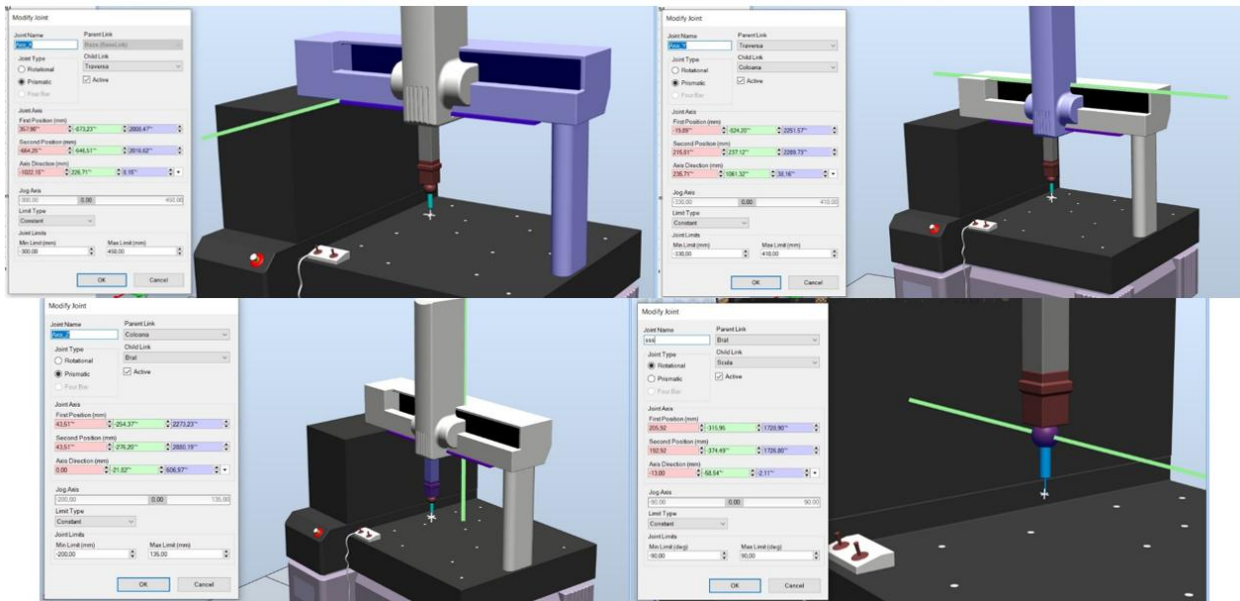


Fig. 7 CMM Machine definition and characteristics declaration

5. Environment specific components definition

Considering the impact of specific components in the definition of motion sequences and space restrictions, such as the position of safety fences and the physical characteristics of the robot dressing. For example, using dedicated simulation software the validation of dressing is possible, elements such as mounting points, cable length and diameter, stiffness of cable can be determined easily by using pre-defined dedicated options. As an example, the stiffness of dressing can be validated by introducing Young's modulus as a material feature. Doing so helps to determine the most suitable robot tool path to avoid unwanted cable twists around the robot arm of possible cable snags in different obstacles along the path [3].

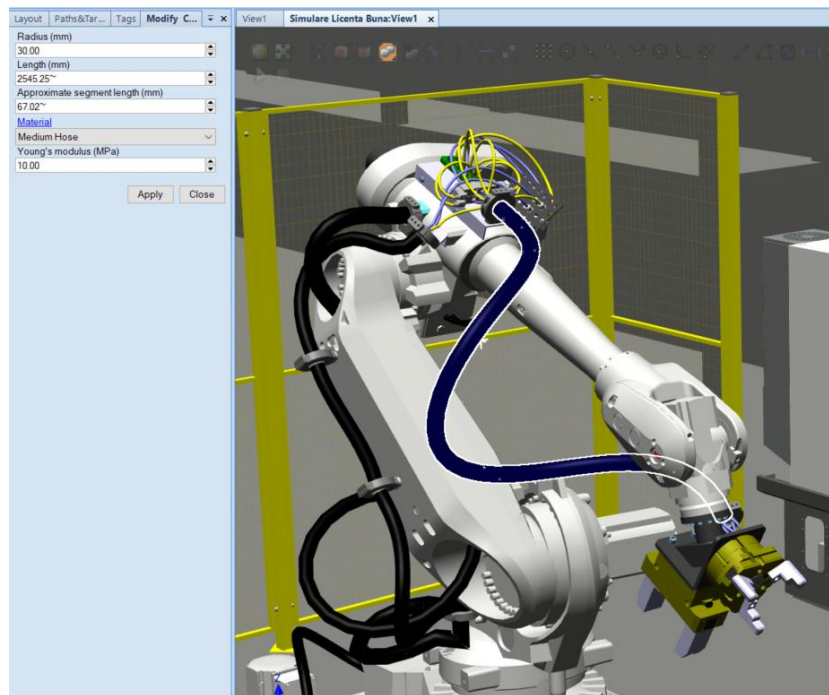


Fig. 8 Cable definition and physical characteristics

6. Simulation events definition and signals mapping

The core of process simulation tools is the variety of events accessible by the developer which permits the integration of real-world behaviors. Alongside this, the usage of different signals and their mapping in dedicated ports of each equipments offers the capability of events synchronization by the cause-effect principle.

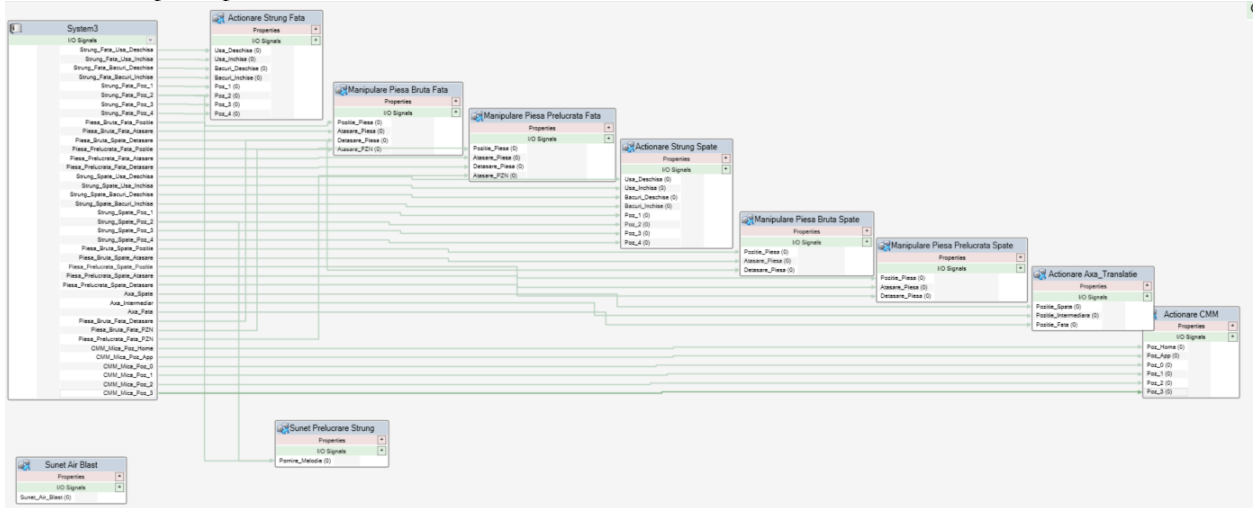


Fig. 9 Station logic scheme

7. Results analysis and comparison

The main goal of robotic process simulation is to be capable of obtaining a set of results such as cycle times, robot arm trajectories and signals diagram. The aim is to be capable to predict the behavior of the system before its physical installation, to optimize the cycle time in order to obtain a as good as possible productivity and to decrease the power consumption on a longer period of time, all of which automatically lead to a more positive return of investment. Statistics such as total power and energy consumption, travel on each individual axis, axis temperature variations, speed variation during runtime, maximum linear acceleration can be analyzed and extracted in order to be imported in software solution for OEE (Overall Equipment Efficiency) and PLM (Product Lifetime Management) which can create a better overall perspective on system's general performances and schedule interventions for preventive maintenance [4].

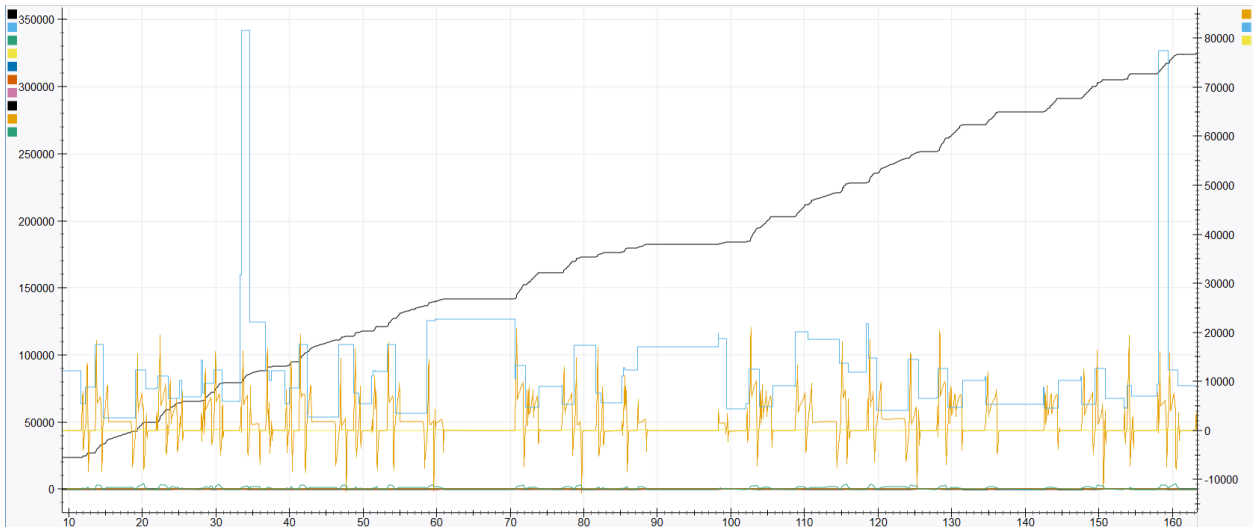


Fig. 10 Signal and robot parameters analysis

Signal Name	Source	Signal Type	Visible	Color	Axis	Unit	Min	Max	Scale	Offset
Near Limit	System3/Mechanical Units/ROB_1 [IRB5720_125_300_04]/Joint	Numeric	☒		Left	deg	3.25	20.00	1	0
Total Motor Energy	System3	Numeric	☒		Left	J	0.0	341790.4	1	0
Total Motor Power	System3	Numeric	☒		Right	W	-12029.72	20445.76	1	0
J1	System3/Mechanical Units/ROB_1 [IRB5720_125_300_04]/Joint	Numeric	☒		Left	deg	-83.33	119.16	1	0
J2	System3/Mechanical Units/ROB_1 [IRB5720_125_300_04]/Joint	Numeric	☒		Left	deg	-55.90	81.75	1	0
J3	System3/Mechanical Units/ROB_1 [IRB5720_125_300_04]/Joint	Numeric	☒		Left	deg	-71.37	56.13	1	0
J4	System3/Mechanical Units/ROB_1 [IRB5720_125_300_04]/Joint	Numeric	☒		Left	deg	-39.70	163.72	1	0
J5	System3/Mechanical Units/ROB_1 [IRB5720_125_300_04]/Joint	Numeric	☒		Left	deg	-43.72	107.73	1	0
J6	System3/Mechanical Units/ROB_1 [IRB5720_125_300_04]/Joint	Numeric	☒		Left	deg	-169.10	104.45	1	0
Near Wrist Singularity	System3/Mechanical Units/ROB_1 [IRB5720_125_300_04]/Joint	Numeric	☒		Left	deg	0.11	20.00	1	0
Maximum Linear Acceleration In World	System3/Mechanical Units/ROB_1 [IRB5720_125_300_04]/TCP	Numeric	☒		Right	mm/s ²	2545.17	98991.58	1	0
Speed In Current Wobj	System3/Mechanical Units/ROB_1 [IRB5720_125_300_04]/TCP	Numeric	☒		Left	mm/s	0.00	4000.82	1	0
Orientation Speed In Current Wobj	System3/Mechanical Units/ROB_1 [IRB5720_125_300_04]/TCP	Numeric	☒		Right	deg/s	0.00	242.08	1	0

Fig. 11 Analysis graphic legend

Time	Total Motor Energy	Total Motor Power	J1	J2	J3	J4	J5	J6	Near Wrist Singularity	Maximum Linear Acceleration In World	Speed In Current Wobj	Orientation Speed In Current Wobj
0.000			0.000	-33.218	56.128	0.000	52.090	0.000	13.870	20.000	6782.712	
0.004	0.000	355.656	0.000	-33.198	56.055	0.000	52.354	0.000	13.943	20.000	30319.624	9.251
0.048	45.262	1885.925	0.000	-33.119	55.764	0.000	53.410	0.000	14.233	20.000	98991.577	161.176
0.072	146.438	4215.674	0.000	-32.961	55.183	0.000	55.521	0.000	14.815	20.000	98991.577	329.409
0.096	240.742	3929.311	0.000	-32.763	54.453	0.000	58.171	0.000	15.545	20.000	98991.577	434.123
0.120	296.723	2332.537	0.000	-32.564	53.720	0.000	60.932	0.000	16.278	20.000	98991.577	473.550
0.144	341.635	1871.332	0.000	-32.365	52.987	0.000	63.493	0.000	17.011	20.000	98991.577	504.953
0.168	385.835	1841.672	0.000	-32.165	52.254	0.000	66.154	0.000	17.744	20.000	98991.577	535.422
0.192	430.117	1845.077	0.000	-31.966	51.521	0.000	68.816	0.000	18.477	20.000	98991.577	565.937
0.216	474.469	1848.025	0.000	-31.767	50.788	0.000	71.477	0.000	19.210	20.000	98991.577	596.618
0.240	518.902	1851.371	-0.001	-31.568	50.055	0.000	74.138	-0.001	19.943	20.000	98991.577	626.343
0.264	563.384	1853.407	-0.001	-31.369	49.322	0.000	76.799	-0.001	20.000	20.000	98991.577	654.797
0.288	607.937	1856.393	-0.001	-31.169	48.589	0.000	79.460	-0.001	20.000	20.000	98991.577	683.939
0.312	652.562	1859.359	-0.001	-30.970	47.856	0.000	82.121	-0.001	20.000	20.000	98991.577	712.757
0.336	697.265	1862.638	-0.001	-30.771	47.123	0.000	84.782	-0.001	20.000	20.000	98991.577	741.228
0.360	742.033	1865.316	-0.001	-30.572	46.390	0.000	87.443	-0.001	20.000	20.000	98991.577	769.332
0.384	786.864	1867.968	-0.001	-30.373	45.657	0.000	90.105	-0.001	20.000	20.000	98991.577	797.038
0.408	831.787	1870.937	-0.001	-30.174	44.924	0.000	92.768	-0.001	20.000	20.000	98991.577	824.352
0.432	876.696	1873.998	-0.001	-29.974	44.191	-0.001	95.427	-0.001	20.000	20.000	98991.577	851.234
0.456	921.579	1876.132	-0.012	-29.758	43.440	0.015	98.092	-0.028	20.000	20.000	98991.577	874.305
0.480	966.773	891.416	-0.057	-29.490	42.687	0.090	100.541	-0.159	20.000	20.000	98991.577	897.217
0.504	946.773	-87.624	-0.156	-29.216	41.951	0.214	102.340	-0.376	20.000	20.000	98991.577	593.093
0.528	964.561	741.157	-0.280	-28.918	41.496	0.384	103.739	-0.675	20.000	20.000	15397.942	541.960
0.552	1009.742	1882.535	-0.471	-28.537	40.904	0.647	104.883	-1.135	20.000	20.000	15397.942	685.052
0.576	1076.369	2776.124	-0.751	-28.042	40.283	1.035	105.658	-1.814	20.000	20.000	15397.942	616.636
0.600	1159.571	3466.756	-1.140	-27.416	39.642	1.572	105.934	-2.754	20.000	20.000	15397.942	584.503
0.624	1266.351	4449.156	-1.640	-26.664	39.006	2.262	105.806	-3.961	20.000	20.000	15397.942	686.729
0.648	1413.464	6129.730	-2.249	-25.782	38.364	3.103	104.725	-5.433	20.000	20.000	15397.942	911.157
0.672	1606.278	8033.926	-2.979	-24.741	37.634	4.106	103.543	-7.188	20.000	20.000	15397.942	1130.296
0.696	1845.905	9884.086	-3.820	-23.530	36.796	5.271	102.169	-9.229	20.000	20.000	15397.942	1342.509
0.720	2127.783	11744.854	-4.781	-22.151	35.820	6.599	100.604	-11.552	20.000	20.000	15397.942	1564.469
0.744	2451.821	13501.584	-5.860	-20.605	34.737	8.088	98.849	-14.159	20.000	20.000	15397.942	1799.132
0.768	2818.798	15290.726	-7.055	-18.891	33.536	9.739	96.904	-17.047	20.000	20.000	15397.942	2047.840
0.792	3227.047	17010.363	-8.366	-17.012	32.219	11.549	94.771	-20.215	20.000	20.000	15397.942	2311.773
0.816	3612.683	18688.172	-9.791	-14.970	30.788	13.515	92.453	-23.657	20.000	20.000	15397.942	2587.764
0.840	3923.423	12947.504	-11.275	-12.842	29.298	15.564	90.038	-27.243	20.000	20.000	15397.942	2777.016
0.864	4126.119	8445.670	-12.770	-10.698	27.796	17.629	87.605	-30.856	20.000	20.000	15397.942	2884.580
0.888	4286.157	6751.571	-14.285	-8.555	26.294	19.693	85.172	-34.469	20.000	20.000	15397.942	2969.383
0.912	4446.879	6613.440	-15.760	-6.411	24.792	21.757	82.739	-38.081	20.000	20.000	15397.942	3050.128
0.936	4608.230	6722.949	-17.255	-4.268	23.290	23.822	80.306	-41.694	20.000	20.000	15397.942	3128.585
0.960	4772.722	6853.804	-18.750	-2.124	21.789	25.886	77.873	-45.307	20.000	20.000	15397.942	3204.935
0.984	4941.186	7019.328	-20.246	0.020	20.287	27.950	75.440	-48.920	20.000	20.000	15397.942	3278.259
1.008	5113.307	7171.724	-21.741	2.163	18.785	30.015	73.007	-52.533	20.000	20.000	15397.942	3351.662
1.032	5289.029	7321.754	-23.236	4.307	17.283	32.079	70.574	-56.146	20.000	20.000	15397.942	3422.276
1.056	5467.634	7441.894	-24.731	6.450	15.781	34.143	68.141	-59.759	20.000	20.000	15397.942	3491.258
1.080	5648.937	7554.275	-26.226	8.594	14.279	36.208	65.708	-63.372	20.000	20.000	15397.942	3558.772
1.104	5833.958	7709.213	-27.721	10.738	12.777	38.272	63.275	-66.985	20.000	20.000	15397.942	3624.976
1.128	6022.518	7866.597	-29.216	12.881	11.276	40.336	60.842	-70.598	20.000	20.000	15397.942	3690.019
1.152	6213.322	7950.255	-30.712	15.025	9.774	42.401	58.409	-74.211	20.000	20.000	15397.942	3754.043
1.176	6407.531	8092.021	-32.207	17.168	8.272	44.465	55.976	-77.823	20.000	20.000	15397.942	3817.138
1.200	6604.556	8201.047	-33.702	19.312	6.770	46.529	53.543	-81.436	20.000	20.000	15397.942	3879.371
1.224	6796.067	7987.959	-35.197	21.455	5.268	48.594	51.110	-85.049	20.000	20.000	15397.942	3940.767
1.248	6967.787	7155.007	-36.686	23.591	3.772	50.650	48.687	-88.664	20.000	20.000	15397.942	3984.459

Fig. 12 General report values and intervals

8.. Conclusions

The propose of conducting this simulation was to recreate as close as possible the digital twin of the real world application in order to determine all the physical restrictions and to analyse the system behaviour and to determine the bottlenecks and the path segments that has to be modified in order to obtain a better overall results in terms of lifetime improvement. The development of off-line simulation created new oportunities in the field of predictive maintainance, industrial management and plant process planification.

9. Bibliography

[1]. Nicolescu, A. – *Implementarea Robotilor Industriali in Sistemele de Productie, note de curs si metodologii de proiectare*, UPB, 2023

[2]. Nicolescu, A., Stanciu, M.D., Popescu D. – *Conceptia si Exploatarea Robotilor Industriali - Vol.1 Tendinte actuale in conceptia si exploatarea RI. Precizia de lucru si precizia volumetrica. Componente organologice specifice. Tehnici si metode de studiu al comportarii elastice si performantelor robotilor industriale*. 156 pag, Editura Printech 2004, Bucuresti, cod CNC SIS 54, ISBN 973-718-007-0, Cota BNR III 263217

[3]. Nicolescu, A. – *Implementarea Robotilor Industriale in Sistemele de Productie, note de curs si metodologii de proiectare*, UPB, 2023

[4]. Nicolescu A., Dobrescu T., Ivan M., Ayram C., Brad S., Doroftei I., Grigorescu S. – *Roboti Industriale, Tehnologii si Sisteme de Productie Robotizate*, 190 pag., Ed Academiei Oamenilor de Stiinta din Romania, 2011, ISBN 978 – 606 – 8371 – 48 – 1, CIP BNR 2011-21962, Cota BNR III 307660

OFFLINE PROGRAMMING AND SIMULATION OF A ROBOTIC PACKAGING CELL IN SIEMENS PROCESS SIMULATE

POPESCU Andrei-Alexandru și TUDOR Alexandru-George (Conducător Științific)

Facultatea: IIR, Specializarea: Robotică, Anul de studii: IV, e-mail:
andrei_popescu_2009@yahoo.com

This paper presents the programming and offline simulation of a robotic packaging cell using Siemens Process Simulate. The purpose of the work was to design a virtual prototype of a flexible robotic cell for packaging plastic containers into cardboard boxes, integrating an ABB IRB 4600-60/2.05 articulated arm robot. The methodology involved importing 3D CAD geometry from CATIA and Siemens NX into Process Simulate, defining kinematics for each subsystem (robot, gripper, conveyors, pistons, sliders), creating event-based logic using signal modules equivalent to PLC programs, and generating Pick & Place trajectories via the Path Editor. The simulation covered the full operational cycle: container rotation and positioning, box opening, loading by the dual gripper (angular grippers and suction cups), and box transport out of the cell. Collision-free trajectories were validated through the Sequence Editor, and the complete cell logic was implemented using photoelectric and proximity sensors interconnected via logical signal equations. The results confirmed that the robotic cell operates correctly within the required cycle time and positioning accuracy constraints, with no collisions detected. The conclusions indicate that Siemens Process Simulate is a highly effective environment for event-based offline programming of complex robotic applications.

CUVINTE CHEIE: simulare offline, celulă robotizată, Process Simulate, programare event-based, ambalare.

1. Introduction

This paper presents the design and offline simulation of a flexible robotic cell for packaging plastic containers into cardboard boxes. The application integrates an ABB IRB 4600-60/2.05 six-axis articulated industrial robot controlled by the IRC5 platform, together with a complex system of conveyors, sensors, and pneumatic mechanisms. The main objective of the project was to develop a complete virtual prototype of the robotic cell in the Siemens Process Simulate environment and validate its operation through event-based simulation, without the need for physical equipment.

The project activities included a comparative analysis of similar robotic solutions, 3D CAD modeling in CATIA V5 and Siemens NX, kinematic definition of each subsystem, development of the operational flowchart, sensor mapping, signal-based logic programming, and generation of robot trajectories. No major structural modifications to the cell were required compared to the initial concept, as the simulation confirmed the absence of collisions and compliance with the imposed functional requirements (productivity: 50 parts/hour, positioning accuracy: ± 0.1 mm, maximum cycle time: 5 s).

2. Current State of the Art

Offline simulation of robotic cells has experienced significant development in recent years, with several specialized software platforms becoming available for both specific robot manufacturers and multi-vendor applications. Reference [1] presents the use of KUKA.Sim Pro software for the simulation and optimization of industrial robot trajectories, highlighting its collision detection capabilities and cycle-time reduction potential. Similarly, Reference [2] analyzes the ABB RobotStudio environment for the offline programming of a welding robot, demonstrating the validation of robotic applications before physical implementation.

Reference [3] describes the FANUC ROBOGUIDE platform, which is used for the simulation of material handling and packaging systems, with particular emphasis on production flow optimization. Reference [4] provides a comparison of multi-vendor simulation environments, particularly RoboDK and Siemens Tecnomatix Process Simulate, concluding that Process Simulate offers superior capabilities for integrating complex manufacturing processes and event-based programming. Furthermore, Reference [5] documents the implementation of signal-based programming and logical modules within Process Simulate for a palletizing cell, similar to the application presented in this work.

These studies justify the selection of Siemens Process Simulate as the development environment for the present project due to its advanced capabilities in simulating complete production systems and integrating automation logic. Additionally, Reference [2] presents a 3D simulation model of an end-effector designed for pick-and-place operations.

3. Design and Simulation of the Robotic Packaging Cell

The first stage of the project consisted of creating a complete 3D CAD model of the robotic cell using CATIA V5 and Siemens NX. Each subsystem was modeled individually, after which the complete assembly was generated and exported in STEP format. The individual STEP files were then imported into Siemens Process Simulate (Fig. 1), where each component was assigned the appropriate type using the “Define Component Type” command: robot, device (actuated mechanism), or part (transported object).

The cell consists of sixteen main components, the most important being the ABB IRB 4600-60/2.05 industrial robot, the dual end-effector (two angular grippers with jaws and two pneumatic suction cups), the box-opening system, a dual belt conveyor, a roller conveyor, a box pushing and positioning system, a container rotation system, and the technical cabinet housing the robot controller, air preparation unit, and automation interface.

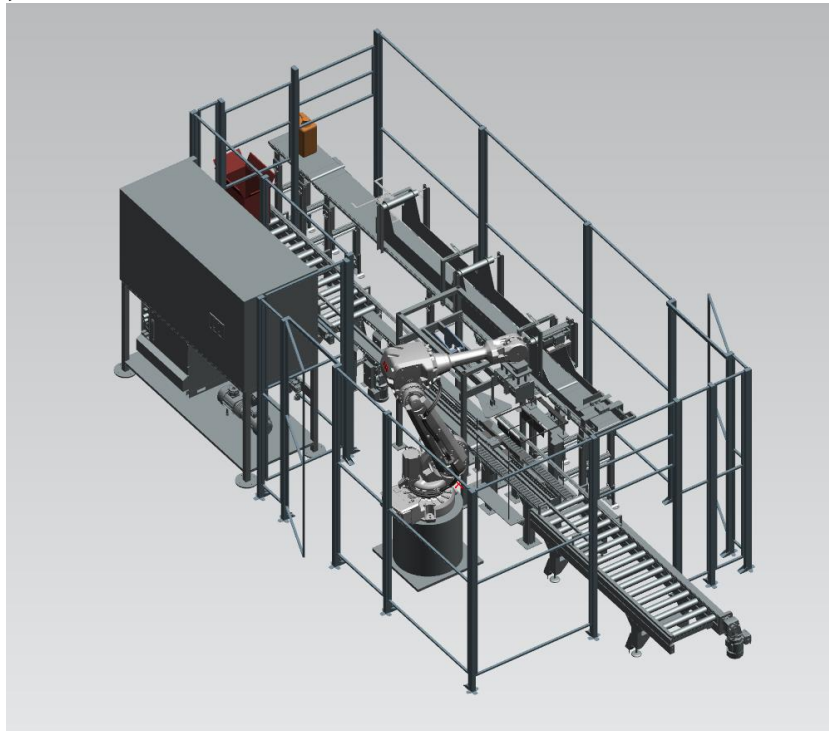


Fig. 1. General view of the robotic cell in Process Simulate

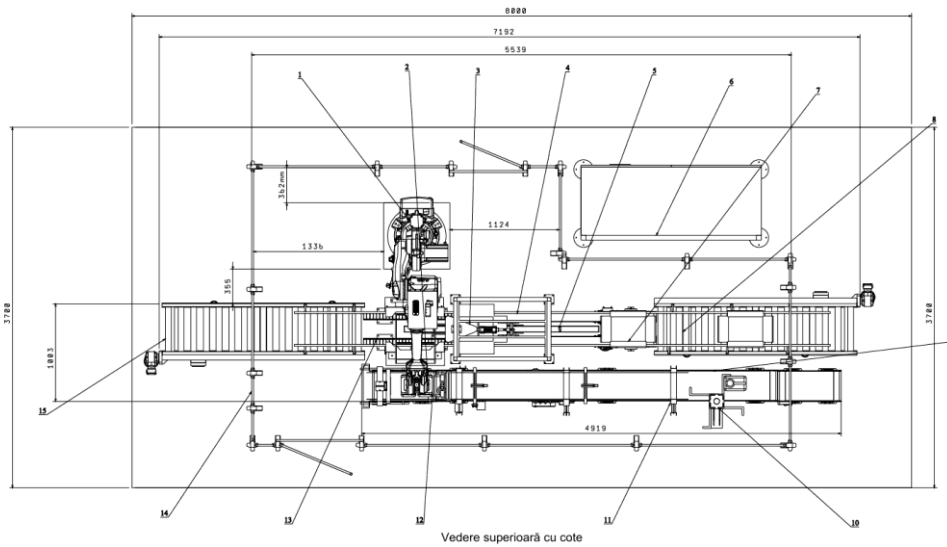


Fig. 2. Footprint layout of the robotic cell

The second stage focused on defining the kinematics of each subsystem within Process Simulate. The ABB IRB 4600-60/2.05 robot model was downloaded directly from the manufacturer's website in .cojt format, which is specific to the Tecnomatix environment and contains all configuration and process files required for accurate simulation behavior. Unlike a simple geometric model, this file behaves as a fully functional robot rather than a generic device (Fig. 3).

The end-effector kinematics were defined using four pneumatically actuated movable jaws attached to a fixed housing (Fig. 4). The box-opening piston was configured with three links and two rotational axes (Fig. 5), corresponding to the rear-axis rotation and the rotation of the opening element. The kinematics of the box-positioning slider and the container-positioning system were implemented by defining translational axes and correctly establishing fixed-to-moving part relationships for each mechanism.

For each subsystem, a series of predefined “Poses” (reference positions) were created, representing the target positions that each mechanism must reach during the operational sequence.

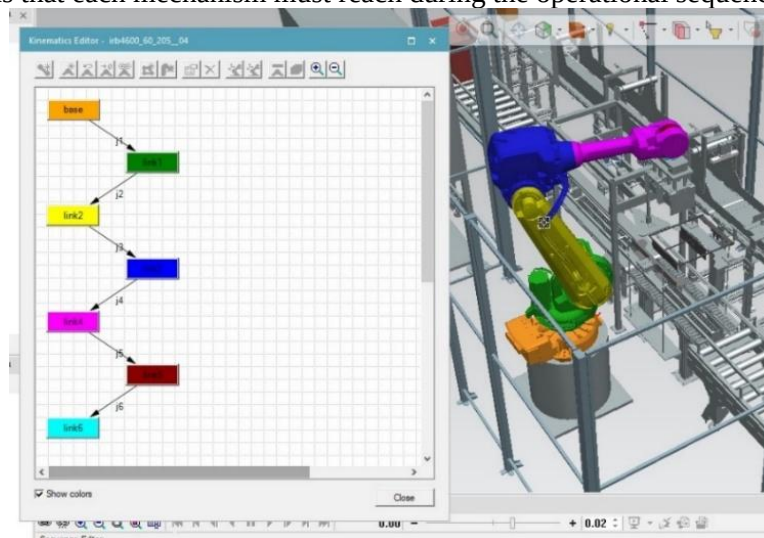


Fig. 3. Industrial Robot Kinematic Definition

OFFLINE PROGRAMMING AND SIMULATION OF A ROBOTIC PACKAGING CELL IN SIEMENS PROCESS SIMULATE

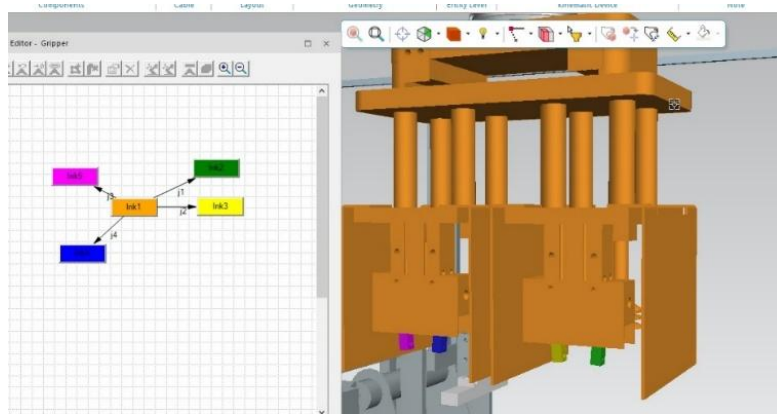


Fig. 4. Gripper Kinematic Definition

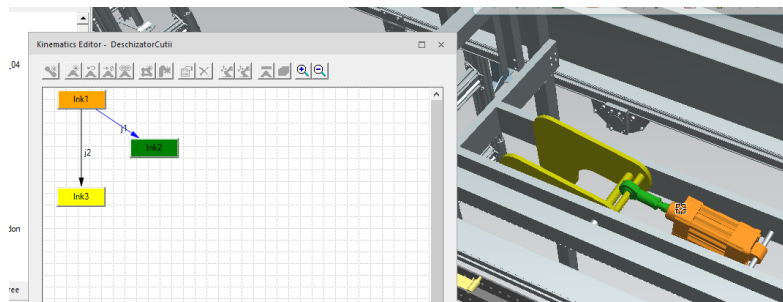


Fig. 5. Piston Kinematic Definition

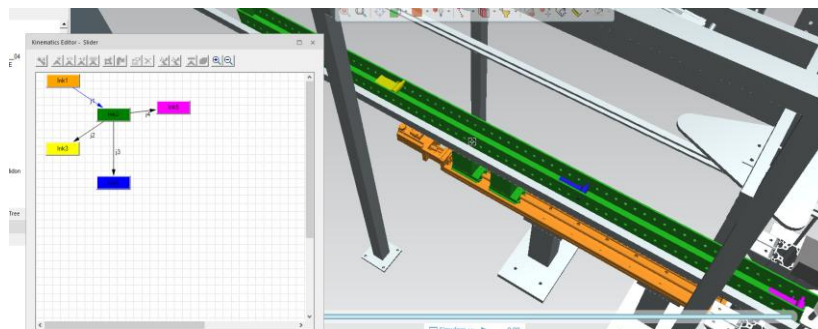


Fig. 6. Slider-Type Mechanism Kinematic Definition

The third stage involved creating the material flow simulation and defining the robot operations. Conveyor movements were implemented using “Object Flow Operations,” specifying the transported object (container or box), start point, destination point, and transition duration. Mechanism actuations (piston, slider, and container rotation system) were programmed using “Device Operations,” allowing transitions between the previously defined positions.

Pick-and-place operations were created by selecting the robot, end-effector, pickup point, and placement location. Intermediate (“via”) positions were manually added using the Path Editor to ensure collision-free trajectories (Fig. 6). Since the complete opening of the cardboard box cannot be physically simulated within Process Simulate, a show/hide visualization technique was used. The partially opened box was hidden and visually replaced by a fully opened version.

All operations were linked together within the “Sequence Editor” (Fig. 7), where temporal dependencies and conditional relationships were established, resulting in the complete operational sequence of the robotic cell. The main functional parameters imposed by the project requirements and validated through simulation are presented in Table 1.

OFFLINE PROGRAMMING AND SIMULATION OF A ROBOTIC PACKAGING CELL IN SIEMENS PROCESS SIMULATE

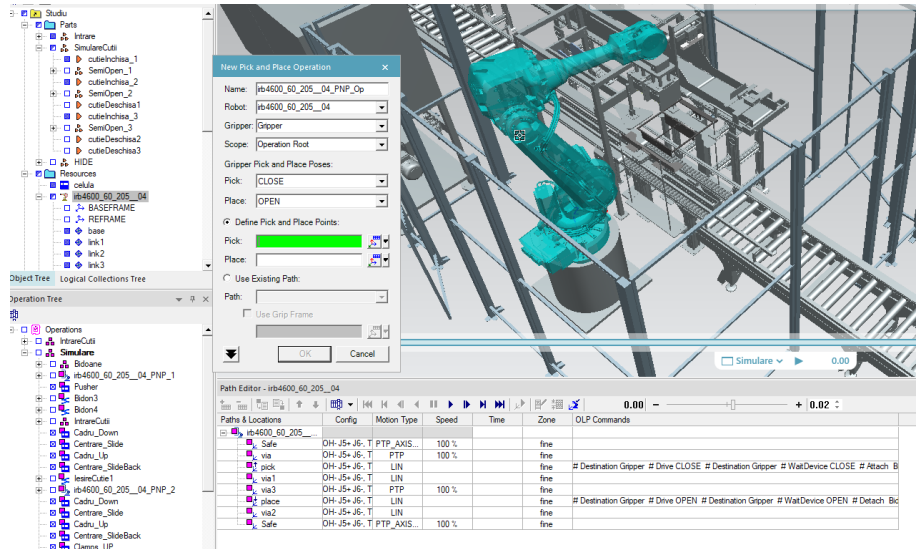


Fig. 6. Pick-and-Place Handling Operation

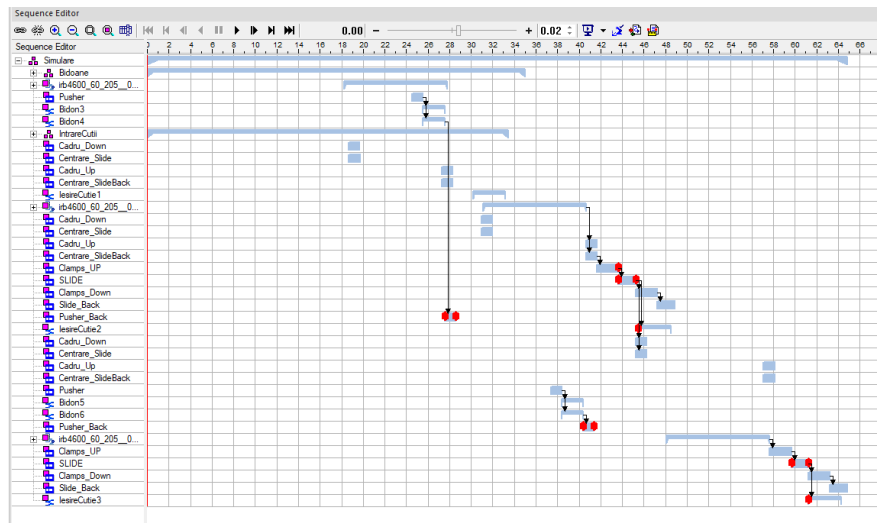


Fig. 7 Sequence Editor Window

Tabelul 1. Required Functional Parameters

Parameter	Unit	Observations	
		Required Value	Status
Productivity	parts/hour/robot	50	Confirmed
Positioning Accuracy	mm	± 0.1	Confirmed

The final stage consisted of implementing the event-based programming of the entire application using the “Line Simulation Mode” available in Process Simulate. Standard robot signals such as *startProgram*, *programNumber*, *emergencyStop*, and *programEnded* were generated, together with application-specific interaction signals including *at_Pick*, *Safe*, *Go_To*, and *Go_From*. A total of seven logical modules equivalent to PLC programs were created (Fig. 8): one for conveyor control, one for robot control, and one for each additional subsystem integrated into the cell. Photoelectric and proximity sensors were defined in Process Simulate and connected to the logical

OFFLINE PROGRAMMING AND SIMULATION OF A ROBOTIC PACKAGING CELL IN SIEMENS PROCESS SIMULATE

modules through Boolean equations.

The robot program was implemented using OLP (Offline Programming) commands inserted in the Path Editor, with signal set/reset actions assigned to each key trajectory point (pickup, placement, waiting positions). The Material Flow Viewer was used to define the relationships between pickup and placement operations for each part, ensuring continuity throughout the simulation. Figure 9 illustrates the Process Simulate interface with active event-based programming.

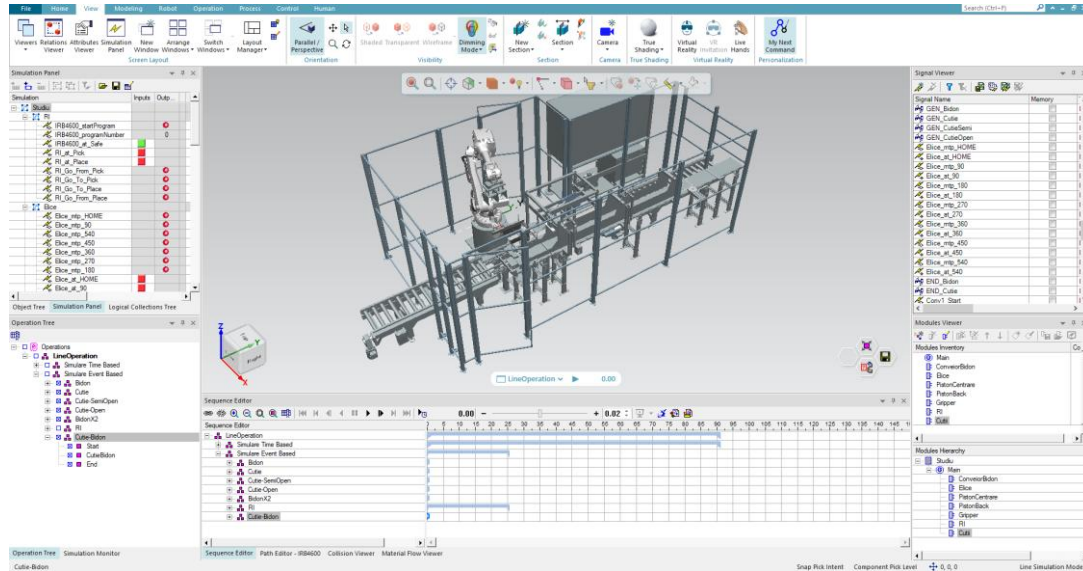


Fig. 8. PLC-Equivalent Logical Modules

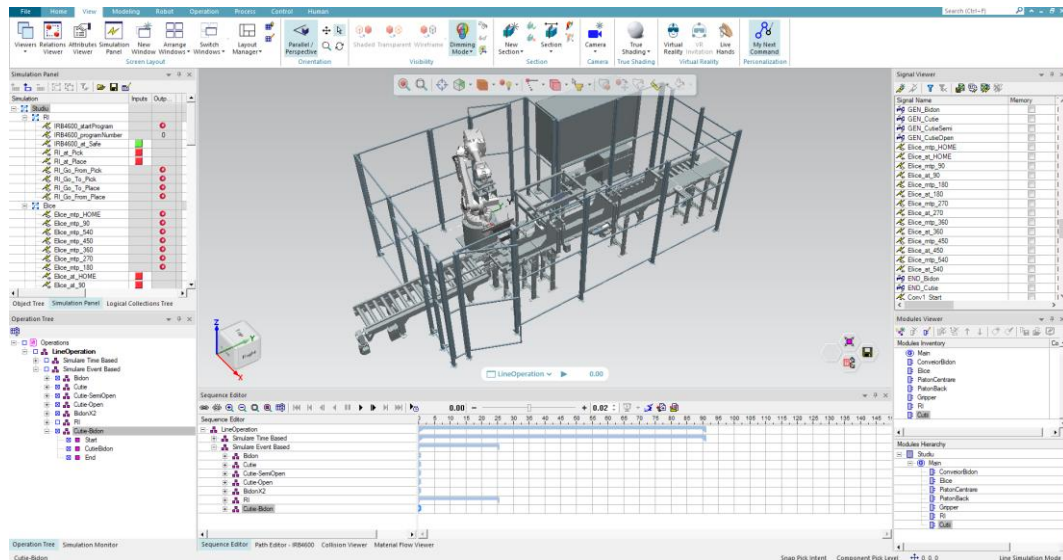


Fig. 9. Process Simulate Interface – Event-Based Programming

A total of twenty-seven sensors were integrated into the simulation, including sixteen detection sensors, four safety sensors, and emergency stop buttons. These sensors were connected to the logical modules through Boolean equations, enabling realistic control logic implementation.

The ABB IRB 4600-60/2.05 robot proved to be an appropriate solution for this application. Its 2051 mm working reach covers the entire operational area, while its 60 kg payload capacity provides a sufficient safety margin with respect to the mass of the end-effector and handled products.

4. Conclusions

This paper presented the design and offline simulation of a flexible robotic packaging cell for plastic containers using the ABB IRB 4600-60/2.05 industrial robot and Siemens Process Simulate. All stages of a complete engineering project were addressed, including comparative analysis of existing solutions, 3D CAD modeling, subsystem kinematic definition, development of the operational flowchart, mapping of twenty-seven sensors (photoelectric, proximity, and safety sensors), event-based programming using seven logical modules, and robot trajectory generation.

The simulation successfully validated the correct operation of the robotic cell without collisions while meeting all imposed functional requirements, including a productivity of 50 parts/hour, positioning accuracy of ± 0.1 mm, maximum cycle time of 5 seconds, and minimum availability of 98%.

The selection of Siemens Process Simulate proved to be justified due to its ability to integrate automation logic directly with kinematic simulation, enabling complete event-based programming and virtual commissioning. Future work will focus on the physical implementation of the robotic cell at a laboratory scale and validation of the generated programs on real industrial equipment.

5. References

- [1]. Siciliano, B. și Sciavicco, L. (2009), *Robotics: Modelling, Planning and Control*, Springer, Londra, ISBN 978-1-84628-641-4.
- [2]. Pan, Z. și Polden, J. (2012), “Recent progress on programming methods for industrial robots”, *Robotics and Computer-Integrated Manufacturing*, 28(2), pp. 87-94, ISSN 0736-5845.
- [3]. Dröder, K. și Bobka, P. (2018), “Simulation-based planning of robot-assisted assembly processes”, în: T. Bauernhansl (ed.), *Handbuch Industrie 4.0*, Springer, Berlin, pp. 321-340, ISBN 978-3-662-45537-1;
- [4]. Neto, P. și Pires, J.N. (2011), “Accelerometer-based control of an industrial robotic arm”, *Proceedings, IEEE International Conference on Robotics and Automation*, Shanghai, Mai 2011;
- [5]. ABB Robotics (2022). *IRB 4600 Product Specification*. ABB Group, Zürich, ISBN/Doc. nr. 3HAC026660-001;
- [6]. Siemens AG (2023). *Tecnomatix Process Simulate – User Manual*, Siemens Digital Industries Software, Plano TX.

CAD AND OLP OF A ROBOTIC CELL FOR PALLETIZING CARDBOARD BOXES WITH FOOD PRODUCTS

DRAGNE Valentin, NICOLESCU Adrian Florin

Faculty of Industrial Engineering and Robotics, Specialization Robotics, Year of study IV,
e-mail:validragne03@yahoo.com

REZUMAT: This paper presents the design, virtual simulation, and automation of a robotic sorting and palletizing cell capable of processing 4 distinct product types. The methodology includes the 3D CAD modeling of a custom clamp end-effector, together with the integration of standard logistics equipment into the robotic cell. The complete geometry was imported into ABB RobotStudio, where the system was tested through offline simulation. The results indicate that the robotic trajectories, pneumatic actuators, and automated guided vehicle were successfully coordinated by means of RAPID programming and smart components. Therefore, the simulation confirmed the functional reliability of the proposed robotic cell, allowed the optimization of robot paths to avoid collisions, and demonstrated the system's efficiency in accordance with the principles of Industry 4.0.

KEYWORDS: Robotic palletizing, Offline simulation, End effector, Design, Industry 4.0.

1. Introduction

The current project involves the creation of a complex robotic sorting and palletizing cell, capable of simultaneously managing 4 distinct types of products. The main goal is the design, virtual simulation, and complete automation of the entire technological flow, minimizing human intervention.

The main stages consisted of selecting the robot (ABB IRB 660), designing the clamp-type gripper, and importing the standard structural elements: part and pallet conveyors, a wrapping machine, a barcode reader, and barrier sensors. For the precise routing of the flow, the modeling of additional components was necessary: pushing systems (deflectors) for diverting rejects, stoppers, and centering systems. The pallet circuit uses an automated rail-guided vehicle (RGV), and a label applicator operates at the end of the line.

All these elements were assembled in a computer-aided design (CAD) environment, optimizing the workspace for the four robotic cells so that the central control panel can efficiently manage the entire process in a compact configuration.

Subsequently, the cell was imported into ABB RobotStudio for simulation and programming. The process included configuring the signals received from the sensors and programming the palletizing trajectories, including the sequences for adding separator sheets between the layers of boxes.

In the final stage, the system was adjusted to perfectly synchronize the communication between the conveyors, the automated vehicle, and the pneumatic drives, thus ensuring fluid operation and the integration of the entire cell into Industry 4.0 standards.

2. Current state

Regarding the design and optimization of end effectors, paper [1] details a design and finite element analysis (FEA) model for a palletizing gripper. The authors evaluate the structural strength and kinematics of the gripping mechanism, ensuring that the applied force is optimal for load handling, taking into account the mass of the stack and the closing clearances. Similarly, paper [2] presents the development of an automated

packaging and palletizing system, focusing on the gripper's efficiency in handling boxes of various sizes at high speeds.

Another essential aspect for reducing implementation times is virtual simulation. Paper [3] presents a complete methodology for the design and simulation of a workstation for a palletizing robot using the ABB RobotStudio virtual environment. The study demonstrates how virtual (offline) programming optimizes the robot's trajectories, prevents collisions, and validates cycle times before the physical implementation of the equipment.

Besides the actual handling, the integration of automated transport systems within the Industry 4.0 concept is crucial for the technological flow. Paper [4] addresses the logistical control of the material flow managed through smart conveyors and automated guided vehicles. The authors highlight the importance of synchronizing the signals received from barrier sensors and barcode readers for fluid control of pallet traffic and the prevention of blockages.

The current project brings together these research directions, proposing an integrated solution. Unlike classical approaches, the proposed system combines the mechanical design of a clamp-type gripper with the advanced simulation of the entire logistical circuit. Specific logistical components, such as pushing systems (deflectors), stoppers, and centering systems, were implemented and modeled, thus ensuring a fully automated sorting and palletizing process with maximum operational efficiency.

3. Modeling the virtual prototype of the end-effector

From a structural standpoint, the motors, the pneumatic valve island, the vacuum ejectors, the cylinders, and the suction cups were integrated using standard models sourced from specialized manufacturers' catalogs. Conversely, the load-bearing mechanical structure, the forks, the side walls, and the connecting elements were entirely designed and 3D modeled at the concept level. The dimensions of these custom components were determined by analyzing a reference video, being estimated through direct comparison with the overall size of the robot and the handled boxes.

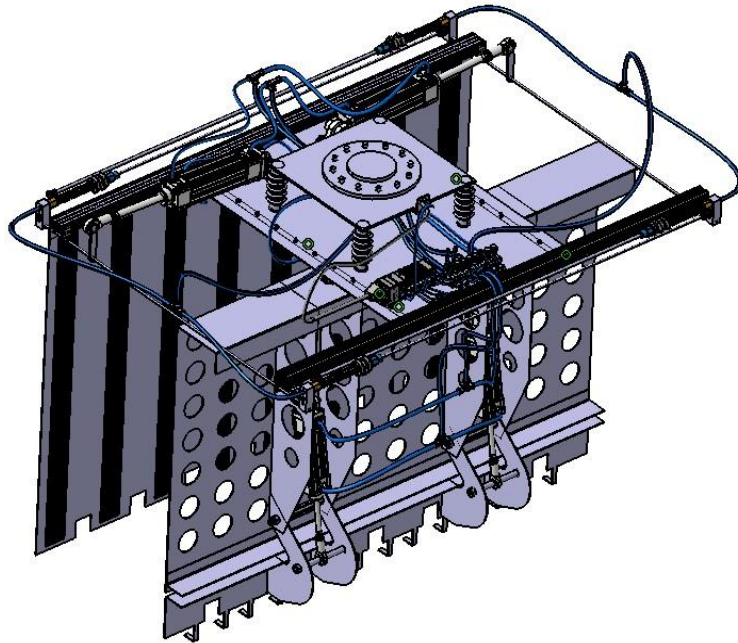


Fig.1. 3D synthesis of the end-effector

4. 3D Modeling of the Robotic Palletizing Cell

The first step in configuring the cell consisted of modeling the end effector and assembling it onto the industrial robot arm, a procedure completed by routing and connecting the pneumatic tubing, as well as the associated electrical wiring:

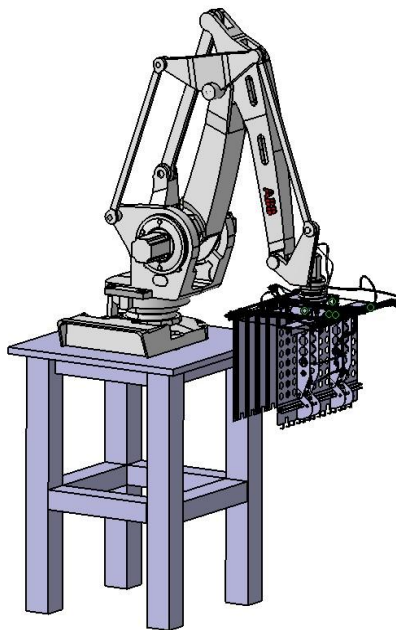


Fig.2. Industrial Robot mounted on the pedestal with the end-effector

attached

To create the virtual assembly of the robotic cell, a series of standard industrial equipment was sourced directly from specialized manufacturers' catalogs and configured according to the application's parameters. This category includes the conveyor network, the perimeter safety fences, the wrapping machine, as well as the barcode reader.

On the other hand, the components that required a strict adaptation to the dynamics of the technological flow were 3D modeled and custom-assembled to achieve optimal shape and functionality. Thus, the centering systems, the rail-guided vehicle (RGV), the pusher systems, and the stoppers were specifically designed to meet the exact requirements of the application and to integrate perfectly into the cell's architecture.

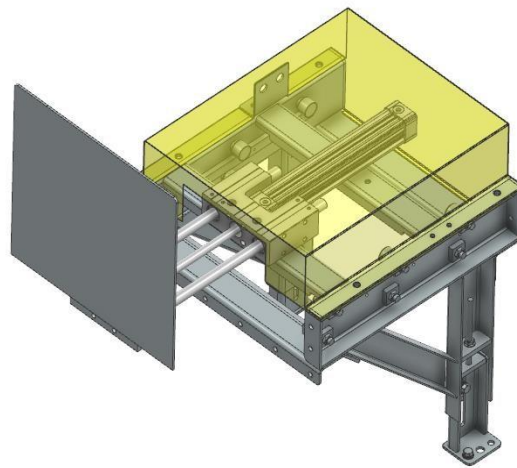


Fig.3. 3D synthesis of the object pusher

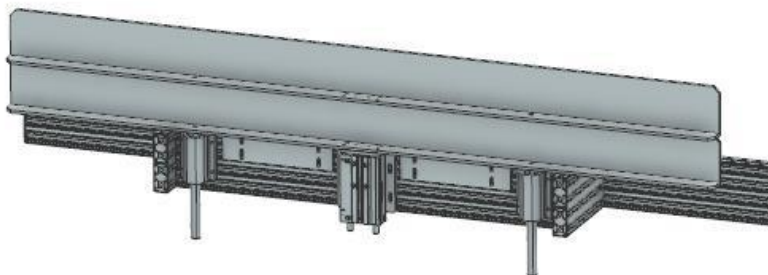


Fig.4. 3D synthesis of the stoppers

CAD AND OLP OF A ROBOTIC CELL FOR PALLETIZING CARDBOARD BOXES WITH FOOD PRODUCTS

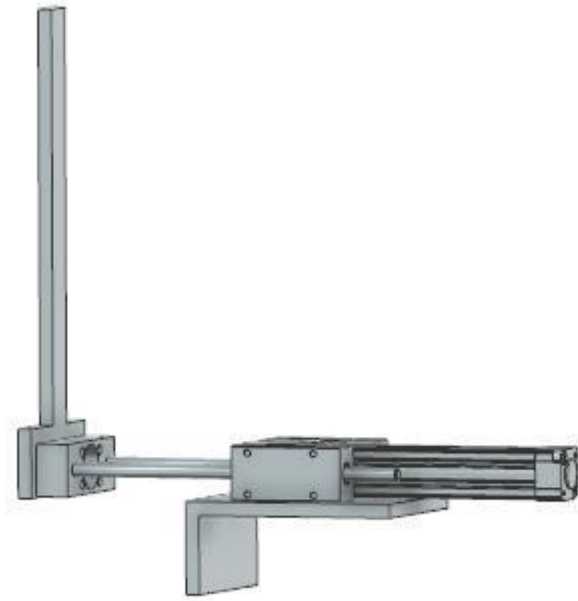


Fig.5 3D synthesis of the pusher for pallet positioning

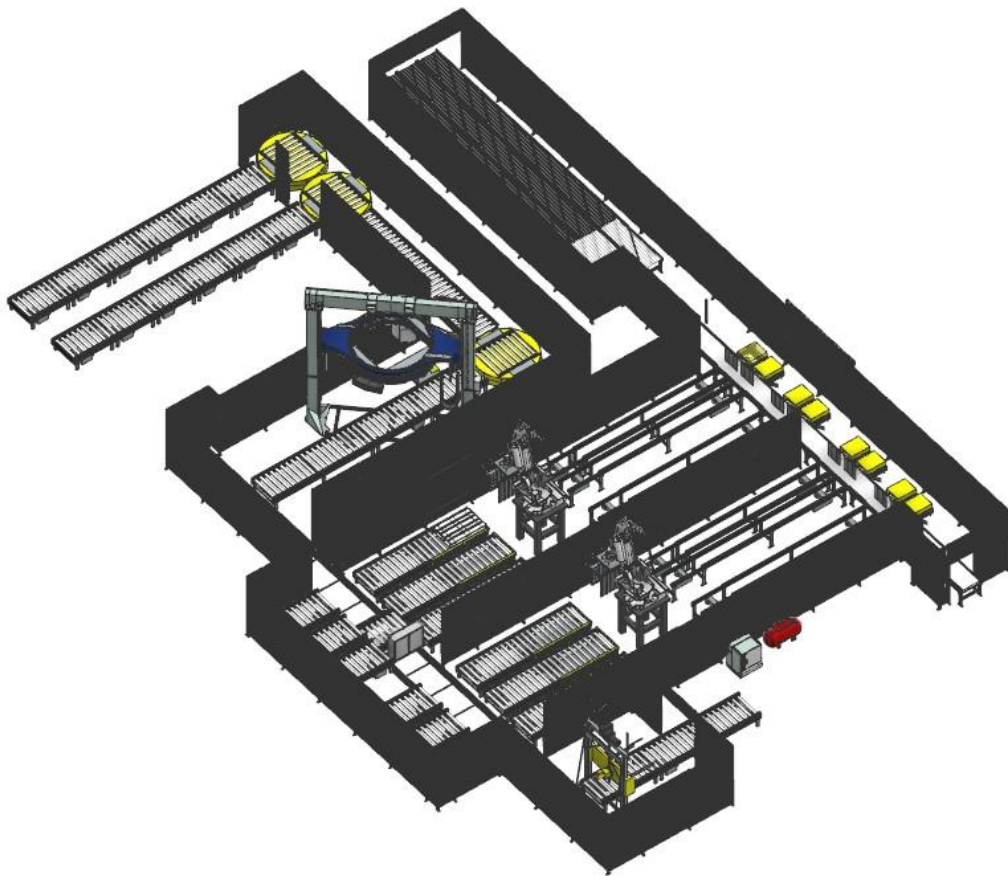


Fig.6. 3D synthesis of the final assembly of the robotic cell

5. Simulation of the Designed Robotic Application Using the ABB RobotStudio

To perform this simulation, it was necessary to create a virtual station in the RobotStudio application, including a controller and the industrial robot model corresponding to the designed cell, the ABB IRB 660.

The offline programming and simulation environment allows the direct import of 3D CAD formats, a procedure through which the clamp-type gripper assembly was inserted into the current project. To be able to simulate the end effector's operation, a new "Tool" type mechanism was created. Within it, the mechanism's axes were defined by selecting the moving components (the gripping jaws), the translation directions, and the opening/closing limits for picking up the boxes in pairs. An essential step was configuring a TCP (Tool Center Point), which establishes the coordinate system attached to the end effector.

Once the mechanism was compiled, predefined positions were set for its actuation. From the "Event Manager" menu, digital output signals from the robot's controller were allocated to command the opening and closing of the gripper.

For the flow simulation of the entire cell, the first step was loading the CAD file containing the complete virtual prototype. Subsequently, the auxiliary external mechanisms were kinematically configured: the centering systems, the stoppers, the pushers for diverting rejects, the wrapping machine, the label applicator, and the RGV system intended for pallets.

The objects handled in the palletizing process were generated using "smart components," allowing the continuous instantiation of the 4 types of boxes, the empty pallets, and the separator sheets. Distinct paths were configured for the advancement of the boxes on the conveyors, the pallet route, and the positioning of the separator sheets, logically integrating barrier sensors and the barcode reader along these routes for flow control.

The final step consisted of defining and correlating the communication signals for each external mechanism, importing the created paths into RAPID, and writing the main routine. Thus, the program that will be fully and synchronously executed by the industrial robot and the adjacent equipment was finalized.

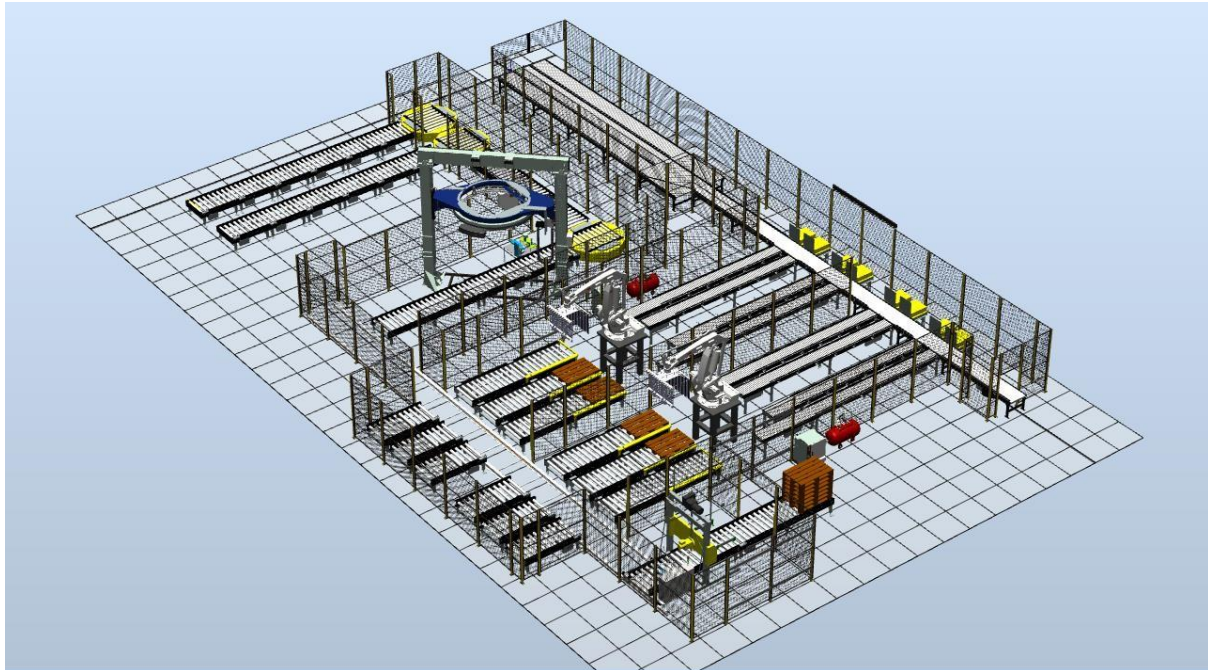


Fig.7 The virtual prototype of the cell imported into RobotStudio

6. Conclusions

In conclusion, the project achieved its objective of designing, simulating, and fully automating a complex robotic palletizing cell, capable of simultaneously managing 4 types of products, thereby minimizing human intervention. The technical solution consisted of integrating the ABB IRB 660 robot, equipped with a multifunctional end effector specifically designed for gripping boxes and handling separator sheets. The cell's architecture efficiently combined standardized industrial equipment, such as the conveyor network and the wrapping machine, with custom-modeled elements, including centering systems, deflectors, stoppers, and an automated rail-guided vehicle (RGV). The concept validation was successfully achieved in the ABB RobotStudio virtual environment, where all mechanisms were kinematically configured, the material flow was generated using "smart components," and the trajectories were programmed in the RAPID language. The perfect synchronization of signals between the controller, sensors, and external mechanisms confirmed the fluid operation of the entire process, attesting to the successful integration of the robotic cell into the specific standards of Industry 4.0.

7. Bibliography

- [1]. Zhang, H., Li, F. și Wang, J. (2018), „Design and finite element analysis of palletizing robot end-effector”, *Journal of Physics: Conference Series*, 1074, 012154.
- [2]. Lin, C. J., et al. (2011), „Development of an automated packaging and palletizing system”, *Robotics and Computer-Integrated Manufacturing*, 27 (4), 745-750.
- [3]. Wang, J., Zhang, L. și Zhao, M. (2018), „Design and Simulation of Palletizing Robot Workstation Based on RobotStudio”, *IOP Conference Series: Materials Science and Engineering*, 423, 012056.
- [4]. Silva, M. F., et al. (2019), „Industry 4.0 concepts applied to automated logistics and guided vehicles”, *IFAC-PapersOnLine*, 52 (13), 2486-2491.

CAD AND OLP OF A ROBOTIC CELL FOR PALLETIZING CARDBOARD BOXES WITH FOOD PRODUCTS

- [5]. ABB Robotics (2022), RobotStudio Operating Manual, ABB Robotics, Västerås, Sweden;
- [6]. Drăghici, G. (1999). Ingineria integrată a produselor. Editura Eurobit, ISBN 973-96065-7-1, Timișoara.
- [7]. ABB RobotStudio – Operating Manual, User Guide, 2023.

COMPUTER-AIDED DESIGN AND OFFLINE PROGRAMMING OF A ROBOTIC PALLETIZING CELL USING MOTOSIM EG-VRC

ILEANA Nicoleta

(Prof.dr.ing. Adrian Florin NICOLESCU)

¹Faculty: IIR, Specialization: Robotics, Year of study: IV, e-mail: nicoleta.ileana58@gmail.com

SUMMARY: This paper presents the assisted design and offline programming–simulation of a robotic palletizing cell using the MotoSim EG-VRC environment. The complete virtual prototype of the cell was built in NX Siemens and imported into MotoSim EG-VRC, where the industrial robot Yaskawa PL500 was configured alongside the YRC 1000 controller. Two simulation stages were developed: a preliminary stage focused on robot pick-and-place trajectories written in INFORM Language, and a full-cell simulation stage integrating the pallet magazine, pallet lifters, conveyors, stoppers, and pushers via Model Script Editor commands (SEE, HID, MOV, DUP, DEL, ACT). The final MASTER program automates the complete palletizing cycle for two full pallets with a third pallet on standby.

KEYWORDS: offline simulation, MotoSim EG-VRC, robotic palletizing, INFORM Language, computer-aided programming

1. Introduction

This paper focuses on designing and implementation of a robotic palletizing cell for syrup boxes, built around the Yaskawa PL500 industrial robot equipped with a high-density foam pad vacuum end-effector (PALLETIZER PRO – FOAM VACUUM GRIPPER, JOULIN), controlled by a YRC 1000 controller. The complete virtual prototype of the cell was developed through 3D CAD in NX Siemens, incorporating conveyors, an automatic pallet magazine, lifters, pneumatic cylinders used as stoppers and pushers, and sensors.

The specific focus of this paper is the offline programming and simulation of the robotic cell's functionality within the MotoSim EG-VRC environment — dedicated software for Yaskawa robots, which enables complete validation of the robot's trajectories and the logic of all cell subsystems prior to physical commissioning. Offline simulation eliminates downtime associated with programming on the real robot, reduces collision risks, and allows cycle time optimization in a virtual environment. The paper presents two distinct stages: (1) the preliminary simulation of the industrial robot's operation (PICK and PLACE movements onto the pallet, with automatic layer height calculation) and (2) the full simulation of the robotic cell, integrating all automated peripheral subsystems coordinated through scripts in the Model Script Editor.

2. Current State of the Art

Offline programming of industrial robots is a widely used method in modern industry. Recent studies [1] show that offline programming reduces commissioning time by 40–60% compared to conventional teach-in programming, eliminating production line downtime. Pan et al. [1] also emphasize the importance of calibrating the virtual model against the real robot for accurate program transfer.

Robotic palletizing cells using industrial robots (4–6 axes) with foam vacuum end-effectors are analyzed in [2], where it is highlighted that the Yaskawa PL500 — with 4 numerically controlled axes, a payload capacity of 500 kg, and a repeatability of ± 0.05 mm — represents an optimal solution for handling a full row of 10 boxes (189.27 kg) within competitive cycle times. Furthermore, dedicated palletizing robots frequently employ MOVL and MOVJ motion types for trajectory optimization, as opposed to purely joint-based MOVJ motions [2].

The MotoSim EG-VRC environment is presented in the Yaskawa technical documentation [3] as a complete simulation solution, including Model Script Editor functionalities (commands SEE, HID, MOV, DUP, DEL, ACT), dual controller configuration, and I/O signal simulation. Recent industrial applications [4] confirm the effectiveness of MotoSim EG-VRC for validating INFORM Language programs prior to deployment on the real robot, including for complex cells with multiple automated subsystems.

COMPUTER-AIDED DESIGN AND OFFLINE PROGRAMMING OF A ROBOTIC PALLETIZING CELL USING MOTOSIM EG-VRC

3. Description of the Robotic Palletizing Cell

The designed robotic cell is intended for palletizing syrup pouch boxes of two types — 2.5 gallons (9.463 kg/unit) and 5 gallons (18.927 kg/unit) — onto standard American 48"×40" pallets (1219.2×1016 mm). The palletizing pattern uses a full row of 10 boxes handled simultaneously by the JOULIN vacuum end-effector, alternating row orientation by 180° every 2 layers to achieve a compact and balanced stack.

The main subsystems of the cell are: (1) Yaskawa PL500 industrial robot on a pedestal; (2) JOULIN PALLETIZER PRO vacuum end-effector (1300×1100 mm); (3) 2 three rowed conveyor belts for box feeding; (4) 2 active roller conveyors (Intralox) for sorting and redirecting boxes at 90°; (5) chain conveyor for pallet transport; (6) automatic pallet magazine (Winkel); (7) pallet lifters; (8) 4 linear pneumatic cylinders used for compaction; (9) stop gate and pusher cylinders; (10) YRC 1000 controller and Allen Bradley HMI.

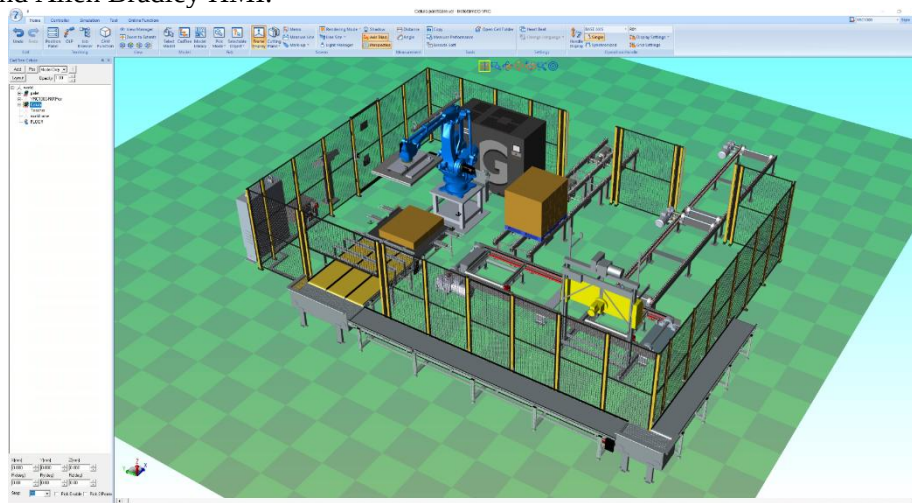


Fig. 1. Virtual prototype of the entire robotic application in the MotoSim EG-VRC environment

4. Preliminary Simulation – Industrial Robot Operation

4.1. MotoSim EG-VRC Environment Configuration

For the preliminary simulation, the entire 3D prototype of the robotic cell was imported into MotoSim EG-VRC, after which the YRC 1000 robot controller and the Yaskawa PL500 robot, placed in a fixed position on the raised base, were configured. The tool coordinate system (TCP) of the vacuum end-effector was defined and the associated kinematic parameters were set.

4.2. INFORM Language Program Structure

The robot programming was carried out in INFORM Language — Yaskawa's proprietary language. A modular architecture was adopted with a main MASTER program that calls 4 subprograms: PICK (pick-up), PLACE (placement), CALCULARE (layer Z-position calculation), and INIT (simulation initialization).

The PICK subprogram defines the pick-up points for boxes from the conveyor and the associated trajectories, using MOVL (linear) and MOVJ (joint) motions. The code line `MDS:PICK invokes the Model Script settings to attach the boxes to the end-effector's TCP, so that subsequent motions (up to placement) are executed relative to the tool coordinate system.

COMPUTER-AIDED DESIGN AND OFFLINE PROGRAMMING OF A ROBOTIC PALLETIZING CELL USING MOTOSIM EG-VRC



Fig. 2. Syntax of the PICK subprogram

The PLACE subprogram defines the box placement points on the pallet using MOVL and MOVS (spline) motions. An IF condition is used to rotate rows by 180° every two layers, following the alternating palletizing pattern. The placed-box counter (I000) is incremented by 1 at each placement. The `MDS:PLACE line detaches the boxes from the TCP and attaches them to the pallet, while simultaneously generating a new box on the conveyor to continue the cycle.



Fig. 3. Syntax of the PLACE subprogram

The CALCULARE subprogram automatically determines the placement height on the Z-axis for each pallet layer. The value is calculated as: $D = I000 \times 200.125 \text{ mm}$ (height of the 5-gallon box), where D is declared as a calculated position on the Z-axis relative to the robot base (Z – 3).

INIT resets all counters to 0, deletes box copies generated during simulation (via a FOR loop), and restores the original model on the conveyor. The MASTER program orchestrates the sequential calling of all 4 subprograms and includes the condition for counting palletized layers. The preliminary simulation demonstrates the correct operation of the Yaskawa PL500 robot in completing a full pallet according to the palletizing pattern.

COMPUTER-AIDED DESIGN AND OFFLINE PROGRAMMING OF A ROBOTIC PALLETIZING CELL USING MOTOSIM EG-VRC

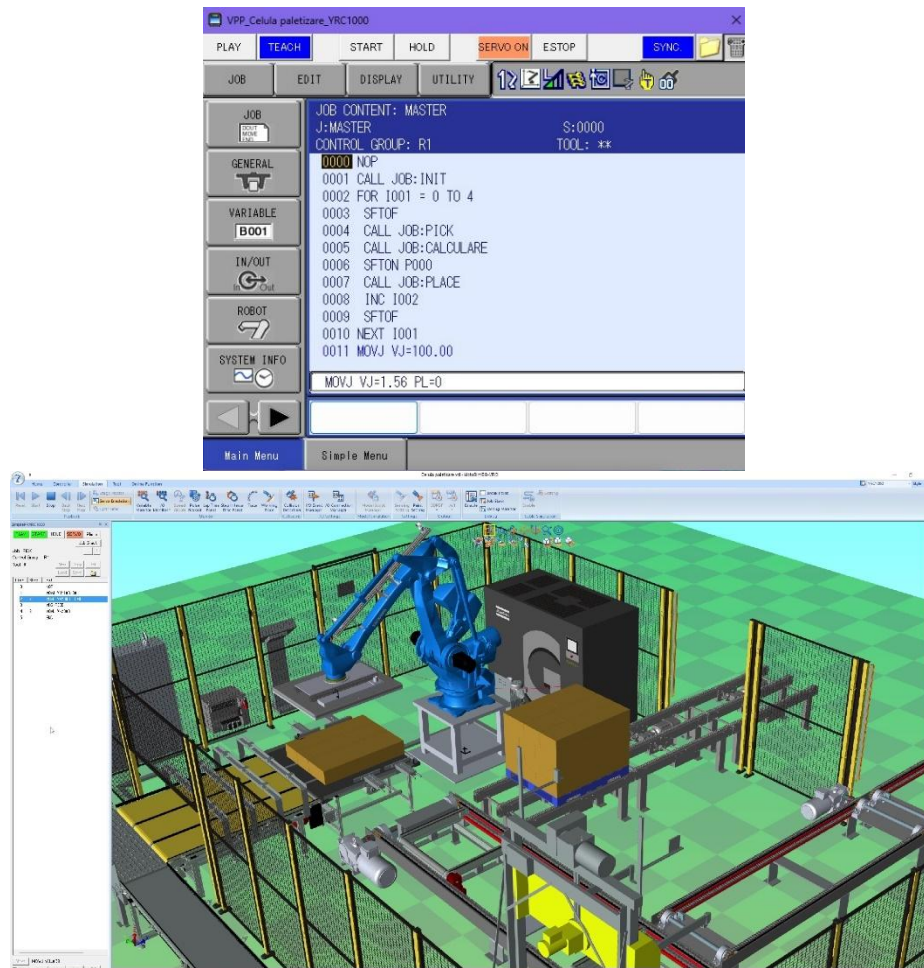


Fig. 4. MASTER program syntax and overview from the preliminary simulation

5. Full Simulation of the Robotic Cell

5.1. Model Script Editor – Commands Used

The simulation of the entire robotic cell's operation extends the preliminary program by adding the motions of all peripheral subsystems, implemented as scripts in the Model Script Editor within MotoSim EG-VRC. The commands used are: SEE – displays the specified entities (appearance of boxes on the conveyor); HID – hides entities (resumes the box feeding cycle); MOV – attaches one entity as a “child” of another (e.g., the pallet to the magazine arms, or to the lifter for transport); DUP – generates copies of an entity (pallets in a stack, boxes for feeding); DEL – deletes an entity (e.g., individual box copies after row compaction); ACT – executes linear/rotational motions on the X, Y, Z axes (movement of boxes on conveyors, actuation of stoppers, pusher, lifter, pallet magazine with all intermediate positions).

5.2. Motion Subprograms for Connected Elements

The MOVEP subprogram coordinates the scripts for feeding the cell with the pallet stack, translating the stack toward the pallet magazine, loading the magazine, and bringing the first empty pallet into the palletizing area. The MOVEP1_2 and MOVEP2 subprograms handle the feeding of pallets 2 and 3 into the palletizing area, each pallet following the same automated logistics flow.

The MISCARI subprogram manages the generation and movement of boxes within the

COMPUTER-AIDED DESIGN AND OFFLINE PROGRAMMING OF A ROBOTIC PALLETIZING CELL USING MOTOSIM EG-VRC

cell from the entry point to the robot's pick-up area. Feeding is performed 10 boxes per palletizing cycle — at the end of each row placed on the pallet, a new set of 10 boxes is automatically generated on the conveyor, ensuring continuity of the production flow.

The INITP subprogram initializes the positions of pallets within the cell (magazine, lifters, palletizing conveyor), while INITC deletes all box copies, repositions them at the start of the feeding line, and resets the counters. Together with INIT (retained from the preliminary simulation), these subprograms allow the simulation cycle to be restarted from the initial state.

The final MASTER program calls all 10 subprograms (PICK, PLACE, CALCULARE, MOVEP, MOVEP1_2, MOVEP2, MISCARI, INIT, INITP, INITC). The robot waits in the PICK subprogram for the WAIT IN#(10)=ON signal — confirmation from the other controller that compaction of the box row is complete — before moving to pick up the layer. All peripheral subsystem motions are assigned to a separate controller, allowing the robot and other systems to move in a coordinated manner through I/O signal exchange.

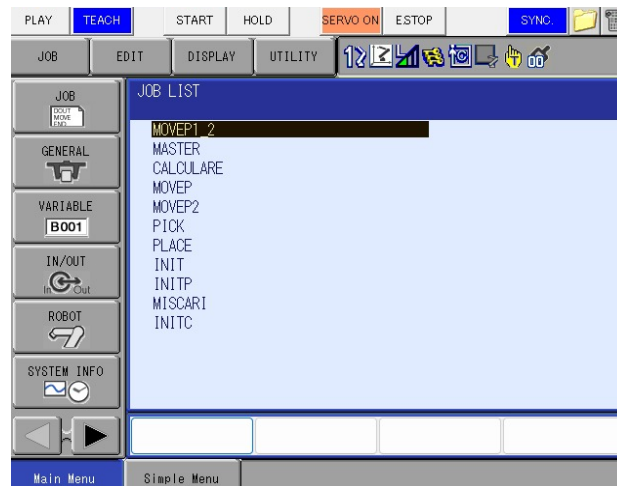


Fig. 5. List of subprograms used for the full simulation



Fig. 6. Overview from the full simulation of the robotic cell

6. Conclusions

Using the presented MASTER program, the robotic cell runs in a fully automated manner, completing the palletizing of 2 full pallets and feeding the third pallet which remains on standby. The main results obtained are summarized in the table below:

**COMPUTER-AIDED DESIGN AND OFFLINE PROGRAMMING OF A ROBOTIC PALLETIZING CELL
USING MOTOSIM EG-VRC**

Table 1. Summary of offline simulation results for the robotic cell

Parameter	Value / Description
Industrial robot used	Yaskawa PL500 (4 axes, 500 kg, ± 0.05 mm)
End-effector	JOULIN PALLETIZER PRO – FOAM VACUUM GRIPPER (1300×1100 mm)
Payload per cycle	10 boxes $\times$ 18.927 kg = 189.27 kg (full row)
Standard pallets used	48"×40" (1219.2×1016 mm) – American standard
Robot motion speed	3000 mm/s (translație principală)
INFORM Language subprograms	10 (PICK, PLACE, CALCULARE, MOVEP, MOVEP1_2, MOVEP2, MISCARI, INIT, INITP, INITC)
Model Script Editor scripts	Over 50 scripts (box movements, pallets, stoppers, pushers, magazine)
Simulation result	2 paleți compleți + al 3-lea palet în așteptare, funcționare complet automată

7. References

[1]. Pan, Z. et al. (2012), "Robotic machining from programming to process control: a complete solution by force control", *Industrial Robot*, vol. 39(1), pp. 36–45.

[2]. Gröger, M. et al. (2021), "Trajectory optimization for palletizing robots with vacuum grippers", *Robotics and Computer-Integrated Manufacturing*, vol. 68, art. 102087.

[3]. Yaskawa Electric Corporation (2023). *MotoSim EG-VRC User's Manual*, Document NX100-MotoSim EG-VRC Rev. 7, Kitakyushu, Japan.

[4]. Nicolescu, A. (2025), *Implementation of Industrial Robots in Production Systems, lecture notes and design methodologies*, UPB, Bucharest.

[5]. Drăgoi, M. (2025), *Fundamentals of Offline Programming of Industrial Robots, lecture notes and application support*, UPB, Bucharest.

[6]. *** Yaskawa Europe GmbH (2024). *Yaskawa PL500 – Product Specifications*, available online: www.yaskawa.eu.com.

Offline Programming of a Robotic Manufacturing Cell for CNC Machine Tending

CHELARU Larisa-Georgiana¹ and NICOLESCU Adrian-Florin (Scientific Advisor)

¹Faculty: IIR, Specialization: Robotics, Year of study: IV, e-mail: larisa.chelaru@stud.fiiir.upb.ro

ABSTRACT: *This paper presents the design and virtual validation of a robotic manufacturing cell dedicated to automated CNC machine tending operations for prismatic mechanical parts. The proposed system integrates a FANUC M20iA/20MT industrial robot with hybrid kinematics, capable of servicing multiple machining centers while minimizing floor-space requirements. A dual-gripper end-effector was designed to enable simultaneous handling of raw and finished parts, improving cycle efficiency. The complete robotic cell was modeled in a 3D CAD environment, including conveyors, CNC machines, vision systems, safety equipment, and auxiliary subsystems. Offline simulation was performed in FANUC ROBOGUIDE to validate robot accessibility, workspace configuration, motion sequencing, and collision avoidance. The study confirms the functional feasibility of the proposed automation concept and highlights the advantages of virtual prototyping in reducing implementation risks, improving layout optimization, and accelerating deployment in flexible manufacturing environments.*

KEYWORDS: *Machine tending, FANUC ROBOGUIDE, manufacturing cell*

1. Introduction

This paper presents the design and virtual validation of a robotic cell intended for the automated tending of CNC machining centers for prismatic mechanical components. The application falls within the field of flexible manufacturing systems, where the use of industrial robots contributes to increased productivity, reduced auxiliary times, and improved repeatability of handling and transfer processes between technological equipment.

The main objective was to develop an integrated robotic application capable of automating the handling flow of blanks and finished parts between several CNC centers, using a FANUC M20iA/20MT industrial robot with hybrid kinematics. The choice of this robot type was determined by its advantages in terms of high accessibility within the workspace, extension of the operating range through the translational axis, and reduction of the overall floor footprint of the application. Moreover, the cycle time was optimized through the use of a dual end-effector capable of simultaneously handling the blank and the finished part.

The project involved several successive stages. In the first stage, a comparative analysis was performed of similar robotic applications and existing constructive solutions for automating machining operations. Subsequently, the general configuration of the application was established and the main equipment was selected, namely the industrial robot and the required peripheral subsystems. Next came the 3D computer-aided design stage for the end-effector, for the auxiliary subsystems for which CAD models were not provided by manufacturers, and for integrating the entire application into a complete virtual prototype. In the final stage, the operation of the application was validated through offline simulation in the FANUC ROBOGUIDE environment, with focus on robot accessibility, handling sequences, collision avoidance, and validation of the overall operating logic.

During development, constructive adaptations of certain subsystems were required, particularly for integrating the designed end-effector, configuring the pneumatic actuation system, creating supports for auxiliary equipment, and modeling auxiliary components. These modifications preserved the general functionality of the application and the compatibility among all integrated subsystems. The paper highlights the advantages of using virtual prototyping and

offline simulation to reduce implementation risks and optimize industrial robotic systems.

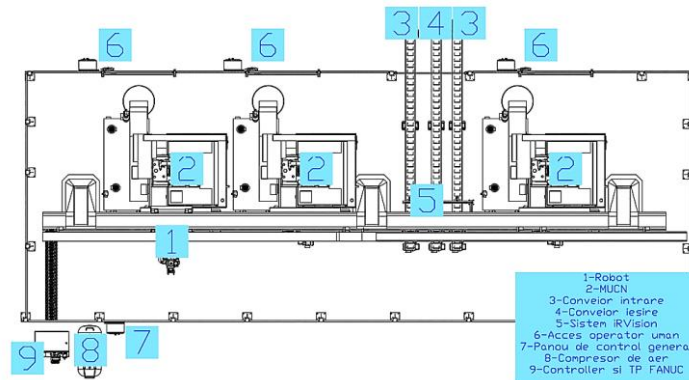


Fig. 1. Subsystems included in the application

2. Current state of the art

Developing robotic systems for manufacturing automation is one of the major research directions in contemporary industrial engineering, especially in the context of the transition toward the concepts of Industry 4.0 and Smart Manufacturing. The integration of industrial robots into flexible manufacturing cells for tending CNC machine tools has become a common solution for reducing auxiliary times, increasing productivity, and improving the repeatability of technological processes. In particular, applications that use industrial robots with hybrid kinematics offer additional advantages by extending the working volume and optimizing the space occupied in the industrial environment. In the literature, numerous studies address the design and operation of integrated robotic systems. In the UPB Scientific Bulletin, Prof. PhD. Eng. Nicolescu and collaborators [1] emphasize the importance of evaluating the accuracy and elastic behavior of industrial robots, showing that the real performance of a robotic system is influenced not only by nominal kinematic characteristics, but also by the structural stiffness of the mechanical assembly. These aspects are essential for handling and automated transfer applications between machine tools, where positioning accuracy directly influences the quality of the technological process.

Also in the UPB Scientific Bulletin, Munteanu [2] analyzes the influence of structural deformability on the accuracy of industrial robots, using numerical methods to model the mechanical behavior of articulated arms. The results demonstrate the relevance of numerical simulation from the design stage, to identify potential constructive limitations and optimize technical solutions.

In papers published at ICMAS, recent research focuses on the use of virtual simulation and offline programming for validating complex robotic systems. In [3], the authors highlight the role of virtual simulation environments in reducing implementation times and minimizing the risks associated with collisions and integration errors in industrial robotic applications. These approaches are relevant to the present project, in which simulation in FANUC ROBOGUIDE is an essential stage of functional validation.

Another ICMAS study [4] analyzes the integration of industrial robots into flexible manufacturing cells for handling and machine-tool tending operations, emphasizing correct configuration of end-effectors and synchronizing the technological flow to reduce system idle times.

In the publications of the Annals of the University of Petroșani, industrial system automation is addressed from the perspective of integrating mechatronic equipment and control systems. In paper [5], the authors present modern concepts regarding the automation of industrial processes and the integration of intelligent handling systems, highlighting the need to accurately model the interaction between mechanical and control components.

Furthermore, [6] presents modern methods for computer-aided design and simulation of complex mechanical systems used in automated industrial environments, emphasizing the advantages of virtual prototyping for validating constructive configurations before physical implementation.

In conclusion, the literature confirms the importance of integrated approaches that combine CAD-assisted design, offline simulation, and functional analysis for the development of industrial robotic applications. This paper follows this direction by proposing a robotic cell for CNC center tending,

using an industrial robot with hybrid kinematics and complete validation in a virtual environment.

3. Kinematics in FANUC ROBOGUIDE

Modeling and validating robot kinematics is one of the essential stages in developing a robotic application, as they enable accessibility verification, motion-trajectory definition, and functional-behavior assessment before physical implementation. In the present paper, the simulation was carried out using the FANUC ROBOGUIDE software environment, a platform dedicated to offline programming and validation of FANUC robotic applications.

The first step in kinematic development is creating a new robotic cell and selecting the robot model required by the application. In the present case, the FANUC M20iA/20MT robot was used; it is a robot with a hybrid kinematic structure, consisting of one linear translational axis and five rotational axes. Once the model is selected from the software library, the ROBOGUIDE environment automatically loads the kinematic model, the constructive axis limits, and the robot-specific operating parameters.

Next, configuring the mounting position and the geometric parameters of the robot. In the analyzed application, the robot was configured in the shelf-mounted variant, to allow easy access to the three machining centers and to reduce the floor space occupied. In the same stage, the stroke of the additional linear axis is defined according to application requirements and the positioning of the subsystems within the cell.

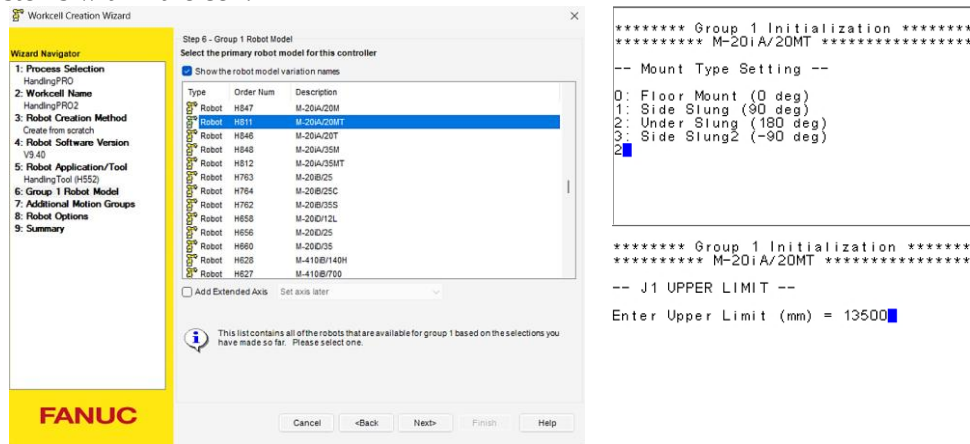


Fig. 2. Selection of the robot configuration in FANUC ROBOGUIDE

After robot configuration, the final end-effector is integrated. For this, the CAD model of the designed end-effector is imported in the ROBOGUIDE environment and attached to the robot flange. The coordinate system associated with the end-effector must be correctly defined, together with the exact positioning of the robot working point (TCP - Tool Center Point). In the developed application, the designed end-effector is a dual one, with two pneumatic grippers arranged at 90°, which required the configuration of two TCP points for the correct handling of blanks and finished parts.

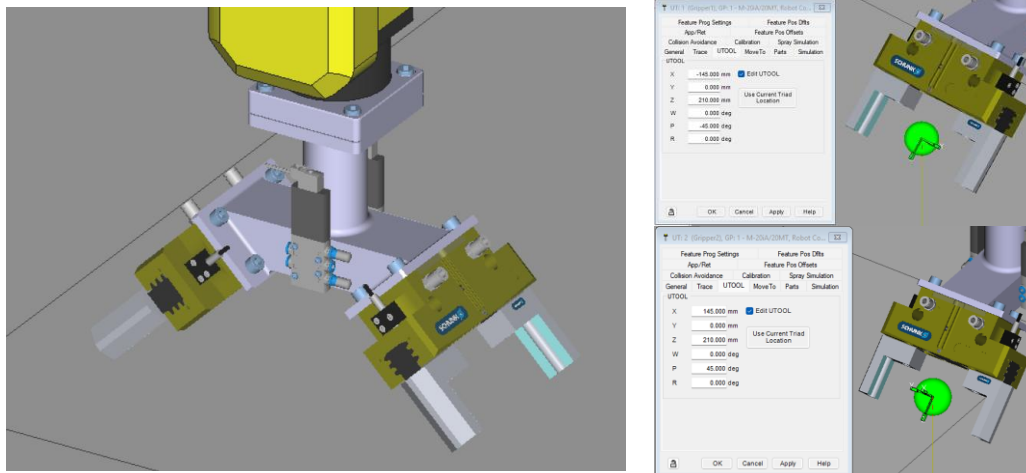


Fig. 3. Insertion of the end-effector and definition of the two TCP points

To configure the robot technological equipment, the dedicated module within the FANUC ROBOGUIDE working environment was used, taking into account the information related to the EE (End Effector) connector, whose role is to divide the signals into output/input signals (Robot Outputs/Robot Inputs) so that they can be interpreted through a single wire connected to it; the EE connector socket is located at the end of segment 2. Next to it is the air connector, which links the air source from the robot base through an internal route. Thus, the length of the air hoses is reduced to preserve the desired pressure and flow rate.

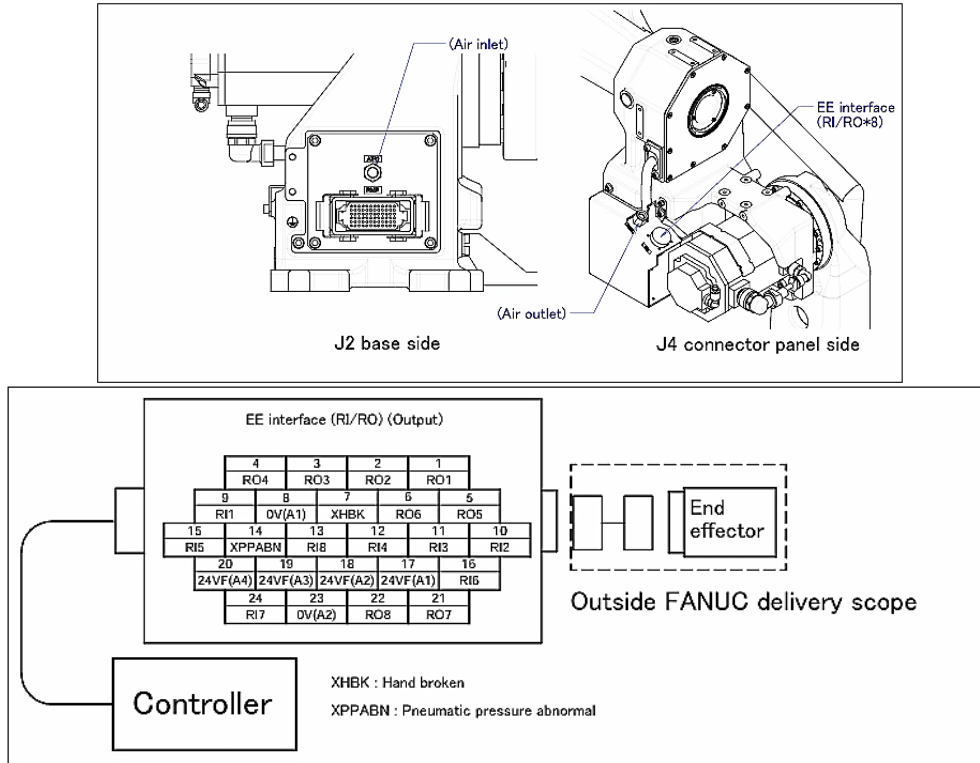


Fig. 4. Information regarding end-effector connection

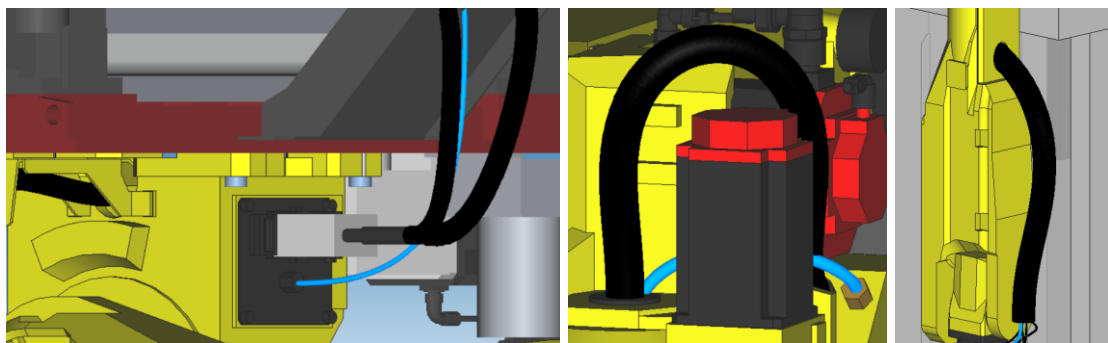


Fig. 5. Coupling of the air source and cables for connection to the controller

Final end-effector actuation involves two separate routes: a pneumatic route (blue hoses), through which air from the source is distributed to the two valves, from which each gripper is supplied, and an electrical route. The electrical route is divided into two subdivisions: a command route (gray wires), through which two RO-type output signals are supplied to the two valves, and a feedback route (black wires), through which four RI-type input signals read the inductive sensors mounted at each end position of the grippers to validate the gripping and, respectively, the placement of the objects.

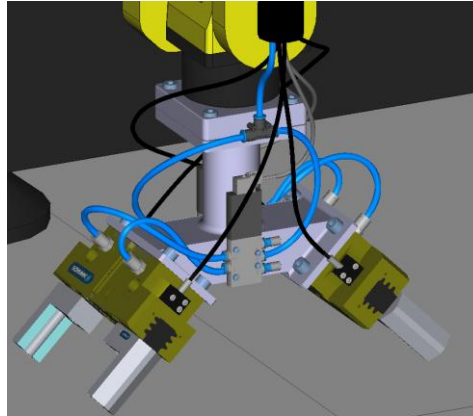


Fig. 6. Complete actuation of the end-effector

For developing the actual kinematics, the characteristic positions of the robot are defined through the teach programming method, using the virtual Teach Pendant interface. Thus, the relevant positions for each technological-cycle stage are recorded: picking parts from the conveyor, intermediate safety positions, access inside the CNC centers, placement positions, and retreat positions. For each point, the optimal kinematic configurations are selected so as to avoid singularities and collisions.

An important aspect of kinematic modeling is the configuration of auxiliary motions and actuation logic. In this application, digital signals were defined for controlling the pneumatic end-effector, opening and closing the doors of the CNC centers, positioning the auxiliary axes, and commanding the part-clamping systems. These signals are integrated into the robot-program sequences for the correct synchronization of the entire application.

Robot motion programming is performed using specific FANUC instructions, such as J (Joint Move) for rapid joint movements and L (Linear Move) for controlled rectilinear motions, which are required in precision handling operations. After the sequences are defined, the application is run in simulation mode for kinematic validation.

The final stage is analyzing the operation and optimizing the kinematics. The accessibility of all working points, the cycle time, possible collisions between the robot and the application subsystems, and the correctness of command synchronization are verified. If problems occur, the recorded positions, axis configurations, or motion paths can be adjusted until stable and efficient operation is achieved and the cycle time is optimized.

By using the FANUC ROBOGUIDE environment, the development of the robot kinematics is carried out in a fully virtual framework, reducing the time required for implementation and enabling validation of the application before commissioning the real system.



Fig. 7. Insertion of all application subsystems

4. Trajectory generation

Trajectory generation is one of the fundamental stages in the development of a robotic application, defining how the robot moves between the characteristic points of the work cycle,

so that the accuracy, safety, and efficiency of the technological process are ensured. In the analyzed application, trajectory generation was performed in the FANUC ROBOGUIDE environment, using the offline programming tools available for configuring the motions of the FANUC M20iA/20MT industrial robot.

The trajectory-generation process begins with identifying the relevant points in the workspace that correspond to the operational stages of the application. These points include the positions for picking blanks from the feed conveyors, intermediate safety points, access positions in the machining centers, part-placement points, and locations for evacuating finished products. Each of these positions is defined by spatial coordinates and specific end-effector orientations, according to handling requirements, as shown in the program diagram in Figure 8.

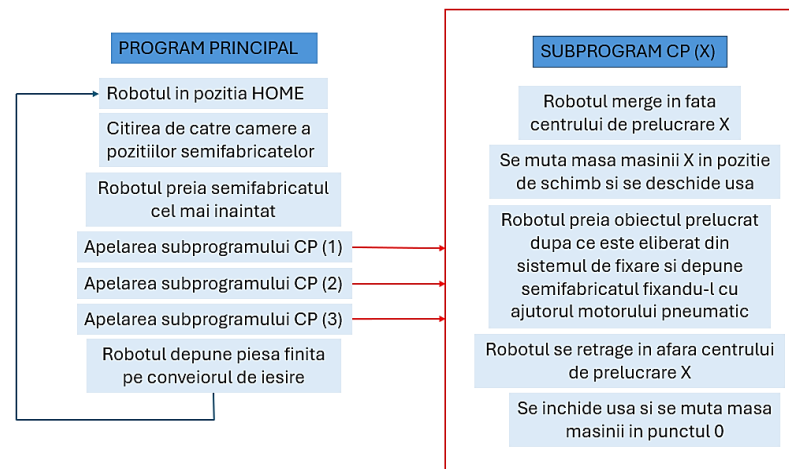


Fig. 8. Block diagram of the studied application

For optimization, each trajectory is assigned specific speed, acceleration, and positioning-accuracy parameters. Free-space movements may use high speeds to reduce cycle time, whereas handling near parts or machines requires reduced speeds and stricter position control.

After all trajectories are defined, they are validated through virtual simulation in FANUC ROBOGUIDE. Collisions, robot kinematic limits, motion smoothness, and synchronization with auxiliary subsystems such as conveyors, clamping systems, or CNC center doors are monitored. If inconsistencies occur, the paths can be adjusted by redefining points or modifying motion parameters.

By generating and optimizing trajectories in the virtual environment, the robotic application can be fully validated before physical implementation, reducing commissioning time and minimizing the risks associated with real system integration.

5. Definition of the dynamic blank-picking position

The system includes four video cameras used to identify the filling level of the two input conveyors, to define a dynamically calculated position for picking a blank from the conveyor that is the most loaded. This assumes that the filling speed of the two conveyors varies, to increasing productivity and smoothing the material flow.

The detection position is therefore calculated in three stages:

- Inductive sensors verify whether at least one of the input conveyors is full (checking whether there is a blank in the last available position on the conveyor);
- If neither conveyor is full, the first camera-set processes (front cameras) are called; these cameras count the blanks within their field of view. The results are compared, and the robot will pick the blank from the most loaded conveyor;

- Because the camera field of view is limited, if the first set of cameras does not detect any blank object, the second set of cameras is used, using the same operating principle.

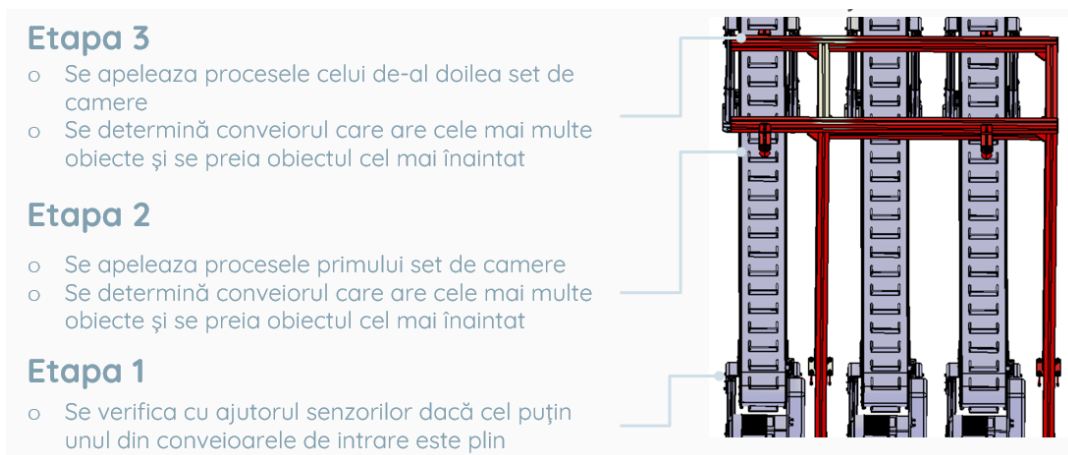


Fig. 9. Identification of the blank-picking position using sensory systems

Implementing the four video cameras required calibration and creation of the corresponding processes.

The vision-process result is the number of detected objects. The program then compares the two values from the cameras in one set and makes the final decision on the next blank-picking position.

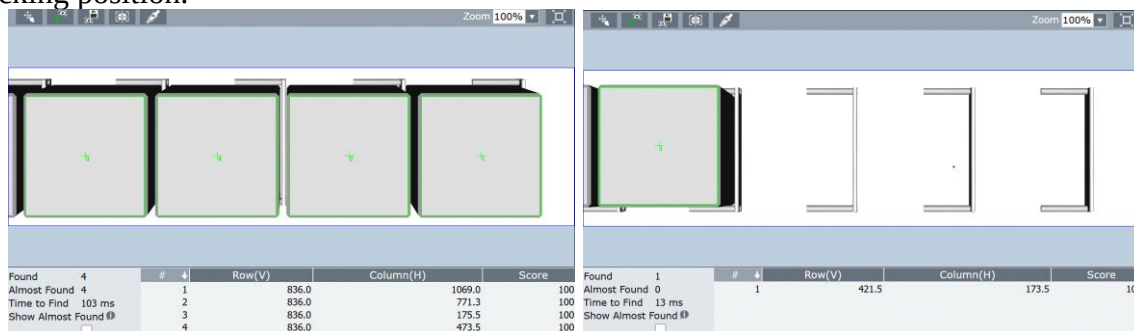


Fig. 10. Example of results obtained through the video-camera process

The above figure shows an example of the results obtained from the two cameras. At the bottom, the number of detected objects can be seen; this number is then introduced into the program logic.

6. Creation of the HMI interface

To facilitate application operation and further project development, a human-machine interface was created and implemented, with the following pages:

- Start page: The main subsystems of the application can be observed and analyzed

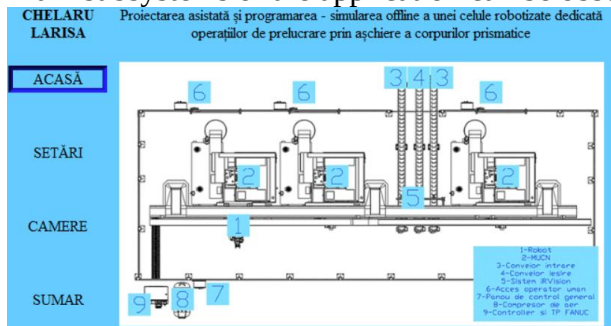


Fig. 11. Start page

- Settings page: Confirmation of the object to be machined is required, assuming that the application may be extended to several part types

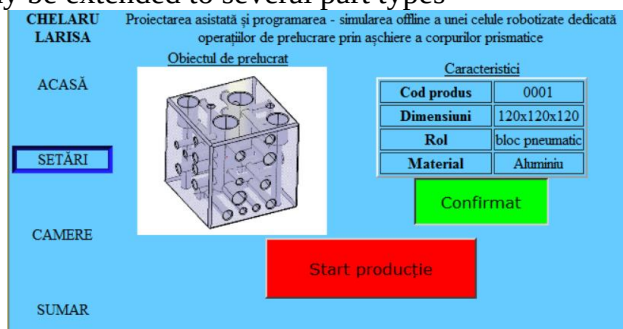


Fig. 12. Settings page

- Camera page: On this page, the status of the pick-position decision can be viewed in real time. If any camera-set process is called, their images are displayed and the final result is highlighted in green.



Fig. 13. Camera page

- Summary page: This page centralizes the main variables of the application, such as the code of the manufactured product, the total processed-part number, each machining-center status, and the status of the safety circuit implemented in the cell

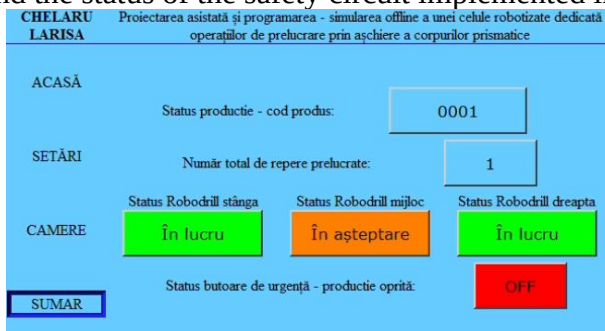


Fig. 14. Summary page

6. Program sequences

The program starts with the robot in the HOME position, then the camera identifies the position of the blanks located on the input conveyors.

The robot picks the most advanced blank and successively calls the CP(1), CP(2), and CP(3) subprograms, for the three machining centers (see fig.8).

In each subprogram, the robot moves in front of the machining center, and the machine table is brought into the exchange position.

The robot extracts the machined part, inserts the new blank, and fixes it using the hydraulically actuated self-centering vise.

Finally, the robot retreats, the machine door closes, the table returns to its initial position, and the finished part is placed on the output conveyor.

Following the previous block diagram from fig.8, several major functional sequences can be extracted:

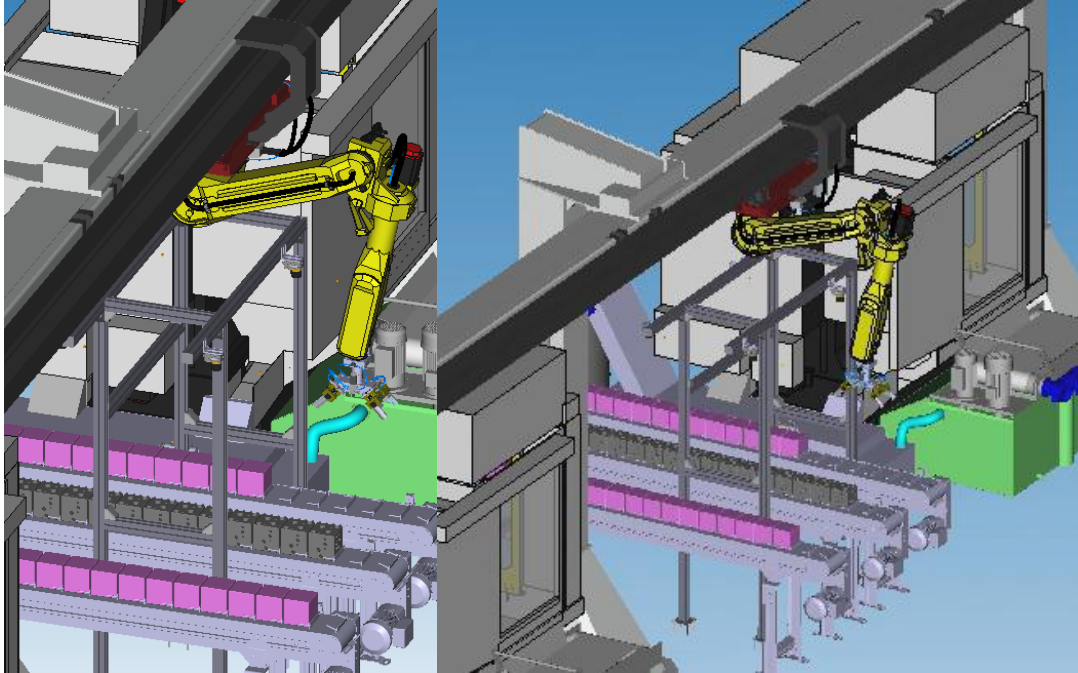


Fig. 15. Waiting position for the decision regarding the object to be picked

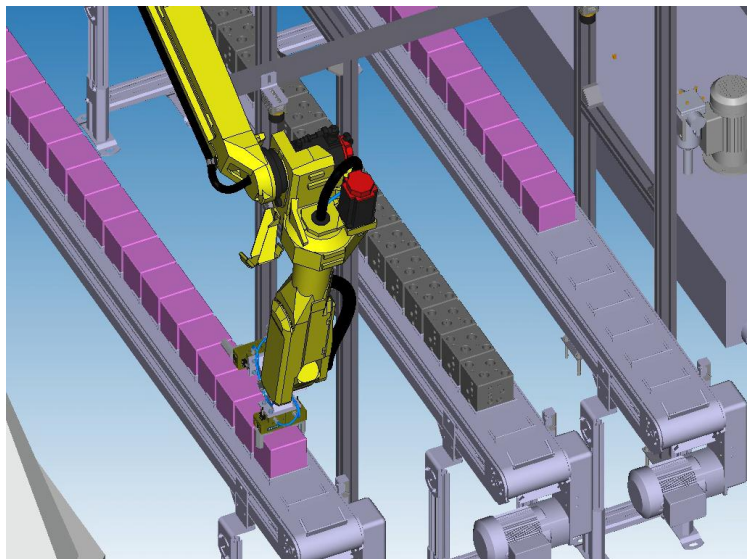


Fig. 16. Picking the object from the most loaded input conveyor

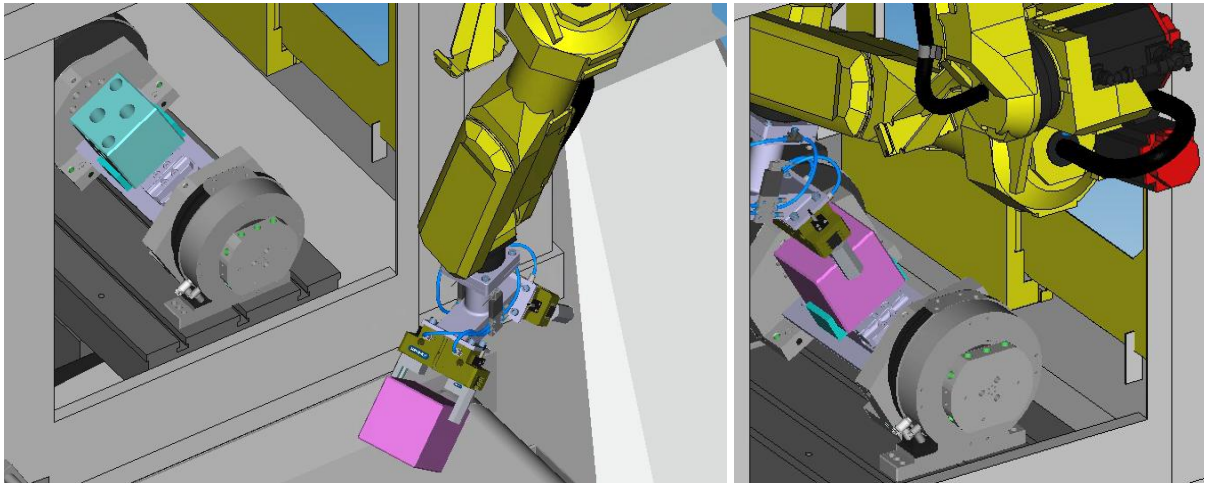


Fig. 17. Part exchange in the first machining center

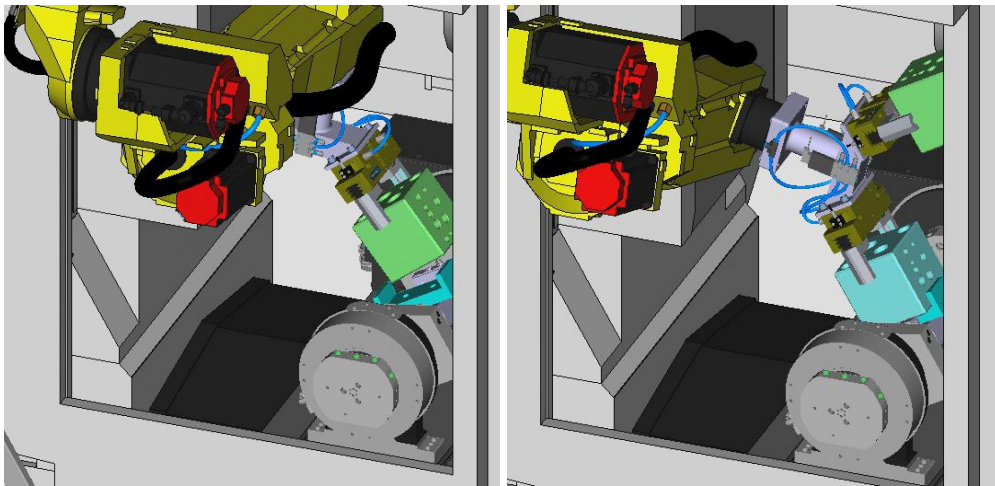


Fig. 18. Part exchange in the second machining center

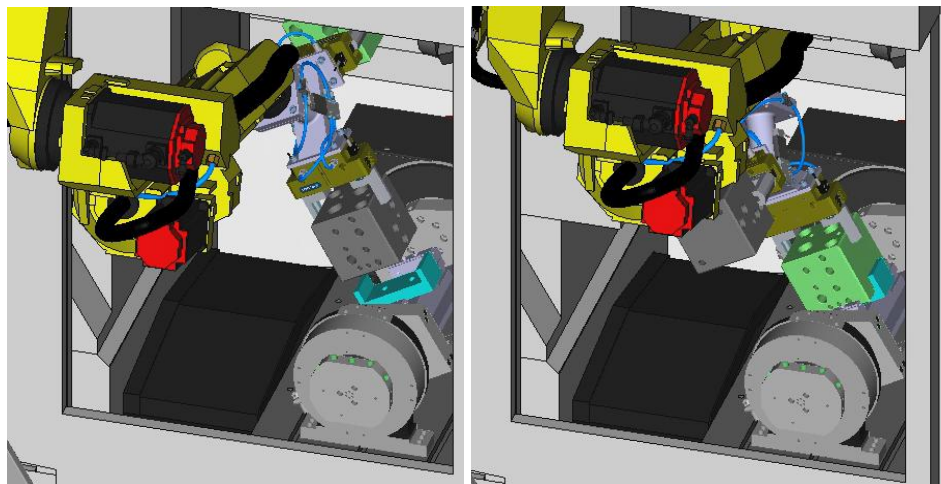


Fig. 19. Part exchange in the last machining center (rotated by 90°)

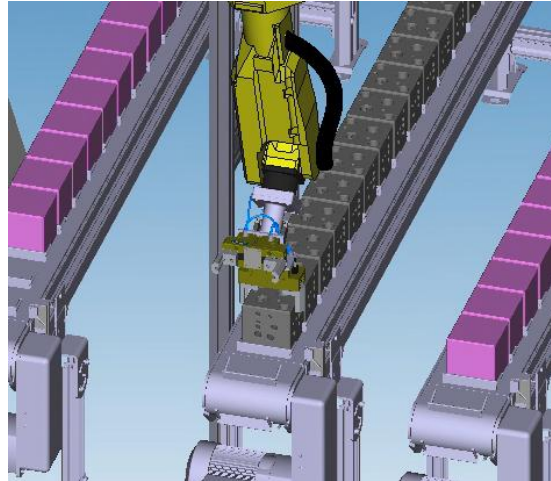


Fig. 20. Placing the finished object on the output conveyor

7. Conclusions

The paper aimed at the design and virtual validation of a robotic cell intended to automate handling and tending operations for CNC machining centers for prismatic mechanical components. By integrating a FANUC M20iA/20MT industrial robot with a hybrid kinematic structure, the aim was to develop a flexible solution able to serve several machine tools in a single robotic application, while optimizing occupied space and technological flow.

The project completed all stages required to develop a complete robotic application, from the analysis of existing solutions and the substantiation of the general concept, continuing with the selection of technological components and the 3D computer-aided design of the specific subsystems, to simulation and functional validation in the FANUC ROBOGUIDE environment. This approach enabled a complete virtual prototype to reproduce the functional behavior of the application before physical implementation.

An important project element was the design of the final end-effector, configured for simultaneous handling of the blank and the finished part, contributing to the reduced auxiliary times and to increased technological-cycle efficiency. Moreover, integrating auxiliary subsystems - conveyors, clamping systems, safety equipment, and systems for monitoring and dynamically determining the blank-picking position - enabled realistic modeling of a complete industrial application.

By generating robot kinematics and trajectories, robot access to all relevant operational points was shown feasible, as was efficient synchronization with peripheral equipment. Offline programming enabled develop and test control logic in a safe virtual environment, reducing the risk of errors during implementation.

The simulation confirmed the viability of the proposed solution, showing that the application can operate safely, without major kinematic conflicts or accessibility problems. Moreover, the use of virtual prototyping highlighted the advantages of offline simulation in layout optimization, reduced commissioning time, and greater integration-process predictability.

In conclusion, the project demonstrates that using an industrial robot with hybrid kinematics in an application for tending CNC centers is an efficient and modern solution for automating manufacturing processes. Future work may focus on extending the application by introducing multiple types of raw parts and dynamically calling both the CNC machining programs and the robot trajectories according to the specific object being processed. Another development direction may involve the integration of an auxiliary inspection module within the application, designed to perform quality inspection and validation of the finished parts before their transfer to the output flow.

7. References

- [1]. Siciliano, B., & Khatib, O. (2016), "Springer Handbook of Robotics", 2nd ed., Springer, Cham, Switzerland, ISBN 978-3319325507.
- [2]. Craig, J. J. (2018), "Introduction to Robotics: Mechanics and Control", 4th ed., Pearson Education, New York, USA, ISBN 978-0133489798.
- [3]. Mourtzis, D., Vlachou, E. & Milas, N. (2016), "Industrial Big Data as a Result of IoT Adoption in Manufacturing", *Procedia CIRP*, Vol. 55, pp. 290-295, ISSN 2212-8271.

- [4]. Tao, F., Zhang, H., Liu, A. & Nee, A. Y. C. (2019), “Digital Twin in Industry: State-of-the-Art”, IEEE Transactions on Industrial Informatics, Vol. 15, Issue 4, pp. 2405-2415, ISSN 1551-3203.
- [5]. Siemens Digital Industries Software (2024), “Tecnomatix Process Simulate Documentation”, Siemens Digital Industries Software, Plano, TX, USA.
- [6]. Groover, M. P. (2020), “Automation, Production Systems, and Computer-Integrated Manufacturing”, 5th ed., Pearson Education, Harlow, U.K., ISBN 978-1292253961.

Assisted design and offline programming-simulation of a robotic cell for palletizing food bags

MÎNTULEASA Eliza-Nicoleta and Ș.l.dr.eng. **ALEXANDRU Tudor-George**,
Prof.dr.eng. **NICOLESCU Adrian Florin**, RSP Department (Scientific Leader)

¹Faculty: IIR, Specialization: Robotics, Year of study: IV, e-mail: elizamintuleasa@gmail.com

***ABSTRACT:** This paper presents the computer-aided design and offline programming of an automated robotic cell for palletizing food product bags using Siemens Process Simulate. The core objective focuses on developing a high-efficiency gripper tailored for handling deformable loads and optimizing the workcell's layout. Kinematic workflows and trajectory paths were generated to ensure collision-free operations and efficient cycle times. Additionally, transient dynamic analysis was integrated to verify the mechanical safety of the auxiliary centering components. The simulation results confirm that the developed profiles and mechanical structures function safely within nominal limits, achieving reliable material flow without requiring subsequent geometric corrections.*

***KEYWORDS:** Process Simulate, End-Effector Design, Robotic Palletizing, Bag Handling, Digital Factory.*

1. Introduction

The modernization of production lines in the food industry requires the implementation of advanced solutions for the automation of packaging logistics processes. Robotic cells for palletizing bags play a critical role in ensuring high productivity, the main challenge being the kinematic management of the cell components to ensure stable placement of stacks on pallets.

The aim of this work is the integrated development, offline programming and functional validation of a robotic palletizing cell. The methodology used combines the constructive design of a gripper adapted to the handling of deformable loads, the configuration of the spatial layout and the complete digital simulation of the trajectories in the Siemens Process Simulate environment. The three-dimensional evaluation demonstrates the viability of the designed assembly, confirming its operation at nominal parameters without the need for structural modifications.

2. Current status

The efficiency of mechanical stacking lines by implementing automated systems is a priority direction in modern industrial logistics. In the specialized literature [1], a high-productivity robotic palletizing cell is analyzed that integrates a four-axis industrial robot ABB IRB 660, specialized in the handling and palletizing of cardboard boxes containing bulk food products. This type of configuration highlights the importance of the kinematic correlation of the arm with the rigid placement recipes to ensure the stability of the stack. In contrast, the study presented in [2] evaluates a flexible robotic cell focused on the dynamic handling of deformable loads, where the design of the end effector plays the main role in the safe handling of products without damaging the flexible packaging. The present work extends this context by directly correlating the constructive parameters of a gripper dedicated to bags with the general architecture of the palletizing cell.

3. Design and construction structure of the end effector (End-Effector)

The final effector (gripper) represents the direct connection component between the ABB IRB 6740 industrial robot and the handled loads, having the main role of taking the food bags from the infeed conveyor and positioning them on the pallet. The mechanical design of this subassembly was guided by the need to ensure a secure grip, capable of maintaining the geometric stability of the flexible bag and preventing its displacement under the action of inertial forces generated during rapid movements

of the robot arm.

From a structural point of view, the effector was designed in the form of a gripper. On its frame, made of cold-formed sheet metal, the lateral guiding and clamping jaws are integrated. These jaws have a concave geometry, specially adapted to the convex profile of bags loaded with bulk products, ensuring a uniform distribution of pressure on the surface of the package and preventing its breakage. The general appearance of the three-dimensional model optimized for the palletizing application is illustrated in figure 1.

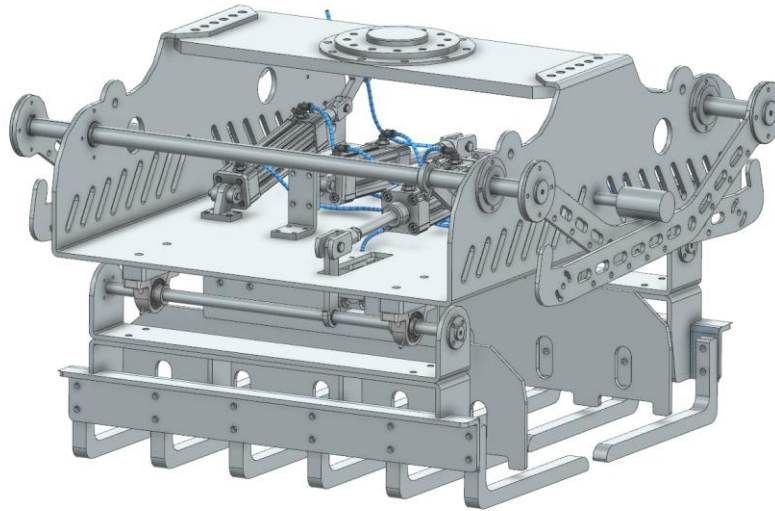


Fig. 1 Final design of the effector

The jaw drive mechanism has been designed to achieve a synchronous closing and opening movement. Synchronization is essential to ensure that the bag is picked up exactly from the central axis of the conveyor and released onto the pallet without lateral positioning deviations. To understand the internal topology of the gripping system, Figure 2 shows a detailed bottom view of the effector to illustrate the active components and guidance systems in the internal structure of the effector.

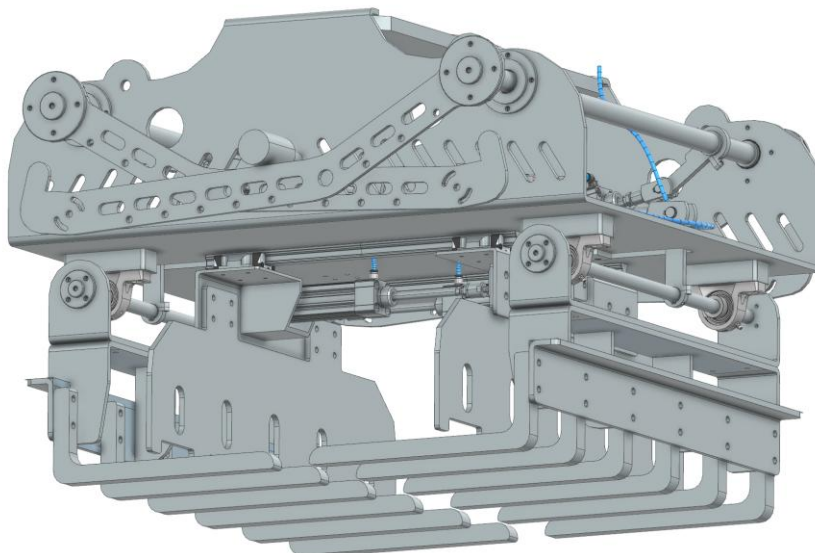


Fig. 2 Bottom view of the effector

The drive system integrated into the geometry includes a centrally positioned pneumatic cylinder to keep the gripper's center of gravity as close as possible to the robot flange axis. The transmission of movement to the jaws is achieved through a mechanism with articulated levers, specially designed to

transform the rotational movement of the shaft into precise translation of the gripping elements. The linear guides used ensure high torsional rigidity at the time of clamping, and the jaw stroke has been dimensioned so that the gripper can handle both bag types and sizes provided for in the design theme, providing flexibility to the entire palletizing cell without requiring the physical change of the effector during production.

4. Integration and configuration of the robotic cell

The spatial configuration of the entire palletizing cell was designed with the main objective of optimizing the placement of components to minimize downtime in the work cycle and to ensure a continuous flow of products. The overall architecture of the cell integrates the industrial robot in a central position, flanked by bag feeding belts and pallet stacking areas.

To ensure full coverage of the ABB IRB 6740 robot over all work points, it was necessary to place it on a rigid overhang structure. This metal frame stiffens the base of the robot and raises its kinematic center, giving the articulated arm the geometric freedom to pick up the bags from a lower height and deposit them up to the maximum height of the final stack without reaching the travel limits of the joints. The three-dimensional placement of the elements within the cell is illustrated in figure 3.

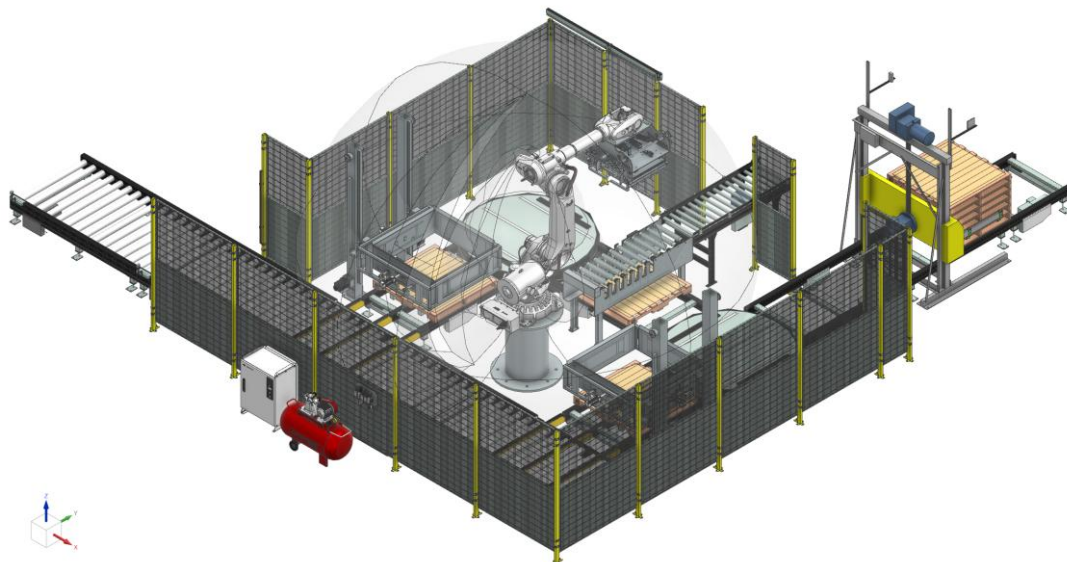


Fig. 3 Robotic Palletizing Cell

A key element in the cell logic is the adjacent pallet guidance and positioning system located on the roller conveyor line. Since the wooden pallets arrive from the automated warehouse with small lateral positioning deviations, the introduction of an external mechanism with pneumatically operated flexible jaws is critical. This system performs forced centering of the pallet on the conveyor axis just before the robot initiates the stacking phase, guaranteeing that each row of bags is placed exactly at the nominal height calculated in the offline program.

The arrangement of the perimeter equipment complies with industrial safety regulations, with a clearly delimited space provided by protective fences with optoelectronic barriers at the complete pallets' evacuation areas. This compact arrangement ensures the fully automated operation of the logistics flow: the bag is brought onto the feed belt, taken over by the specialized gripper, oriented in space by the robot and deposited on the pallet already centered by the pneumatic mechanism, achieving a fluid and efficient production cycle.

5. Simulation, offline programming and validation of the dynamic behavior of the technological flow

The functional validation of the entire robotic cell required the use of advanced offline programming techniques and three-dimensional digital simulation to faithfully reproduce the real industrial technological cycle. The configuration of the kinematic interactions between the components, the definition of the logical sequences and the generation of the robot trajectories were carried out in the Siemens Process Simulate development environment. The main objective of the simulation was to optimize the movements during the successive phases of bag pickup, orientation and deposit, ensuring the fluid operation of the line, avoiding kinematic singularities and eliminating any risk of collision between the gripper and the perimeter elements.

The programming algorithm developed in the digital environment defined optimized trajectory points to reduce sudden accelerations at the ends of the stroke and to streamline the total cycle times. The robot takes the bag from the feeding conveyor area by activating the end effector-specific signals, adjusts its spatial orientation during the movement and positions it according to the palletizing recipe. The kinematic trajectory generated in Process Simulate and the spatial location points of the manipulated objects are illustrated in figure 4.

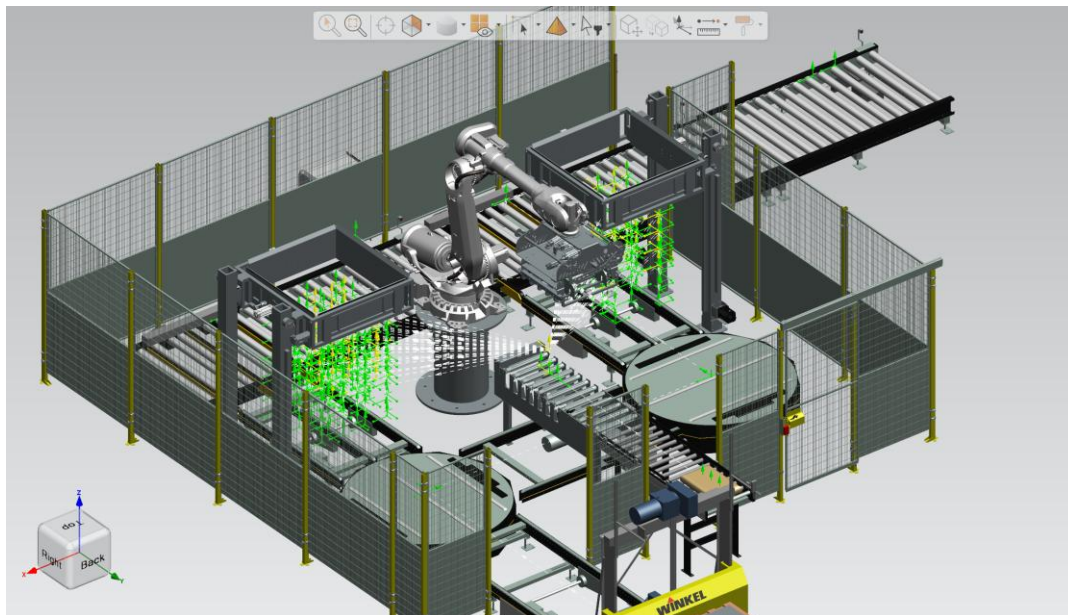


Fig. 4 Trajectories made in Process Simulate

The stacking process is carried out in a fully automated logical sequence. Since food bags are highly flexible, the offline programming has provided release points calculated so that the stacking ensures a stable stacking of the rows. The sequence of steps within the palletizing cycle, highlighting the progressive generation of layers and the management of the task flow in Process Simulate, is shown in Figure 5.

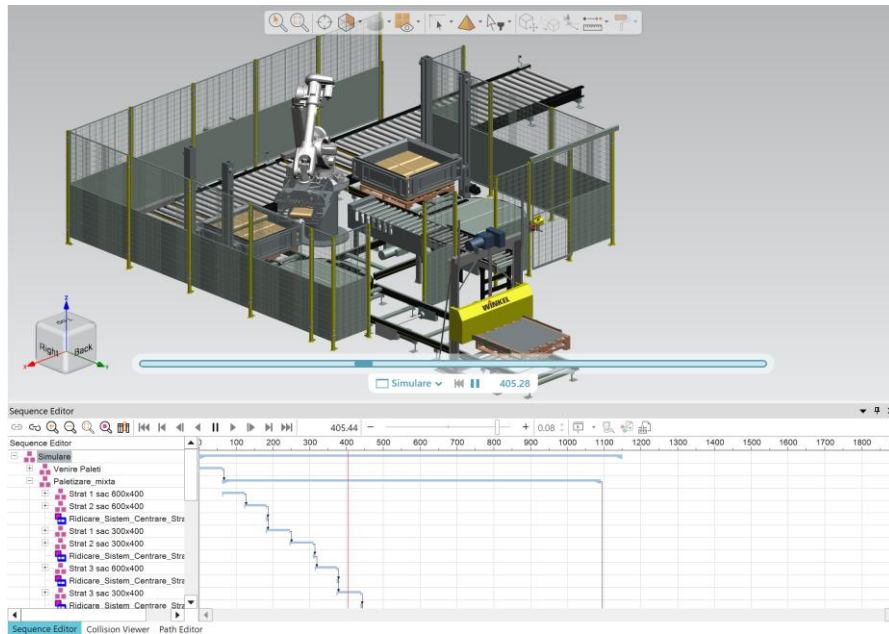


Fig. 5 Successive steps in creating a stack of bags through offline simulation

6. Conclusions

This work demonstrated the efficiency of using computer-aided design techniques and digital simulation tools for the development of an industrial bag palletizing cell. The proposed mechanical architecture for the specialized gripper ensures stable load handling, the concave geometry of the jaws providing optimal pressure distribution without deforming the flexible packaging, while the lightweight structure maintains a low dead weight.

The kinematic validation carried out in the Siemens Process Simulate environment confirms the fluidity of the technological flow. The rigorous definition of the operative trajectories led to the elimination of collision points during the stack formation.

7. Bibliography

- [1]. Craig, J. J. (2018), *"Introduction to Robotics: Mechanics and Control"*, 4th ed., Pearson Education, New York, USA, ISBN 978-0133489798.
- [2]. Siciliano, B., & Khatib, O. (2016), *"Springer Handbook of Robotics"*, 2nd ed., Springer, Cham, Switzerland, ISBN 978-3319325507.
- [3]. Groover, M. P. (2020), *"Automation, Production Systems, and Computer-Integrated Manufacturing"*, 5th ed., Pearson Education, Harlow, U.K., ISBN 978-1292253961.
- [4]. Siemens Digital Industries Software (2024), *"Tecnomatix Process Simulate Documentation"*, Siemens Digital Industries Software, Plano, TX, USA.
- [5]. Monkman, G. J., Hesse, S., Steinmann, R., & Schunk, H. (2007), *"Robot Grippers"*, Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, ISBN 978-3527406197.

[6]. Bi, Z., & Wang, L. (2020), “*Advances in 3D Digital Simulation and Virtual Commissioning for Manufacturing Systems*”, *International Journal of Advanced Manufacturing Technology*, 107(5–6), pp. 2425–2440, ISSN 0268-3768.

Computer ASSISTED DESIGN AND PROGRAMMING - OFFLINE SIMULATION OF A ROBOTIC PALLETIZING CELL FOR AGRICULTURAL FERTILIZER BAGS

Nagîț FABIAN-MIHAI, Alexandru TUDOR-GEORGE, Nicolescu FLORIN-ADRIAN

Faculty: IIR, Specialisation: Robotics, Year of study: IV, e-mail: fabiannagit@gmail.com

SUMMARY: This project focuses on the 3D computer-aided design and offline simulation of a flexible robotic cell dedicated to the palletizing of bulk fertilizer bags. The methodology involved the 3D synthesis of a custom hybrid fork-style end-effector and the modeling of non-standard peripheral subsystems, such as a riser base, a bag pusher, and customized conveyor alterations. The entire layout was subsequently integrated into Siemens Tecnomatix Process Simulate, where kinematic joints, virtual sensors, and offline programming (OLP) trajectories were defined. Through extensive virtual testing and automated clash detection, all potential spatial collisions and kinematic singularities were successfully identified and eliminated. The study concludes that this digital twin approach effectively validates the technological flow, confirming that specialized modifications—including parallelipedic rollers and mechanical stoppers—ensure a flawless, automated 90-degree alternating layer stacking sequence without physical prototyping risks.

KEY WORDS: robotic cell, palletizing, offline simulation, 3D modelling

1. Introduction

The project studied aims at the 3D assisted synthesis and functional simulation of a robotic cell intended for palletizing operations of fertilizer bags. The application involves the use of an articulated arm industrial robot, which has an attached "fork style" effector, which takes the bags from the conveyor and places them neatly on the pallets. The development stages consisted of substantiating the model based on a real application, 3D modeling of the subsystems missing from the CAD bases (socket, pusher, separator table) and assembling the virtual prototype. The system was then transferred to the Siemens Tecnomatix Process Simulate environment for defining the kinematic couplings and programming the work cycle. The integration required various design changes: replacing cylindrical rollers with paralleliped rollers to even out the bags, installing fixed stops to avoid collisions with the claws, adding pallet locking modules to the transfer unit, and designing a new clamping flange to adapt the effector to the ABB IRB 660 robot model used in the simulation

2. Stadiul actual

In the literature, CAD design and offline simulation (OLP) are the fundamental pillars in the development of robotic cells. The 3D virtual environment allows for the assembly of systems, programming of trajectories and validation of interactions before physical construction, drastically reducing costs and implementation times.

In the paper [1], the methodology for offline simulation of a cell for palletizing products with variable geometry (bags) is presented. The authors analyze the optimization of the workspace and conveyors to reduce cycle times. Similarly, the paper [2] describes the use of OLP for robots with complex effectors (fork style). The research focuses on the automatic detection of collisions (clash detection) between the gripping claws and the stops, completely eliminating hardware programming errors. Another approach, discussed in [3], analyzes the integration of automated logistics flows, highlighting the importance of temporal synchronization in offline simulation between conveyor speed and pneumatic system activation (suction cups for separators, locking modules).

By correlating these concepts, the present project uses modern geometric modeling

Computer ASSISTED DESIGN AND PROGRAMMING - OFFLINE SIMULATION OF A ROBOTIC PALLETIZING CELL FOR AGRICULTURAL FERTILIZER BAGS

and virtual programming techniques. This approach allows for the complete validation of the technological flow of orderly palletizing of fertilizer bags, ensuring a fluid operation, without risks of spatial collision or kinematic singularities.

3. The modeling and the offline simulation of the studied robotic cell

The design and validation of the virtual prototype of the palletizing cell involves a rigorous sequence of stages, starting from the three-dimensional modeling of the mechanical components to the programming of the operating logic in the virtual environment. The main steps taken to obtain the functional system were:

- Step 1. 3D-assisted synthesis of the hybrid effector

The first step consisted of the detailed modeling of the technological equipment of the industrial robot. A "fork style" effector (with claws) was designed for bag handling, integrating structural elements, bearings, rods, jaws and a bag pressing system. Also, to make the effector multifunctional, a vacuum system (rods with suction cups operated by pneumatic cylinders) necessary for handling cardboard separators was added. To adapt the effector to the flange of the ABB IRB 660 robot, available in the simulator, an intermediate flange was specially modeled.

- Step 2. non-standard components

Since certain components were not available in the standard CAD bases, they were modeled from scratch.

- Elevation base: Since the effector must be able to handle bags of different heights, a base with adjustable height was designed.

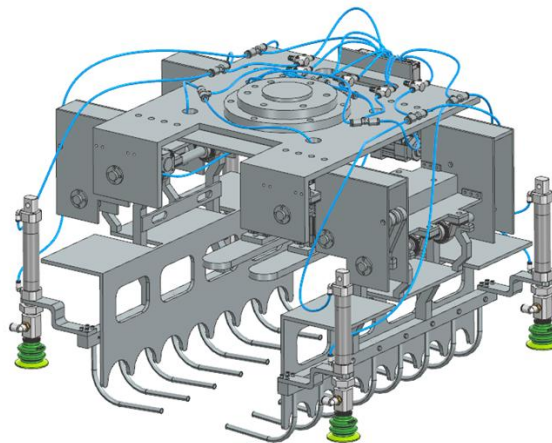
- Transfer conveyor: To ensure a uniform flow of bags, a conveyor with variable speed was designed to uniformize the flow of bags to the claws to penetrate them.

- Transfer unit: The transfer unit was designed to block the pallet and to move it to the next stage.

- Pusher and separator: The pusher and separator were designed to push the bags and to separate them from the pallet.

- Step 3. Creating the virtual prototype

After individual modeling, the complete assembly of the flexible manufacturing cell layout followed. The virtual prototype included the robot, infeed and outfeed conveyors, transfer unit, WINKEL pallet dispenser, Phoenix wrapper, sensors and protective fences. The placement of the equipment strictly followed the configuration of the real application identified in the foundation phase.



al CAD bases, they were

per layers of the pallet.

parallelepiped rollers (to

led to allow the effector

ling pneumatic modules

ing the bags and a table

re designed.

- Step 4

1

Simula

Pallet

Classifi

dispencer

• The

Table for

Separators

F

D

Controller IRC

5

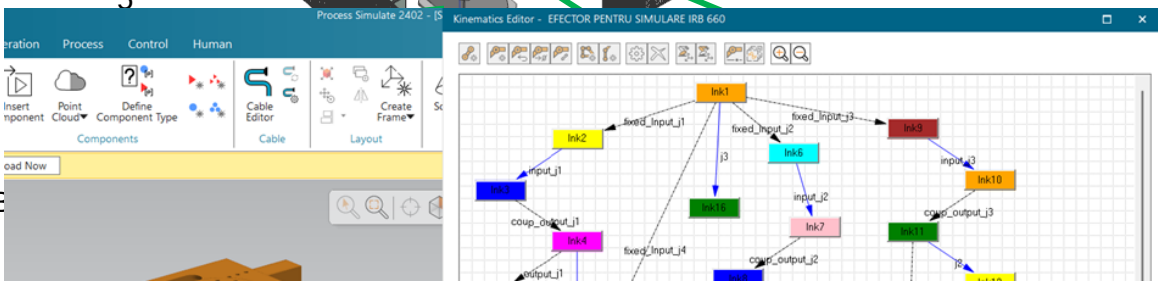
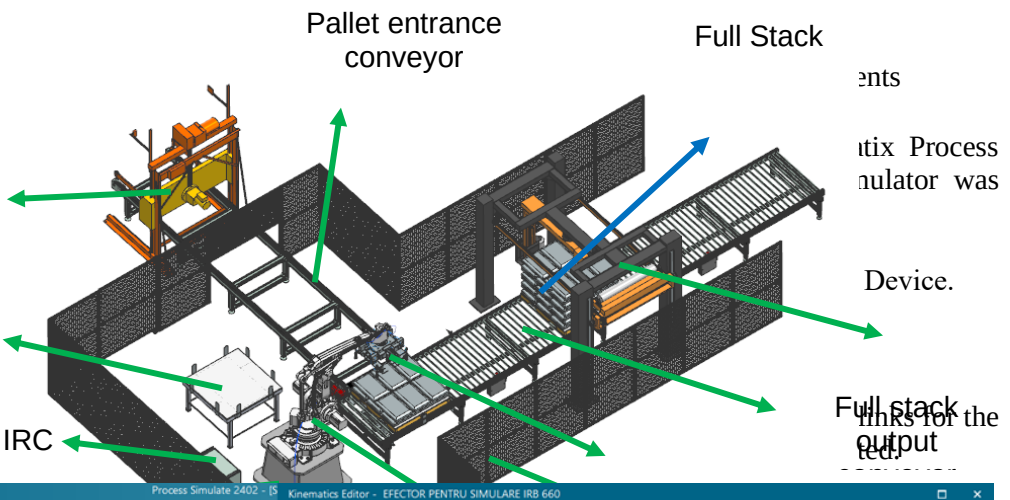


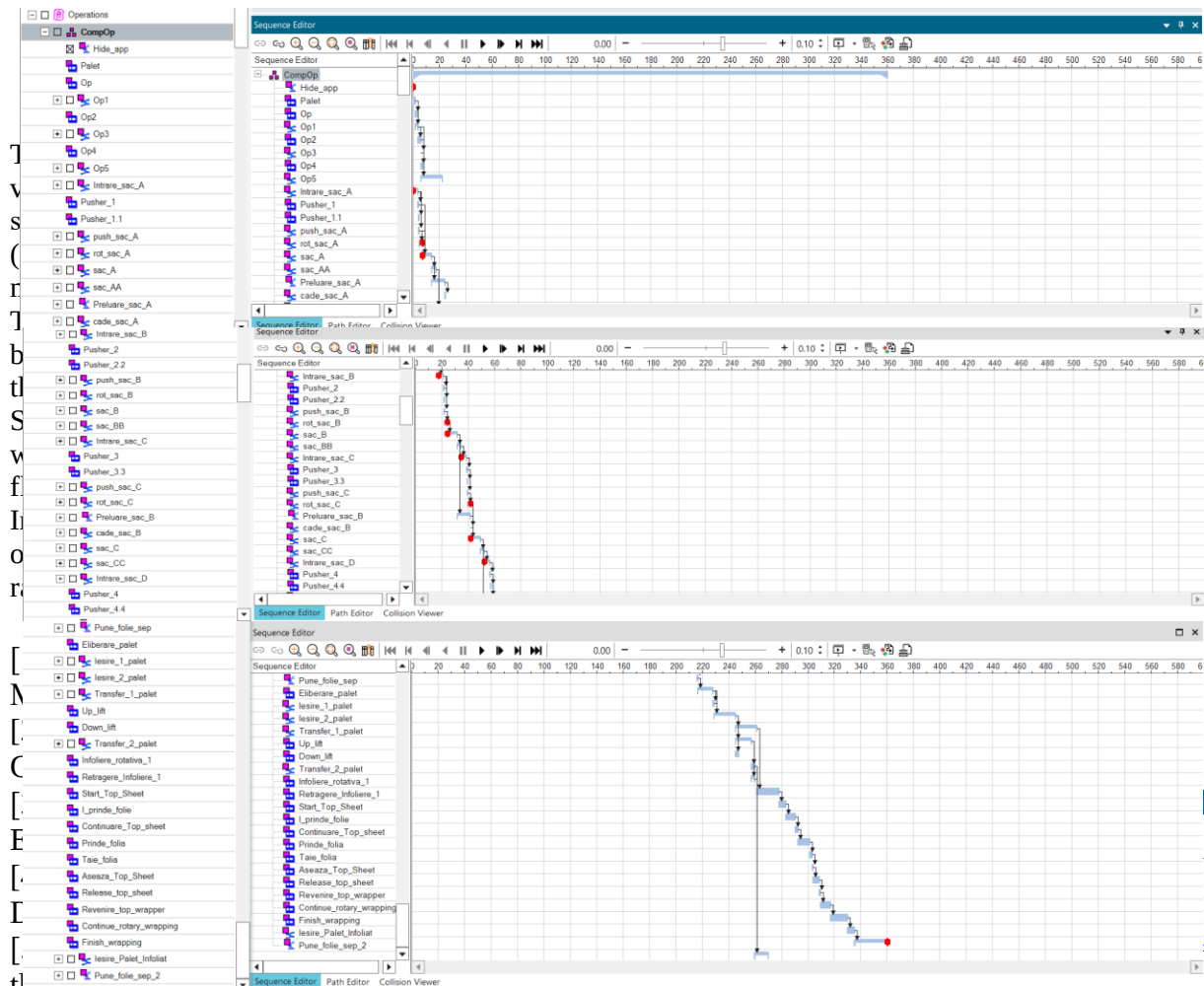
Fig. 3. Defining effector kinematic torques in Process Simulate

- Offline programming: The start and end positions (Home/End) were set and the robot trajectories were generated point by point (including alternating the placement of bags at 90 degrees on different rows).

Step 6. Synchronizing processes using sensors

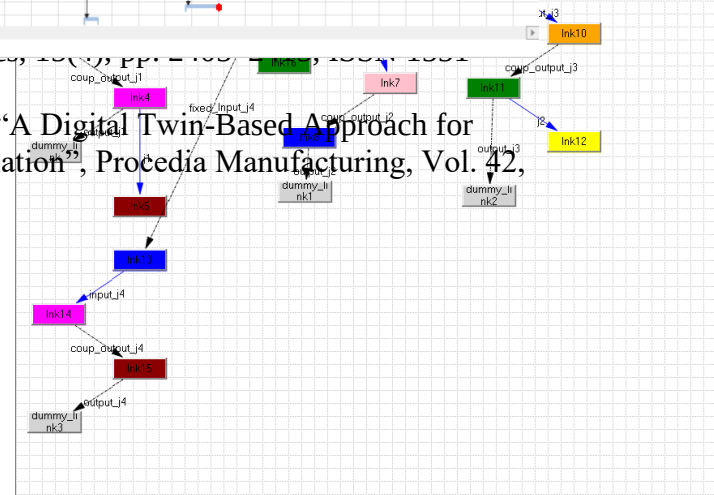
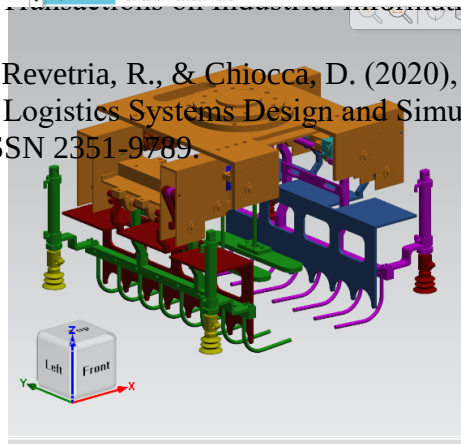
For a continuous and realistic simulation of the technological flow, the operating logic was conditioned by the triggering of virtual sensors:

- Capacitive sensors were used to detect the passage of bags/pallets, triggering the stop of the belts or the start of the robotic arm.
- Travel limiters (mechanical sensors/buttons) ensured the stop of the pallets at a fixed point or controlled the mechanisms for lowering the transfer unit from the chain flow to the roller flow.



3203.

[6]. Guizzi, G., Revetria, R., & Chiocca, D. (2020), "A Digital Twin-Based Approach for Production and Logistics Systems Design and Simulation", *Procedia Manufacturing*, Vol. 42, pp. 367–374, ISSN 2351-9789.



COMPUTER-AIDED DESIGN AND PROGRAMMING - OFFLINE SIMULATION OF A ROBOTIC CELL FOR PALLETIZING FOOD PRODUCT BOXES

ȘELARU Carmen Cristina Ioana¹ and TUDOR Alexandru (Scientific Advisor)

¹Faculty: IIR, Specialization: Robotics, Year of study: IV, Group: 641BB, e-mail:
selaru.carmen.ioana@gmail.com

ABSTRACT: This paper presents the synthesis and functional simulation of an industrial robotic cell dedicated to palletizing cardboard boxes containing granular products. The work starts from the analysis of reference palletizing cells and continues with the selection of the industrial robot, the design of the technological equipment, and the integration of conveyors, safety devices, and auxiliary subsystems. The proposed solution uses an ABB IRB 660-180/3.15 palletizing robot equipped with a multifunctional clamp-style end-effector capable of handling product boxes, cardboard separators, and pallets. The 3D model includes the robot support, the pneumatic gripper, the pallet and separator magazines, the conveyor network, the wrapping area, and the perimeter safety system. The functional simulation validates the main motions of the robot, the end-effector, the positioning mechanism, the transfer units, and the wrapping machine. The study confirms that the proposed layout can ensure a coherent, safe, and productive palletizing flow.

KEYWORDS: palletizing; robotic cell; ABB IRB 660; clamp-style end-effector; 3D simulation.

1. Introduction

Palletizing is the operation of arranging products in an ordered manner on pallets, in successive layers, so that the resulting load can be handled, stored, and transported safely. In modern industrial applications, this operation is increasingly performed by industrial palletizing robots because they provide high flexibility, repeatability, rapid adaptation to different products, and reduced operator intervention in repetitive or physically demanding work areas.

The aim of this paper is to substantiate, model in 3D, and simulate the operation of a robotic cell for palletizing boxes containing granular products. The designed application includes a box infeed area, two stack-forming areas, a transfer system for loaded pallets, a wrapping area, and an outfeed area. In addition, the cell integrates pallet and cardboard separator magazines, a perimeter safety system, sensors for flow control, and a control station for operation and maintenance.



Fig. 1. Reference model for the robotic palletizing application

The project topic starts from a real palletizing system in which the robot handles parallelepiped boxes, separators, and pallets. Based on this reference model, the proposed solution was adapted for a cell with a single product infeed and two stack-forming positions, with final discharge toward a wrapping unit. The concept aims to obtain a continuous technological flow, with reduced idle times and repeatable product positioning.

2. Current State of the Art

Several cell architectures are used in robotic palletizing applications, ranging from compact mini pallet cells, intended for production lines with limited space, to systems with multiple stack-forming stations and complex conveyor circuits. Compact cells generally include a centrally located industrial robot, an infeed conveyor, a palletizing station, pallet and separator magazines, and safety elements. These systems are efficient when the product range is relatively stable and when end-of-line automation is required.

For heavier loads and fast cycles, dedicated palletizing robots are used, with an articulated-arm architecture and a closed kinematic chain. They generally have four numerically controlled axes and an internal mechanism that maintains the end-effector orientation during motion. Compared with universal six-axis robots, palletizing robots sacrifice part of their orientation freedom, but gain speed, rigidity, and payload capacity. For this reason, they are suitable for repetitive handling of boxes, bags, crates, or prepacked products.

		Table 1. Common pallet types used in robotic applications
Pallet type	Common dimensions	Role in palletizing applications
EUR 1	800 x 1200 mm	standard European pallet, frequently used in logistics
EUR 2	1200 x 1000 mm	variant for larger loads and supporting surfaces
UK / industrial	1200 x 1100 mm	solution adapted for bulky stacks and dedicated flows

The end-effectors used in palletizing are selected according to the shape and rigidity of the product. For cardboard boxes, vacuum end-effectors or jaw-type end-effectors are frequently used, while for bags or deformable products claw systems or hybrid systems are applied. In the present application, the choice of a clamp-style end-effector is justified by its ability to grip boxes laterally and by the possibility of extending it to handle separators and pallets, thus becoming a multifunctional technological device.

3. Design Rationale of the Robotic Application

The designed cell model is based on an application for palletizing powdered sugar boxes, in which the products arrive on the infeed conveyor, are detected by sensors, and are then picked by the robot. Each package is composed of grouped cardboard boxes, and the total mass of the manipulated product requires the use of a robot with a high payload and a reach sufficient to cover both the pickup area and the two pallet deposition areas.

The robot selected for integration is the ABB IRB 660-180/3.15, an industrial palletizing robot with four numerically controlled axes, a payload of 180 kg, and a reach of 3.15 m. The robot is mounted on a raised support in order to increase accessibility to the conveyors, magazines, and stack-forming positions. This choice resulted from comparing several dedicated palletizing robots, taking into account payload, repeatability, self-weight, axis speed, and integration within the available space.



Fig. 2. ABB IRB 660 industrial robot used for palletizing

Table 2. Synthetic comparison of the analyzed palletizing robots

Industrial robot	NC axes	Payload	Reach / range	Repeatability
Yaskawa MPL160II	4	160 kg	3159 mm	±0.05 mm
FANUC M-410iC/185	4	185 kg	3143 mm	±0.05 mm
Kawasaki CP300L	4	250 kg	3255 mm	±0.5 mm
ABB IRB 660-180/3.15	4	180 kg	3150 mm	±0.05 mm

With respect to the application requirements, the ABB IRB 660 offers a favorable compromise between payload capacity, reach, and working speed. The 180 kg payload provides a sufficient reserve for the mass of the boxes, the mass of the end-effector, and possible product variations. The working reach allows access to the two stack-forming areas, while the high repeatability contributes to maintaining a constant geometry of the product layers.

4. Computer-Aided 3D Synthesis of the Technological Equipment

The robot tooling is represented by a clamp-style end-effector designed to handle cardboard boxes, separators, and pallets. The assembly includes lateral gripping elements, pneumatic actuation mechanisms, suction cups for separators, and oscillating arms for pallet pickup. Through this configuration, the robot can perform several successive operations without changing the end-effector, which reduces cycle time and simplifies the architecture of the cell.

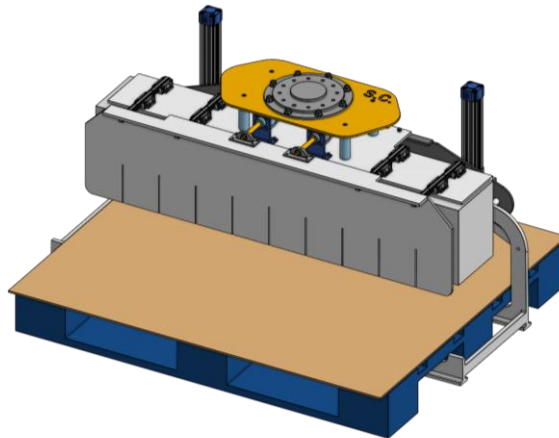


Fig. 3. 3D model of the multifunctional clamp-style end-effector

The end-effector mechanisms are actuated pneumatically. The 5/2 directional control valves, vacuum ejectors, suction cups, fittings, and linear pneumatic actuators generate the gripping, supporting, and release motions. For cardboard separators, the vacuum system ensures distributed and controlled gripping, while for pallets, pallet-support elements are used and can oscillate between the rest position, the pickup position, and the deposition position on the conveyor.

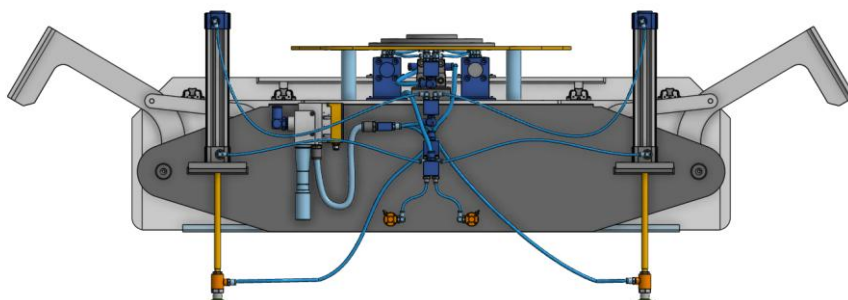


Fig. 4. Integration of pneumatic mechanisms on the end-effector

The 3D design also included components for which no manufacturer CAD databases were available. These include the box positioning mechanism on the infeed conveyor, the raised robot support, the guides for the pallet and separator magazines, and the pallet guiding parts on the stack-forming conveyors. These components ensure repeatable product positioning and eliminate deviations that could lead to pickup errors or unstable stacks.

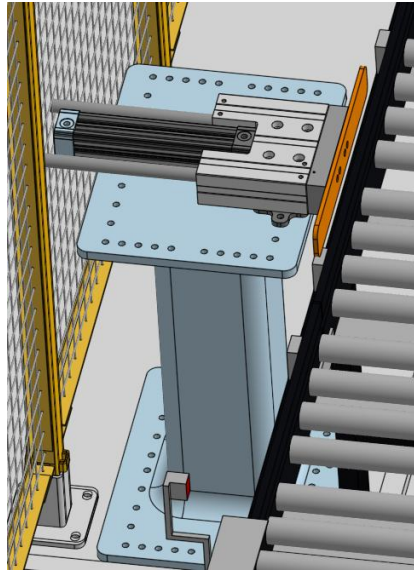


Fig. 5. Box positioning mechanism on the infeed conveyor

5. Integration of Subsystems into the Robotic Cell

The general cell assembly includes the robot and its technological equipment, product infeed conveyors, stack-forming conveyors, the transfer conveyor, lifting units, the rotary-arm wrapping machine, the control area, and the perimeter guards. Roller conveyors with their own drive are used for box infeed, while wider conveyors, adapted to pallet dimensions and transported load, are used for pallet handling.

The functional flow can be summarized as follows: boxes enter the cell on the infeed conveyor, are aligned and stopped in the pickup area, and are then picked by the robot with the end-effector and deposited alternately in one of the two forming areas. After each layer is completed, the robot can add a cardboard separator. When the stack is complete, the pallet is transferred to the wrapping area, where the load is stabilized for transport. After wrapping, the pallet is discharged toward the outfeed area.

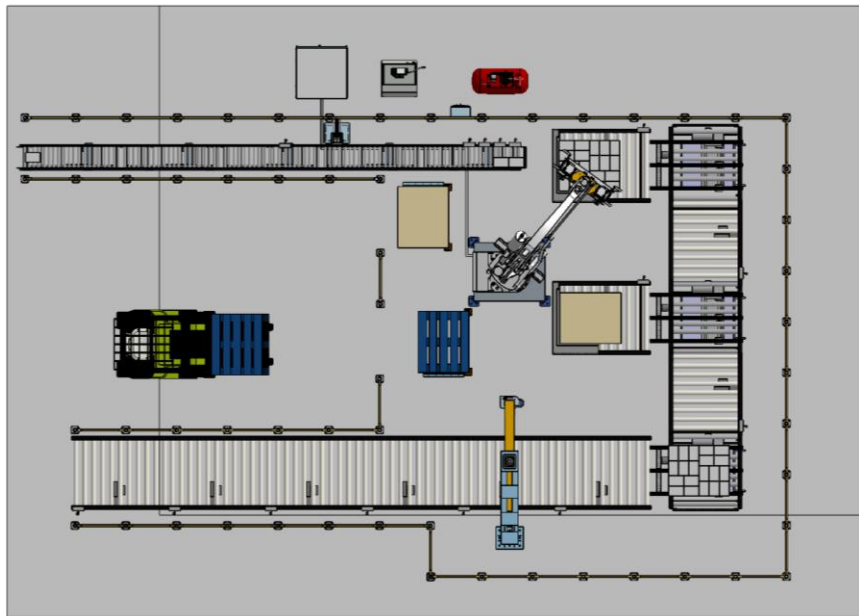


Fig. 6. Virtual prototype of the robotic palletizing cell

To ensure safe operation, the cell is delimited by protective fences and light curtains. The robot controller, control panel, teach pendant, compressor, air preparation unit, and automation cabinet are located in the control area. This separation allows the operator to monitor and adjust process parameters without entering the robot work area during automatic operation.

Table 3. Main subsystems integrated into the robotic application

Subsystem	Functional role in the cell
Infeed conveyor	transports boxes to the pickup area and enables their detection by sensors
Robot + end-effector	picks boxes, separators, and pallets and performs stack formation
Pallet/separator magazines	ensure the periodic supply of the auxiliary elements required for the cycle
Forming and transfer conveyors	support the stacks and move them toward the wrapping area
Fences and light curtains	limit access to the hazardous area and ensure stopping in case of intrusion
Control station	enables control, diagnostics, adjustment, and manual or automatic operation

6. Functional Simulation of the Application

The functional simulation was carried out to verify the main motions of the robot, the end-effector, and the auxiliary subsystems. Kinematic constraints were defined in the working environment for the axes of the industrial robot, for the gripping mechanisms of the end-effector, for the box positioning mechanism, for the lifting units, and for the wrapping machine. The purpose of the simulation was not only to present motion, but also to verify the operating sequence logic.

COMPUTER-AIDED DESIGN AND PROGRAMMING - OFFLINE SIMULATION OF A ROBOTIC CELL FOR PALLETIZING FOOD PRODUCT BOXES

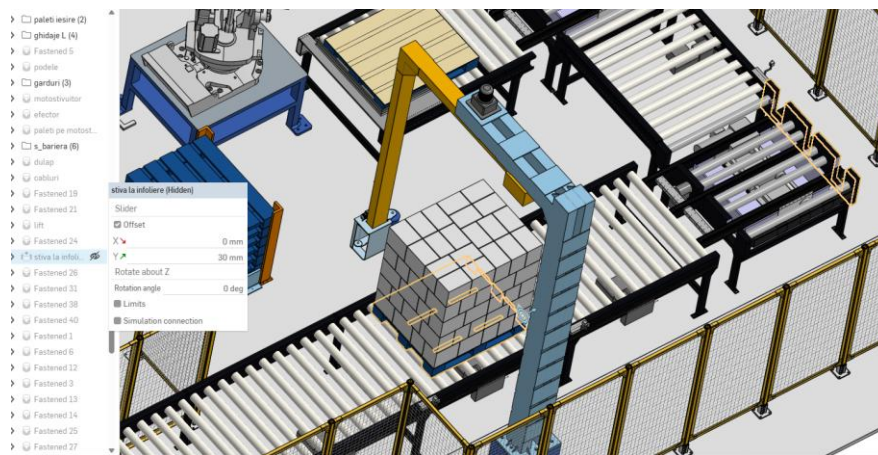


Fig. 7. Wrapping and complete-stack transfer area

During the simulation stage, the end-effector was verified in the zero position, in the pallet-handling position, in the separator pickup position, and in the box-gripping position. Areas with a risk of interference between the robot, conveyors, guards, stacks, and auxiliary equipment were monitored. The transfer motions of the complete stacks and the integration with the wrapping area were also analyzed so that the loaded pallet could follow the flow without unjustified stops.

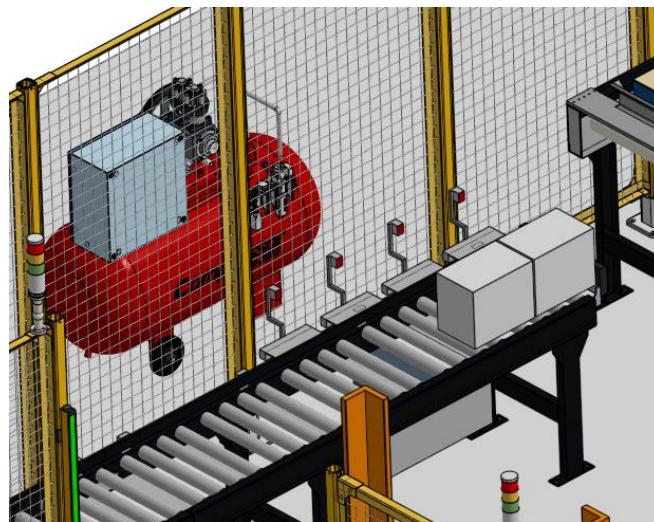


Fig. 8. Manipulated-object pickup area

The simulation results confirmed the coherence of the proposed architecture. The robot can access the critical positions of the application, the end-effector can execute the pickup and deposition sequence, and the conveyors provide the link between the main areas of the cell. For practical implementation, the virtual model must be completed by verifying cycle times, performing a detailed sensor analysis, and validating the automation logic in relation to the real states of the equipment.

7. Conclusions

The paper synthesized the concept of a robotic cell for palletizing operations, starting from the analysis of existing industrial applications and from the requirements of a cell intended for handling boxes containing granular products. Dedicated industrial palletizing robot solutions were compared, and the ABB IRB 660-180/3.15 variant was considered suitable due to its payload, reach, repeatability, and compatibility with the space requirements of the application.

The main contribution of the project is the integration of a multifunctional end-effector capable of

handling not only the palletized products, but also the cardboard separators and the pallets. This approach reduces the need for auxiliary devices and enables a more complete automation of the cycle. The 3D modeling included both manufacturer-provided equipment and components designed specifically for the application: supports, guides, positioning mechanisms, and adaptation elements.

The functional simulation validated the essential motions of the robot and of the cell subsystems. The proposed configuration provides a logical flow from box infeed to wrapped-pallet outfeed, with clear separation of the work, control, and safety areas. Before real implementation, cycle-time optimization, verification of pneumatic loads, sensor reliability analysis, and finalization of the control program for the entire cell are recommended.

8. References

- [1]. Nicolescu, A. (2025), Implementation of Industrial Robots in Production Systems, course notes and design methodologies, UPB, Bucharest.
- [2]. Nicolescu, A. F. and Cristoiu, C. A. (2020), Implementation of Industrial Robots in Production Systems. Laboratory and Project Guide, Politehnica Press Publishing House, Bucharest, ISBN 978-606-515-915-0.
- [3]. Bucureşteanu, A. (2013), Pneumatic Elements and Systems for Actuating Industrial Robots, Printech Publishing House, Bucharest, ISBN 978-606-23-0081-4.
- [4]. Cristoiu, C. A. and Nicolescu, A. F. (2020), NX CAD-Basic - Computer-Aided Design, Politehnica Press Publishing House, Bucharest, ISBN 978-606-515-914-3.
- [5]. Nicolescu, A. F. (2005), Industrial Robots, Didactic and Pedagogical Publishing House RA, Bucharest, ISBN 973-30-1244-0.
- [6]. *** ABB Robotics, IRB 660 Palletizing Robot, manufacturer technical documentation; ESS Technologies, Robotic Mini Pallet Cell, system presentation documentation.

Computer-Aided Design and Offline Programming–Simulation of a Robotic Palletizing Cell

SÎRBULESCU Maria Denisa¹ and Prof. dr. Ing. NICOLESCU Adrian Florin

¹ Faculty of Industrial Engineering and Robotics, Specialization Robotics, Year of study IV,
e-mail: denisa.srbulescu@yahoo.com

REZUMAT: This project presents the design and offline simulation of a robotic palletizing cell for refrigerated product boxes using an ABB IRB 660 industrial robot. The objective was to develop and validate an automated palletizing system in a virtual environment, based on an existing concept extracted from a reference video and adapted to the proposed application. The methodology involved CAD import, 3D modeling of the complete cell, and integration of conveyors, robot, end-effector, and auxiliary systems in ABB RobotStudio. The offline simulation allowed verification of correct coordination between components and proper execution of the palletizing cycle. Minor adjustments were required to optimize trajectories and signal logic, ensuring smooth operation without conflicts. Overall, the results confirm that the virtual cell functions correctly and reproduces the intended palletizing process.

CUVINTE CHEIE: robot paletizare, ABB RobotStudio, simulare offline, celula robotizata, end-effector.

1. Introduction

Palletizing is the operation of arranging various categories of products in an organized three-dimensional pattern on standardized load carriers known as pallets. The arrangement is performed both horizontally, in the form of layers of uniform height, and vertically, through the stacking of multiple layers. The handled products may include cardboard boxes with a prismatic shape, bags containing bulk materials such as granules or powders, as well as shrink-wrapped multipacks of products (e.g., bottles of water or cooking oil). These items can be manipulated individually or in groups by industrial robots or automated palletizing systems. As one of the most widespread industrial robot applications, palletizing is characterized by a wide variety of products with different shapes, dimensions, and weights, as well as numerous pallet stacking configurations, which significantly increase the complexity of the process. Furthermore, the integration of industrial robots into palletizing operations leads to increased productivity, improved process consistency, and reduced reliance on human operators. [1]

In this context, the project aims to develop a robotic cell dedicated to the palletizing of boxes containing refrigerated products, using an articulated industrial robot and peripheral equipment specific to automated handling applications. The palletizing cell was developed based on an existing model presented in a reference video, which constituted the starting point of the general concept. On this basis, the stages of 3D modeling of the conveying system, industrial robot, end-effector, and auxiliary equipment were carried out, followed by offline programming and simulation in ABB RobotStudio. The purpose of these stages was to verify the correct operation of the system and optimize the work sequences.

Throughout the development of the project, the general structure of the cell, the conveying system, and the palletizing solution were maintained according to the reference model, with only the adjustments and optimizations necessary for integration and simulation in the virtual working environment being implemented.



Fig. 2. The robotic palletizing cell in the research video

2. Current state

At present, robotic palletizing applications represent an important field of industrial automation, being widely used in the food, logistics, and pharmaceutical industries. The development of simulation and offline programming has increased design efficiency, allowing the testing and optimization of robotic cells before implementation.

In the scientific literature, there are numerous studies regarding the simulation and offline programming of industrial robots. An important role is played by software environments such as ABB RobotStudio, which allow realistic simulations through the Virtual Controller, similar to the real controller, facilitating testing and optimization without interrupting production. A relevant study is “AdaptPack Studio: An Automated Intelligent Framework for Offline Factory Programming”, which highlights the advantages of offline programming in reducing development time and optimizing trajectories, including automatic program generation and collision avoidance. Another paper, “AdaptPack Studio Translator: Translating Offline Programming to Real Palletizing Robots”, analyzes the transfer of offline programs to real industrial robots, showing a significant reduction in implementation time for palletizing applications.

The importance of CAD model integration is highlighted in “Offline Programming of an ABB Robot Using Imported CAD Models in the RobotStudio Software Environment”, where the advantages of using 3D models for trajectory definition and collision detection before implementation are emphasized. Furthermore, the paper “Research on Simulation Design of Robotic Handling and Stacking of Various Shaped Materials” presents the use of ABB RobotStudio for the modeling and simulation of palletizing cells, concluding that virtual simulation improves design efficiency and reduces commissioning time.

Based on the analyzed studies, it can be observed that the current trend in industrial robotics is the integration of virtual simulation, CAD models, and offline programming in the design of robotic cells. The present work follows this direction, having as its objective the design and offline simulation of a robotic palletizing cell in ABB RobotStudio.

3. Modeling the virtual prototype of the end-effector

For the designed palletizing application, a vacuum-based pneumatic solution was selected due to the advantages it offers in handling cardboard boxes, such as non-invasive gripping, operational safety, and the possibility of achieving fast working cycles. This solution enables product handling without deforming or damaging the packaging.

The gripping system is implemented using pneumatic gripping units (area grippers) mounted on the lower structure of the end-effector. These units are controlled individually through solenoid valves, allowing the flexible handling of a variable number of boxes, ranging from one to four

simultaneously.

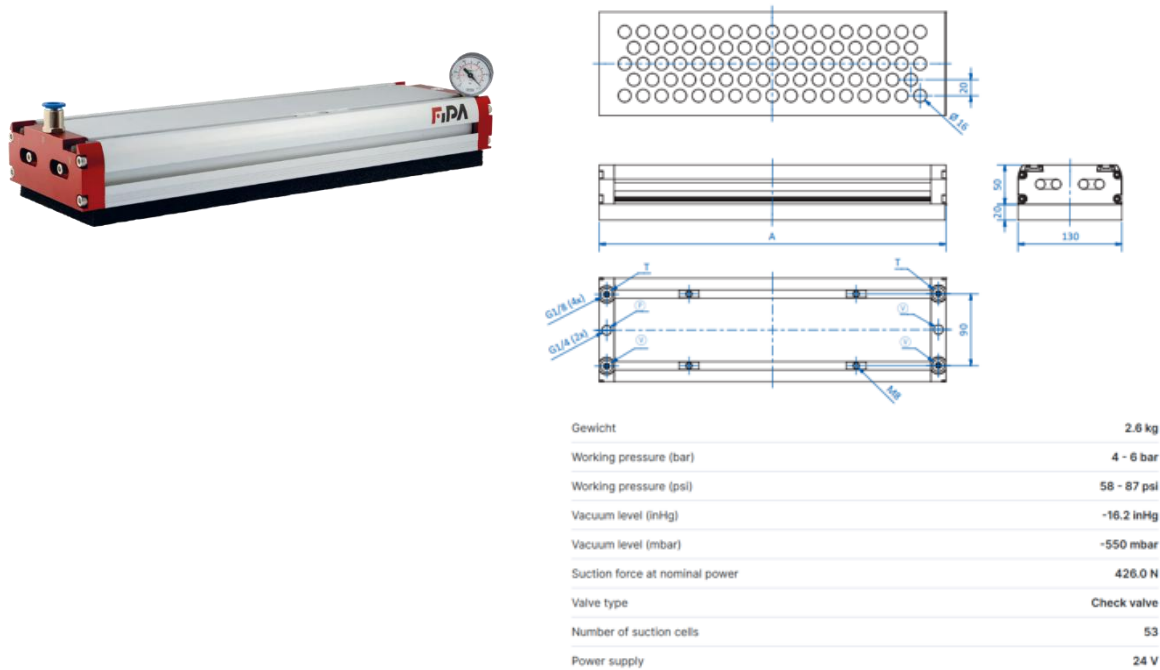


Fig. 3. Area Gripper integrated into the end-effector [7]

The end-effector is of a single-purpose design, intended for handling cardboard boxes used for the storage of refrigerated products. The assembly was designed in CATIA V5 using both standardized components downloaded from manufacturers' CAD databases and custom-designed components developed specifically for the application. The upper structure consists of the robot mounting flange and a protective enclosure housing the pneumatic and electrical components, while the lower structure is made of aluminum profiles on which the vacuum gripping systems are mounted. The end-effector is capable of simultaneously handling up to four boxes with approximate dimensions of $400 \times 300 \times 200$ mm and a mass of approximately 10 kg each. Furthermore, each support plate of the pneumatic gripping units is equipped with a proximity sensor that confirms the presence of the object during the picking operation.

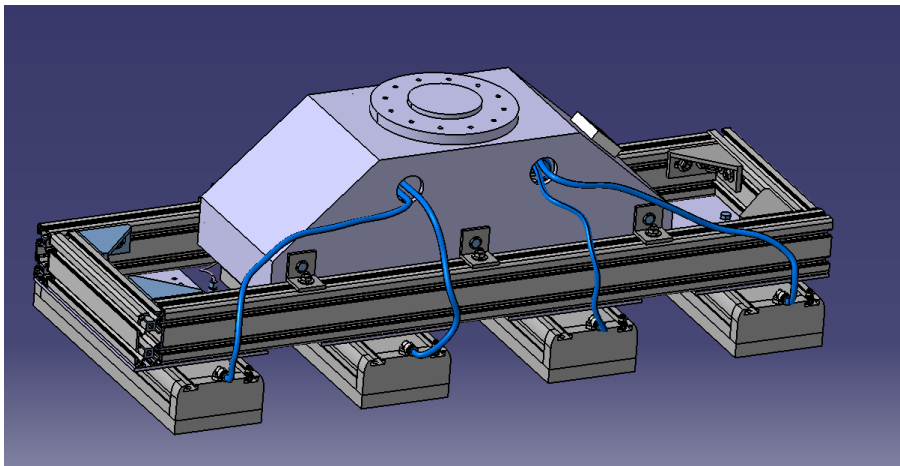


Fig. 4. The virtual prototype of the end-effector

4. 3D Modeling of the Robotic Palletizing Cell

The three-dimensional modeling of the robotic palletizing cell was carried out in the CATIA V5 CAD environment, which was used for the design and integration of all equipment constituting the application. The 3D modeling process aimed to obtain a complete virtual

representation of the cell, necessary for verifying the integration of the equipment, the robot workspace, and the overall operation of the palletizing system. Furthermore, the cable routing was also designed in order to verify its integration within the cell and to prevent interference with the moving components during operation.

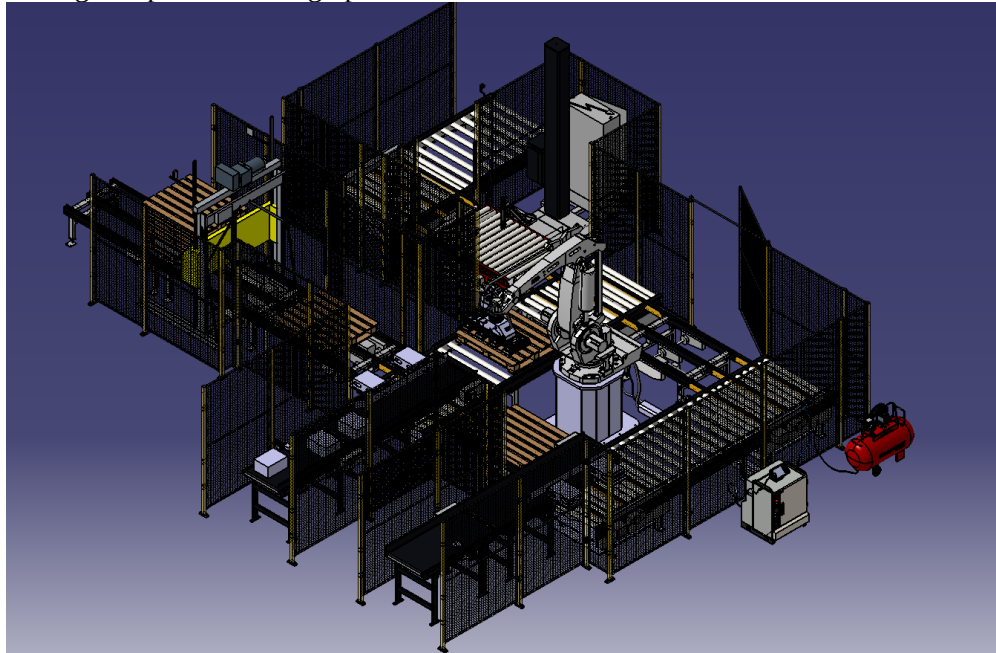


Fig. 5. Virtual prototype of the robotic palletizing cell

During the modeling process, both CAD models downloaded from manufacturers' databases and components specifically designed for the developed application were used. The imported elements included the main standardized equipment of the cell, namely the conveyor systems, the industrial robot, the wrapping machine, and the pallet distribution system (pallet dispenser). The use of these CAD models reduced the design time and enabled the creation of a virtual representation closely matching the configurations commonly used in industrial environments.

The robot selected for the application is the ABB IRB 660, an industrial robot specifically designed for palletizing operations, characterized by a high payload capacity and excellent dynamic performance. It offers a positioning repeatability of approximately ± 0.1 mm and a reach of approximately 3.15 m, allowing efficient coverage of the entire workspace. The robot is equipped with a vacuum end-effector designed for the handling of cardboard boxes, enabling the simultaneous manipulation of up to four products during a single operating cycle.

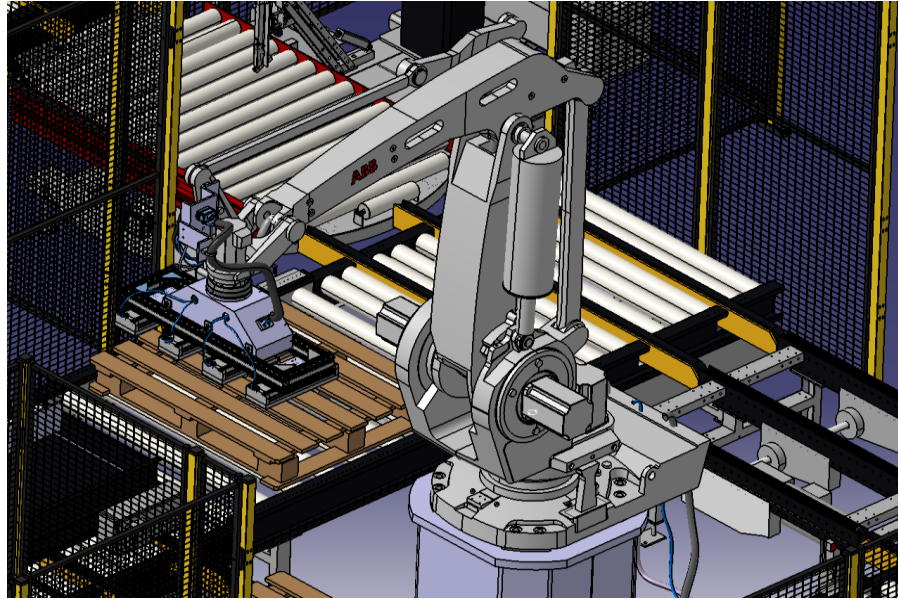


Fig. 6. ABB IRB 660 robot with technological equipment

In addition to the standardized components, the auxiliary elements required for product flow control, such as pusher systems and stop-gate mechanisms, were individually modeled. These components were designed to ensure the correct positioning, separation, and transfer of the boxes along the conveying line toward the palletizing area. The integration of all components into the overall cell assembly enabled the development of a functional configuration and the preparation of the model for the offline simulation stage of the robotic application.

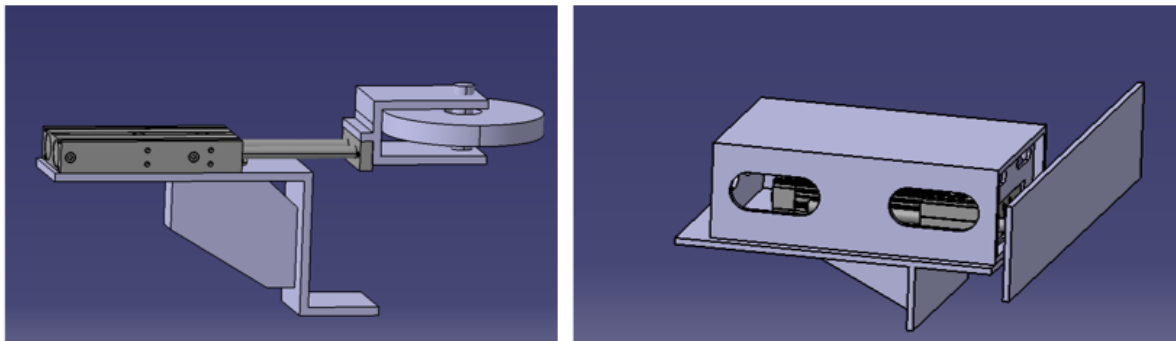


Fig.6. 3D Molded elements for palletizing application

5. Simulation of the Designed Robotic Application Using the ABB RobotStudio

The palletizing cell was imported from the CATIA V5 CAD environment into ABB RobotStudio using an STP file format. To perform the partial simulation, a workstation was configured in RobotStudio, including the controller and the ABB IRB 660/180 robot corresponding to the designed cell.

For the end-effector simulation, a Tool mechanism was defined, in which the TCP (Tool Center Point) was configured and defined as the center of the quadrilateral formed by the pneumatic gripping units.

For the complete simulation of the cell, the mechanisms of the external equipment (centering systems, stop gates, wrapping machine, and pallet positioning system) were defined by specifying their axes of motion (translation/rotation) and operating ranges. Using SmartComponents, the interaction between the equipment was simulated, enabling the automatic generation of product flows. Paths were defined for two generic categories involved in the palletizing process: the handled products and the pallets. Since the cell includes two box infeed lines and two palletizing stations, the robot trajectory was defined for each operating scenario in order to verify the accessibility of all required positions and to avoid collisions.

Through the Event Manager, digital output signals were defined and assigned for controlling the various actuators of the auxiliary equipment.

In the final stage, the signals associated with each mechanism were configured, the paths were imported into the RAPID programming language, and the main program (Main) controlling the execution of the entire palletizing sequence was developed.

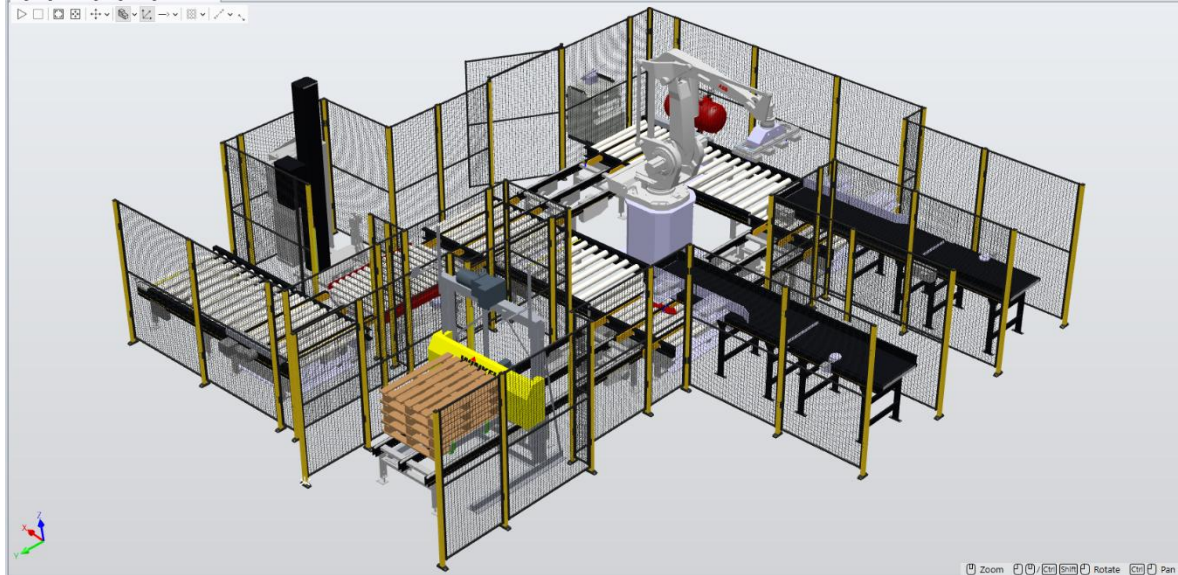


Fig.7. Robotic palletizing cell in ABB Robostudio

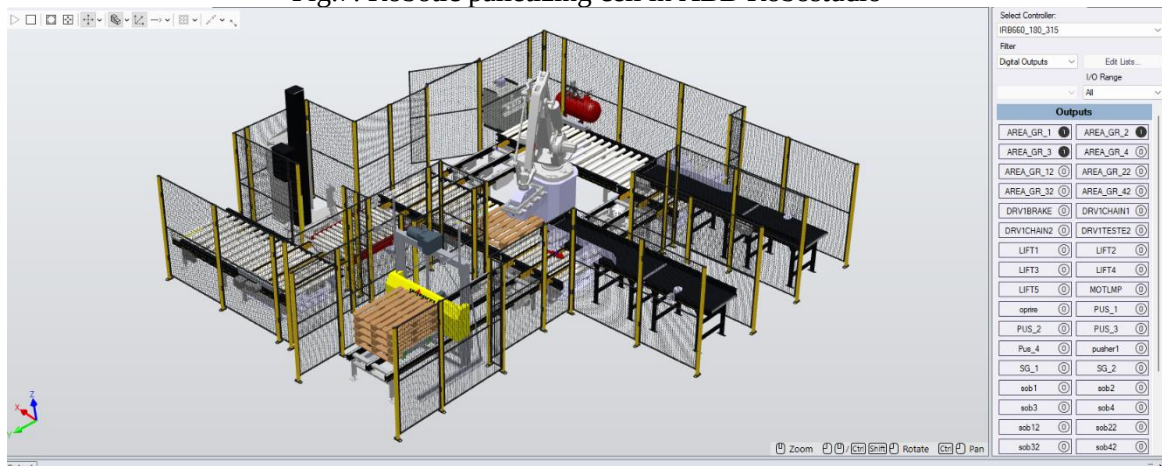


Fig.8. Offline simulation of the palletizing cell

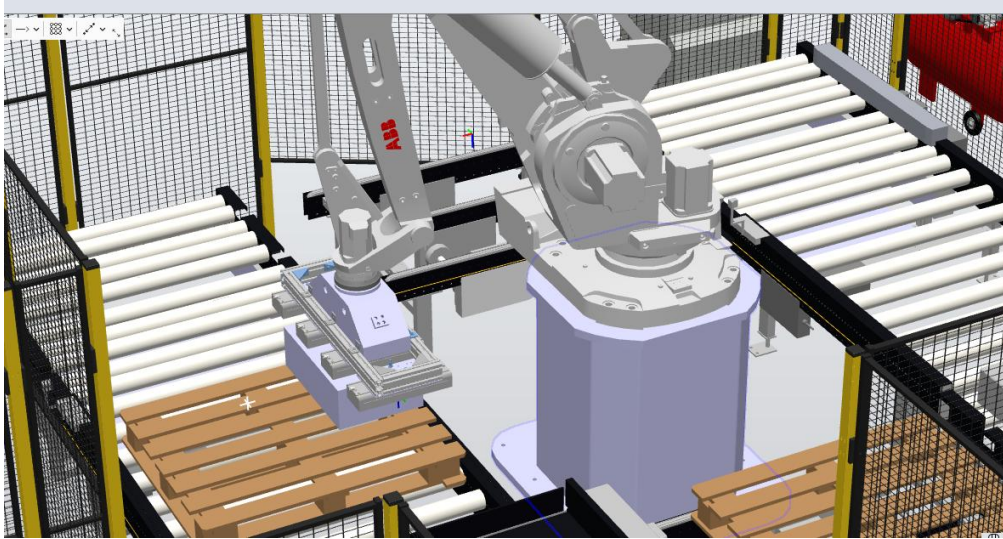


Fig. 9. Sequences from the offline simulation of the palletizing cell

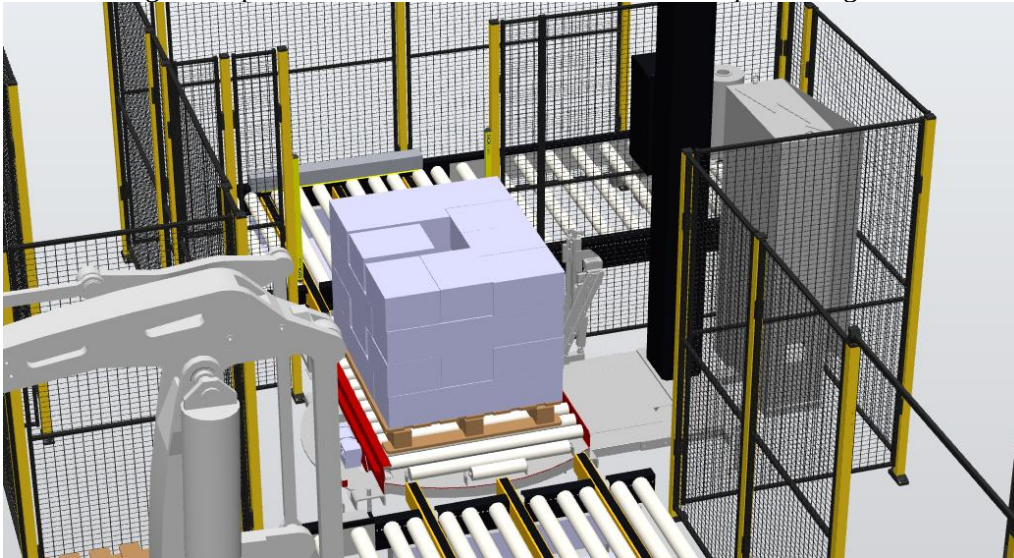


Fig.10. Sequences from the offline simulation of the palletizing cell - wrapping

6. Conclusions

The present thesis had as its main objective the computer-aided design and offline simulation of a robotic cell for the palletizing of boxes containing refrigerated products, using the ABB RobotStudio environment. Within the project, an analysis of robotic palletizing applications and existing solutions in industry was carried out, highlighting the important role of automation in optimizing logistics and handling processes.

Based on an existing model taken from a reference video, a complete robotic cell was conceived and implemented, including the ABB IRB 660 industrial robot, the conveying systems, the end-effector, and the auxiliary equipment. The 3D modeling of the components enabled the coherent integration of all elements into a functional virtual system.

During the offline simulation stage, the Tool mechanisms, the TCP point, the working paths, and the objects to be palletized were defined, using SmartComponents and digital signals for process coordination. Programming in the RAPID language enabled the control of the operating sequences of the cell and the integration of all equipment into an automated flow.

The simulation carried out allowed the verification of the correct operation of the system, the identification and elimination of possible motion conflicts, as well as the optimization of the robot trajectories. The obtained results confirm that the designed cell meets the requirements of the palletizing application and operates correctly in the virtual environment.

In conclusion, the thesis demonstrates the efficiency of using computer-aided design and offline

programming in the development of robotic cells, highlighting the major advantages of simulation before physical implementation, such as reducing development time, increasing safety, and optimizing system performance.

7. Bibliography

[1]. Nicolescu, A. – *Implementarea Robotilor Industriali in Sistemele de Productie*, note de curs si metodologii de proiectare, UPB, 2025

[2]. Nicolescu, A. – *Conceptia si Exploatarea Robotilor Industriali 2*, note de curs si metodologii de proiectare, UPB, 2024

[3]. Oyekan, J., Smith, S. (2018), “Offline programming and simulation of industrial robots for manufacturing applications”, *Robotics and Computer-Integrated Manufacturing*, 54, pp. 45–56, ISSN 0736-5845.

[4]. Johansson, M., Eriksson, P. (2019), “Offline programming of ABB robotic systems using RobotStudio”, *Proceedings, International Conference on Robotics and Automation (ICRA), IEEE, Montreal, Canada, May 2019*.

[5]. Drăghici, G. (1999), *Ingineria integrată a produselor*, Editura Eurobit, Timișoara, ISBN 973-96065-7-1.

[6]. ABB (2021), *RobotStudio User Guide*, ABB Robotics, Västerås, Sweden.

[7]. <https://fipa.com/en/products-and-solutions/vacuum-automation/area-grippers/>

(accessed on 18.05.2026)

COMPUTER-AIDED DESIGN AND PROGRAMMING – OFFLINE SIMULATION OF A ROBOTIC PALLETIZING CELL

STĂNESCU Bogdan, and Prof.dr.ing. NICOLESCU Adrian Florin

¹Faculty: IIR, Specialization: Robotică, Year of study: IV, e-mail: bogdanstanescu2004@yahoo.com

Summary:

In this paper, it is presented the way in which a robotic cell for palletizing with one input and one output was designed and modeled. Afterwards, the procedures to import said cell in a simulation and offline programming software in which it was the trajectories of the industrial robotic arm were studied. The study focuses on the designed of assemblies both relating to the technological equipment of the industrial robot, as well as the other assemblies in the cell, than making them communicate with each other via the robot controller. In the simulation software, this is made using sensors and signals, that are also used in making the assemblies function. The obtained results demonstrate the advantages of using offline programming and virtual simulation for automated palletizing applications.

Keywords: industrial robot, palletizing, offline programming, design, assembly

1. Introduction

This project involves the realization, for the purpose of physical implementation, of a robotic palletizing cell with one input and one output consisting of packs of plastic bottles. The main purpose is the design, offline simulation/programming of the entire robotic cell, as well as its automation so that there is no need for human operators. The main stages of the project were the selection of the industrial robot, the realization of the multifunctional end-effector assembly of the robot, the import of the structural elements of the cell (product conveyor, pallet conveyor) in cases where these could be obtained from manufacturers, and the modeling based on technical data of the structural elements that were not available to be imported (pallet stacker). After obtaining these elements, it was necessary to design/model assemblies for which neither sources nor technical data were found (object pusher, simple portal for separator handling).

The stages mentioned above are related to the design of the cell in a computer-aided design environment, which is finalized by the distribution of the elements so that they are within the useful working space of the industrial robot, while also occupying as little space as possible, in order to be easily implemented.

After that, the designed cell was imported into the offline simulation/programming environment, namely ABB RobotStudio. The programming process was composed of two stages: preparation for programming and the actual programming.

There were no modifications within the cell.

2. Current State

In the context of the development of modern automated systems, robotic palletizing and automatic handling cells represent an important direction of research and industrial implementation. Numerous studies have analyzed the design, simulation and offline programming of industrial robots, as well as their integration into processes specific to Industry 4.0.

In the paper “Offline Programming of an ABB Robot Using Imported CAD Models in the RobotStudio Software Environment”, carried out by Radovan Holubek, Daynier Rolando Delgado Sobrino, Peter Košťál and Roman Ružarovský, the use of the ABB RobotStudio software for the offline programming of an ABB industrial robot through the import of CAD models of equipment into a virtual environment is presented. The authors highlight the advantages of using virtual simulation, such as collision detection, trajectory optimization and reduction of the time required for the physical implementation of the robotic system. The study demonstrates the importance of integrating CAD models into the development process of robotic cells.

A similar approach is presented in the article “An Off-line Programming System for Palletizing

Robot”, developed by Youdong Chen, Ling Li and Wei Tang. In this paper, a dedicated offline programming system for palletizing robots is developed, with the emphasis placed on the automatic generation of stacking models and on robot trajectory planning. The authors underline the fact that offline programming allows the testing and validation of system operation before physical implementation, contributing to cost reduction and increased flexibility of the industrial process. In the paper “AdaptPack Studio: an automated intelligent framework for offline factory programming”, an intelligent software framework intended for the development and simulation of robotic palletizing cells is proposed. The authors present a modular library for the rapid configuration of the elements of a robotic cell and for carrying out the offline programming of industrial robots. The obtained results show a significant reduction in the design and integration time of automated palletizing systems.

Also, in the paper “Incorporation, Programming and Use of an ABB Robot for the Operations of Palletizing and Depalletizing at an Academic Research Oriented to Intelligent Manufacturing Cell”, the integration of an ABB robot into an intelligent manufacturing cell for palletizing and depalletizing operations is analyzed. The authors aim to optimize the technological flow and automate the product handling process, highlighting the importance of robot simulation and programming before real implementation.

Similar to the studies presented above, the present project aims at the realization of a robotic palletizing cell with one input and one output for handling packs of plastic bottles. The project involves the mechanical design of the cell, the realization of a multifunctional end-effector, the modeling of auxiliary components and the integration of all equipment into an offline simulation and programming environment using ABB RobotStudio.

3. Design of the end-effector assembly

The end-effector for the articulated arm robot is multifunctional and can handle, using the “Clamp Style” concept, up to 5 pre-packaged juice packs, each having a weight of 6 kg. Using the “Fork Style” concept, it can handle pallets through the use of the corresponding forks.

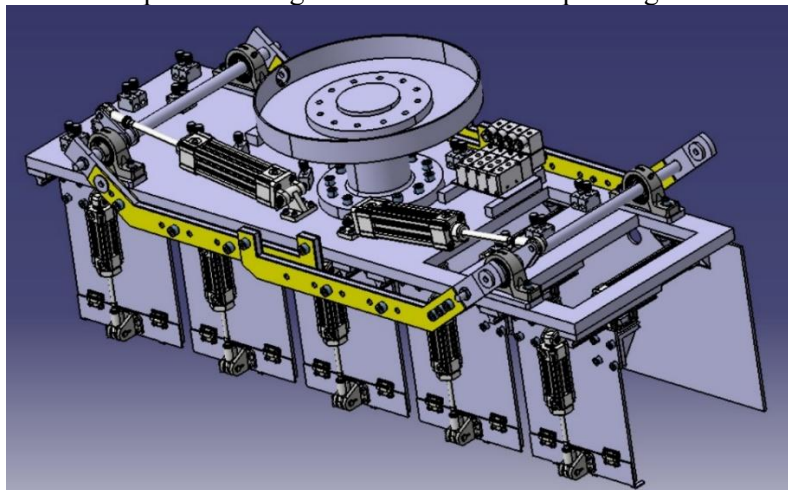


Fig. 1 3D Synthesis of the End-Effector

The handling of objects is carried out with the help of 5 plates individually actuated by linear pneumatic motors, equipped with electromagnetic sensors for position detection and stopping at the established stroke. The linear motion is guided by ball bearing blocks, which ensure precision and reduced friction. The objects are supported from below by vertical pneumatic motors, provided with rod-eye joints and hinges, which allow the rotation of the plates by 90° and their secure fixation.

The claws for pallet handling are actuated by two linear motors with mechanical joints and pillow block support units with rigid bearings. The synchronization of the movement is achieved through an articulated bar system.

The end-effector of the gantry robot is built on an aluminum frame and includes a vacuum system with four suction cups for handling cardboard separators. The gripping of the separator is

confirmed by a capacitive sensor that detects the presence of the object.

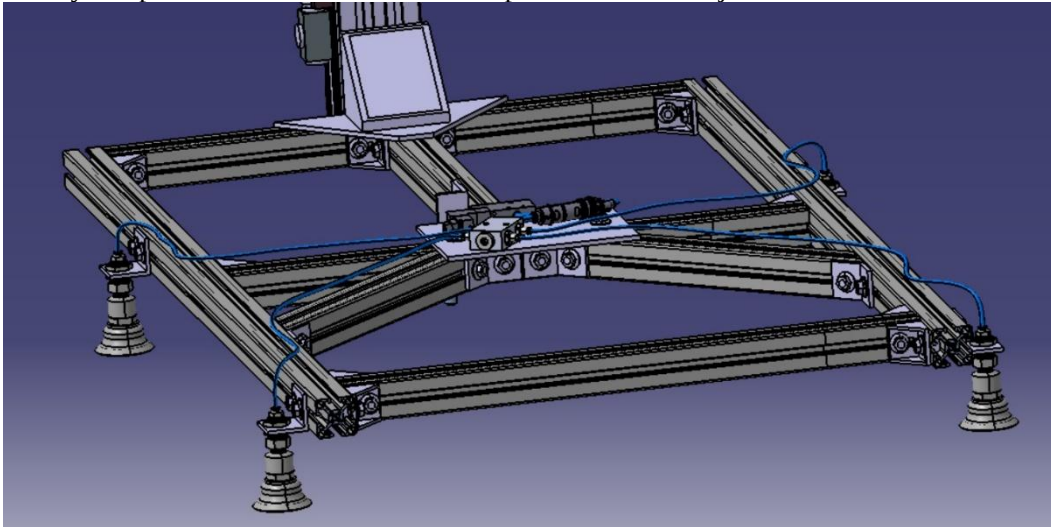


Fig. 2 3D Synthesis of the End-Effector for the Gantry Robot

4. Modeling of the Robotic Cell

The first step in modeling the cell is the modeling of an aluminum profile frame for the simple gantry robot, followed by the assembly of the developed end-effector on the industrial robot, as well as the installation of the pneumatic tubes and electrical cables:

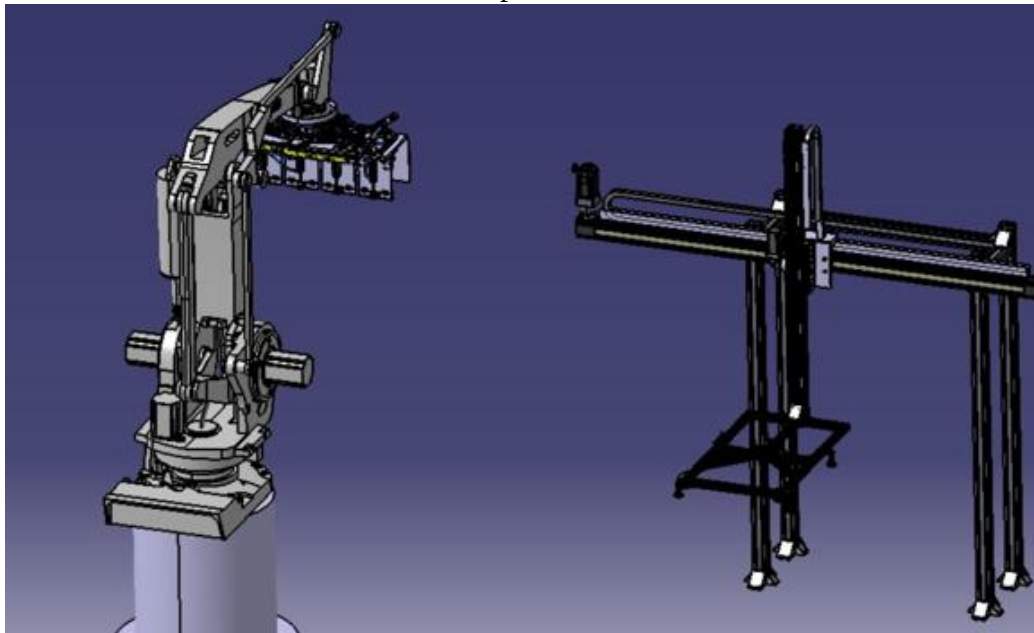


Fig. 3 Industrial Robots Mounted with the End-Effectors

The conveyors used for the entry of pallets and objects, as well as the conveyor intended for pallet discharge, were obtained from manufacturers. A wrapping system was also integrated, used for securing and stabilizing the load on the pallet. The pushers used for positioning the pallets and objects, as well as the blade-type stoppers, were modeled.

**COMPUTER-AIDED DESIGN AND PROGRAMMING – OFFLINE SIMULATION OF A ROBOTIC
PALLETIZING CELL**

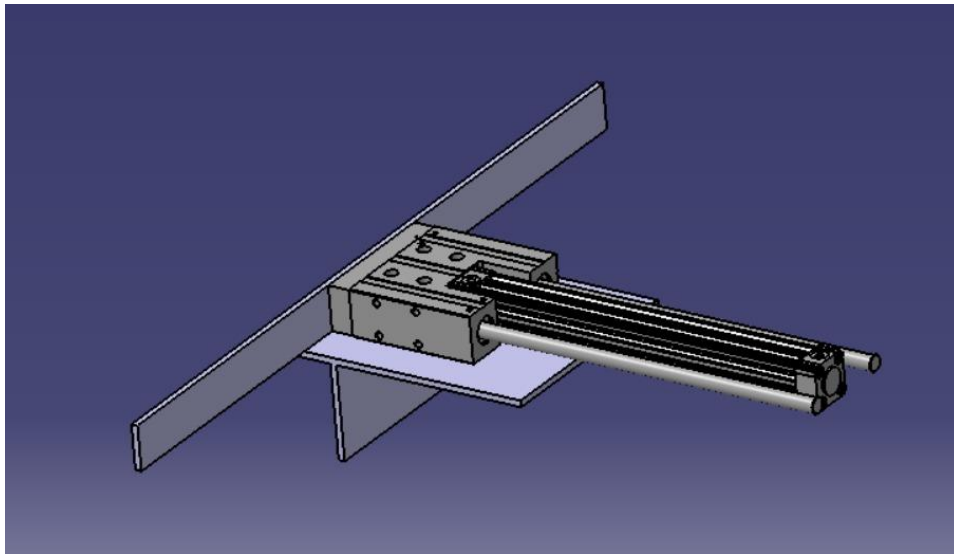


Fig. 4 Pusher for Object Positioning

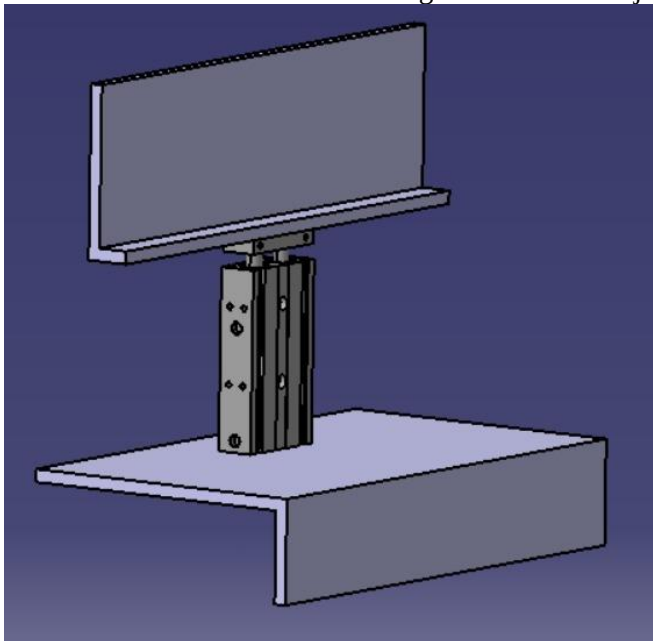


Fig. 5 Stop Gate for Stopping the Continuous Flow of Objects

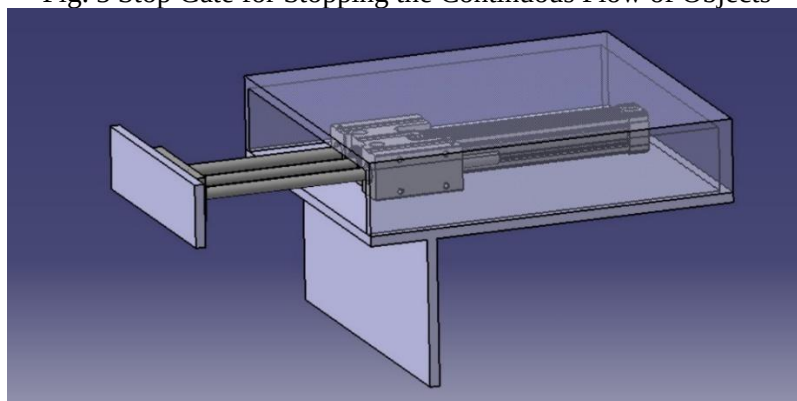


Fig. 6 Pusher for pallet Positioning

COMPUTER-AIDED DESIGN AND PROGRAMMING – OFFLINE SIMULATION OF A ROBOTIC PALLETIZING CELL

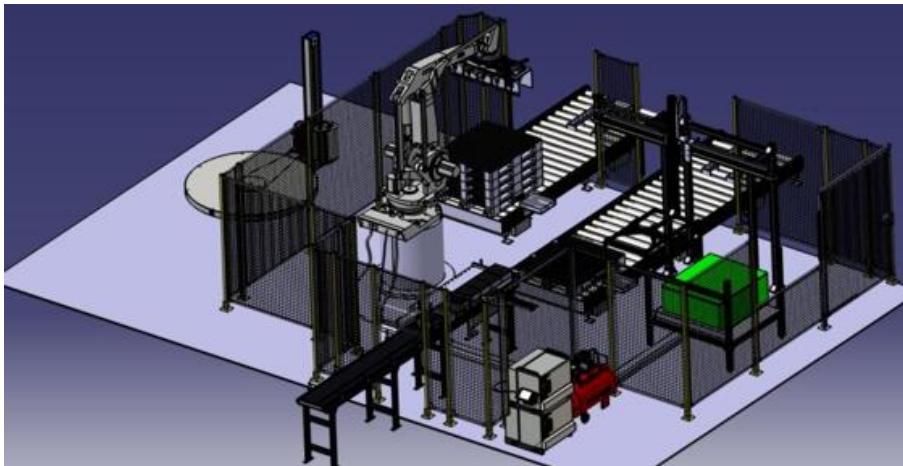


Fig. 7 Final Assembly of the Robotic Cell

5. Offline Simulation/Programming

For the realization of this partial simulation, it was necessary to create a station in the RobotStudio application that included a controller and the robot model corresponding to the designed cell, ABB IRB 660/180, as well as a drive for actuating the simple gantry robot. The offline programming-simulation environment allows the conversion of other native 3D CAD formats, which led to the direct insertion of the end-effector assembly into the created project. In order to simulate the end-effector, it is necessary to create a new mechanism of the “Tool” type. Within it, the mechanism axes will be created by selecting the components, the rotation/translation axis, and the maximum angles or distances.

For the realization of the simulation of the entire cell, the first step is loading the CAD file containing the virtual prototype of the entire cell to be designed. After that, the external mechanisms are created, such as the centering systems, stoppers, wrapping machine mechanisms, and the pallet centering system.

The objects to be palletized will be created using “SmartComponents” in order to create the number of objects required for building the stack. Three paths will be created: one for pallets, one for the packs of plastic bottles, and one for the separator, integrating barrier-type sensors along their routes.

The final step consists of creating the signals required for each mechanism, importing the created paths into RAPID, and writing the main program to create the program that will be executed by the industrial robot.

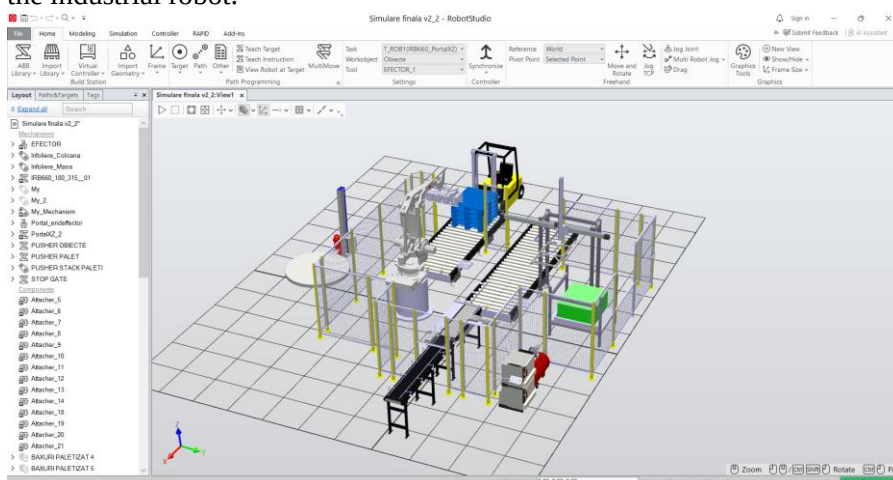


Fig. 8 Virtual Prototype of the Cell Imported into RobotStudio

**COMPUTER-AIDED DESIGN AND PROGRAMMING – OFFLINE SIMULATION OF A ROBOTIC
PALLETIZING CELL**

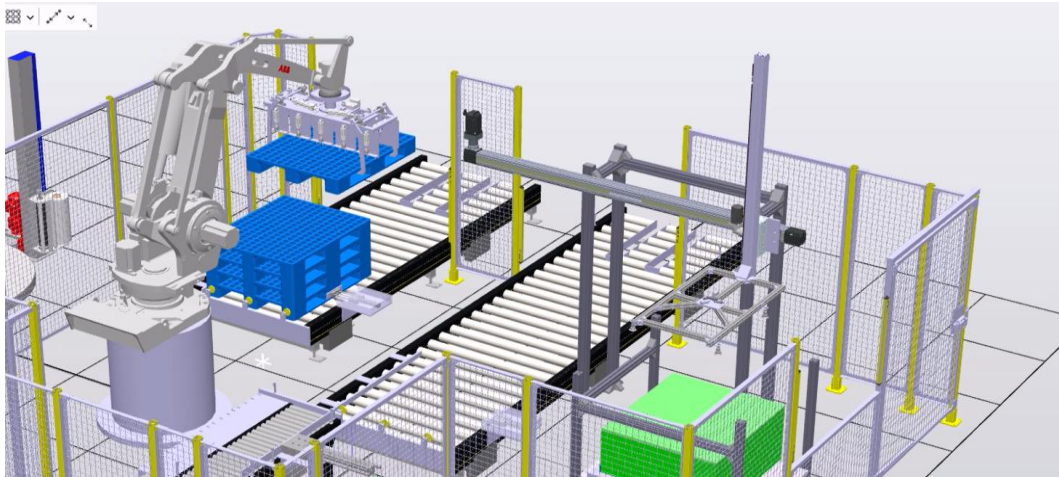


Fig. 9 Pallet Pickup

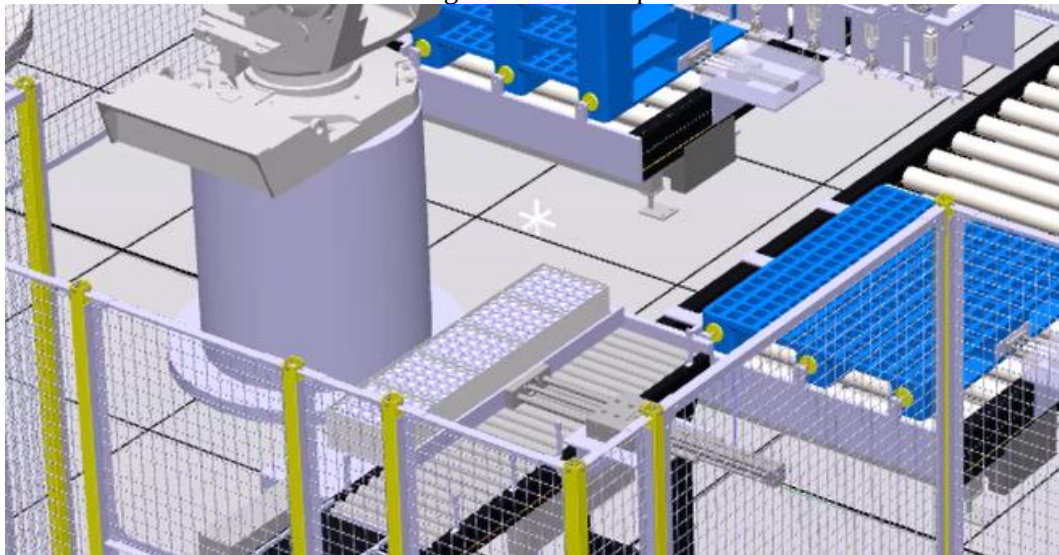


Fig. 10 Use of the Pusher for Positioning the Objects

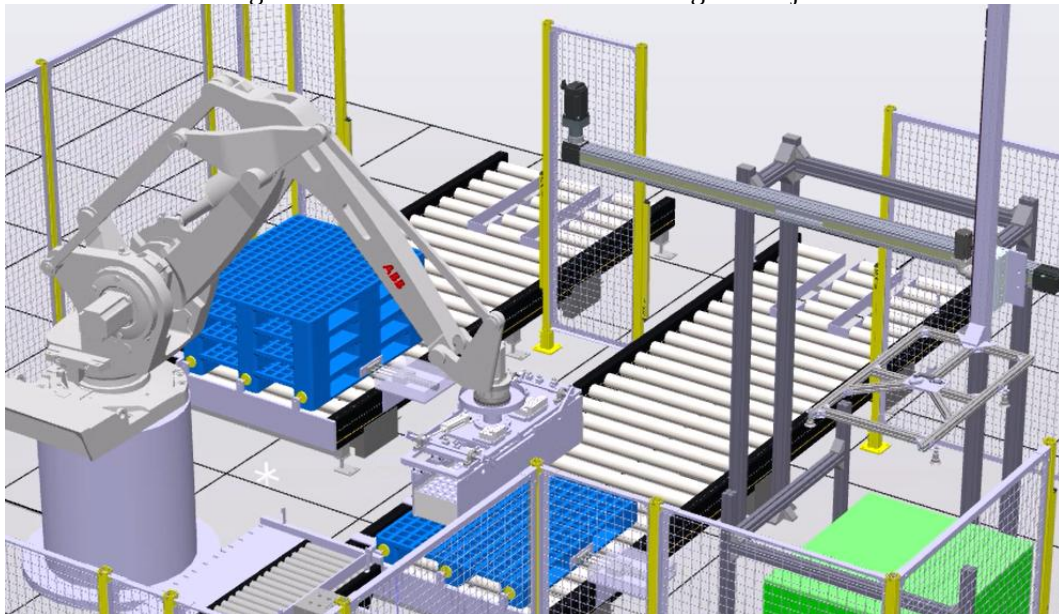


Fig. 11 Placement of the First Object

**COMPUTER-AIDED DESIGN AND PROGRAMMING – OFFLINE SIMULATION OF A ROBOTIC
PALLETIZING CELL**

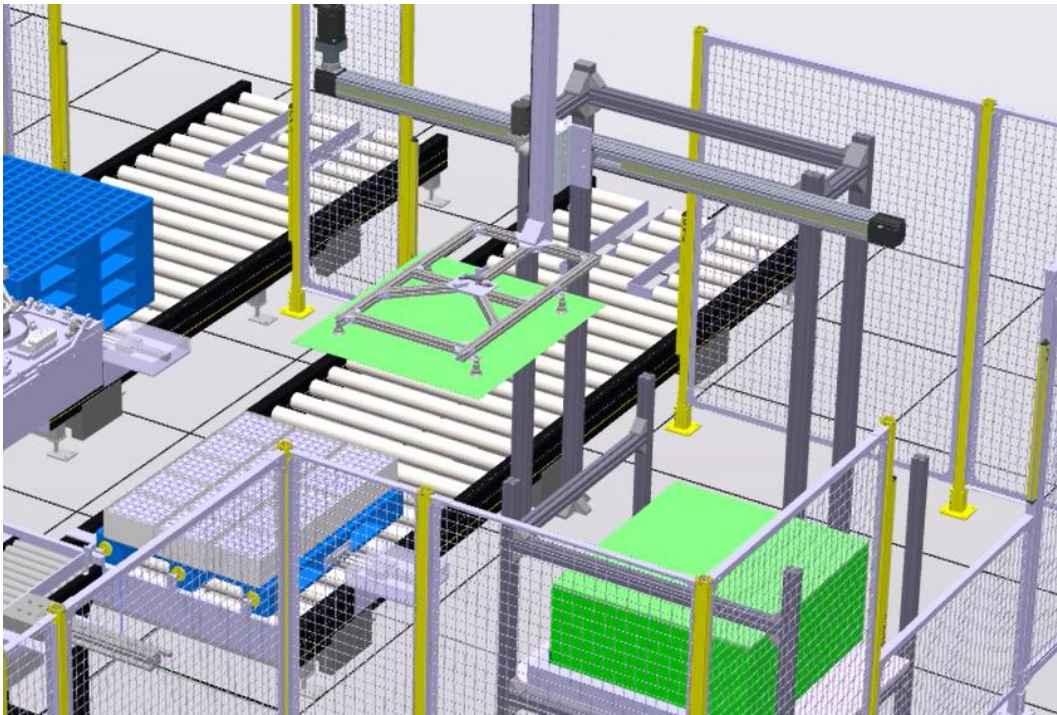


Fig. 12 Placement of the sheet

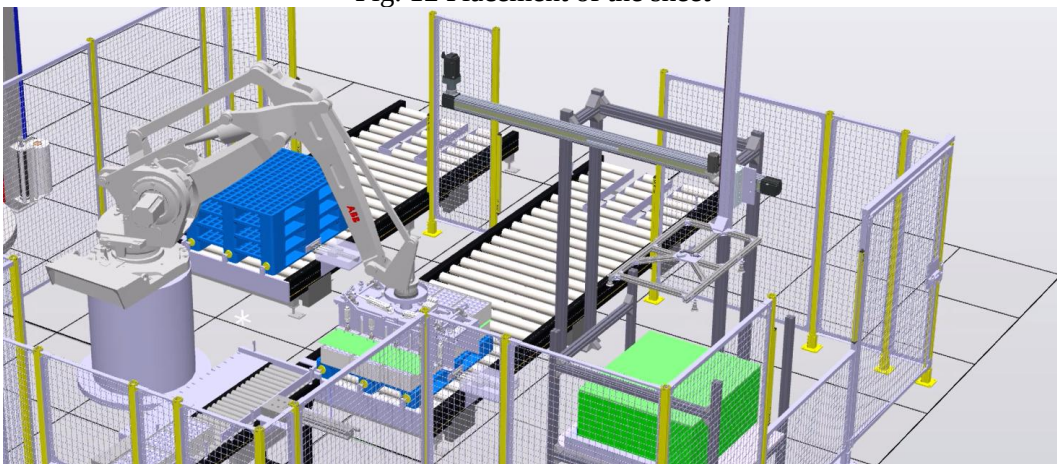


Fig. 13 Placement of the Objects Rotated by 90 Degrees

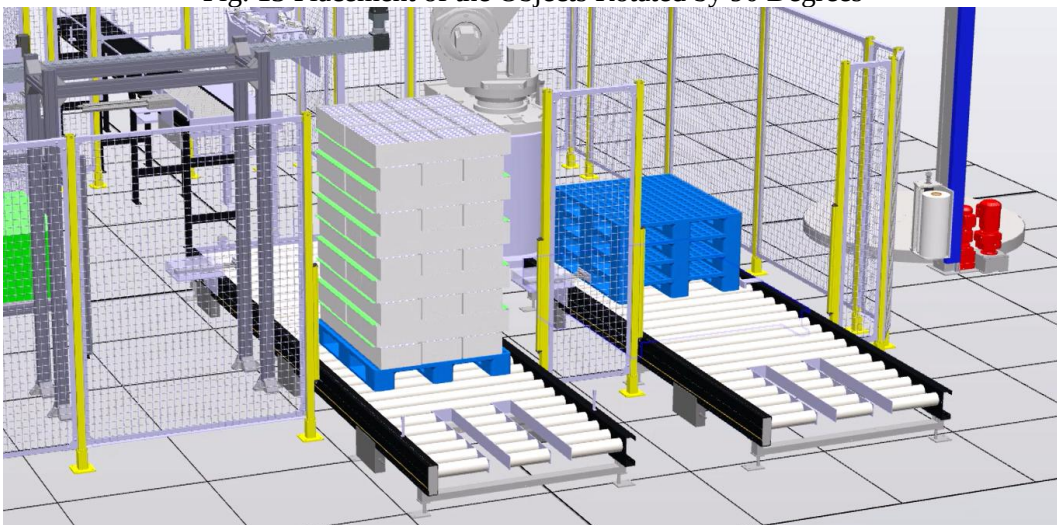


Fig. 14 Movement of the Full Stack

6. Conclusions

Within this paper, the computer-aided design and offline simulation/programming of a robotic palletizing cell for handling packs of plastic bottles were carried out. A multifunctional end-effector as well as a monofunctional one were designed, and all the necessary subsystems were modeled and integrated into a complete virtual assembly.

The robotic cell was imported and simulated in ABB RobotStudio, where the mechanisms, robot paths, and operating logic of the automated system were defined. The use of offline programming allowed the verification of the cell operation and the validation of the solution before a possible physical implementation.

The obtained results highlight the advantages of using computer-aided design and virtual simulation in the development of modern robotic systems, contributing to the optimization of the design process and the reduction of implementation time.

7. Bibliografie

- [1]. Monkman, G. J., Hesse, S., Steinmann, R. & Schunk, H. (2007), “Robot Grippers”, Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, ISBN 978-3527406197.
- [2]. Corke, P. (2017), “Robotics, Vision and Control: Fundamental Algorithms in MATLAB”, 2nd ed., Springer, Cham, Switzerland, ISBN 978-3319544120.
- [3]. Tao, F., Zhang, H., Liu, A. & Nee, A. Y. C. (2019), “Digital Twin in Industry: State-of-the-Art”, IEEE Transactions on Industrial Informatics, Vol. 15, Issue 4, pp. 2405-2415, ISSN 1551-3203.
- [4]. Hoebert, T., Seibel, S., Amersdorfer, M., Vincze, M., Lepuschitz, W. & Merdan, M. (2024), “A Framework for Enhanced Human–Robot Collaboration during Disassembly Using Digital Twin and Virtual Reality”, Robotics, Vol. 13, Issue 7, Article 104, ISSN 2218-6581.
- [5]. Siemens Digital Industries Software (2024), “Process Simulate X Robotics Standard – Robot Programming and Simulation Documentation”, Siemens Digital Industries Software, Plano, TX, USA.
- [6]. Babcsinski, M., Cruz, F., Duarte, N., Santos, S., Alves, S. & Neto, P. (2024), “Offline Robot Programming Assisted by Task Demonstration: an AutomationML Interoperable Solution for Glass Adhesive Application and Welding”, Robotics and Computer-Integrated Manufacturing, preprint edition.

MODAL ANALYSIS OF AN ARTICULATED SUPPORT FOR MOBILE DEVICES

IVAN Toni-Marian, POPESCU Diana, ALEXANDRU Tudor-George

Faculty: IIR, Specialization: IE, Year of study: IV, e-mail: ivantoni2003@gmail.com

ABSTRACT: *This paper presents the modal analysis of an articulated support designed to support mobile devices on the motorcycle's handlebars. The 3D CAD model was developed*

in CATIA V5, based on the dimensions of the latest models of smartphones. A simplified numerical model was created in ANSYS, in which the articulated structure was represented by beams, while the clamping components remained as 3D solid elements. The smartphone's mass was introduced and positioned according to its center of gravity. The structure was considered fixed to the motorcycle's handlebar, the first six natural frequencies and vibration modes were extracted and compared with the excitation range generated by the motorcycle's engine during operation. The comparison indicated a potential risk of resonance under different working conditions. Based on the analysis, modifications and vibration mitigation solutions were proposed to improve the structural dynamic performance

KEYWORDS: *CAD, simulation, modelling, modal analysis, frequency analysis.*

1. Introduction

In today's world, mobile supports are mandatory accessories designed to make navigation easier for the driver. For motorcycles, these supports are very effective for route monitoring, quick access to different apps or changing the music. However, the operating environment of the motorcycle is characterized by dynamic loads generated by the engine, road irregularities and vibrations transmitted throughout the structure of the motorcycle. These things could damage the smartphone, so it becomes a necessity to choose a support that ensures the safety of the device.

This paper verifies, through modal analysis, the dynamic behavior of an articulated support for mobile devices designed for motorcycles. The purpose of this study is to identify the natural frequencies and vibration modes of the assembly and to evaluate the possibility of the resonance phenomenon occurring in relation to the frequencies generated by the motorcycle's engine. The resonance can lead to the amplification of displacements and accelerations transmitted to the smartphone, affecting the integrity of the support, the stability of the phone, or even its sensitive electronic components.

To make this study, the first step was to develop a 3D model of the assembly, using the CAD software CATIA V5, using the dimensions of the existing smartphones. The support configuration was made taking into consideration constructive solutions used in real supports, including articulated elements and fixing mechanisms used to calibrate the position of the device.

The next step was to simplify the geometry and transpose it into a numeric model in ANSYS, to carry out the modal analysis. The articulated structure was modeled using beam elements, to reduce the computational cost and to maintain the global mechanic relevance, while the clamping and the complex geometric elements were left in the 3D state. The mobile device was represented as a concentrated mass element, defined by the mass and the center of gravity of the analyzed phone. The boundary conditions determined the assembly rigidly fixed on the handlebar.

The modal analysis generated the results of six vibration modes and their natural frequencies, which were compared with the frequency range specific to the operating modes of the motorcycle's engine. Based on the results, the risk of resonance occurrence was assessed and constructive measures were proposed to improve the dynamic behavior of the support, such as increasing the structural rigidity, modifying the geometric configuration or introducing

vibration damping solutions.

2. Literature overview

The analysis of the dynamic behavior of mechanical structures subjected the vibration excitations in an important field of research in mechanical engineering, especially for the applications in which the structural integrity and functional stability are influenced by periodic dynamic demands. In case of mounting systems for mobile devices mounted on vehicles, vibrations generated by engine's operation and interaction with road surface can lead to undesirable phenomena, such as amplification of displacements, damage to components, or even loss of functionality of the transported device. For this reason, the modal analysis and the study of natural frequencies are essential tools for evaluating the behavior of these structures from the design stage.

Many research articles published in the U.P.B. Scientific Bulletin explore the topic of vibrations and how mechanical structures respond dynamically. For example, in his work, Mazilu [1] examines how a wheel and rail interact when they're subjected to repeated mechanical forces. He points out that if the frequency of these forces matches the natural frequency of the system, it can have a major effect on how the structure behaves. Although the study refers to railway systems, the methodology for evaluating the resonance risk is also relevant for lightweight structures mounted on vehicles. In another contribution, Mazilu [2] analyzes the structural response of an elastic component subjected to vertical dynamic forces, demonstrating the importance of modal characterization in evaluating the behavior of periodically loaded structures. The results support the need for correlation between real operating conditions and the dynamic characteristics of the analyzed assembly.

The characterization of the dynamic behavior of structures and materials is also addressed by Benmansour et al. [3], who investigate the static and dynamic response of composite structures using experimental and numerical methods. The study emphasizes the importance of identifying the vibration eigenmodes and the mechanical response under dynamic conditions, an aspect applicable also to small articulated structures.

In the work proposed by Zheng et al. [4], modal analysis is used as a tool for structural optimization of a mechanical equipment, in order to avoid critical operating frequencies. The authors demonstrate that modifying the geometric and constructive parameters can lead to the displacement of the natural frequencies outside the excitation range, a methodology applicable also in the case of articulated supports exposed to engine vibrations.

The issue of flexibility of articulated structures and its impact on dynamic behavior is addressed in the works of Nicolescu et al. [5], which highlights the relationship between structural rigidity, functional precision and dynamic stability of articulated mechanical systems. Although the study focuses on industrial robots, the principles regarding the influence of structural flexibility on dynamic response are directly relevant to the topic of the present work.

Munteanu [6] also uses the finite element method to analyze the mechanical behavior of articulated structures, demonstrating the efficiency of simplifying numerical models for evaluating structural performance and overall stiffness. This approach justifies the use of simplified modeling with beam-type elements for the articulated support analyzed in this paper. In conclusion, current research confirms the importance of modal analysis and numerical simulations for evaluating the dynamic behavior of mechanical structures subjected to vibrations. The present work is in this direction, proposing the modal analysis of an articulated support for mobile devices mounted on a motorcycle, with the aim of identifying the resonance risk and formulating constructive solutions to improve dynamic performance

3. Numerical approach

- Pre-processing and solving

The preprocessing stage represents an essential phase in the realization of a correct numerical analysis, because the accuracy of the results is dependent on the quality of the mathematical model and on the way the initial conditions are defined. Regarding this study, the preprocessing stage aimed to prepare the structural geometric and structural model of the articulated support for mobile devices in order to perform a modal analysis in the ANSYS Workbench. The process began with the creation of a complete 3D geometric model in CATIA V5,

starting from the dimensional specifications of the latest smartphones, so that the geometry of the holder reflects the real conditions of use (see Fig.1).

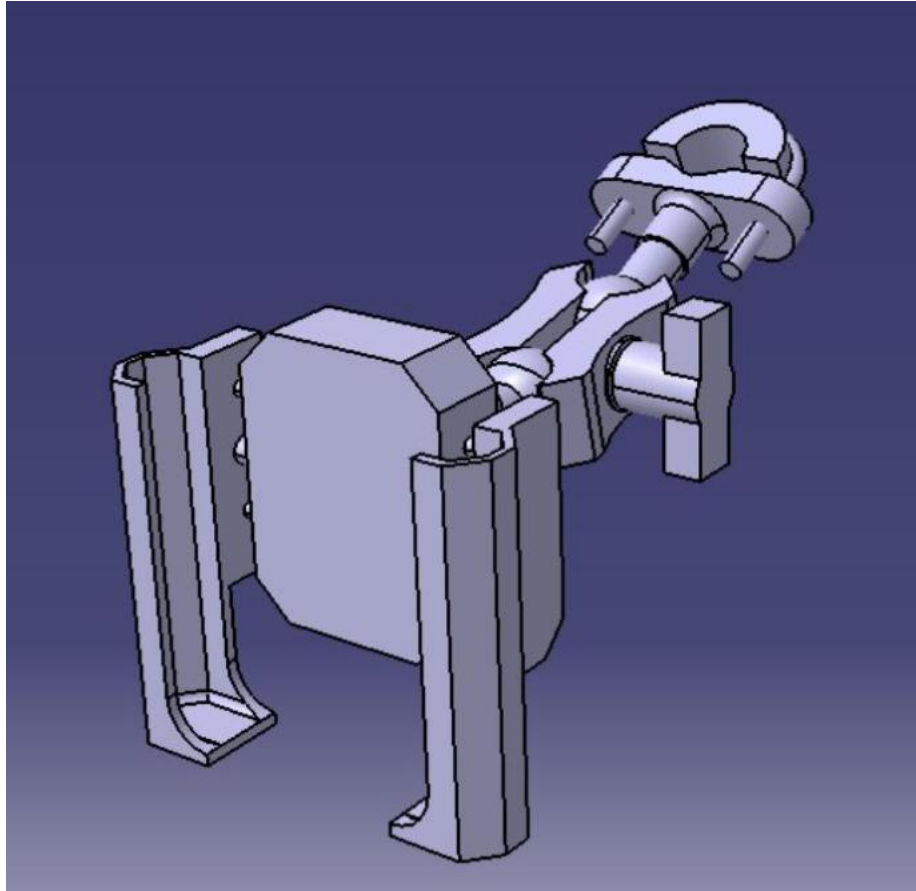


Fig.1. The initial geometry in CATIA V5

The constructive configuration of the holder was designed taking as a reference commercial motorcycle handlebar mounting systems, including the fixing elements, articulated segments and components intended for positioning and supporting the mobile device.

Because the assembly was complex and it needed computational cost optimizations, the 3D model underwent a structural simplification process. For this purpose, the ANSYS Design Modeler module was used to convert 3D components with predominantly axial and flexural behavior to 1D beam elements and 2D shafts. This approach allowed for the preservation of the global mechanical behavior of the articulated structure, while significantly reducing the size of the numerical model and the time required for processing.

The elements with a complex geometry and an important functional role in transmitting forces, such as the handlebar mounting components and the interface elements between the articulated segments, were maintained in 3D representation and modeled using solid elements.

Thus, the final model had a hybrid structure, combining the efficiency of 1D and 2D elements with the precision of 3D modeling in critical areas.

The discretization of the model was performed in a controlled manner, depending on each component's structural specifications. For the 3D solid elements, a tetrahedral mesh was used, which is suitable for modeling complex geometries and irregular surfaces. For the structurally simplified elements, the discretization was achieved using beam elements, defined based on the geometric properties of the cross-sections and the material characteristics. This strategy allowed obtaining an optimal compromise between numerical accuracy and computational efficiency.

The interaction between the model components was defined by configuring the structural contacts

based on fixed kinematic couplings, which correspond to the real assembly mode. This method allowed for the realistic simulation of the transfer of forces and structural continuity between the articulated elements considered rigidly connected in the hypothesis of the analyzed model. The mounted mobile device was modeled using a concentrated mass element, defined by the mass of a real smartphone and the known position of its center of gravity. This simplification allowed the representation of an inertial influence of the device upon dynamic computations of the assembly, without introducing additional geometric complexities (see Fig.2).

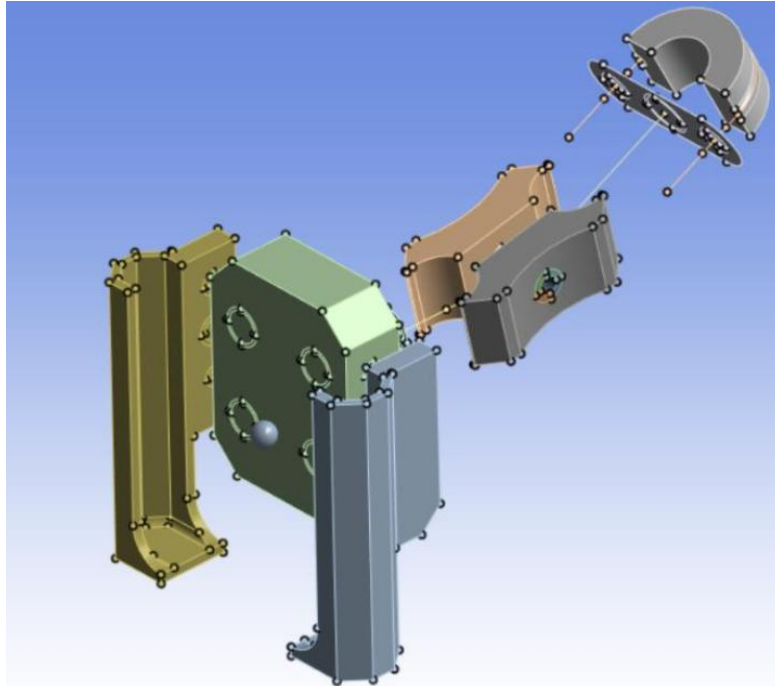


Fig.2. The simplified simulation model including 1D,2D, 3D bodies and a 0D element to capture the mass of the smartphone

The boundary conditions were established so that they reproduce the hypothesis of a system in real working conditions. The handlebar mounting area was considered rigidly embedded, assuming the handlebar's structure has a significantly superior rigidity compared to the analyzed support.

Finally, the resulting numerical model was verified from a geometric point of view, of the connectivity of the elements and of the consistency of the defined structural properties, to ensure the validity of the subsequent modal analysis. Through this preprocessing stage, an efficient and representative model was obtained, capable of describing the dynamic behavior of the assembly under relevant operating conditions.

Modal analysis is a method used to determine the dynamic behavior of a mechanical structure by identifying the natural frequencies and modes of vibration. In the case of this work, this analysis is essential for assessing the resonance risk of the articulated support intended to mount a smartphone on the handlebars of a motorcycle.

Solving the general equation of motion leads to an eigenvalue problem, from which the natural frequencies of the structure and the associated modes of vibration are determined.

The natural frequencies indicate the values at which the structure tends to oscillate naturally, and the eigenmodes describe the shape of the structure's deformation at these frequencies.

In the context of this work, these results are compared with frequencies generated by the force of the motorcycle's engine. If the natural frequencies and the engine's frequencies overlap, the phenomenon of resonance can appear, which could amplify the vibrations and may affect the stability of the support or the integrity of the mobile device

- **Post-processing**

The natural frequencies of the system under study are between 27.13 Hz and 386.31 Hz. Below, these frequencies are compared with the operating frequencies of the motorcycle.

It is concluded that there is a risk of resonance at the frequency of 133.33 Hz. The vibration modes for frequencies 1 to 6 are shown in the figure 3.

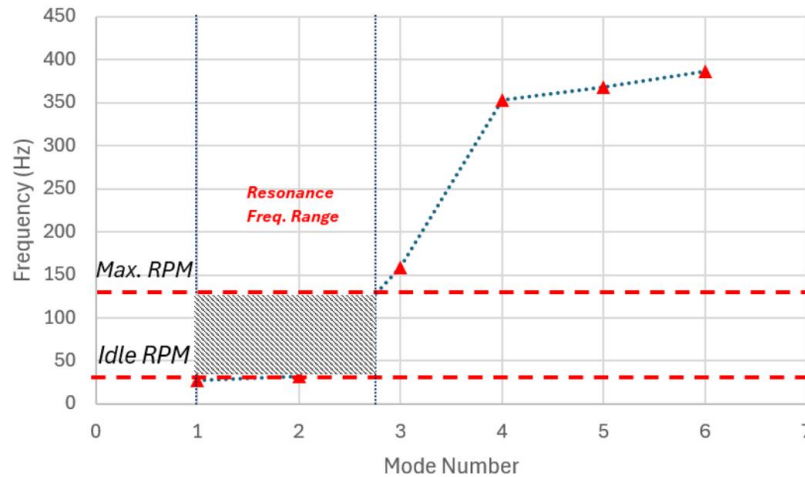


Fig. 3 Representation of the operational and own frequencies of the studied assembly

4. Conclusion and further research directions

The work aimed to evaluate the dynamic behavior of an articulated support intended for mounting a mobile device on the handlebars of a motorcycle, using modal analysis. The geometric model was created in CATIA V5, and the simplified numerical model was developed in ANSYS, using a combination of beam and solid elements, to obtain an optimal compromise between the accuracy of the results and computational efficiency. The analysis allowed the determination of the first natural frequencies and the eigenmodes of vibration of the structural assembly, highlighting the dynamic behavior of the support under conditions of rigid fixation on the handlebar. The comparison of the eigenfrequencies obtained with the excitation range generated by the operation of the motorcycle engine indicated the possibility of the occurrence of the resonance phenomenon in certain operating modes. The results confirm the need for dynamic evaluation of these systems from the design phase, as excessive vibrations can affect both the stability of the mobile device and the durability of the mechanical assembly. To reduce this risk, it is recommended to optimize the structural rigidity, modify the geometric configuration or integrate vibration damping solutions.

5. Bibliography

- [1] Mazilu, T., A Dynamic Model for the Impact Between the Wheel Flat and Rail, U.P.B. Sci. Bull., Series D, Vol. 69, No. 2, 2007;
- [2] Mazilu, T., The Rail's Response to the Action of Vertical Sliding Force, U.P.B. Sci. Bull., Series D, Vol. 68, No. 2, 2006;
- [3] Benmansour, S.A., Benmoussat, A., El Mahi, A., Bourdim, A., Static and Dynamic Characterization of a Woven Eco-Composite: Young's Modulus and Acoustic Emission Analysis, U.P.B. Sci. Bull., Series D, Vol. 82, Iss. 1, 2020;
- [4].Zheng, Q.C., Lu, K.J., Zhu, P.H., A Novel Lightweight Optimization Design of Gear Shaper, U.P.B. Sci. Bull., Series A, Vol. 81, Iss. 4, 2019;

- [5]. Nicolescu, A. et al., The Influence of Total Elastic Displacement of Translation Joints on Gantry Robot Accuracy, U.P.B. Sci. Bull., Vol. 6, Iss. 4, 2011;
- [6]. Munteanu, G., Analysis of the Accuracy Parameters for the Articulated Arm Industrial Robots Using Modern Instruments and Software, U.P.B. Sci. Bull., Series D, Vol. 72, Iss. 4, 2010

TRANSIENT FEA ANALYSIS OF A CENTERING SYSTEM FOR CONTAINERS WITH GRANULAR PRODUCT

STAN-IACOB Adrian, Tudor-George ALEXANDRU and Adrian Florin NICOLESCU

Faculty: Industrial Engineering and Robotics, Specialization: Robotics, 4th Year-mail:
staniacobadrian3@gmail.com

REZUMAT:

This study represents a transient analysis of a centering system of carton boxes or buckets with building materials in computer assisted engineering software ANSYS. The scope for the project was to determine whether the tensions and displacements in structure of the jaws of the system result in any placing errors and jeopardize the robotic palletizing cell in which is implemented. To study the stress in the jaws, they needed to be transformed from 3D in 2D and afterwards be discretized in quadrilaterals. Since the shape of the system is relatively simple, the discretization was satisfactory only by adjusting the number of elements. The tensions and displacements were evaluated by the software which confirm the positioning accuracy, even resulted that the system is a bit oversized and the thickness of the plates of the jaws could be lowered.

KEYWORDS: ANSYS, centering system, palletizing, analysis, transient regime

1. Introduction

The present project proposes the analysis of a centering system for containers with construction materials used within a robotic palletizing application. The purpose of the study was the transient finite element analysis of the system in order to determine the equivalent stresses and displacements that could influence positioning accuracy. Therefore, the study aimed to determine whether the jaw thickness is appropriate or if dimensional modifications could be applied.

The analysis was performed using the computer-aided engineering software ANSYS.

The first stage consisted of preprocessing the assembly by removing fastening elements, simplifying geometries where necessary, merging components that were not relevant for the analysis and converting the jaws from 3D to 2D geometries.

After solving the model, the post-processing stage allowed the extraction of relevant data and the formulation of conclusions regarding the jaw dimensions. Although the results indicated that the analyzed elements were oversized, no modifications were applied to the assembly.

2. Current State of Research

Robotic palletizing is frequently used in industry for fast and accurate product handling, especially when products have large masses or shapes requiring proper guidance and positioning. In the case of containers with construction materials, centering systems ensure product alignment before gripping or pallet placement. Therefore, jaw rigidity and deformation directly influence positioning accuracy.

Paper [1] presents the structural and manufacturing optimization of a robotic end-effector intended for palletizing operations. The author focuses on obtaining a reliable and low-cost gripper capable of handling cylindrical components. The study is closely related to the present project because it investigates the structural behavior of robotic elements in direct contact with handled products.

Paper [2] analyzes the kinematic, static and dynamic behavior of a robotic gripper mechanism. The authors use finite element analysis to determine equivalent stresses and deformations in multiple working positions. This approach is similar to the one used in the present project, where the centering system is analyzed under transient conditions in order to determine jaw stresses and displacements.

Paper [3] presents a gripper designed for pick-and-place operations in the fertilizer industry, intended for handling heavy bags. The modeling was performed in Autodesk Inventor, while the load simulations were carried out in ANSYS. The study highlights the importance of numerical verification for components directly interacting with the handled product.

Paper [4] investigates positioning errors of a robotic gripper by considering elastic deformations, geometric errors and dynamic effects. The study demonstrates that component rigidity plays an essential role in positioning accuracy, which is also important for the centering system analyzed in this project.

TRANSIENT FEA ANALYSIS OF A CENTERING SYSTEM FOR CONTAINERS WITH GRANULAR PRODUCT

Based on these studies, finite element analysis can be considered an important method for validating robotic handling and centering systems. In general, the analyzed parameters include von Mises equivalent stresses, total displacements, safety factors and the influence of deformations on positioning accuracy. Therefore, the ANSYS analysis of the container centering system allows proper evaluation of jaw thickness and dimensional optimization possibilities.

3. Pre-processing

In order to prepare the assembly for analysis, it was necessary to simplify the geometry in ANSYS DesignModeler. Fastening components such as bolts were removed, after which missing or improperly imported geometries were recreated.

The guide rails were reconstructed using the “Extrude” command applied to the rail profile along its length, followed by the deletion of unnecessary surfaces.

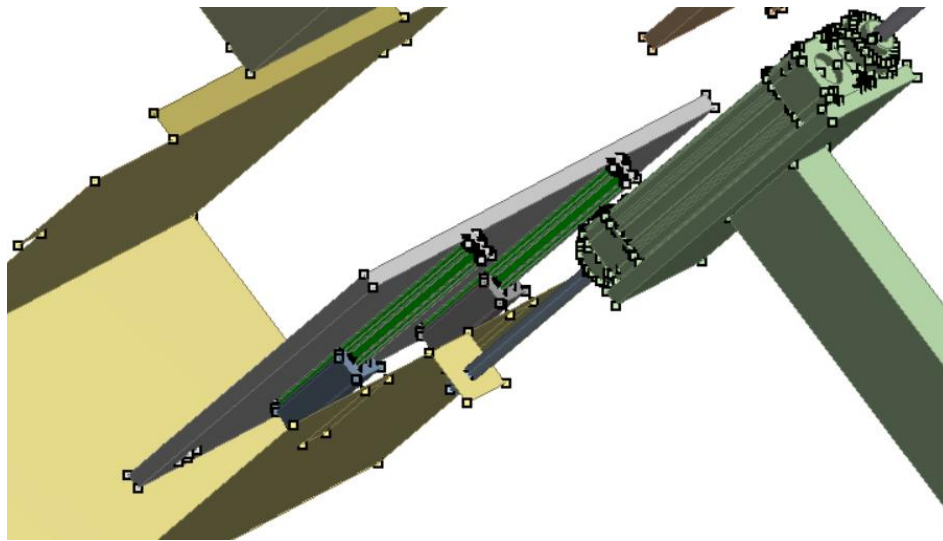


Fig. 7 Extrude command for rail correction

To simplify the definition of kinematic joints and contacts, the “Named Selection” command was used. The surfaces involved in joints and contacts were named according to the following convention:

- Tn_fix or mobile for translational joints, where n represents the joint number;
- Fn_fix or mobile for fixed joints;
- Cn_fix or mobile for contacts.

In order to define “Frictional” contacts between the jaws and the box, additional surfaces had to be created and later hidden during simulation. These surfaces were then imprinted onto the box geometry.

TRANSIENT FEA ANALYSIS OF A CENTERING SYSTEM FOR CONTAINERS WITH GRANULAR PRODUCT

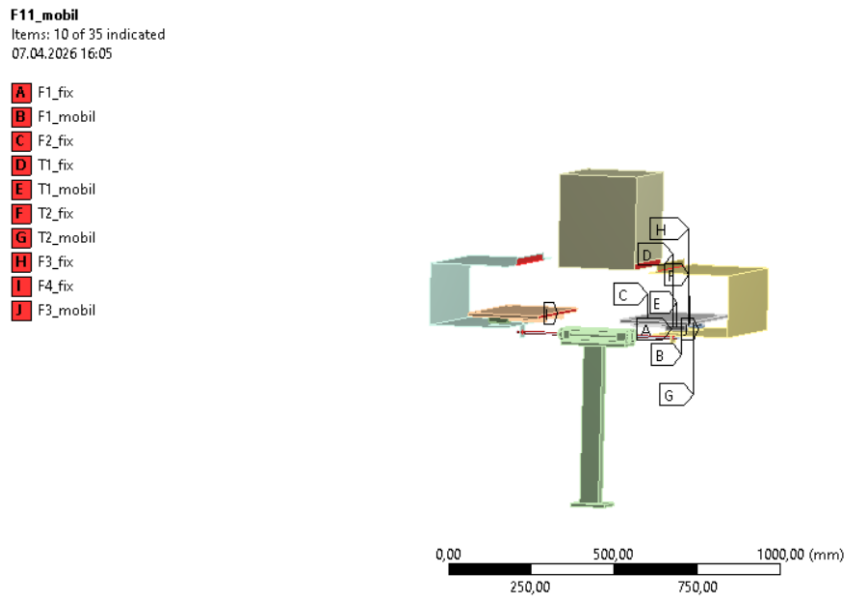


Fig. 8 All the surfaces that will be part of joints or contacts

Regarding the materials used, it was first necessary to identify the relevant flexible components. All components except the jaws were considered rigid. The box also had to be considered in order to accurately reproduce its real mass. Structural steel was assigned to all rigid components except the box. For the jaws, the aluminum alloy AlMg4.5Mn (5083 aluminum alloy) was selected from the ANSYS material library to match the real material used. For the boxes, a custom material was created based on the calculated density.

$$V = 240 \cdot 300 \cdot 300 = 21,6 \cdot 10^4 \quad (1)$$

$$\rho = \frac{m}{V} = \frac{22kg}{21,6 \cdot 10^4 mm^3} = 7850 \frac{kg}{m^3} \quad (2)$$

Tabel 1 Used Materials in ANSYS

Material	Alloy	Density	Yield strength
Aluminum	AlMg4.5Mn	2770 kg m ³	280 MPa
Structural Steel	OL43	7850 kg m ³	250 MPa
Box	Construction material	1091,3 kg m ³	/

To generate the motion diagrams of the linear pneumatic actuators, their 100 mm stroke and the required 2-second closing time were considered. A Python script was used to calculate the instantaneous velocities at 0.1-second intervals, resulting in the motion diagram used within the simulation.

TRANSIENT FEA ANALYSIS OF A CENTERING SYSTEM FOR CONTAINERS WITH GRANULAR PRODUCT

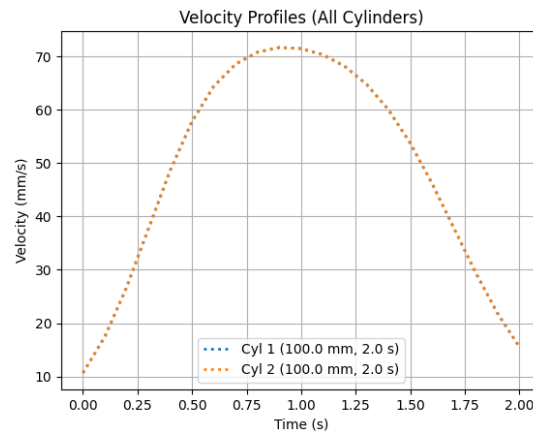


Fig. 9 Motion diagram for pneumatic actuators

4. Solving

The force convergence diagram generated during the simulation indicates the convergence behavior of the numerical solution. The blue line represents the sum of reaction forces, while the purple line corresponds to the calculations performed by the software. Each variation in the purple line represents an iteration.

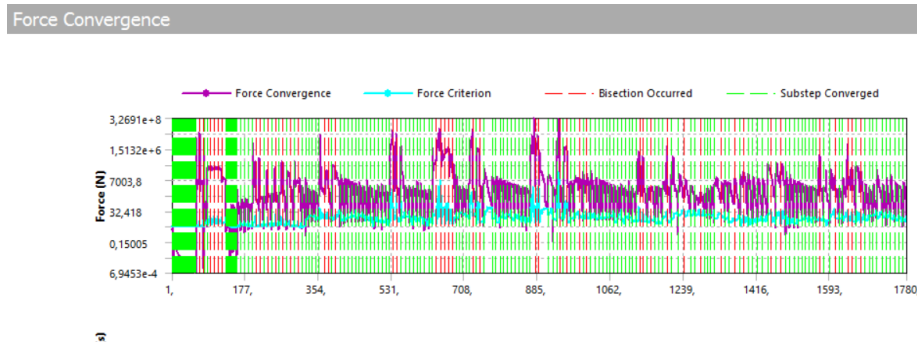


Fig. 10 Force Convergence Diagram

The vertical green lines indicate successful convergence moments, while the red lines represent time-step bisections automatically generated by the software in order to improve convergence using smaller time increments.

By analyzing the diagram, the impact moments between the box and the jaws can be identified through the concentration of red vertical lines. The most important contact events occur around 1.3 seconds and 1.7 seconds, corresponding to the impacts between the box and the first jaw, respectively the second jaw.

5. Post-processing

The displacement diagrams obtained for both jaws were identical, meaning that only one diagram had to be analyzed.

After exporting the displacement values for each time step, a Python script was created in order to generate higher-resolution plots compared to Microsoft Excel.

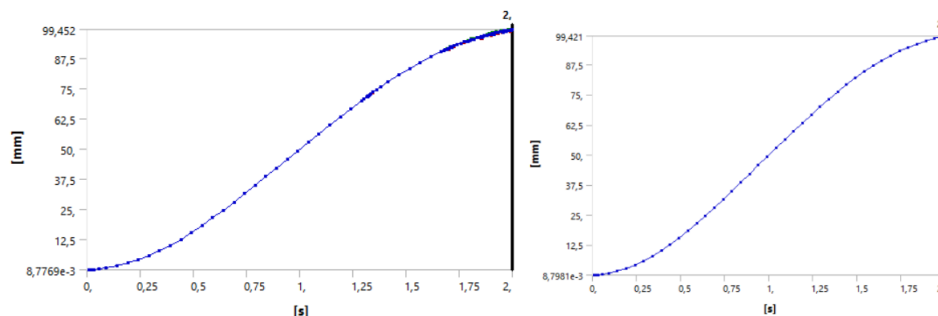


Fig. 11 Displacement diagram for the pushing jaw, real vs. ideal

TRANSIENT FEA ANALYSIS OF A CENTERING SYSTEM FOR CONTAINERS WITH GRANULAR PRODUCT

The stress analysis indicated that the maximum equivalent stress occurred in the bent region of the jaw plate, reaching approximately 33.669 MPa around 1.85 seconds.

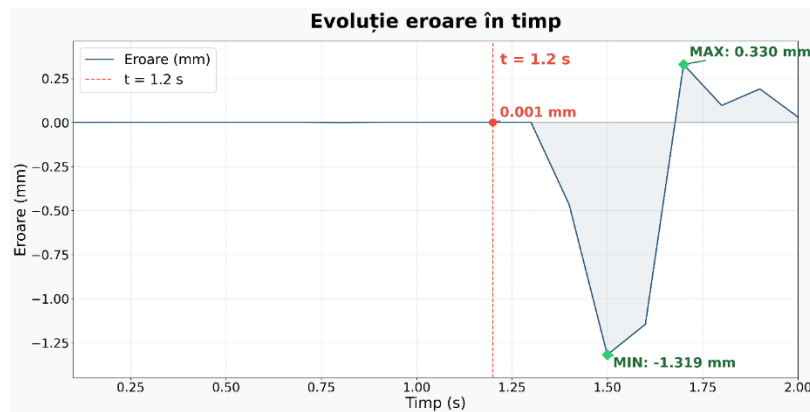


Fig. 12 Error diagram for displacement of the pushing jaw

The maximum displacement obtained during the simulation was approximately 1.319 mm. However, this displacement was elastic, and the final displacement value decreased below 0.1 mm after the motion ended.

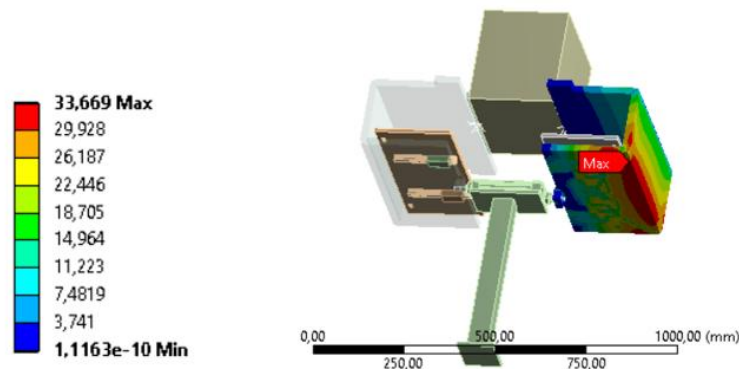


Fig. 13 The place of the highest equivalent stress

The resulting safety factor can be computed as:

$$V = 240 \cdot 300 \cdot 300 = 21,6 \cdot 10^4 \quad (1)$$

6. Conclusions

The analysis started from the 3D model of the centering system intended for boxes containing granular products. The imported STEP assembly required geometry refinement inside ANSYS, including repairing surface bodies, recreating missing geometries and merging elements in order to simplify the analysis process.

The motion diagrams of the two linear pneumatic actuators were generated using a Python script, considering a 100 mm stroke and a total motion duration of 2 seconds.

The final solution indicated a maximum displacement of 1.319 mm, although this deformation was elastic and decreased below 0.1 mm at the end of the simulation. The maximum equivalent stress reached 33.669 MPa, resulting in a safety factor of approximately 8.

Such deformations can be considered acceptable because the required positioning accuracy is

approximately 0.5 mm. Furthermore, the robotic end-effector uses additional gripping jaws equipped with suction cups that provide secondary positioning and prevent box damage during handling. Additionally, the current 20 mm plate thickness makes conventional bending manufacturing processes difficult, indicating that dimensional optimization could be considered in future developments.

7. Bibliography

[1]. Anastasiu, A.I. (2023), “Structural and fabrication process optimization for a robot effector”, Student Scientific Communications Session 2023, Faculty of Industrial Engineering and Robotics, National University of Science and Technology Politehnica Bucharest, Bucharest;

[2]. Badadhe, S., Singh, A. P., Roy, S. and Tiwari, M. (2018), “FEM Based Dynamic Analysis Of Robot End Gripper Mechanism”, International Journal of Engineering Science Invention, Invention Journals, published on December 3, 2018;

[3]. Sumbodo, W., Pambudi, R. and Setiadi, R. (2025), “Design and Analysis of Pick and Place Robot Gripper in the Fertilizer Industry”, Reports in Mechanical Engineering, 6(2), pp. 53-64, ISSN 2683-5894;

[4]. Dmytriv, V., Dmytriv, I., Horodetskyi, I., Hutsol, T., Kukharets, S., Cesna, J., Bleizgys, R., Pietruszynska, M., Parafiniuk, S. and Kubon, M. (2024), “A Method for Simulating the Positioning Errors of a Robot Gripper”, Applied Sciences, 14(14), 6159, EISSN 2076-3417;

Transient FEA ANALYSIS of a Box-Pushing System Integrated into a Palletizing Application

CRAIOVEANU David, ALEXANDRU Tudor-George

Faculty: IIR, Specialization: Robotics, Year of study: IV, e-mail: davicraioveanu@gmail.com

***ABSTRACT:** This paper presents the structural and dynamic validation of a pneumatic pusher system used for deflecting or positioning objects on an industrial conveyor belt. The primary objective is to evaluate the transient mechanical behavior, structural integrity, and stress distribution of the flexible components during impact. The numerical methodology, implemented in Ansys Workbench (Transient Structural), utilizes a hybrid modeling approach: mid-surface shell elements for the support frame and the pusher plate, and solid elements for the pneumatic cylinder casing and rod. Kinematic joints (translational and fixed) were established to accurately replicate the physical constraints, while a displacement-based joint load controlled the actuation cycle. The numerical results post-processed from the simulation indicated that the equivalent Von Mises stresses and structural deformations remain significantly below the material's yield strength. The simulation confirms the structural viability of the design under dynamic operational conditions without requiring geometric modifications.*

***KEYWORDS:** Transient Structural, Ansys Workbench, Mid-Surface, Pneumatic Pusher, Joint Displacement, Shell-Solid Interaction.*

1. Introduction

In the context of production line automation and modern logistics flows, the rapid and accurate handling of products on conveyor belts is a critical requirement. Pusher-type diverter and sorting systems (mechanical pushers) are widely used due to their short response time and high reliability. However, during the dynamic impact between the pusher plate and an object moving on the conveyor, transient stresses and vibrations are generated within the structure, which may affect system accuracy or lead to material fatigue.

The aim of this study is to numerically validate, through the Finite Element Method (FEM), a pneumatically actuated pushing assembly. The transient analysis seeks to determine the contour maps of equivalent stresses and elastic deformations in the most highly loaded components, identified as the mounting bracket and the front impact plate. The adopted methodology includes the simplification of the three-dimensional CAD model, the conversion of thin-walled components into mid-surfaces, the proper definition of kinematic joints, and the implementation of the cylinder rod motion profile.

2. Literature overview

The dynamic optimization of peripheral components in industrial conveying systems has become the subject of extensive research aimed at increasing operational throughput and reducing energy consumption. In the scientific literature, the analysis of flexible systems subjected to impact loading is predominantly addressed through nonlinear dynamic simulations. Recent studies indicate that the use of rigid-body dynamics (RBD) models provides only kinematic information, such as velocities and accelerations, making the adoption of flexible-body models essential for the assessment of structural integrity and mechanical strength.

A common challenge in computer-aided engineering is the efficient management of computational hardware resources. The application of hybrid solid-shell discretization techniques (volume-surface modeling) has proven to be an optimal approach for structures that combine thin-walled profiles with massive components. The present work follows this research direction by proposing a computational methodology that reduces simulation runtime in ANSYS Workbench while maintaining a high level of accuracy in the prediction of local stresses within the cylinder-to-structure connection regions involving flexible structural elements.

3. Numerical approach

The evaluation of the structural and kinematic performance of the centering system required the development of an advanced numerical model capable of accurately reproducing the dynamic interactions occurring during operation. This section details the geometry pre-processing stages, the configuration of the transient analysis algorithms, and the interpretation of the contour plots obtained from the simulation results. All stages were carried out using the ANSYS Workbench finite element analysis platform, with the resulting data playing a key role in the validation of the assembly.

- *Pre-processing and solving*

The initial three-dimensional geometry of the assembly was subjected to a rigorous topological simplification (defeaturing) process to remove minor geometric details, such as small fillets, chamfers, and threads, which would have unnecessarily increased the number of finite elements. To ensure high computational efficiency, the components whose thickness was small relative to their other dimensions—namely the mounting bracket and the pusher plate—were converted into mid-surfaces. The pneumatic cylinder housing and piston rod were retained as solid bodies. In ANSYS Mechanical, the structural stiffness behavior of the mounting bracket and pusher plate was defined as Flexible, allowing the calculation of stresses and deformations, whereas the cylinder components were assigned standard elastic material properties.

The connection between the shell elements (surfaces) and the solid elements (volumes) was established in the projection regions through Multi-Point Constraint (MPC) contacts, ensuring the proper transfer of translational and rotational degrees of freedom across the interface.

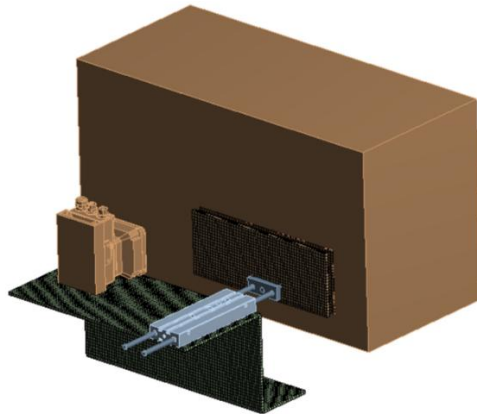


Fig. 1 Details regarding the mesh

The meshing process was performed using a controlled finite element mesh. Face sizing controls were applied to the flexible shell surfaces, with particular attention given to the regions surrounding the rod and cylinder housing projections, where stress concentrations were expected to occur.

The transient dynamic structural analysis (Transient Structural) solves the general equation of motion of the system by considering the mass, stiffness, and damping matrices. The Large Deflection option was enabled to properly account for the geometric nonlinearities induced by the rod motion and the flexural deformation of the plates.

The kinematic constraints were defined through three fundamental joints:

Fixed Joint (Ground-to-Casing): A fixed connection between the pneumatic cylinder housing and the assembly support structure.

Translational Joint (Casing-to-Rod): A translational joint guiding the linear motion of the rod inside the cylinder housing along the actuation axis.

Fixed Joint (Rod-to-Plate): A fixed connection between the rod end and the pusher plate, while preserving the flexible structural behavior of the plate.

The total simulation time was defined according to the pusher stroke (e.g., 1.0 s). The motion was prescribed through a Joint Load condition applied to the translational joint, selecting the Displacement option and defining a tabular displacement profile corresponding to the pneumatic cylinder motion (e.g., from 0 mm to 150 mm during the first 0.4 s, followed by a dwell period and subsequent

Transient FEA ANALYSIS of a Box-Pushing System Integrated into a Palletizing Application

retraction).

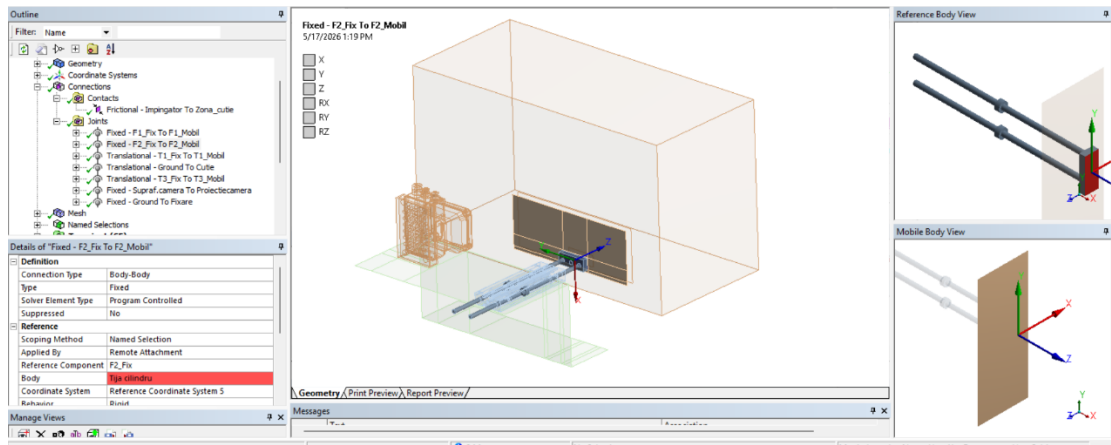


Fig. 2 Configuration of the joints

To ensure numerical convergence during the impact event with the virtual object on the conveyor, the Auto Time Stepping feature was employed, with an initial time step of 1×10^{-3} s and a minimum time step of 1×10^{-5} s.

3.3. Post-processing

Upon completion of the numerical simulation, contour plots of the structural response quantities of interest were extracted, and their evolution was monitored throughout the entire centering cycle. Displacement Analysis: The post-processed results indicate that the total displacement is predominantly governed by the rigid-body motion of the rod and the pusher plate along the actuation axis. To evaluate the actual structural response, the local elastic displacements were analyzed after removing the overall rigid-body motion component. For both the mounting bracket and the pusher plate, the resulting elastic displacements were found to be microscopic, demonstrating the excellent structural stiffness of the selected profiles.

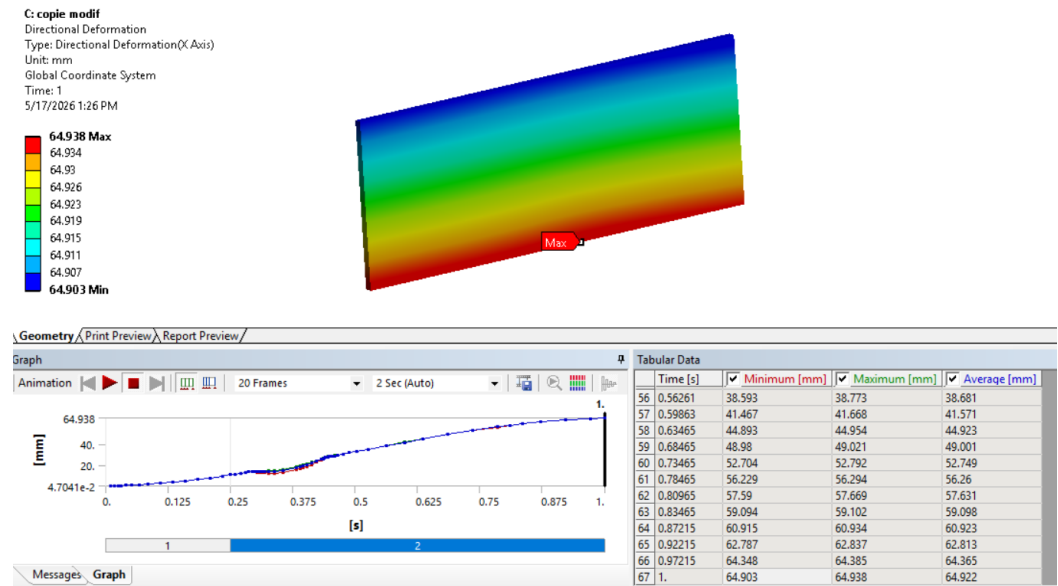


Fig. 3 Total deformation plot of the pusher plate

Transient FEA ANALYSIS of a Box-Pushing System Integrated into a Palletizing Application

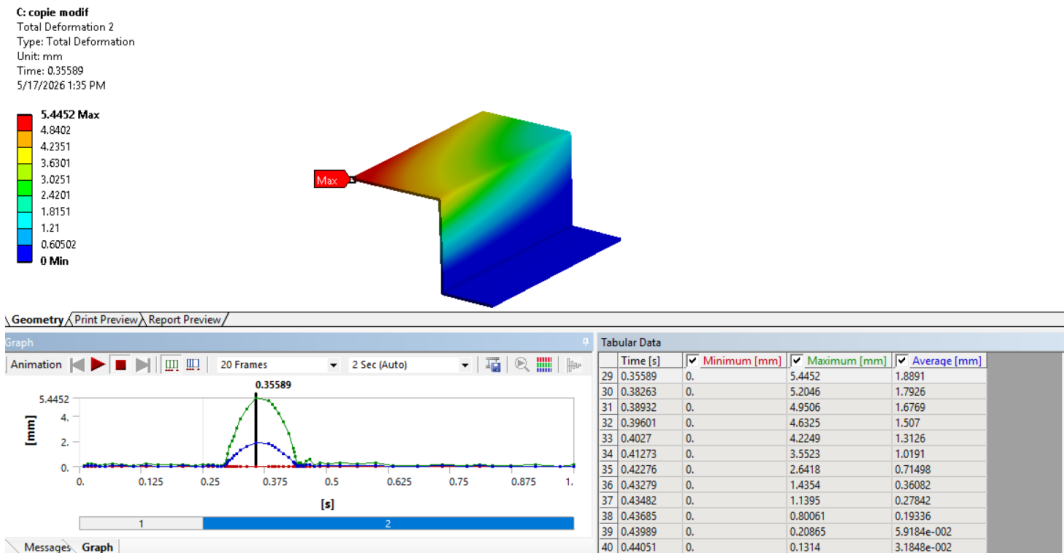


Fig. 4 Total deformation plot of the bracket

Von Mises Equivalent Stress Analysis: The stress contour plots confirm that peak stress values occur transiently at the beginning of the stroke, due to inertial effects, and during the impact event with the load on the conveyor. The maximum stresses are concentrated in the pusher plate, particularly in the projection region where the solid rod transfers the applied force to the flexible shell component, and in the mounting bracket around the cylinder housing attachment area.

All recorded stress levels remain well below the yield strength of the materials used, ensuring a high factor of safety and guaranteeing that the entire assembly operates exclusively within the elastic regime, without any permanent deformation.

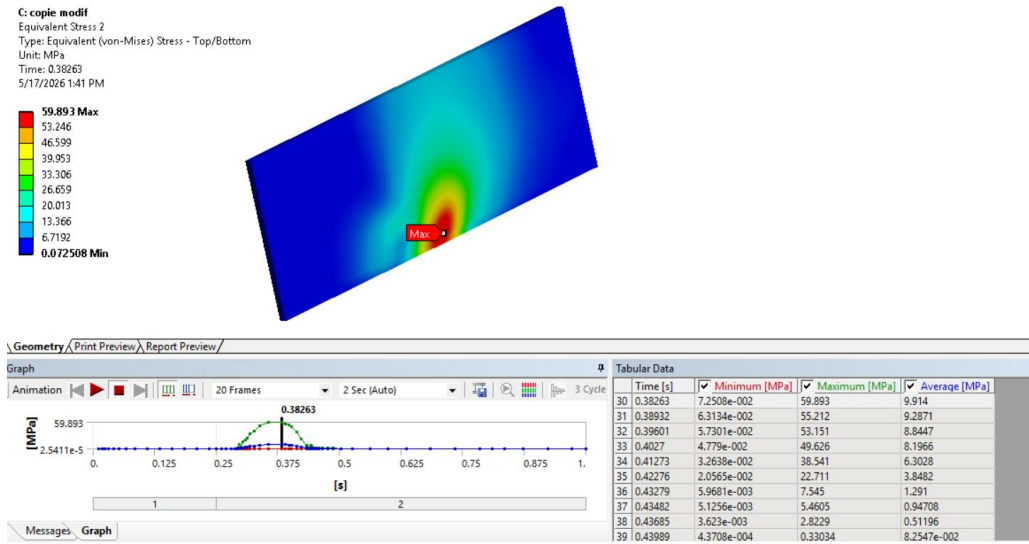


Fig. 5 Stress plot for the pusher plate

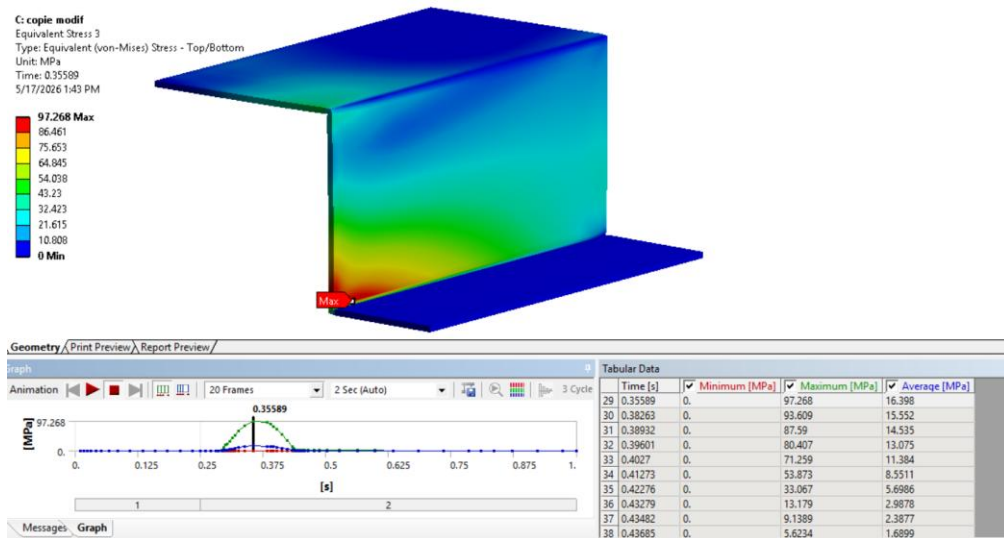


Fig. 6 Stress plot for the bracket

4. Conclusion and further research directions

Through the transient dynamic structural analysis performed in the ANSYS Workbench environment, the mechanical behavior of an industrial pusher system was successfully validated. The hybrid shell–solid modeling approach, combined with the definition of the mounting bracket and pusher plate as flexible bodies, enabled an accurate prediction of the stress state during the critical phases of motion and impact, while simultaneously optimizing computational resources.

The low levels of elastic displacement and Von Mises stress obtained for the flexible components demonstrate that the original design is properly dimensioned, successfully withstands the operational dynamic loads, and does not require structural modifications or material thickening. The proposed configuration is therefore fully suitable for physical implementation.

To further optimize long-term performance and extend the service life of the equipment under intensive operating conditions, the following recommendations are proposed:

1. Selection of a pneumatic cylinder with adjustable end-of-stroke cushioning, in order to reduce the initial inertial shock transmitted to the pusher plate during actuation.
2. Installation of a protective polymer layer (e.g., polyurethane or elastomer) on the active face of the pusher plate. This layer would absorb micro-impacts occurring during direct contact with conveyed objects and protect the surface against abrasive mechanical wear.

5. Bibliography

- [1]. Bathe, K. J. (2014), “Finite Element Procedures for Nonlinear Dynamic Analysis”, 2nd ed., ADINA R&D Inc., Watertown, MA, USA, ISBN 978-0979004902.
- [2]. Belytschko, T., Liu, W. K., Moran, B., & Elkhodary, K. (2014), “Nonlinear Finite Elements for Continua and Structures”, 2nd ed., John Wiley & Sons, Hoboken, NJ, USA, ISBN 978-1118632703.
- [3]. Pupăză, C., & Parpală, R. C. (2013), “Modelare CAD-FEM”, Politehnica Press, București, ISBN 978-6065154414.
- [4]. Pupăză, C., & Parpală, R. C. (2011), “Modelare și analiză structurală cu ANSYS Workbench”, Politehnica Press, București.
- [5]. Rao, S. S. (2017), “The Finite Element Method in Engineering”, 6th ed., Butterworth-Heinemann, Oxford, U.K., ISBN 978-0128143643.
- [6]. Structural Dynamics and Materials Conference (2021), “Transient Dynamic Analysis of Flexible Mechanical Systems Using Finite Element Methods”, Journal of Physics: Conference Series, Vol. 1780, No. 1, ISSN 1742-6596.

Structural validation of a palletizing end-effector through transient MEF analysis

PĂUNESCU Adrian-Georgian¹ and ALEXANDRU Tudor-George

¹Faculty: IIR, Specialization: Robotics, Year of study: IV, e-mail: paunescugeorgian@gmail.com

SUMMARY: This paper presents the structural validation of a multifunctional end-effector used in a robotic palletizing application for motor oil boxes produced by ARDECA Lubricants. The study aimed to evaluate the structural behavior of the end-effector using transient finite element analysis.

The geometry was modeled in CATIA, verified in Siemens NX, and imported into ANSYS for simulation. The analysis evaluated equivalent stresses and displacements in the critical structural components. The results confirmed that the stresses remained within the allowable limits of the EN AW-6082 aluminum alloy, validating the proposed structural solution.

KEYWORDS: End-effector, ANSYS, CATIA, NX

1. Introduction

The paper presents the structural validation of an end-effector used in a robotic application for palletizing boxes of motor oil cans from the ARDECA Lubricants company. The analyzed end-effector is a multifunctional one, which integrates functionalities specific to the "Clamp style", "Fork style" and vacuumatic effectors, having the role of manipulating and transporting the boxes within the palletizing process.

The aim of the project was to validate the structural design of the end-effector through a transient finite element analysis, aiming to evaluate the equivalent stresses and displacements during operation. The analysis was carried out for areas considered structurally critical, such as the upper support structure and the claws used for handling boxes.

To carry out the project, the following stages were followed: importing and simplifying the geometry, defining the kinematic couplings, discretizing the model, defining the motion cyclograms, applying the simulation conditions, solving the model and interpreting the results obtained in the post-processing stage. Following the analyses performed, the displacements and stresses in the analyzed structure were evaluated, without the need for additional constructive modifications.

2. Current status

In the specialized literature there are numerous researches oriented towards the design, simulation and structural validation of robotic systems used in palletizing and automatic handling applications. Special attention is paid to end-effectors, since they represent the element in direct contact with the manipulated product and are subjected to important mechanical stresses during the work cycles. In the paper [5], the numerical analysis of the load-bearing structure of a universal head used for palletizing industrial bags is presented. The study uses the finite element method to reproduce real operating conditions and evaluate the structural behavior of the assembly during the handling process. The results obtained allowed the identification of critical areas and the verification of the strength of the structure in relation to the applied stresses. Similarly, in the paper [6], the strength of a gripper intended for a palletizing robot for VDA 4500 containers is analyzed. The authors performed the load calculation, the choice of pneumatic elements and the strength analysis of the main components of the effector.

Based on the approaches presented in the literature, this paper presents the structural validation of a multifunctional end-effector used in a robotic palletizing application for oil cans. The analysis is performed in transient mode using the finite element method, tracking the equivalent stresses and resulting displacements in critical areas of the load-bearing structure.

3. Presentation of the effector and flexible bodies

As mentioned in the introductory chapter, the end-effector analyzed in the paper is used in a robotic application for palletizing boxes with engine oil cans from the ARDECA Lubricants company. It is a multifunctional effector, which integrates functionalities specific to the "Clamp

Structural validation of a palletizing end-effector through transient MEF analysis

style", "Fork style" and vacuum effectors, having the role of manipulating and transporting the boxes during the palletizing process. The effector can be seen below.

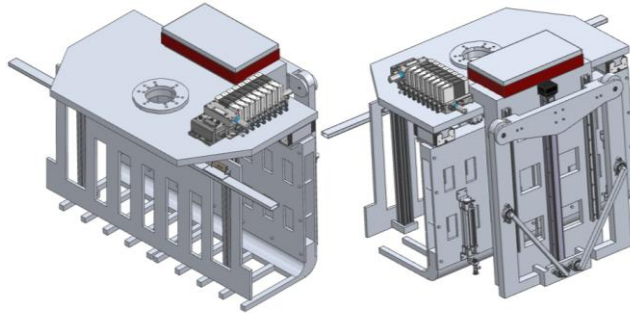


Fig. 1. The studied effector

Two main flexible bodies were considered in the structural analysis, represented by the claws used to support the boxes and the upper structural element called the “ceiling”. These components are made of EN AW-6082 aluminum alloy and were selected to evaluate the equivalent stresses and displacements occurring during the operating cycle.

The following figure presents the three-dimensional models of the analyzed flexible bodies.

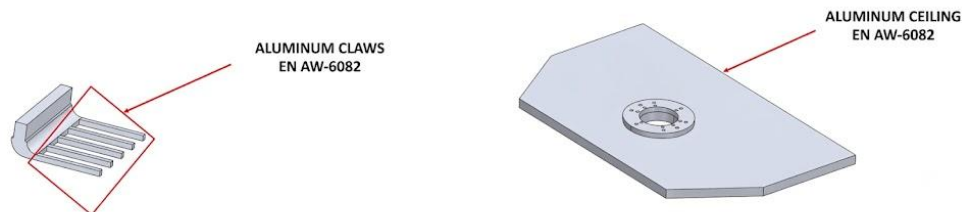


Fig. 2. Flexible bodies

To define the mechanical properties of the claws, the EN 755-2 standard was consulted, corresponding to extruded profiles and bars made of aluminum alloys, and the 20×20 mm section was selected. In the case of the upper structural element, the mechanical properties defined in the EN 485-2 standard for 15 mm thick aluminum sheets and plates were used.

The mechanical characteristics used in the analysis for both components are:

- Tensile strength R_m : 295 MPa
- Yield strength $R_{p0.2}$: 250 MPa
- Elastic modulus E : 70 GPa
- Density: 2700 kg/m^3

4. Pre-processing

4.1. Importing geometry

The geometry of the end-effector, presented previously, was created in the CATIA CAD environment based on the estimated dimensions from the video materials related to the analyzed application, using proportional relationships reported to known dimensions. After completing the modeling, the assembly was exported in STEP format and checked in the Siemens NX program to validate the correctness of the geometric transfer between different software applications. Following the verification, it was imported into Ansys.

4.2. Simplification of geometry

Structural validation of a palletizing end-effector through transient MEF analysis

After importing the assembly into ANSYS, the geometry simplification step was performed, necessary to reduce the complexity of the numerical model and streamline the simulation process. In this step, components that do not significantly influence the results of the structural analysis, such as bolts, nuts, washers, bearings or caps, were eliminated, using "Body Delete" operations. Also, certain subassemblies were joined by Boolean operations of the "Unite" type, in order to reduce the number of bodies and simplify the contacts between components. For the flexible bodies analyzed, dimensional simplifications were also applied, by switching from three-dimensional modeling to 2D and 1D approaches. Thus, the upper structural element called "ceiling" was converted into a surface representation, and the claws were remodeled using linear elements with equivalent rectangular section.

The following figure presents the main operations used in the process of simplifying the geometry of the analyzed model.

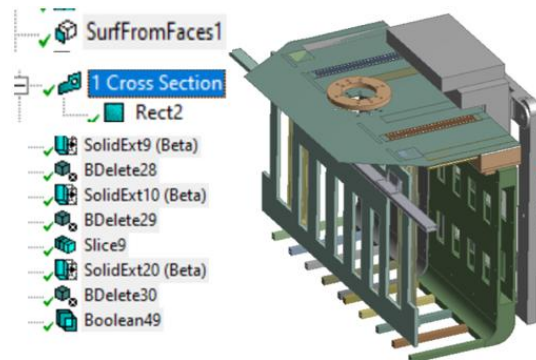


Fig. 3. Simplified geometry

4.3. Definition of kinematic couplings

The kinematic couples were defined in the Design Modeler interface of ANSYS and subsequently assigned in the Mechanical environment, as shown in the image below.

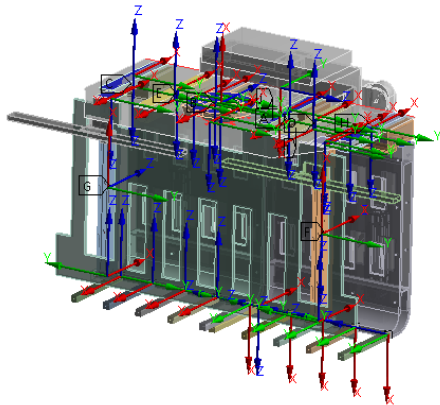


Fig. 4. Defined kinematic couples

4.4. Performing discretization

The element that was discretized was the ceiling, for which "Face Meshing" was used. Below we will present how the discretization resulted.

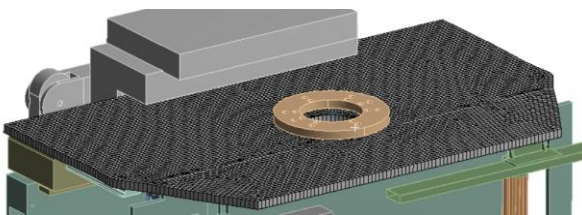


Fig. 5. Discretization of the "ceiling"

4.5. Definition of motion cyclograms

Four pneumatic cylinders are used in the project: two for the translation of the jaws and two for the actuation of the presser feet. Due to the symmetrical configuration of the effector, two distinct motion cyclograms were defined. To generate them, a Python script was created, used to determine the speed variation as a function of time based on the stroke, actuation time and time step of the cylinders.

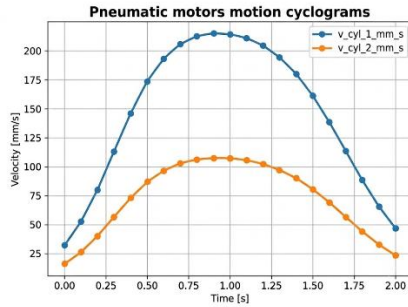


Fig. 6. Cyclograms provided by Python code

4.6. Application of simulation conditions

The following conditions were defined: the gravitational force generated by the manipulated boxes, the movement cyclograms corresponding to the controlled axes, as well as the gravitational acceleration applied during the lifting process, together with the lifting sequence.

5. Solution

Structural transient analysis is based on the general equation of motion for dynamic systems with multiple degrees of freedom, obtained by discretizing the structure using the finite element method and applying the principle of dynamic equilibrium [1][2]. This relationship expresses the balance between inertial forces, damping forces, elastic forces, and time-dependent external loads:

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{F(t)\} \quad (5.1.)$$

Where:

- $[M]$ - The mass matrix that describes the inertial properties of the structure
- $[C]$ - Damping matrix that models mechanical energy dissipation
- $[K]$ - The stiffness matrix that characterizes the elastic behavior of the material
- $\{u\}$ - The displacement vector that represents the main unknowns, and its derivative in time leads to velocities and accelerations
- $\{F(t)\}$ - Vector of time-dependent loads

6. Post-processing

6.1. Total displacement graph

The graph of total displacements is presented below, the maximum value recorded over the entire sequence being 474.5 mm.

Structural validation of a palletizing end-effector through transient MEF analysis

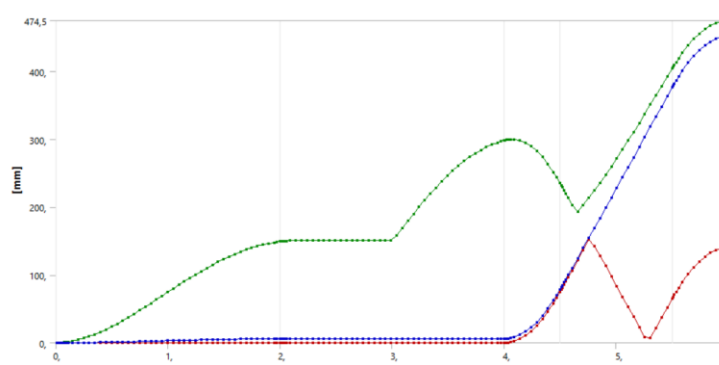


Fig. 7. Displacement graph

6.2. Calculation of position deviations

After presenting the graph of total displacements, the position deviations were calculated, and the graph presented below highlights the fact that the maximum deviation obtained exceeds the nominal accuracy of the robot.

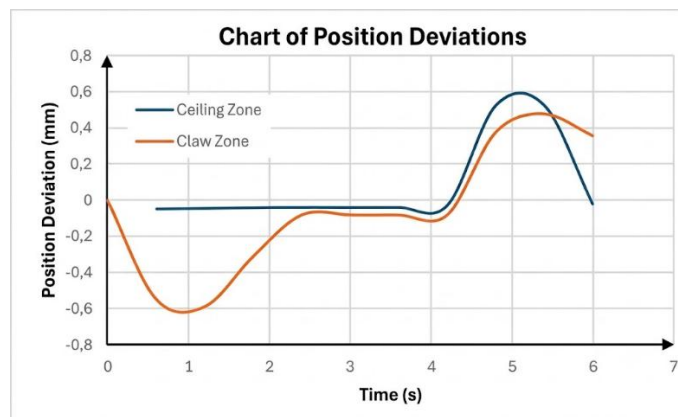


Fig. 8. Position deviation graph

6.3. Equivalent stresses graph

The analysis also evaluated the equivalent stresses, the maximum value obtained being 42.247 MPa. Considering that the maximum allowable stress is approximately 250 MPa, according to the characteristics of the EN AW 60682 material, it can be stated that the result falls within optimal operating parameters.

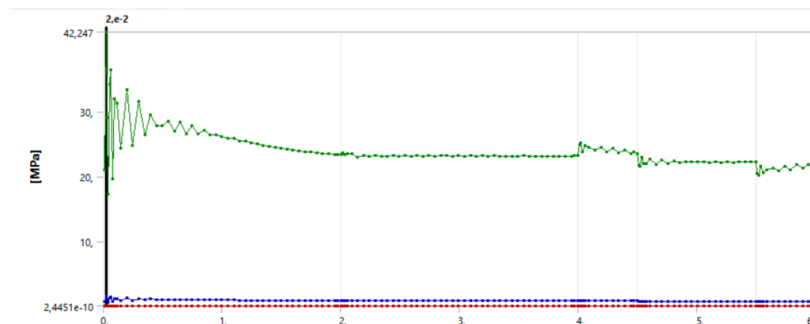


Fig. 9. Equivalent stresses graph

6.4. Bar stress graph

As can be seen from the graph below, the maximum stress value is -6.5 MPa.

Structural validation of a palletizing end-effector through transient MEF analysis

Considering that the tensile strength limit for the EN AW-6082 material is approximately 295 MPa, it can be stated that the result falls within optimal operating parameters.

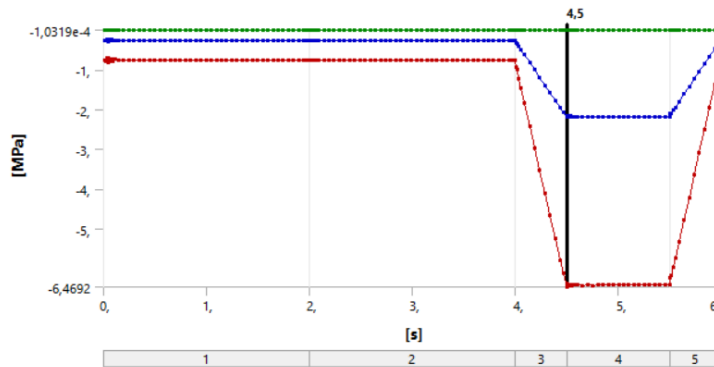


Fig. 10. Graph of stresses occurring in bars

7. Conclusions

The structural analysis performed highlighted that the stress values obtained are within the permissible limits of the material used, confirming the validity of the proposed constructive solution. The results obtained also demonstrated the appropriate behavior of the analyzed structure under the operating conditions imposed on the palletizing application.

8. Bibliography

[1]. Bathe, KJ (1996), *Finite Element Procedures*, Prentice Hall, Upper Saddle River, NJ, ISBN 0-13-301458-4.

[2]. Zienkiewicz, OC and Taylor, RL (2005), *The Finite Element Method: Its Basis and Fundamentals*, Butterworth-Heinemann, Oxford, ISBN 978-0750663205.

[3]. *** CEN (2008), *EN 485-2: Aluminum and aluminum alloys – Sheet, strip and plate*, European Committee for Standardization.

[4]. *** CEN (2008), *EN 755-2: Aluminum and aluminum alloys – Extruded rod/bar/tube*, European Committee for Standardization.

[5]. Chodurski, M., Dębski, H., Samborski, S., Teter, A. (2015), "Numerical strength analysis of the load-bearing frame of a palletizing robot's universal head", *Eksploracja i Nyatydosc – Maintenance and Reliability*, DOI: 10.17531/ein.2015.3.7.

[6]. Urbonas, K., Rimovskis, S., Sabaliauskas, A. (2021), "Analysis of Strength of VDA 4500 Type Container Palletizing Robot Manipulator Gripper", *Journal of Young Scientists*, DOI: 10.15388/JMD.2021.11.

Simulation of the RESISTANCE SPOT welding process based on FEM coupled analyses

PEREPELEA Iulia and ALEXANDRU Tudor-George

Faculty: IIR, Specialization: Robotics, Year of study: IV, e-mail: iuliaperpelea@yahoo.com

SUMMARY: This paper presents the thermal and structural simulation of a resistance spot welding process applied to a Body in White automotive component. The studied assembly consists of two overlapping DP600 steel sheets welded using a C-type spot welding end-effector. The numerical model was prepared by simplifying the CAD geometry, defining the contact regions, generating the finite element mesh and applying transient thermal and structural boundary conditions. Temperature curves were generated in Python and introduced into ANSYS as time-dependent thermal loads. The results showed that the maximum temperature is concentrated in the weld nugget area, while the mechanical deformations remain localized near the electrode contact zone. The electrode axis remains approximately aligned, and the equivalent stresses do not exceed the material yield strength.

KEYWORDS: spot welding; DP600; thermal analysis; structural analysis; finite element method

1. Introduction

Spot pressure welding is one of the most widely used joining technologies in the automotive industry, especially for the production of Body in White structures.

The objective of this work is to simulate the spot pressure welding process applied to an assembly in the front area of the car body. The analyzed reference is part of the basic structure of the vehicle.

The aim of the work is to perform a coupled thermal-structural analysis, through which the effects of the welding process on the temperature distribution, displacements and equivalent stresses are monitored. The main steps consisted of remodeling and simplifying the geometry, defining the welding parameters, discretizing the model, generating temperature curves, performing the transient thermal analysis and using the thermal results as input data for the transient structural analysis.

2. Current status

Simulation of the spot welding process is used to understand the thermal, mechanical and electrical phenomena that occur in the contact area between electrodes and sheets and involve rapid local heating, heat transfer, mechanical pressure, variation of material properties with temperature and formation of the welded core.

In the paper [1], a study of spot welding for DP600 is presented, using a coupled thermo-mechanical analysis by the finite element method. The influence of welding parameters on the properties of the molten zone and the welded joint is investigated.

Another relevant study [2] develops a numerical thermal-electrical-mechanical-metallurgical model for the spot welding of two DP600 sheets. The simulation results were compared with experimental measurements of the weld core size, residual stresses and hardness.

In [3], a sequential electrical-thermal-metallurgical and mechanical numerical model was proposed for spot welding simulation. The study emphasizes the importance of contact conditions, electrical contact resistance and fine discretization in the interface area between the sheets, as these aspects directly influence the formation of the welded core.

Compared to these works, the present study aims to apply thermal and structural analysis to a simplified car body assembly, including both welded sheets and electrodes of a C-type welding effector. Temperature curves are generated through a Python program and subsequently introduced into ANSYS, to allow the correlation of thermal effects with the structural response of the assembly.

3. Model pre-processing

3.1. Geometric model

The geometric model of the assembly was created starting from a frontal area of the car body, adapted for the analysis of the spot welding process. The reference was adapted and simplified in Siemens

Simulation of the RESISTANCE SPOT welding process based on FEM coupled analyses

NX, keeping only the relevant area for the contact between the electrodes and the sheets.

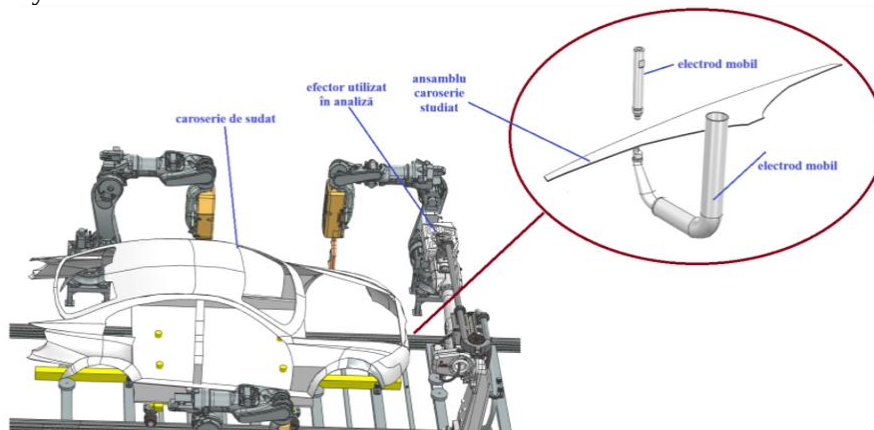


Fig. 1. The studied ensemble

The welding effector used is type C, the configuration being adapted from the constructive solutions from ARO Welding Technologies.

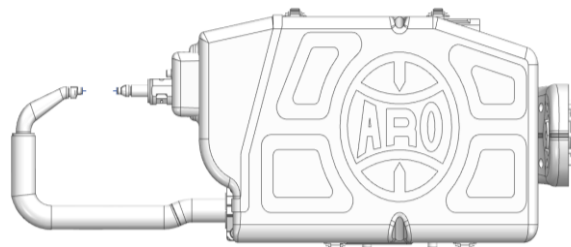


Fig. 2. CAD base of the adapted welding effector

The geometry was adapted so that the electrodes were correctly positioned relative to the welding area, and the opening of the effector allowed the inclusion of the sheet stack. The weld spot was modeled as a lenticular solid, positioned between the two overlapping sheets.

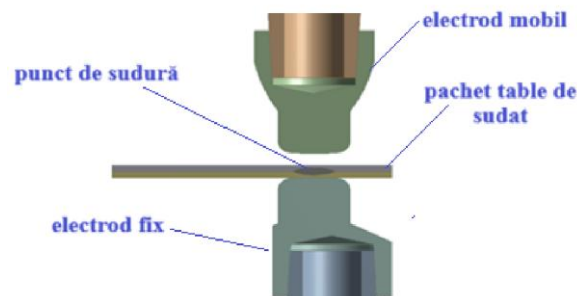


Fig. 3. The components of the studied area

3.2. Simplification of geometry

To reduce the calculation time and obtain a stable model, the geometry was simplified. Constructive details that do not directly influence the welding process, such as secondary edges, areas far from the welding point and geometric elements with no role in the transmission of heat or mechanical stress, were eliminated.

The model was approached three-dimensionally, but a symmetrical partial sectioning was used to more clearly highlight the contact between the electrode, the sheet metal and the welded point.



Fig. 4. Simplified geometry of the analyzed model

3.3. Meshing

Meshing was performed with tetrahedral finite elements. For areas with large temperature and voltage variations, namely the welded point and the contact between the electrodes and the sheets, a finer discretization was applied.

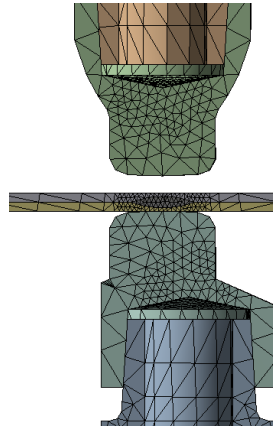


Fig. 5. Discretization of the ensemble

The maximum element size was set to 5 mm, and the final model contained 9461 nodes and 33055 elements.

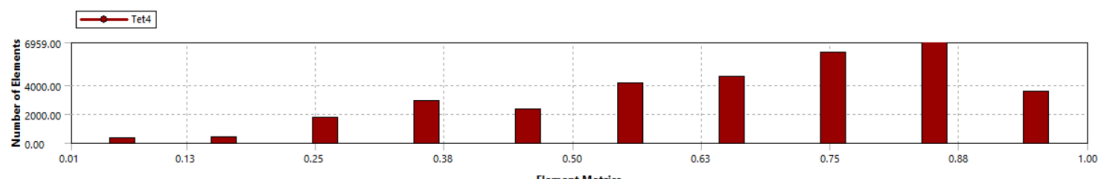


Fig. 6. Finite element quality graph

3.4. Interaction conditions and kinematic couplings

Three kinematic couplers were defined within the model: a translational coupler for the mobile electrode and two fixed couplers for the fixed electrode and the reference support area.

Simulation of the RESISTANCE SPOT welding process based on FEM coupled analyses

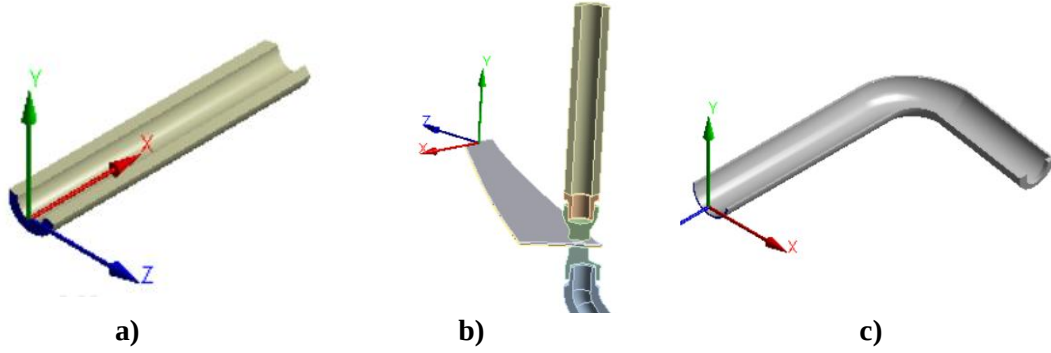


Fig.6. Kinematic torques defined for a) mobile electrode b) welded workpiece c) fixed electrode

Contact regions between electrodes, sheets and the weld point were also defined. The contact between the weld point and the sheets was considered to be bonded.

4. Simulation and solution conditions

The temperature curves were generated using a Python program that used temperature-dependent properties for the DP600 material. The process was divided into three stages: initial contact, rapid heating, and cooling.

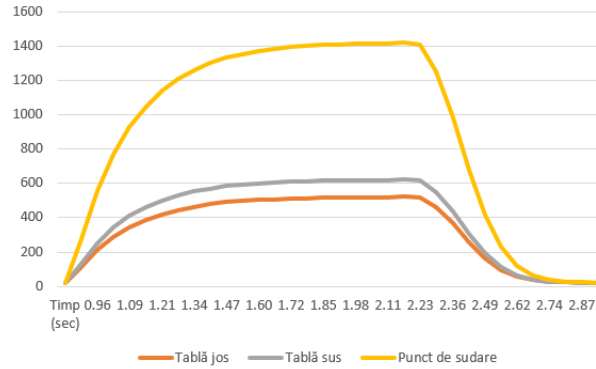


Fig. 7. Temperature curves generated in Python

In the thermal analysis, the initial temperature was considered equal to the ambient temperature of 22°C. Convection conditions were defined at the level of the sheets and electrodes, to take into account the heat exchange with the external environment, and the maximum temperature was reached in the area of the welded point at approximately 1417.8°C.

In the next step, the results of the thermal analysis were imported into the transient structural analysis where the pressing pressures, the translational stroke of the mobile electrode and the locking conditions for the fixed electrode and the support area of the workpiece were defined. Gravitational acceleration was also activated to take into account the self-weight of the components.

5. Solution

In the solution stage, basic theoretical relationships were used to describe the heating of the material, the heat exchange with the external environment, and the expansion produced by temperature variation.

$$Q = m \cdot c \cdot \Delta T \quad (1)$$

$$\dot{Q} = h \cdot A \cdot (T_s - T_\infty) \quad (2)$$

$$\Delta L = \alpha \cdot L_0 \cdot \Delta T \quad (3)$$

where: Q is the amount of heat, m is the mass of the material, c is the specific heat, ΔT is the temperature variation, $\dot{Q}$ is the heat flux transmitted by convection, h is the convection coefficient, A surface area, T_s surface temperature, T_∞ ambient temperature, ΔL length variation, α linear thermal expansion coefficient, and L_0 initial length.

6. Post-processing of results

6.1. Temperature distribution

The results of the thermal analysis show that the maximum temperature is concentrated in the weld point area.

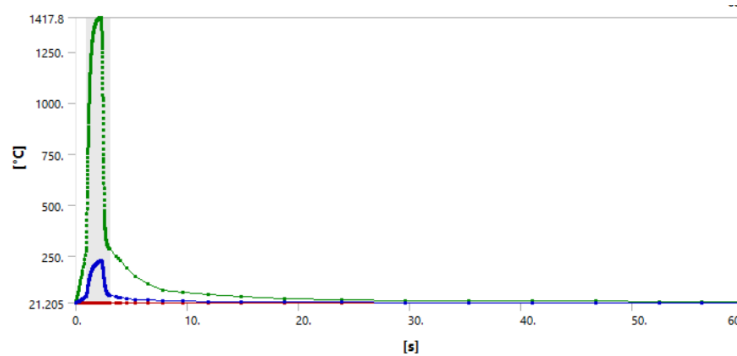


Fig. 8. Temperature variation curve over time

In the maximum temperature stage, the welded core reaches approximately 1417.8°C according to the generated curve, and in the distribution in ANSYS the observed value is approximately 1353.7°C.

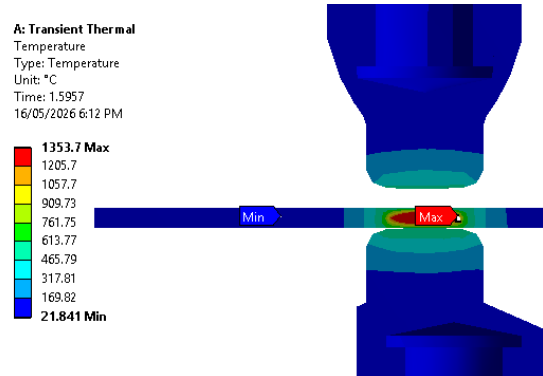


Fig. 9. Temperature distribution in the stages of the welding process

6.2. Total displacements

In the structural analysis, the total displacements of the assembly were monitored, separately at the reference level and at the electrodes level. The maximum displacements occur in the contact area between the electrodes and the sheet, where the pressing pressure acts directly. Regarding total displacements, the maximum value obtained is 0.0084563 mm.

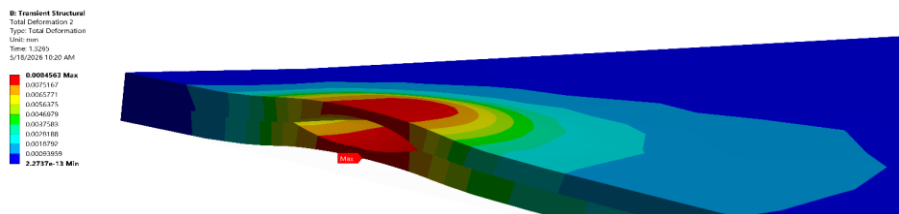


Fig.10. Total displacements

6.3. Von Mises equivalent stresses

To evaluate the mechanical stress of the assembly, the von Mises equivalent stresses were analyzed, separately at the reference level and at the electrode level.

Simulation of the RESISTANCE SPOT welding process based on FEM coupled analyses

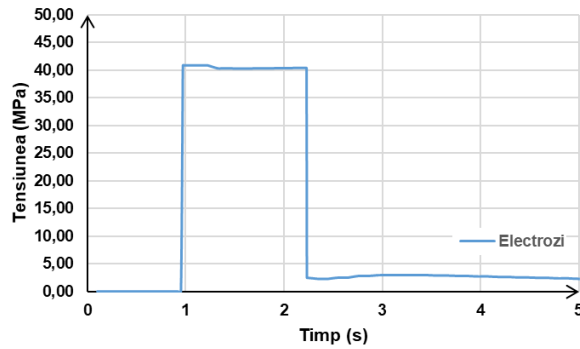


Fig.11. Equivalent voltage graph

The safety factor was determined by relating the yield strength to the maximum equivalent stress:

$$c = \frac{\sigma_c}{\sigma_{ech,max}} = \frac{350}{40,89} = 8,56 \quad (4)$$

7. Conclusions

In this paper, a simulation of the spot pressure welding process was performed for a simplified car body assembly, consisting of two DP600 sheets with a thickness of 1 mm each. The study aimed to correlate the thermal behavior with the structural response of the assembly.

The model was prepared by remodeling and simplifying the geometry in Siemens NX, defining the weld point, positioning the electrodes and discretizing the model with tetrahedral elements. The temperature curves were generated using a Python program and were subsequently introduced into ANSYS.

Thermal results showed that the maximum temperature occurs in the weld point area, and the temperature gradually decreases after the heating stage, through heat transfer to the sheets, electrodes and the external environment.

Based on the results obtained, no joint defects caused by electrode displacement or misalignment are expected. Therefore, for the results obtained, the geometry analyzed does not require any modifications.

8. Bibliography

[1]. Eshraghi, M., Tschopp, MA, and Felicelli, SD (2014), "Effect of resistance spot welding parameters on weld pool properties in a DP600 dual-phase steel: A parametric study using thermomechanically-coupled finite element analysis", *Materials & Design*, 56, pp. 387–397.

[2]. Bézi, Z. (2014), "Experimental and numerical analysis of resistance spot welding process of DP600 steel sheets", *Bulletin of the Transilvania University of Braşov, Series I: Engineering Sciences*.

[3]. Raoelison, RN, Fuentes, A., Rogeon, P., and Dechalotte, F. (2014), "Modeling and numerical simulation of the resistance spot welding of zinc coated steel sheets using finite element method", *Mathematics and Computers in Simulation*, 105, pp. 17–34.

[4]. Bergman, TL, Lavine, AS, Incropera, FP and DeWitt, DP (2011), *Fundamentals of Heat and Mass Transfer*, 7th Edition, John Wiley & Sons

[5]. <https://www.arotechnologies.com/>

DESIGN AND TRANSIENT STRUCTURAL ANALYSIS OF BAG PALLETIZING END-EFFECTOR

CRÎȘU Dragoș-Constantin, Tudor-George ALEXANDRU, Nicoleta Elisabeta PASCU

¹Faculty: IIR, Specialization: Robotics, Year of study: IV, e-mail: dragos.crisu@gmail.com

This paper presents the structural verification of the end-effector of a robotic bag palletizing cell using the Finite Element Method in transient structural regime in ANSYS. The CAD geometry was imported from CATIA V5, simplified from 700 to 22 bodies, with the claws and fixing plate converted to 2D shell surfaces. Kinematic joints and S-curve motion profiles were defined to replicate the gripping and lifting sequence, yielding a mesh of 14,633 nodes and 13,351 elements. Stress analysis confirmed structural integrity, with safety factors of 22.02 for the claws and 1.30 for the fixing plate within EN AW-6082 admissible limits. However, position deviations of 2.90 mm and 2.40 mm were recorded, exceeding typical industrial tolerances and indicating that stiffness improvements such as reinforcing ribs or increased plate thickness are recommended. KEYWORDS: end-effector, palletizing, finite element method, transient mode, aluminum alloy.

1. Introduction

The present project is in the context of a robotic palletizing cell, an industrial application frequently encountered in modern production lines, in which an industrial robot takes 25 kg cement bags from a conveyor and stacks them on a pallet according to a predefined pattern. The central component analyzed is the robot's end-effector, a complex mechanical assembly equipped with gripping claws and a fixing plate, actuated by linear pneumatic pistons.

The main objective of the project was to evaluate the mechanical and dynamic behavior of the assembly under conditions as close as possible to real ones, as well as to verify the structural capacity of the critical components — the gripping claws and the fixing plate — during the operation of lifting the bag from the conveyor. The analysis was performed in transient mode, using the Finite Element Method (FEM).

To achieve this objective, several successive stages were traversed. The starting point was importing the geometry, the CAD model made entirely in CATIA V5 being taken to the simulation environment. Subsequently, the geometry was simplified by eliminating components irrelevant for the analysis — screws, bearings, pneumatic actuators — and by unifying the remaining bodies through Boolean operations. This was followed by the definition of the kinematic couplings between the mobile components, the configuration of the motion profiles corresponding to the bag lifting sequence, and the discretization of the assembly. Finally, the necessary boundary conditions and loads were applied, the transient analysis was run, and the results were interpreted in the post-processing stage, tracking the total displacements, position deviations, and tension distributions in the critical components.

The analysis confirmed the structural integrity of the selected materials and geometry; however, the obtained positioning deviations indicate that additional stiffness improvements may be required.

2. Current status

Structural analysis of robotic end-effectors using the finite element method is an increasingly widespread practice in modern engineering, allowing the evaluation of the mechanical behavior of critical components before manufacturing and reducing the risk of structural failure in service. In the specialized literature, there are numerous studies that address the verification of the structural integrity of industrial grippers, with a focus on stress distribution, total deformation and the identification of critical areas, providing a solid methodological framework for projects similar to the present one.

From the perspective of structural analysis of gripping elements, the paper [1] presents

a detailed approach to the mechanical behavior of gripper claws, modeled in CATIA V5 and analyzed in ANSYS. The study compares the Von Mises stress distribution and total deformation of the claws for different constructive variants of them, under identical loading and constraint conditions. The results demonstrate that by geometric modifications of the claw structure, the maximum stress can be significantly reduced — from 527 N/mm² to 241 N/mm² — without compromising the gripping capacity, confirming the usefulness of FEM as a tool for optimizing critical components of an end effector.

A similar structural analysis, focused on the complete assembly of an aluminum alloy end-effector, is presented in paper [2]. The effector components — base plates, clamping blocks and fasteners — are analyzed in ANSYS both under static and dynamic loading conditions, tracking the stress distribution and nonlinear behavior of the material. The paper emphasizes the importance of correct material selection and section geometry to guarantee the structural integrity of the assembly under real operational conditions.

The structural verification of a lifting gripper under combined loads is addressed in the paper [3], where an elastic analysis is performed by FEM according to the EN 13001 standard, taking into account both the vertical lifting forces and the bending moments generated by the eccentricity of the center of gravity of the manipulated object with respect to the gripping point. This approach is relevant to the present project, where the cement bag generates, during lifting, combined stresses at the level of the effector claws. Finally, the paper [4] deals with the static analysis by FEM in ANSYS of a gripper with an aluminum structure, determining the maximum Von Mises stress and the total deformation of the assembly, with the explicit identification of the critical areas located at the pivot axes of the gripping mechanism.

3. Summary of the analysis procedure

The starting point of the simulation was the complete CAD model of the palletizing effector, created in CATIA V5 and initially consisting of approximately 700 bodies. Importing the geometry into the ANSYS environment was the first step in the workflow, immediately followed by a series of simplification operations aimed at reducing the computational complexity of the model without affecting the accuracy of the results.

During the simplification stage, carried out in the Design Modeler module of ANSYS, all components with a functional role that was not significant for the structural analysis were eliminated: assembly screws, bearings, components dedicated to the actuation and control of pneumatic pistons. The linear guides, components with complex geometry, were completely remodeled as a single unified body. After that, the remaining bodies were grouped by Boolean union operations into two main assemblies — the static part of the effector and the claw module — reducing the total number of bodies to twenty-two. The claws and the fixing plate, identified as flexible elements subject to structural analysis, were converted from 3D to 2D surfaces using the MidSurface command, thus optimizing their discretization process.

Defining the contacts between the components was a crucial step in preprocessing. Using the Named Selection command, the contacts were organized into three categories—fixed, translational, and rotational. A total of seven kinematic couplings were defined, one for each pneumatic piston and one for the lifting sequence of the entire assembly.

The motion cycloramas of the pneumatic pistons were generated through a Python script that calculates S-curve kinematic profiles, ensuring smooth transitions between phases and avoiding sudden starts or stops of the mechanisms. Three pairs of pneumatic pistons were defined, with a time step of 0.05 seconds. The script generated both an overlaid graphical representation of the velocity profiles of all pistons (Fig. 1), and a CSV file with the velocity values on a common time axis, which was later used to configure the couplings in the simulation parameters.

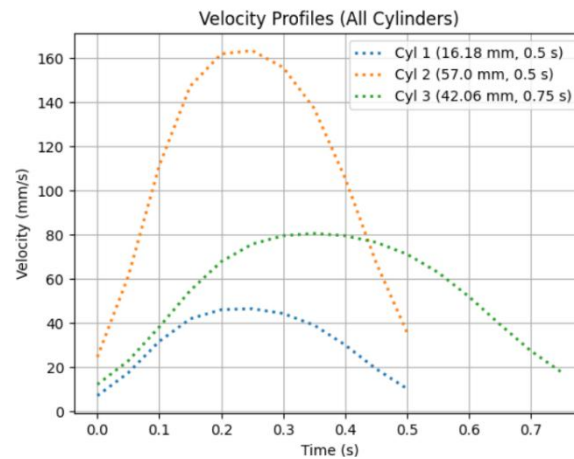


Fig. 1. Graph of the motion cycloramas for the three pairs of pneumatic pistons

The assembly discretization was performed in the Transient Structural module, where the flexible components — the claws and the fixing plate — were passed from the rigid to the flexible state. A global dimensioning parameter of 5 mm was applied, and the quadrilateral mesh generation method was imposed on the contact surfaces between the fixing plate and the pneumatic piston, in order to uniform the distribution of elements in the critical areas. The final mesh obtained consists of 14,633 nodes and 13,351 elements. (Fig. 2)

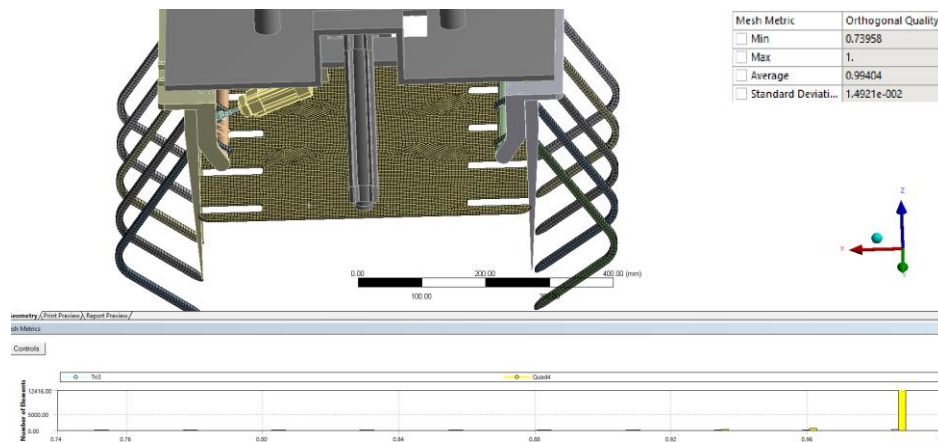


Fig. 2. The final result of the discretization

The simulation conditions were structured into four main time steps, corresponding to the stages of the operating cycle: translation and opening of the claws, lowering of the fixing plate, gripping of the bag and lifting of the assembly. The forces applied to the effector claws — 242.5 N, corresponding to the weight of the 25 kg bag distributed on the contact surface — and the pressure exerted on the fixing plate were introduced after 1.4 seconds of the simulation, marking the moment when the effector actually comes into contact with the bag.

The transient simulation results were evaluated through four main sets of results: total displacements of the assembly, position deviations from the rigid model, Von Mises equivalent stress distribution in the claws, and stress distribution in the mounting plate.

The total displacements of the assembly reflect the complete trajectory of the effector during

the simulated cycle, from the initial position, through the phase of closing the claws and lowering the plate, to the final, raised position.

The position deviations were determined by comparing the displacements obtained in the Rigid Dynamics analysis — in which all components are considered perfectly rigid — with those obtained in the Transient Structural analysis, where the claws and the fixing plate are modeled as flexible elements. The difference between the two analyses highlights the contribution of the elastic deformations of the components to the positioning accuracy of the effector. The maximum deviation determined at the claw assembly level is 2.90 mm, recorded at second 0.13 of the simulation, during the closing stage of the effector. At the fixing plate level, the maximum deviation is 2.40 mm, recorded during the lifting phase of the assembly. (Fig. 3.)

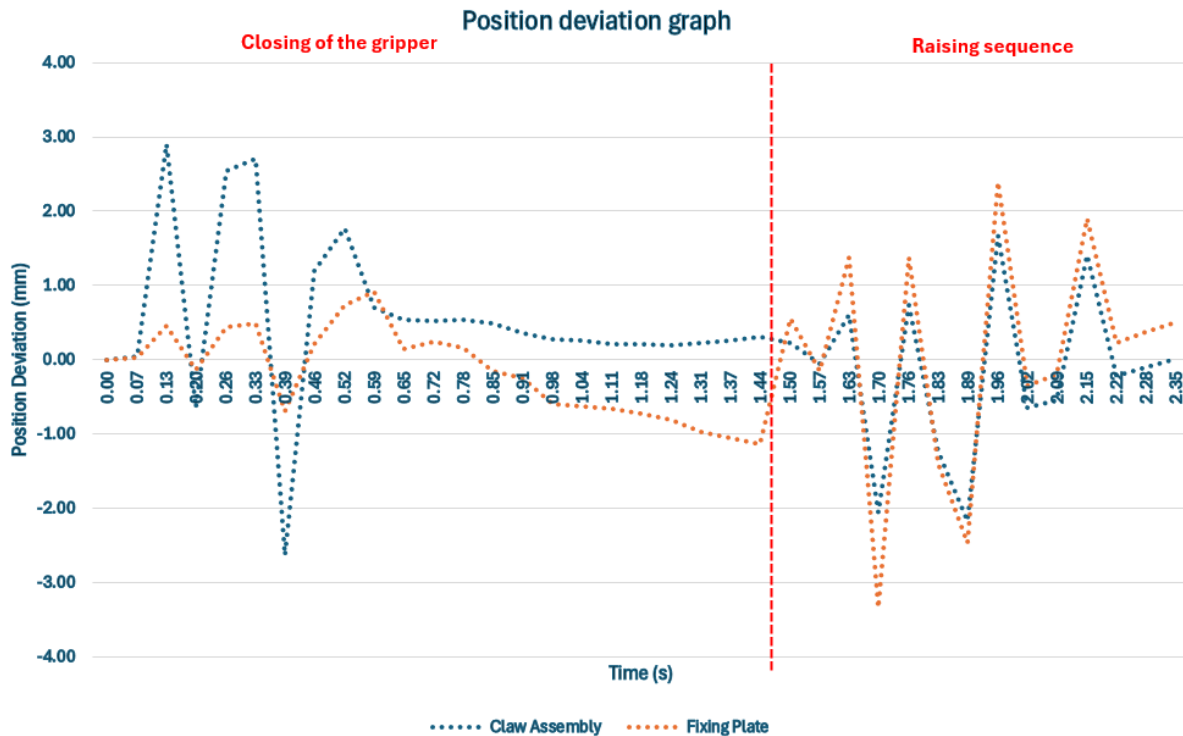


Fig. 3. Graph of position deviation

The analysis of the tensions in the claws revealed a maximum Von Mises equivalent stress of 13.398 MPa, recorded during the lifting stage of the effector. Relating this value to the tensile strength of the EN AW-6082 alloy material used in the claws' manufacture ($R_m = 295$ MPa), the following safety coefficient is obtained:

$$n = 295 / 13.398 = 22.02$$

The high value of the safety coefficient confirms that the claws operate with a very generous structural margin, the stresses developed during the palletizing operation being well below the material yield limit of the material. (Fig. 4)

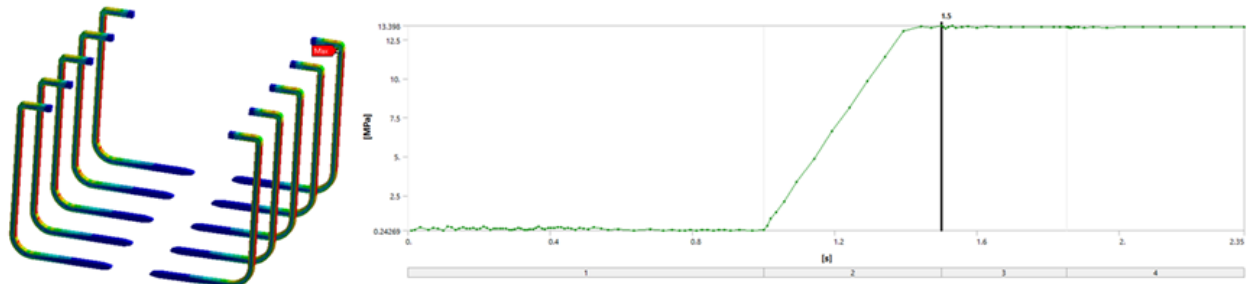


Fig. 4. Visualization of the moment of maximum stress in the claws

The stress analysis in the fixing plate revealed a more demanding structural situation. The maximum Von Mises equivalent stress determined in this component is 238.95 MPa, recorded in the final phase of lifting the bag. The safety coefficient calculated against the tensile strength of the plate material ($R_m = 310$ MPa, EN AW-6082 alloy 3 mm sheet) is:

$$n = 310 / 238.95 = 1.30$$

Although the safety factor is superunitary — indicating that the plate does not structurally fail under the simulated conditions — the available margin is significantly less than that of the claws, the fixing plate representing the critical component of the assembly in terms of structural strength.

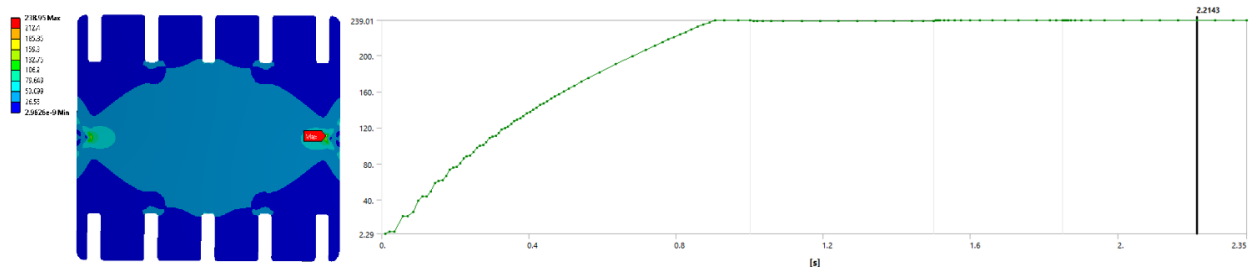


Fig. 5 Visualization of the moment of maximum stress in the fixing plate

Correlating the results of the two types of analysis, it is observed that, although the stress values remain within the allowable limits of the materials, the determined position deviations — 2.90 mm at the claws and 2.40 mm at the fixing plate — are considerably larger than the typical tolerances of an industrial palletizing operation, where the precision of positioning on the pallet is essential for the stability of the stack. These deviations are the direct consequence of the relatively low rigidity of the analyzed elements, especially the fixing plate with a thickness of 3 mm. Possible solutions to reduce them include stiffening the structure by adding reinforcing ribs or increasing the thickness of the flexible elements, with any changes to be validated through a new simulation iteration.

4. Conclusions

This paper presented the verification of the constructive solution of the end-effector in a robotic cell for palletizing 25 kg cement bags, through transient analysis using the Finite Element Method. The workflow included the simplification of the CAD geometry, the definition of the kinematic couplings and the motion cycloramas, the discretization of the assembly and the application of the simulation conditions corresponding to the lifting operation.

In terms of structural strength, the results confirm the viability of the constructive solution for both critical components. The clamping claws present a maximum Von Mises stress of 13.398 MPa, significantly below the allowable limit of the EN AW-6082 alloy (295 MPa), corresponding to a safety coefficient of 22.02. The fixing plate develops higher stresses — 238.95 MPa — with a safety factor of 1.30, still remaining within the allowable limits of the material.

The analysis of the position deviations, however, highlighted a problem of the rigidity of the assembly. The elastic deformations of the flexible components generate deviations of 2.90 mm at the

claws and 2.40 mm at the fixing plate compared to the rigid model, values that exceed the typical tolerances of an industrial palletizing operation. These deviations indicate that the current constructive solution, although structurally safe, does not ensure the positioning accuracy required in operation.

As directions for improvement, it is recommended to stiffen the claws and the fixing plate by adding reinforcing ribs or increasing their thickness, followed by a new simulation iteration to validate the proposed changes.

5. Bibliography

[1] Nirmale, A. P., Patil, S., Handikherkar, V., & Phalle, V. M. (2018, December). *Mechanical and Finite Element Analysis of Gripper Finger*. In *Proceedings of TRIBOINDIA-2018 An International Conference on Tribology*. [2] "Finite Element Analysis of End Effector for an Automated Equipment of High-Speed Forming of Thin-Layered Composite Materials", ResearchGate, 2021.

[2] Mozzillo, R., Bachmann, C., Fanelli, P., Janeschitz, G., & Steinbacher, T. (2023). *Structural assessment of the gripper interlock of the DEMO breeding blanket transporter*. *Heliyon*, 9(8).

[4] Bhelsaikar, A. H., Chothe, P., Atpadkar, V., Ohol, S., & Roy, D. *Finite Element Analysis of a Novel Robotic Gripper: Grabo*