

Simularea și analiza cinematică a unei aplicații robotizate de gestionare a intrărilor și ieșirilor dintr-un AS-RS

NASTASE Robert-Paul

Facultatea de Inginerie Industrială și Robotică, Specializarea Robotică, Anul de studii 3, e-mail: chessroby@yahoo.com

Conducător științific: S.I. dr. Ing. Andrei Mario IVAN

REZUMAT: Subiectul din prezentarea de astăzi îl reprezintă simularea și analiza cinematică a unei aplicații robotizate de gestionare a intrărilor și ieșirilor dintr-un AS-RS. Simularea va fi realizată în programul ABB RobotStudio și totodată se vor prezenta pașii parcurși spre îndeplinirea acestuia. În prima parte a prezentării vom însuși astfel noțiuni introductive legate de operația de asamblare și manipulare cât și o scurtă prezentare a programului utilizat. În cea de-a doua parte vom trece la însușirea elementelor de programare utilizate în simulare cât și la eventuale direcții de lucru în viitorul apropiat. Aplicația dată, astfel, reprezintă o cale deschisă la viitoare îmbunătățiri.

1. Introducere

Scopul lucrării este acela de a realiza o simulare a unei aplicații de manipulare și asamblare a unor semifabricate folosind programul ABB RobotStudio. Asadar funcția logică a aplicației este cea de a asambla obiectele în mediul industrial și tot odată amplasarea acestora pe conveioare, acestea la rândul lor având scopul de a transfera obiectul asamblat către un sistem automat de stocare. Automatizarea acestui sistem ar permite creșterea randamentului într-o firmă și totodată o economie ridicată a costurilor. Operația de manipulare poate fi împărțită în: mecanizată, automatizată, semiautomatizată și informatizată. În aplicația dată dispunem de o manipulare automatizată ce se poate realiza prin intermediul unor echipamente precum Roboți Industriali. Automatizarea procesului de pick and place cu roboți grăbește procesul de ridicare a pieselor și de plasare a acestora în locații noi, crescând rata de producție. Cu o varietate mare de efectori disponibili, roboții de pick and place pot fi personalizați pentru a se potrivi cerințelor specifice de producție. Mutarea produselor mari, mici, grele sau greu de manipulat poate fi o sarcină ușoară de automatizat în linia din fabrică. Consistența este, de asemenea, un beneficiu al utilizării unui sistem de pick and place. Roboții pot fi programați și echipați cu ușurință pentru a deservi în cadrul mai multor aplicații, dacă este necesar. Vom utiliza ABB RobotStudio datorită flexibilității acestuia, programul în sine fiind conceput pentru o foarte mare varietate de aplicații cât și compoziția librărie fiind foarte vastă.

2. Robotul industrial și programul de simulare utilizat

Pentru această aplicație am utilizat robotul ABB IRB 120, Catia V5, robot industrial cu 6 axe de tip braț articulat. Programul ABB RobotStudio, ne permite utilizarea unei vaste biblioteci și astfel cu ajutorul unui controller virtual vom putea realiza simularea unei celule robotizate.

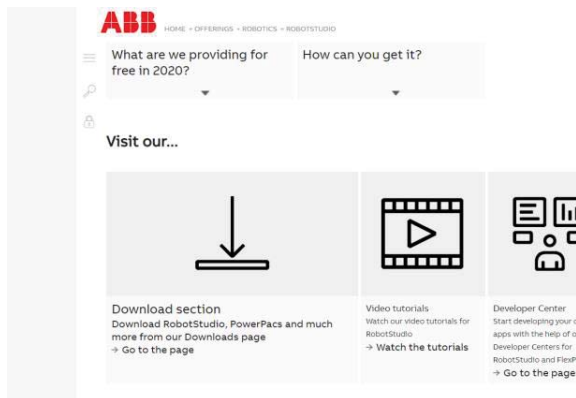


Fig. 1. Site ABB – descarcare program simulare



Fig. 2. Robotul ABB IRB 120

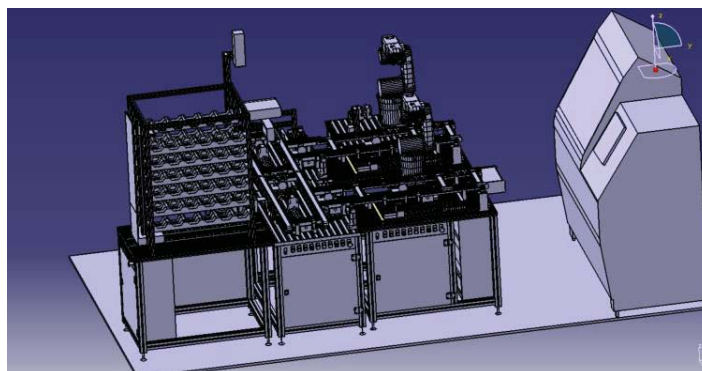
Specificații tehnice:
 Sarcina portanta maximă: 0.3 kg
 Raza maxima atinsă: 580 mm
 Numar axe comandate numeric: 6
 Repetabilitate: $\pm 0,01$

Tabelul 1. Specificații cuple robot

Axa	Domeniu	Viteza Maximă
1	+165° la -165°	250°/s
2	+110° la -110°	250°/s
3	+70° la -110°	250°/s
4	+160° la -160°	320°/s
5	+120° la -120°	320°/s
6	+400° la -400°	420°/s

3. Aplicația de manipulare

Primul pas va fi realizarea unei celule robotizate cu ajutorul căreia vom realiza simularea aplicației date. Celula data este formata dintr-un CNC si 2 roboti, fiecare avand conveioare pentru transportul ansamblurilor/semifabricatelor.



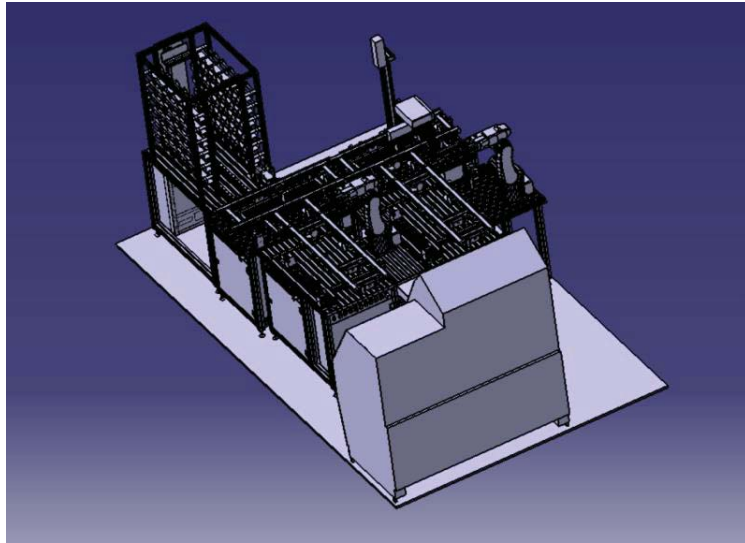


Fig. 3. Celula robotizata

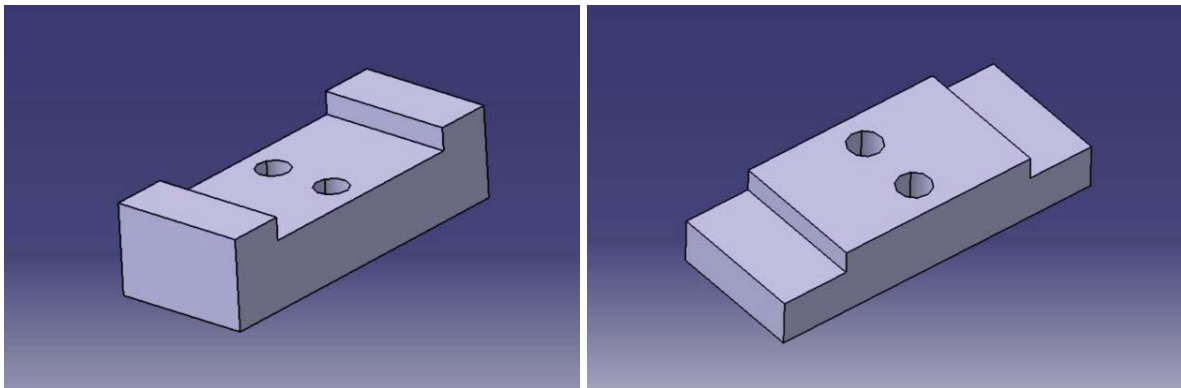


Fig. 4. Piesele ce urmeaza a fi asamblate/manipulate

Dupa realizarea celulei si a pieselor de prelucrat urmeaza implementarea acesteia in programul propriu zis.

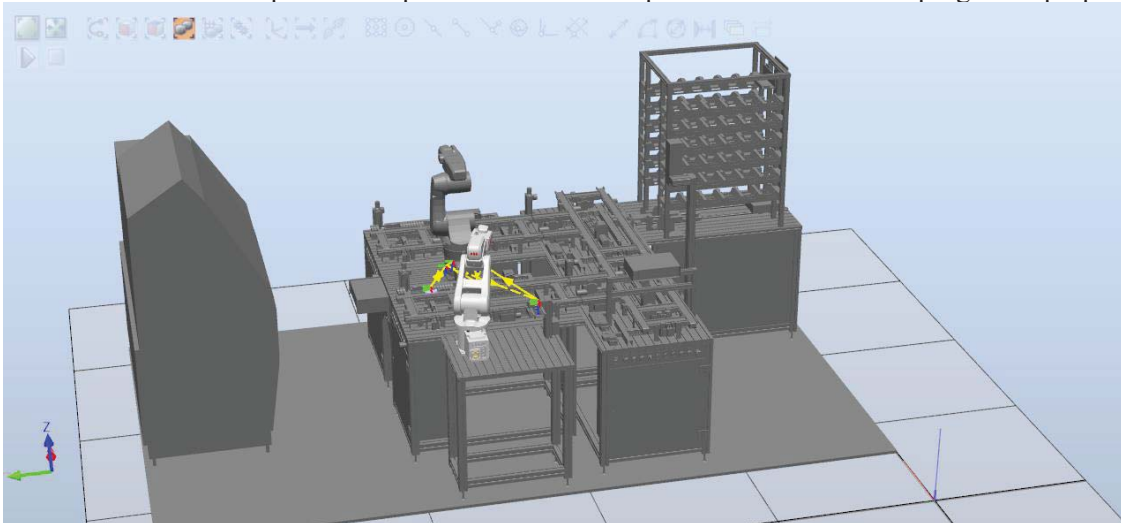


Fig. 5. Celula introdusa in program

Odata cu acest pas urmeaza realizarea programarii si astfel vom avea nevoie sa introducem mai multe puncte utilizate atat la preluare cat si la amplasare. Punctele vor fi de forma urmatoare: 2 pentru pick 1 (piesa initiala), 2 pentru mijloc deoarece vom avea 5 preluari 1 singur punct de mijloc nu va fi suficient, 1 pentru home, 2 pentru place si 2 pentru pick 2 (piesa ce se asambleaza peste piesa 1). Programul va arata asa:

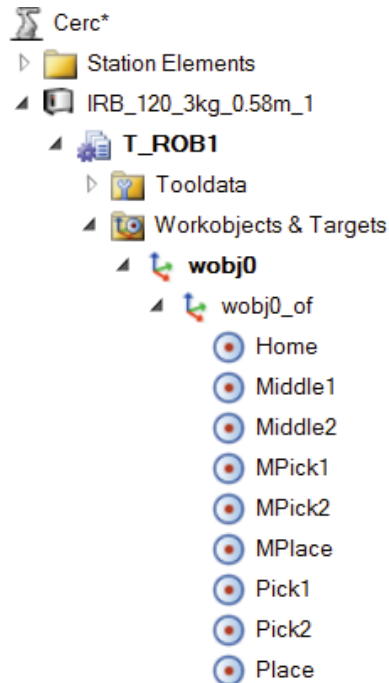


Fig. 6. Punctele utilizate in program

In cea de-a 2 parte a simularii urmeaza atribuirea miscarilor robotului asupra punctelor create si anume:



Fig. 7. Punctele utilizate in program

Un ultim pas spre realizarea simulării va fi astfel declararea „variabilelor” ca și introducerea de senzori. Variabilele în aplicația dată vor fi obiectele preluate și acestea date vor fi de forma: „CollisionSensor” - detectează coliziunea, setat la o toleranță de 10 mm; „Attacher” - atașează un obiect, atribuit atât obiectului 1 ca și 2; „Detacher” – detașează un obiect, atribuit atât obiectului 1 ca și 2;

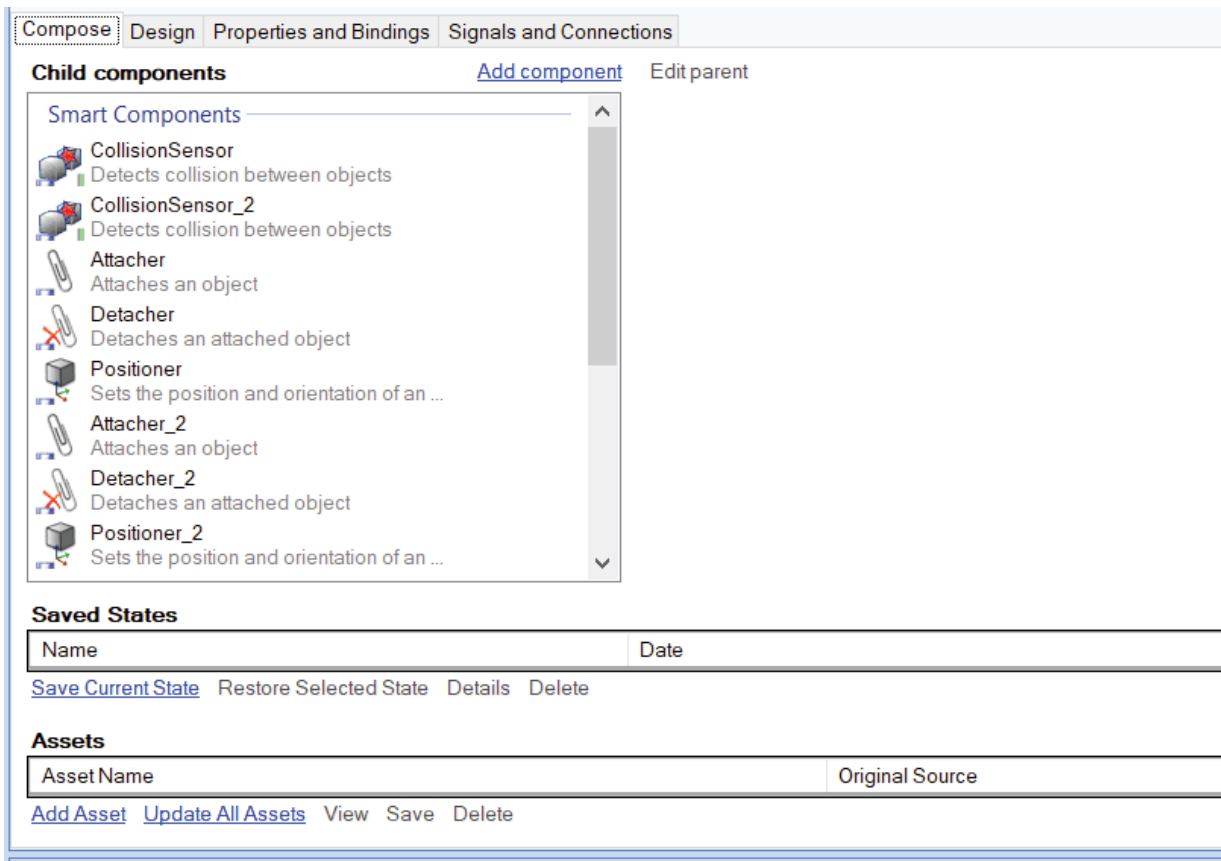


Fig. 8. Declararea „variabilelor”

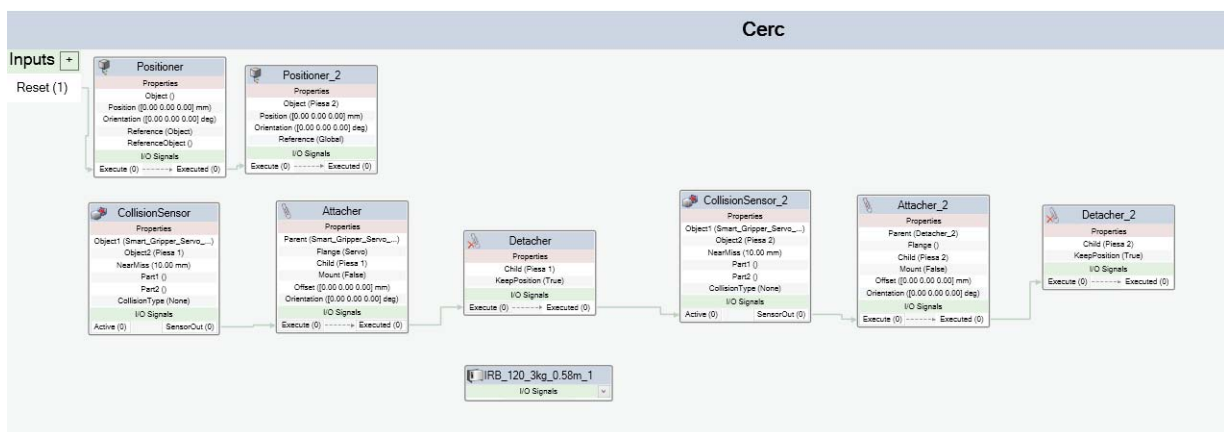


Fig. 9. Schema logica

Ultimul pas de care este important să nu uităm este introducerea unei variabile de reset care realizează repetabilitatea programului:

Compose Design Properties and Bindings **Signals and Connections**

I/O Signals

Name	Signal Type	Value
Reset	DigitalInput	1

[Add I/O Signals](#) [Expose Child Signal](#) [Edit](#) [Delete](#)

I/O Connections

Source Object	Source Signal	Target Object	Target Signal or Property
Cerc	Reset	Positioner	Execute
Positioner	Executed	Positioner_2	Execute
CollisionSensor	SensorOut	Attacher	Execute
Attacher	Executed	Detacher	Execute
Detacher	Executed	CollisionSensor_2	Active
CollisionSensor_2	SensorOut	Attacher_2	Execute
Attacher_2	Executed	Detacher_2	Execute

Fig. 10. Introducerea variabilei de reset

Pentru simulare va trebui sa intram in bara rapid la sectiunea Synchronize de unde vom bifa urmatoarele:

Cerc

Compose Design

I/O Signals

Name
Reset

[Add I/O Signals](#) [Edit](#) [Delete](#)

I/O Connections

Source Object
Cerc
Positioner
CollisionSensor
Attacher
Detacher
CollisionSensor_2
Attacher_2

[Add I/O Connection](#)

Synchronize to RAPID

Name	Synchronize	Module	Local	Storage class	Inline
IRB_120_3kg_0.58m_1	<input checked="" type="checkbox"/>		<input type="checkbox"/>		
T_ROB1	<input checked="" type="checkbox"/>		<input type="checkbox"/>		
Paths & Targets	<input checked="" type="checkbox"/>		<input type="checkbox"/>		
Path_20	<input checked="" type="checkbox"/>	Module1	<input type="checkbox"/>		
ToolData	<input checked="" type="checkbox"/>		<input type="checkbox"/>		
Servo	<input checked="" type="checkbox"/>	CalibData	<input type="checkbox"/>	PERS	
WorkObject	<input checked="" type="checkbox"/>		<input type="checkbox"/>		

Fig. 11. Meniul Synchronize

Suntem gata sa apasam butonul de start:

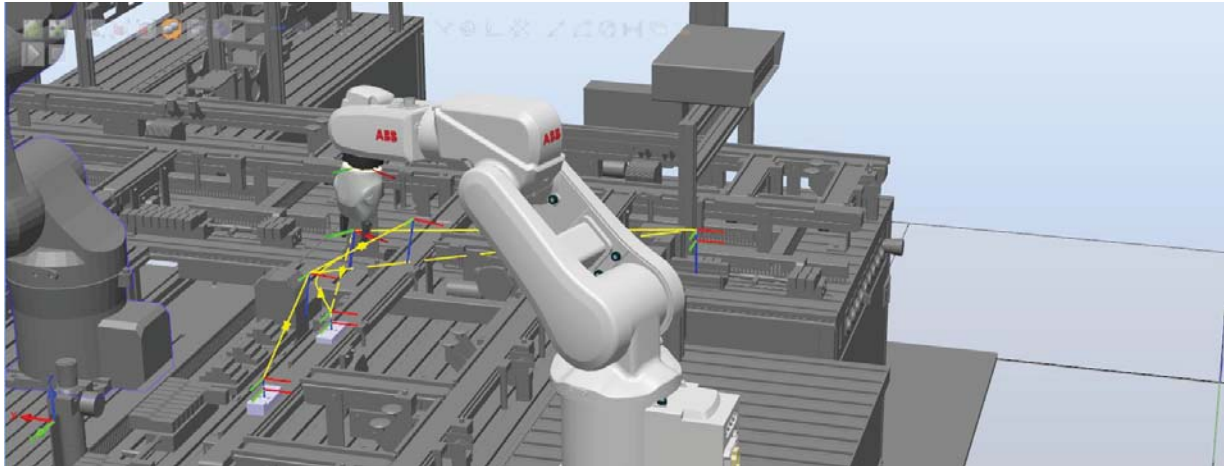


Fig. 12. Programul in urma simularii

Din nefericire simularea data este una imperfecta, eroarea insa neputand fi identificata in timp util.

4. Concluzii

Desi simularea nu a iesit cum era in plan, aceasta in viitorul apropiat se va putea implementa la un nivel mai complex ca, cel al unei lucrari de licenta. Viitoare directii de lucru ar fi: simularea conveioarelor, a robotilor si nu in cele din urma al sistemului AS-RS.

5. Bibliografie

- [1]. <https://www.youtube.com/watch?v=0aa7l4qzpnI>
- [2]. https://www.youtube.com/watch?v=i1Q_kxe5Og0
- [3]. https://www.youtube.com/watch?v=2ZzmK_egDuU&list=RDQMx-IW8h04We8&index=25
- [4]. <https://www.youtube.com/watch?v=0aa7l4qzpnI>
- [5]. https://en.wikipedia.org/wiki/Pick-and-place_machine
- [6]. <https://new.abb.com/products/robotics/robotstudio>