

PROGRAMAREA SI SIMULAREA OFFLINE A UNEI CELULE DE SUDARE CU ARC ELECTRIC REPERE DE DIMENSIUNI MARI PENTRU CONSTRUCTII DE HALE INDUSTRIALE UTILIZAND MEDIUL DE LUCRU ABB ROBOT STUDIO. PROGRAMAREA PE BAZA DE SEMNALE

GRIGORE D. Adrian-Georgian

Facultatea:Facultatea de Inginerie Industriala si Robotica, Specializarea:Robotica, Anul de studii:Master II,
e-mail: grigoreadrianunsr@gmail.com

Conducător științific: Prof. dr. ing. Adrian NICOLESCU

REZUMAT: În celula robotizată prezentă în acest studiu se realizează sudarea cu arc electric a unor grinzi metalice de dimensiuni mari utilizate în construirea de hale industriale. Pentru sudarea acestor grinzi sunt utilizați 3 roboți industriali model ABB IRB 6640 cu echipament de sudare și un robot ABB IRB 6640 cu efector de tip Gripper magnetic, precum și alte echipamente ce ajută la alimentarea celulei cu semifabricat dar și la poziționarea corectă a acestuia.

1. Introducere

Pentru a putea realiza programarea și simularea offline a celulei de sudare cu arc electric repere de dimensiuni mari pentru construcții de hale industriale utilizând mediul de lucru ABB ROBOT STUDIO am urmărit ca puncte principale realizarea fluxului tehnologic pentru a putea determina legăturile logice dintre componentele celulei și necesitatea semnalelor digitale de intrare și ieșire pentru corelarea mișcărilor componentelor; implementarea soluțiilor de verificare a prezenței semifabricatelor în puncte cheie ale celulei, acest punct reprezentând studierea necesității unor senzori pentru a detecta poziția corectă a grinzilor metalice pentru a fi preluate și reorientate dar și pentru prezența plăcilor metalice ce urmează a fi manipulate de robotul cu gripper magnetic și ultimul punct realizarea programului pentru simularea offline a celulei.

2. Fluxul tehnologic al celulei

Celula prezentată are în componență 4 roboți industriali ABB IRB 6640(fig.1-Cifra 1), 3 dintre aceștia fiind echipați cu efector de tip torță pentru sudare cu arc electric și echipamentul necesar procesului de sudare iar unul dintre acești roboți este echipat cu efector magnetic pentru preluarea semifabricatelor de tip plăci metalice debitate în prealabil; pentru a se putea deplasa pe o distanță maximă de 19m roboții au nevoie de o axa adițională de translație la sol iar pentru reperele de tip grindă sunt necesare 3 echipamente de reorientare cu deplasare la sol(fig.1-Cifra 2), astfel a fost realizată o bază comună cu sistem individual de deplasare a celor 4 roboți și a celor 3 sisteme de reorientare (fig.1-Cifra3). Pentru manipularea grinzilor este necesar un echipament de tip pod rulant(fig.1-Cifra 4) controlat prin comandă numerică, un conveior (fig.1-Cifra 5) , un sistem pentru depozitarea produsului finit(fig.1-Cifra 7) și sisteme de protecție pentru oprirea accesului personalului în interiorul celulei(fig.1-Cifra 6).

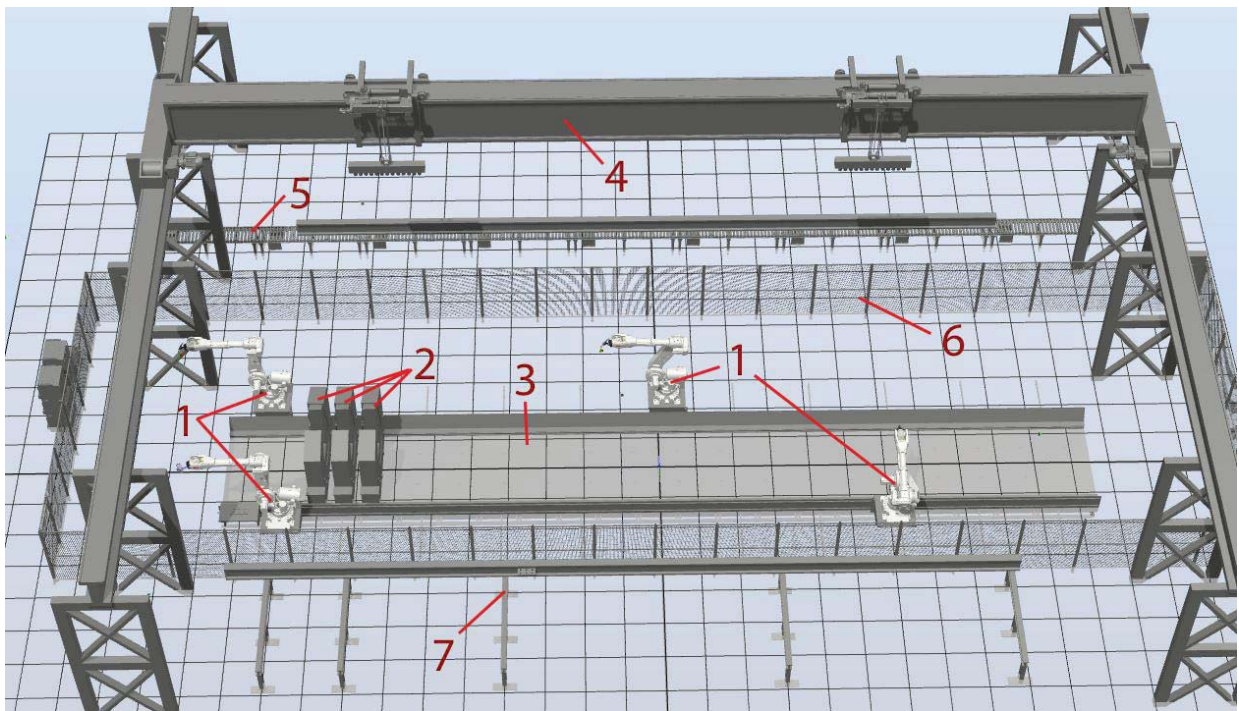


Fig. 1. Componente celulă

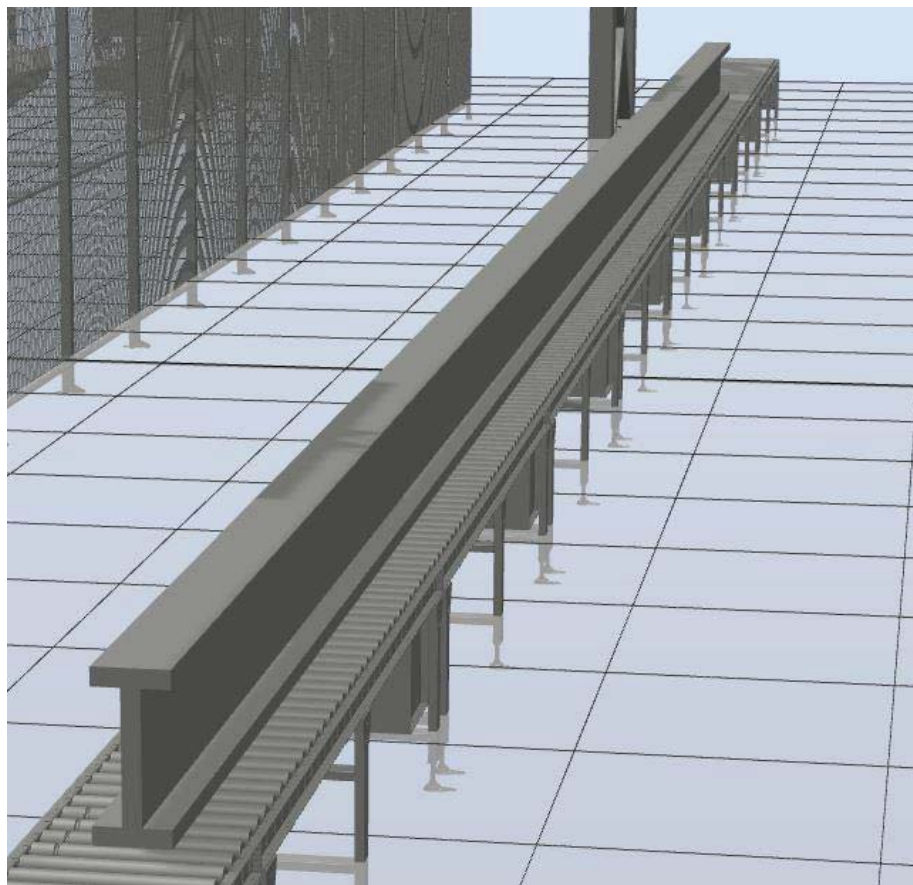


Fig. 2. Reper tip grindă metalică de dimensiuni mari

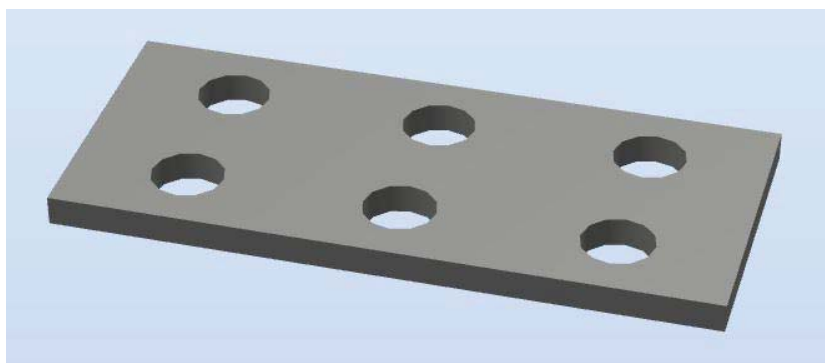


Fig. 3. Reper tip placă metalică debitată

Accesul reperului de tip grindă în celulă se realizează prin intermediul conveiorului, aici sunt necesari senzori inductivi ce vor indica dacă grinda este fizic prezentă în poziția stabilită pentru preluarea acesteia de către macaraua de tip pod rulant și îi vor transmite mai departe acesteia un semnal de confirmare pentru a putea fi acționată. Din această poziție grinda este preluată de macara, trece prin câteva poziții prestabilite, iar în momentul în care aceasta ajunge deasupra sistemelor de reorientare, le transmite un semnal pentru ca acestea să se poziționeze pentru a prelua grinda metalică. Odată poziționată grinda va fi verificată de un alt set de senzori inductivi ce vor transmite un semnal de confirmare către controller iar acesta va comanda sistemele de reorientare să fixeze și să orienteze grinda în poziția de sudare. După ce grinda este în poziția de sudare este transmis un semnal către controller să acționeze poziționarea roboților de sudare și robotului de manipulare să preia reperul de tip placă metalică. Pentru a prelua placa metalică, robotul va verifica întâi prezența acesteia în punctul de preluare cu ajutorul unui set de senzori inductivi. După ce aceștia trimit un semnal pozitiv, robotul preia placa și o manipulează către poziția de sudare după care trimite un semnal primului robot de sudare care poate începe procedeul. După ce robotul de sudare fixează placa pe grindă trimite un semnal către robotul de manipulare care va pleca să preia placa următoare, acest procedeu se repetă și pentru următoarele plăci până la ultima. În momentul în care toți cei 3 roboți de sudare au terminat procesul pentru locurile accesibile aceștia transmit câte un semnal controllerului. Acesta așteaptă activarea celor 3 semnale pentru a acționa sistemele de orientare la 180 de grade. După ce grinda a fost reorientată cei 3 roboți primesc un semnal și încep programul de sudare pentru zonele rămase iar la finalizare ei trimit un semnal către controller care acționează sistemele de orientare ce se repun în poziția de preluare, dar și către macara, care se pregătește de preluarea grinzii, însă nu se va coborî până nu primește semnal de la sistemele de orientare că acestea sunt în poziția corectă. De aici macaraua preia grinda și o manipulează către locul de depozitare și transmite un semnal pentru reînceperea procesului.

3. Implementarea soluțiilor de verificare a prezenței semifabricatelor în puncte cheie ale celulei

Prezența semifabricatelor în pozițiile corecte este necesară pentru ca procesul să se desfășoare în mod corect, precis și fără continuarea acestuia într-un mod eronat care ar duce atât la distrugerea semifabricatelor cât și a echipamentelor. Soluția adoptată pentru verificarea acestor poziții este una simplă și anume plasarea unor senzori inductivi calibrați corespunzător unei detecții precise a semifabricatelor.

În figura numărul 4 sunt prezentate zonele necesare a fi verificate cu ajutorul senzorilor inductivi: În zona cu numărul 2 se va verifica prezența și amplasarea grinzii metalice cu ajutorul a 6 senzori inductivi. Se va folosi un număr de 6 senzori pentru a putea verifica grinzi de diferite lungimi astfel controllerul va citi un număr corespunzător de senzori în funcție de lungimea efectivă a grinzii, informație ce este primită din celula anterioară acesteia ce are ca rol debitarea grinzilor.

În zona numărul 3 se va verifica cu ajutorul a 2 senzori inductivi pe fiecare echipament de reorientare dacă poziția grinzii este corectă pentru ca acestea să poată bloca și reorienta grinda.

În zona numărul 4 se va verifica cu ajutorul unui sensor inductiv dacă există sau nu în poziție o placă pentru a fi preluată.

În zona numărul 1 sunt prezente echipamentele de tip controller unde se vor procesa toate datele necesare precum și semnalele provenite de la senzori sau echipamente.

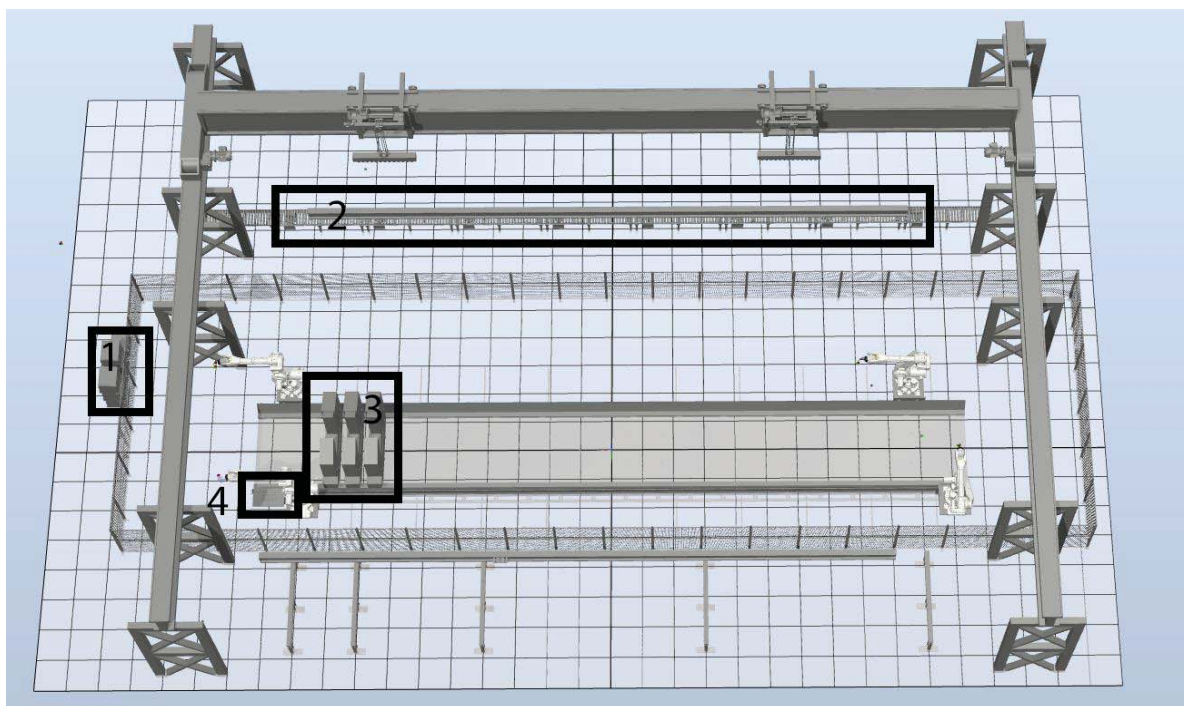


Fig. 4. Pozitii cheie ce necesita a fi verificate.

4. Realizarea programului pentru simularea offline a celulei.

Pentru realizarea programului robotul ce realizează manipularea reperelor de tip plăci a fost folosit drept controller Master iar restul echipamentelor Slave, astfel toate semnalele au fost centralizate către controllerul acestui robot.

Pentru a putea realiza traiectoriile roboților în mediul de simulare offline ABB Robot Studio este necesar să definim o serie de Target-uri pe care apoi le apelăm cu comenzi de tip “MoveJ” sau “MoveL” diferența dintre aceste două mișcări fiind faptul că prima trimite robotul din punctul actual în punctul următor fără să țină cont de traiectorie, alegând cele mai scurte mișcări ale fiecărei axe iar cea de a doua comandă realizează o traiectorie directă printr-o linie dreaptă între punctul actual și următorul punct. Un target este definit prin 3 vectori (fig.5 X-roșu,Y-verde,Z-albastru) ce îi conferă atât poziția în spațiu cât și direcția.



Fig. 5. Exemplu de Target.

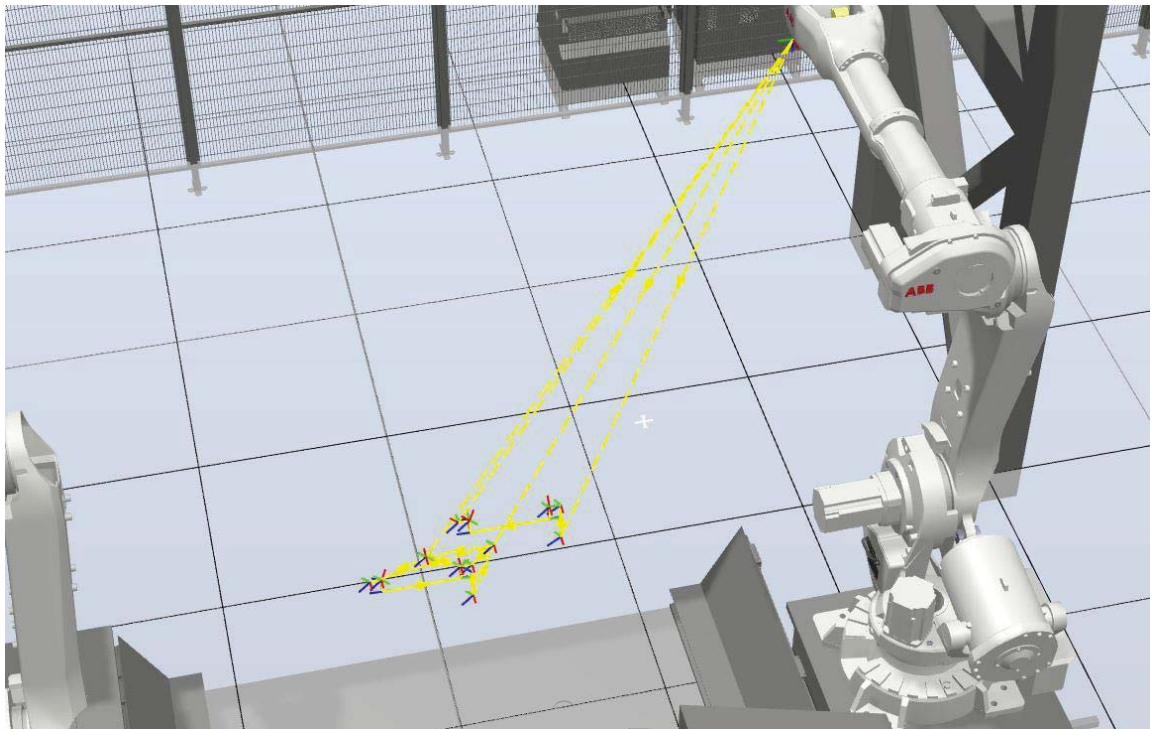


Fig. 6. Traiectorie robot de sudare.

După ce au fost definite punctele pentru fiecare robot în parte, se realizează traiectoriile(fig.6) inserându-se mișcările în program(fig.7)

- ⚡ WaitTime 1.01
- ➡ MoveL homeR1
- ➡ MoveL Target_10_3
- ➡ MoveL Target_10
- ⚡ Wait Time 0.2
- ⚡ SetDO attachplaca1_gr,1
- ➡ MoveL Target_10_3
- ⚡ SetDO rob1,1
- ➡➡ MoveJ Target_20_2
- ➡ MoveL Target_20
- ⚡ WaitDO rob1,0
- ⚡ SetDO attachplaca1_gr,0
- ⚡ Wait Time 0.3
- ⚡ SetDO attachplaca1,1
- ➡ MoveL Target_20_2
- ➡➡ MoveJ homeR1
- ➡ MoveL Target_10
- ⚡ SetDO attachplaca2_gr,1
- ⚡ Wait Time 1.01
- ➡ MoveL Target_10_3

Fig. 7. Exemplu de acțiuni.

Pentru a putea realiza rețeaua de senzori și semnale trebuie să introducem semnalele în controllerul robotului master, însă în mediul offline nu este suficientă doar definirea acestor semnale (fig.8) ci este necesară și definirea unor acțiuni ce au loc în urma activării sau dezactivării unui semnal digital(fig.9). Aceste acțiuni sunt definite prin alegerea unui semnal(exemplu OR_1), o condiție pentru a porni acea acțiune(exemplu: Signal OR_1 = true) și una sau mai multe acțiuni ce se realizează după îndeplinirea acelei condiții(exemplu: se vor acționa cele 3 sisteme de reorientare pentru a ajunge în poziția numărul 1 și apoi semnalul OR_1 să redevină false)

Type	Name	Type of Signal	Assigned to Device	Signal Identification Label	Device Mapping	Category	Access Level	Default Value	Filter Time Passive (ms)	Filter Time Active
Access Level	AS1	Digital Input	PANEL	Automatic Stop chain(X5.11 to X5.6) and (X5.9 to X5.1)	13	safety	ReadOnly	0	0	0
Cross Connection	AS2	Digital Input	PANEL	Automatic Stop chain backup(X5.5 to X5.6) and (X5.3 to X5.1)	14	safety	ReadOnly	0	0	0
Device Trust Level	attachgrd	Digital Output			N/A		All	0	N/A	N/A
Industrial Network	attachplaca1	Digital Output			N/A		All	0	N/A	N/A
Route	attachplaca2	Digital Output			N/A		All	0	N/A	N/A
Signal	attachplaca3	Digital Output			N/A		All	0	N/A	N/A
Signal Safe Level	attachplaca3_gr	Digital Output			N/A		All	0	N/A	N/A
System Input	cr_1	Digital Output			N/A		All	0	N/A	N/A
System Output	cr_2	Digital Output			N/A		All	0	N/A	N/A
	cr_3	Digital Output			N/A		All	0	N/A	N/A
	cr_4	Digital Output			N/A		All	0	N/A	N/A
	cr_5	Digital Output			N/A		All	0	N/A	N/A
	cr_6	Digital Output			N/A		All	0	N/A	N/A
	cr_7	Digital Output			N/A		All	0	N/A	N/A
	cr_home	Digital Output			N/A		All	0	N/A	N/A
	DRV1BRAKE	Digital Output	DRV_1	Brake-release coil	2	safety	ReadOnly	0	N/A	N/A
	DRV1BRAKEFB	Digital Input	DRV_1	Brake Feedback(X3.6) at Contactor Board	11	safety	ReadOnly	0	0	0
	DRV1BRAKEOK	Digital Input	DRV_1	Brake Voltage OK	15	safety	ReadOnly	0	0	0
	DRV1CHAIN1	Digital Output	DRV_1	Chain 1 Interlocking Circuit	0	safety	ReadOnly	0	N/A	N/A
	DRV1CHAIN2	Digital Output	DRV_1	Chain 2 Interlocking Circuit	1	safety	ReadOnly	0	N/A	N/A
	DRV1EXTCONT1	Digital Input	DRV_1	External customer contactor (X2d) at Contactor Board	4	safety	ReadOnly	0	0	0
	DRV1FAN1	Digital Input	DRV_1	Drive Unit FAN1(X10.3 to X10.4) at Contactor Board	9	safety	ReadOnly	0	0	0
	DRV1FAN2	Digital Input	DRV_1	Drive Unit FAN2(X11.3 to X11.4) at Contactor Board	10	safety	ReadOnly	0	0	0
	DRV1K1	Digital Input	DRV_1	Contactor K1 Read Back chain 1	2	safety	ReadOnly	0	0	0
	DRV1K2	Digital Input	DRV_1	Contactor K2 Read Back chain 2	3	safety	ReadOnly	0	0	0
	DRV1LM1	Digital Input	DRV_1	Limit Switch 1 (X2a) at Contactor Board	0	safety	ReadOnly	0	0	0
	DRV1LM2	Digital Input	DRV_1	Limit Switch 2 (X2b) at Contactor Board	1	safety	ReadOnly	0	0	0

Fig. 8. Definiere semnale în controller.

Events	Activation	Trigger Type	Trigger System	Trigger Name	Trigger Parameter	Action Type	Action System	Action Name	Action Parameter	Time (s)
Add...	On	I/O	Controller37	cr_home	1	Multiple			Multiple	9:20:18 PM
Delete	On	I/O	Controller37	cr_1	1	Multiple			Multiple	
Copy	On	I/O	Controller37	cr_2	1	Multiple			Multiple	
Refresh	On	I/O	Controller37	cr_3	1	Multiple			Multiple	
Export...	On	I/O	Controller37	cr_4	1	Multiple			Multiple	
Import...	On	I/O	Controller37	cr_5	1	Multiple			Multiple	
	On	I/O	Controller37	cr_6	1	Multiple			Multiple	
	On	I/O	Controller37	cr_7	1	Multiple			Multiple	
	On	I/O	Controller37	resetgrinda	1	Multiple			Multiple	9:20:18 PM
	On	I/O	Controller37	attachgrd	1	Attach Object		Attach Object	Pod rulant -> grinda	
	On	I/O	Controller37	attachgrd	0	Detach Obj...		Detach Object	Pod rulant <- grinda	4:39:58 PM
	On	I/O	Controller37	OR_home	1	Multiple			Multiple	9:20:18 PM
	On	I/O	Controller37	OR_1	1	Multiple			Multiple	
	On	I/O	Controller37	OR_2	1	Multiple			Multiple	
	On	I/O	Controller37	OR_3	1	Multiple			Multiple	
	On	I/O	Controller37	OR_4	1	Multiple			Multiple	
	On	I/O	Controller37	attachor	1	Attach Object		Attach Object	Sistem Reorientare 1 -> grinda	
	On	I/O	Controller37	attachor	0	Detach Object		Detach Object	Sistem Reorientare 1 <- grinda	4:39:58 PM
	On	I/O	Controller37	resetplaci	1	Multiple			Multiple	9:20:18 PM
	On	I/O	Controller37	attachplaca1	1	Attach Object		Attach Object	grinda -> placa1	
	On	I/O	Controller37	attachplaca2	1	Attach Object		Attach Object	grinda -> placa2	
	On	I/O	Controller37	attachplaca3	1	Attach Object		Attach Object	grinda -> placa3	

Trigger: I/O Signal Trigger

Activation: Comments:

Signal Name	Signal Type
ES2	DI
fn1	DO
fn2	DO
GENERAL_RESET	DO
GS1	DI
GS2	DI
MAN1	DI
MAN2	DI
MANFS1	DI
MANFS2	DI
MONPB	DI
MOTLMP	DO
OR_1	DO

Signal Source: Controller37

Trigger Condition: Signal is true (1) Signal is false (0)

Action: Move Mechanism to Pose

Added Actions:

Seq. No	Action
1	Move Mechanism to Pose
2	Move Mechanism to Pose
3	Move Mechanism to Pose
4	Change I/O

Cyclic

Station signal to set when Pose reached:

Mechanism:	Name	Type
Sistem Reorientare 1	Pose1	

Set to True Set to False

Fig. 9. Definiere acțiuni pentru semnale.

Se implementează toate mișcările și acțiunile necesare fluxului și după verificarea vizuală a acestora se sincronizează modulul stației de lucru cu controllerul virtual, astfel în modulul “RAPID” putem să verificăm și să ajustăm codul celor 4 programe ce pot fi implementate fizic direct pe roboți.

```

2  □  CONST robtarget homeR1:=[[12167.45561847,1962.69993033,2671.045],[0.353558358,-0.612381039,0.612363832,-0.353548423],[1,0,-2,0],[9E+09,9E+09,9E+09,9E+09]];
3  CONST robtarget Target_10_3:=[[10914.104,1814,781.795],[0,-0.707106781,0.707106781,0],[0,0,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
4  CONST robtarget Target_10:=[[10914.104,1814,681.795],[0,-0.707106781,0.707106781,0],[0,0,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
5  CONST robtarget Target_20_2:=[[10252,148.902,1082.049],[0.707106781,0,-0.707106781,0],[0,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
6  CONST robtarget Target_20:=[[10152,148.902,1082.049],[0.707106781,0,-0.707106781,0],[0,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
7  CONST robtarget Target_30:=[[11867.457,1962.701,2671.045],[0,1,0,0],[1,0,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
8  CONST robtarget Target_40_2:=[[11032,168.902,1407.049],[0.5,0.5,-0.5,0.5],[0,-2,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
9  CONST robtarget Target_40:=[[11032,168.902,1207.049],[0.5,0.5,-0.5,0.5],[0,-2,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
10 CONST robtarget Target_100:=[[8912,148.902,1382.049],[0.707106781,0,0.707106781,0],[-1,0,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
11 CONST robtarget Target_110:=[[8912,148.902,1082.049],[0.707106781,0,0.707106781,0],[-1,0,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
12 CONST robtarget Target_90:=[[9112,148.902,1082.049],[0.707106781,0,0.707106781,0],[-1,0,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
13 PROC Path_10()
14
15     MoveL homeR1,v1000,z100,GripperMagnetic\Wobj;mwobj0;
16     SetDO cr_1,1;
17     WaitTime 1.01;
18     SetDO cr_2,1;
19     WaitTime 1.01;
20     SetDO cr_3,1;
21     WaitTime 1.01;

```

Fig. 10. Program cod in RAPID.

În programul din figura 10 sunt 4 tipuri de comenzi pe care se bazează această simulare: definirea unei constante (definirea punctelor target) prima paranteză a acestei comenzi reprezintă poziția în spațiu în coordonatele X Y Z, a doua paranteză reprezintă orientarea în spațiu a axelor, a 3a reprezintă configurația de mișcare a axelor iar ultima paranteză reprezintă axele externe ale robotului (exemplu axa de translație la sol) care pot fi în număr de maxim 6. Al doilea tip de comandă este comanda de mișcare, aceasta apelează punctul definit anterior, definește viteza cu care se va mișca robotul (v1000), toleranța de atingere a punctului apelat (z100) și apelează sistemul de coordonate al efectorului robotului. Comanda “SetDO” îi setează unui semnal digital valoarea True sau False (1 sau 0). Ultimul tip de comandă este comanda WaitTime sau WaitDO, această comandă îi spune robotului să aștepte un timp definit sau valoarea unui semnal digital înainte de a continua programul.

5. Concluzii

În final după realizarea acestor pași, simularea procesului poate fi rulată în mediul de lucru ABB Robot Studio și are ca avantaje urmărirea unor timpi reali, vizualizarea unor imperfecțiuni ce ar putea apărea în realitate și astfel pot fi anticipate dar și faptul că acest proiect poate fi implementat ușor cu mici modificări.

6. Bibliografie

- [1]. Nicolescu, A. – Implementarea Robotilor Industriali in Sistemele de Productie, note de curs si metodologii de proiectare, UPB, 2018.
- [2]. <http://new.abb.com/products/robotics/controllers/irc5/irc5>
- [3]. <http://agtrobotics.com/beam-rotator>
- [4]. <https://forums.robotstudio.com>