

SINTEZA ASISTATĂ 3D A UNEI CELULE DE VOPSIRE ROBOTIZATĂ A CAROSERIILOR ȘI PROGRAMAREA-SIMULAREA SA OFFLINE

GHENEA Alexandru – Claudiu¹

Conducător științific: Conf. dr. ing. **Diana POPESCU**

REZUMAT: În cadrul prezentei lucrări s-a realizat sinteza asistată parametrizată 3D CAD al unei celule complete de vopsire robotizată în câmp electrostatic având ca punct de referință analiza comparativă a mai multor celule similare destinate aceluiași scop. Modelul virtual 3D al celulei a fost integrat apoi într-un mediu de lucru dedicat, și anume ABB Robot Studio, în vederea realizării unui studiu preliminar al posibilităților de programare și simulare offline. Astfel, au fost detaliate și ilustrate pas cu pas etapele principale de lucru cu această aplicație, plecând de la importul componentelor celule robotizate, poziționarea acestora în spațiul de lucru, urmată de generarea de către software-ul utilizat a codului în format text pentru întreaga aplicație.

CUVINTE CHEIE: sinteză asistată parametrizată 3D, celulă vopsire caroserii auto, Robot Studio

1 CARACTERISTICILE APLICAȚIEI DE VOPSIRE A CAROSERIILOR AUTO

Roboții industriali utilizați în operații de acoperire a suprafețelor prin vopsire, utilizează scule de tip pistol de vopsire care includ o duză prin care se realizează pulverizarea vopselei sau împrăștierea acesteia sub formă de pulbere. Împrăștierea vopselei se poate realiza cu ajutorul aerului comprimat sau, în cazul vopsirii electrostatice, pistolul este legat la un pol al unei surse de curent, iar obiectul ce urmează a fi vopsit la celălalt pol. Astfel, particulele de vopsea se încarcă electrostatic și sunt atrase de obiect, acestea ajungând în cele din urmă chiar și pe suprafețele mai "ascunse". Comanda roboților de vopsire se face prin programe multipunct (MP) sau prin programe de traiectorie continuă (CP) [3].

2 STADIUL ACTUAL

În vederea realizării modelului virtual 3D parametrizat al aplicației de vopsire robotizată, primul pas a fost studiul aplicațiilor similare din domeniu, urmată de realizarea unei celule proprii și de programarea-simularea online a acesteia prin utilizarea unui program dedicat – ABB Robot Studio. Astfel, a fost generat codul pentru întreaga aplicație, cât și un film în care este prezentat modul de funcționare al întregii aplicații.

2.1 Celule flexibile de vopsire similare celei de proiectat

În figura 1 este prezentată o imagine a unei celule de vopsire în câmp electrostatic a caroseriilor din industria auto.

Operația de vopsire este realizată cu ajutorul a doi roboți de tip braț articulat având o configurație dedicată acestei operații. Aceștia au 6 grade de libertate, sunt deplasabili la sol pe o structură portantă care este ghidată pe o cale de rulare atașată pereților laterali ai celulei.



Fig. 1. Model de aplicație de referință [3]

În figura 2 este prezentată o altă celulă de vopsire robotizată care utilizează 4 roboți de tip braț articulat cu structură hibridă. Aceștia au structură hibridă întrucât primul grad de la bază i-a fost înlocuit de o translație de-a lungul unei structuri pe care roboții sunt suspenși. În cadrul acestei celule se regăsește și un sistem perirobotic de curățare a pistolului de vopsit, un fel de sistem de management al pistolului de vopsit.



Fig. 2. Model aplicație similară celei de proiectat [3]

¹ Specializarea Robotică, Facultatea IMST; E-mail: alexandru.claudiu.ghenea@gmail.com

2.1.1 Analiza comparativă a variantelor constructive similare de roboți industriali

S-a realizat și un studiu comparativ al mai multor variante de roboți industriali cu o construcție similară și care pot fi integrați într-o celulă de vopsire în câmp electrostatic.

Modelul prezentat în figura 3 aparține firmei FANUC și are cinci axe comandate numeric, fiind proiectat pentru viteze mari de deplasare. Acest model are segmentul doi al brațului de lungime de 1.80 m.



Fig. 3. Robot de tip braț articulat al firmei FANUC [1]

Robotul FANUC dedicat operației de vopsire combină funcțiile avansate ale modelului 5400 cu un design compact, rezultatul fiind adaptarea rapidă la orice tip de sistem de vopsire [3].

Modelul ABB IRB 580 este cel mai ușor de programat din clasa sa (figura 4), integrând un sistem unic de proces de reglare în buclă închisă, control de mare viteză și ajustări ale fluxului de vopsea și aer [3]. Spațiul de lucru al acestui robot este prezentat în figura 5, iar tabelul 1 prezintă principalele sale caracteristici tehnice.



Fig. 4. Robot ABB IRB 580 [2]

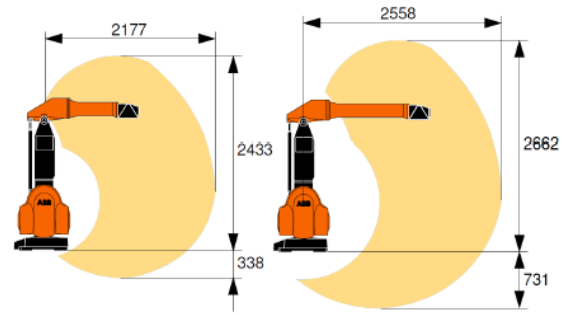


Fig. 5. Spațiul de lucru al robotului ABB IRB 580 [2]

Tabel.1. Caracteristici tehnice ABB IRB580

Mișcare axă	Spațiu de lucru	Viteza maximă
1. Rotație la bază	+1500 -1500	1120/s
2. Brat vertical	+750 -700	1120[°/s]
3. Brat orizontal	+35 0 -600	1120[°/s]
4. Rotație braț 1	nelimitată	4650[°/s]
5. Pitch	+180 -180	3500[°/s]
6. Rotație flansa(Roll)	nelimitată	5350[°/s]

2.1.2 Simularea aplicației în ABB Robot Studio

În cadrul imaginilor de mai jos sunt prezentate zonele în care se pot amplasa (poziționa) componentele din cadrul celulei cât și acțiunile ce se revarsă asupra acestora.

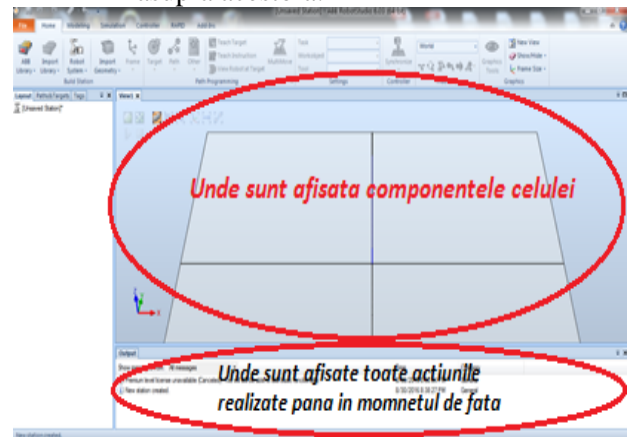


Fig.6. Amplasarea componentelor celulei

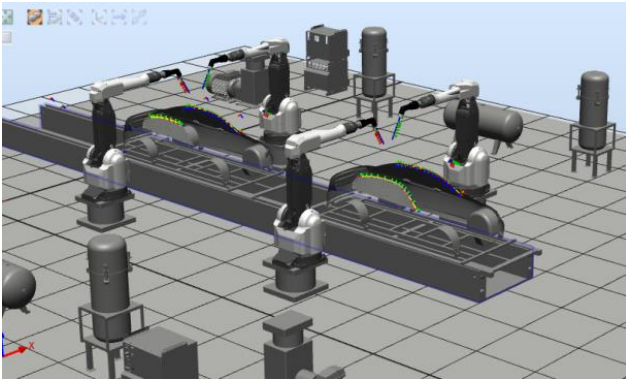


Fig.7. Aplicația completă din ABB Robot Studio

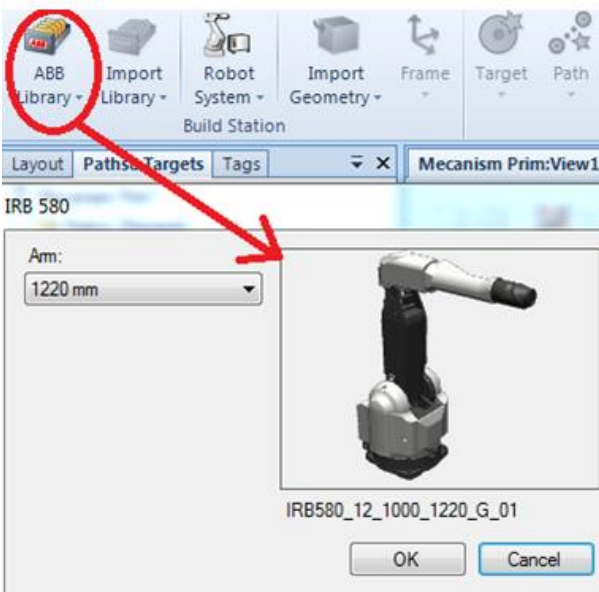


Fig.8. Inserarea modelului de robot în ABB Robot Studio

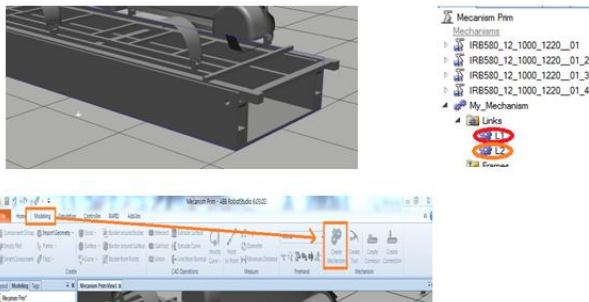


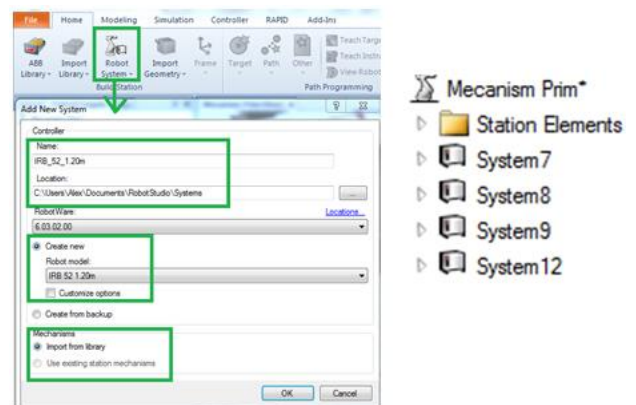
Fig. 10.Realizarea mecanismelor

Pentru a realiza mișcarea de aducere a caroseriilor auto în postul de vopsire,s-a realizat un mecanism de translație între structura de susținere a acestora și structura de bază. Acest s-a importat din cadrul programului unde există o librărie predefinită cu majoritatea modelelor de roboți. În momentul importului se poate defini și lungimea segmentului doi al brațului robotului.



Fig. 9.Importarea elementelor

Elementele componente ale celulei pot fi importate sub format STP.Se recomandă ca acestea sa fie introduse sub forma unor subsamblate alcătuite din mai multe componente .



3

Fig. 11.Crearea systemului robotului(Controller-ul)

Pentru a funcționa și pentru a realiza mișcări,fiecarui robot i se atribuie un controller.Acest lucru se face din comanda Robot System și selectăm New Controller.Practic este specificat căru robot i se atribuie și făcând acest lucru elimini modul de a face fiecărui robot un mecanism,acesta îl prefigurează automat ca și mecanism pe robot și primește și semnalele de intare permițind să știe când să pună in mișcare robotul pentru a vopsi.

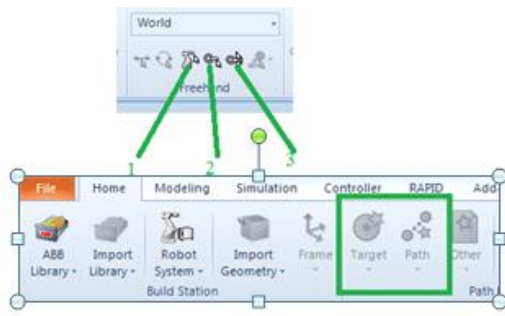


Fig. 12. Generarea traiectoriei și realizarea mișcării

Mișcarea robului se poate face selectând una din cele trei comenzi, acestea fiind cele trei moduri și anume linear, circular sau se poate selecta cel de al treilea mod în care mișcăm robotul de-a lungul unei traiectorii iar acesta memorează poziția punctelor pe care le-a atins.

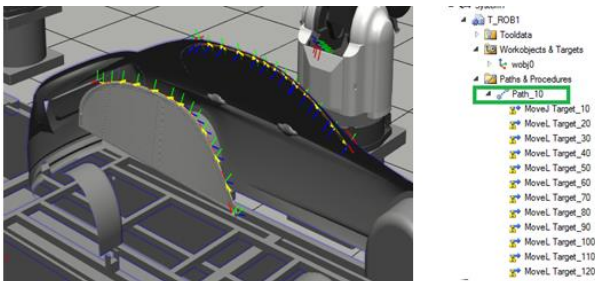


Fig. 13. Traiectorii generate pe obiect

În imaginea de mai sus se pot observa fizic traiectoriile generate pe obiect cât și sensul în care acestea trebuie urmate. Acestea se găsesc în cadrul fiecărui sistem la opțiunea "Path" și sunt sub forma de target-uri, numărul acestora variind în funcție de mișcarea pe care robotul trebuie să o facă. Pentru a urma această traiectorie trebuie să selectăm opțiunea "Move along path" dând click dreapta pe Path și apoi selectând această comandă.

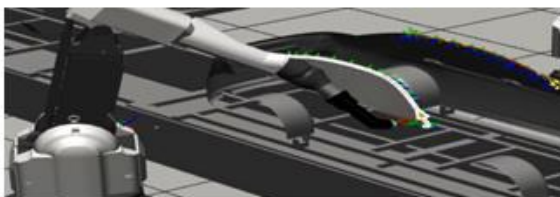


Fig. 14. Realizarea simulării mișcării de-a lungul traiectoriei

Această simulare se face în cadrul fiecărui sistem (controller) acesta se selectează apoi se selectează meniul "Simulation" apoi se generează comanda de mișcare a fiecărui robot. Aceasta se înregistrează tot din cadrul aceluiași meniu cu "Record simulation". Practic se realizează un filmuleț.

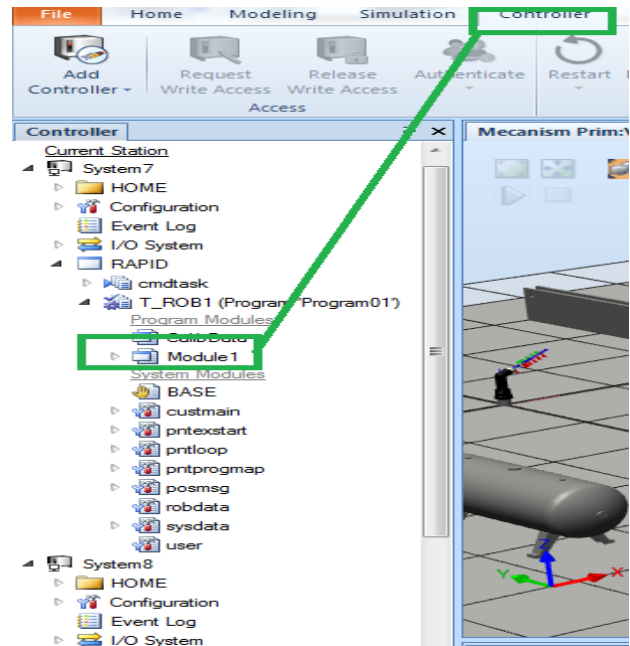


Fig. 15. Generarea automată a programului fiecărui robot

Pentru a putea vizualiza partea de cod a fiecărui robot, se selectează meniul controller apoi selectăm carui modul (robot) dorim să îi vizualizăm programul dând dublu click pe acesta. Partea de cod se va genera automat

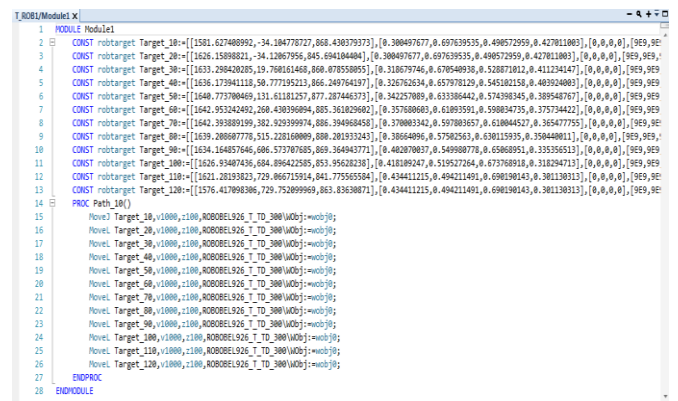


Fig. 16. Program generat în Rapid pentru primul robot

```

1  T_ROB1Module1 x
2  1  MODULE
3  2  CONST robtarget Target_18=[1585.286344655,-738.6946119,866.-507716786],[0.294157065,0.693833486,0.487652414,0.4467208
4  3  CONST robtarget Target_20=[1438.22648933,-789.612979329,844.50374141],[0.294157065,0.693833486,0.487652414,0.4467208
5  4  CONST robtarget Target_28=[1638.98455268,-641.176538989,863.372602788],[0.323110951,0.670805586,0.525170061,0.413385
6  7  CONST robtarget Target_40=[1646.989162431,-551.813513686,879.239295963],[0.348933891,0.647209536,0.55547876,0.396118
8  8  CONST robtarget Target_48=[1492.8192148,-428.189919143,898.042786663],[0.337879126,0.633127436,0.583361636,0.378972
9  7  CONST robtarget Target_60=[1655.785893369,-286.397218342,893.79215639],[0.369255681,0.607837766,0.601451504,0.362345
10 8  CONST robtarget Target_70=[1583.384940469,-143.57937832,896.04418863],[0.389440959,0.587421997,0.622348232,0.351498
11 9  CONST robtarget Target_80=[1452.584940469,-62.277297989,882.783178241],[0.392894509,0.566245476,0.642511862,0.33833628
12 10 CONST robtarget Target_90=[1646.881833993,18.8386461,869.109168429],[0.418121332,0.52787843,0.672383997,0.31931828
13 11 CONST robtarget Target_100=[1637.39181636,76.490391413,893.199151484],[0.420818976,0.486227566,0.703899122,0.2994548
14 12 CONST robtarget Target_110=[1622.86731246,151.259889876,822.71455573],[0.446289247,0.439393392,0.728867897,0.276785
15 13 CONST robtarget Target_120=[1683.385108867,222.82696499,787.45938145],[0.464895489,0.40478643,0.743471622,0.2595268
16 14 CONST robtarget Target_130=[1578.397714032,305.576965283,734.353218743],[0.487378763,0.375399915,0.751485149,0.238240014
17 15 CONST robtarget Target_140=[1549.78480187,401.115712123,663.362282472],[0.507373285,0.36184295,0.748689424,0.2227288
18 16 CONST robtarget Target_150=[1525.45866877,488.689105333,593.476941794],[0.518663874,0.361927044,0.744353776,0.214322317
19 17 CONST robtarget Target_160=[1479.8940132,491.487798999,611.982314763],[0.518663874,0.361927044,0.744353776,0.214322317
20 18 PROC Path_18()
21 19 MoveJ Target_18,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
22 20 MoveJ Target_20,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
23 21 MoveJ Target_28,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
24 22 MoveJ Target_40,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
25 23 MoveJ Target_48,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
26 24 MoveJ Target_60,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
27 25 MoveJ Target_70,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
28 26 MoveJ Target_80,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
29 27 MoveJ Target_90,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
30 28 MoveJ Target_100,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
31 29 MoveJ Target_110,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
32 30 MoveJ Target_120,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
33 31 MoveJ Target_130,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
34 32 MoveJ Target_140,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
35 33 MoveJ Target_150,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
36 34 MoveJ Target_160,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
37 35 ENDPROC
38 36 ENDMODULE

```

Fig. 17. Program generat in Rapid pentru al doilea robot

```

9  T_ROB1Module1 x
10  9  MODULE
11  10 CONST robtarget Target_08=[1652.584494683,-62.277297989,882.781678263],[0.392894509,0.56415474,0.642511862,0.338336283]
12 11 CONST robtarget Target_09=[1646.881833993,18.8386461,869.109168429],[0.418121332,0.52787843,0.672383997,0.31931828]
13 12 CONST robtarget Target_100=[1637.39181636,76.490391413,893.199151484],[0.420818976,0.486227566,0.703899122,0.2994548]
14 13 CONST robtarget Target_110=[1622.86731246,151.259889876,822.71455573],[0.446289247,0.439393392,0.728867897,0.276785]
15 14 CONST robtarget Target_120=[1683.385108867,222.82696499,787.45938145],[0.464895489,0.40478643,0.743471622,0.2595268]
16 15 CONST robtarget Target_130=[1578.397714032,305.576965283,734.353218743],[0.487378763,0.375399915,0.751485149,0.238240014]
17 16 CONST robtarget Target_140=[1549.78480187,401.115712123,663.362282472],[0.507373285,0.36184295,0.748689424,0.2227288]
18 17 CONST robtarget Target_150=[1525.45866877,488.689105333,593.476941794],[0.518663874,0.361927044,0.744353776,0.214322317]
19 18 CONST robtarget Target_160=[1479.8940132,491.487798999,611.982314763],[0.518663874,0.361927044,0.744353776,0.214322317]
20 19 PROC Path_18()
21 20 MoveJ Target_18,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
22 21 MoveJ Target_20,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
23 22 MoveJ Target_28,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
24 23 MoveJ Target_40,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
25 24 MoveJ Target_48,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
26 25 MoveJ Target_60,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
27 26 MoveJ Target_70,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
28 27 MoveJ Target_80,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
29 28 MoveJ Target_90,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
30 29 MoveJ Target_100,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
31 30 MoveJ Target_110,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
32 31 MoveJ Target_120,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
33 32 MoveJ Target_130,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
34 33 MoveJ Target_140,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
35 34 MoveJ Target_150,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
36 35 MoveJ Target_160,v1000,z100,ROBOBEL926_T_T0_380(MObj)=wobj0;
37 36 ENDPROC
38 37 ENDMODULE

```

Fig. 18. Program generat in Rapid pentru al treilea robot

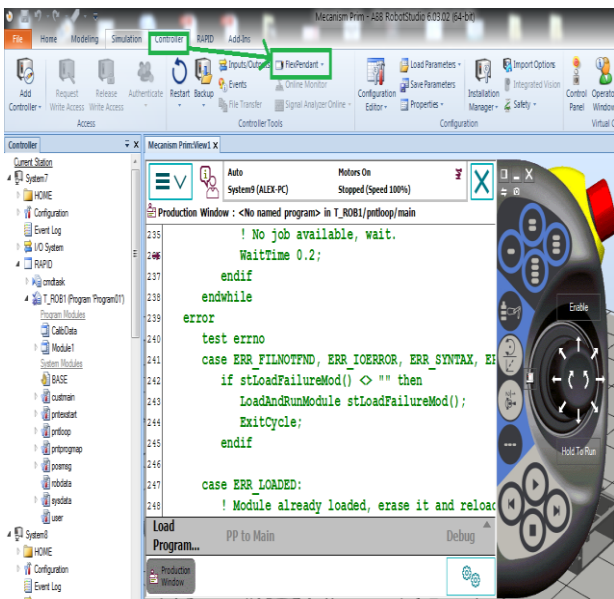


Fig. 19. Model virtual al teach-ului și vizualizarea programului

În cadrul imaginii este prezentat modelul virtual al teachpendantului. În cadrul interfeței virtuale sunt prezentate principalele comenzi ce pot pune în mișcare structura mecanică cât și codul la nivel textual ce poate fi modificat.

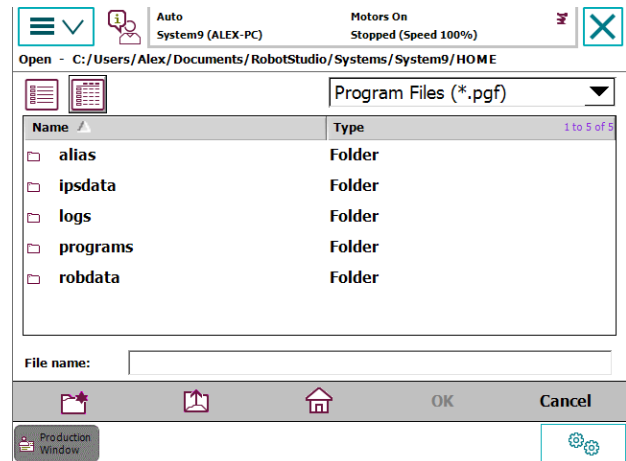


Fig. 20. Model virtual al teach-ului și posibilitatea de încărcare a unor programe deja predefinite

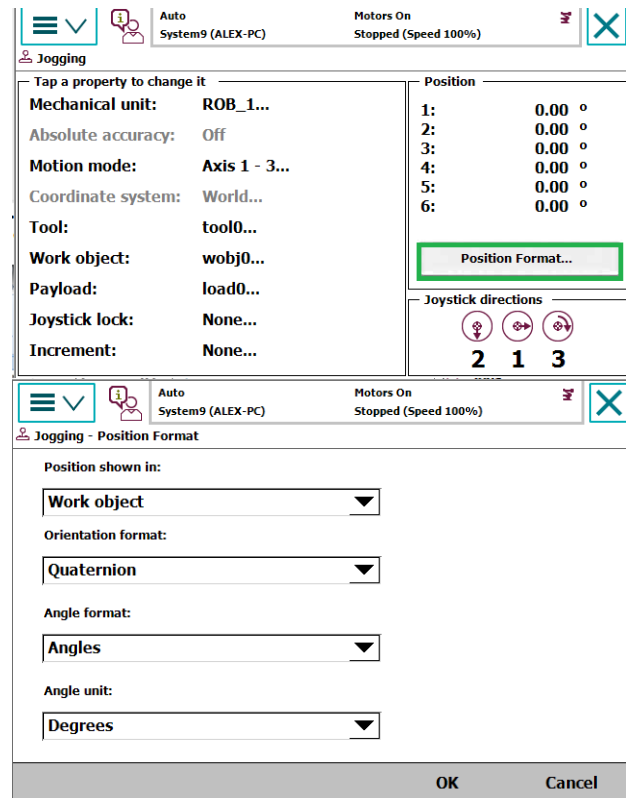


Fig. 21. Meniul jogging

În cadrul meniului jogging este prezentată entitatea (robotul), scula acestuia, numărul de axe ce pot fi comandate cât și sarcina portantă. Prin accesarea comenzii "Position Format" se pot face

modificări asupra mai multor parametri care țin de orientare.

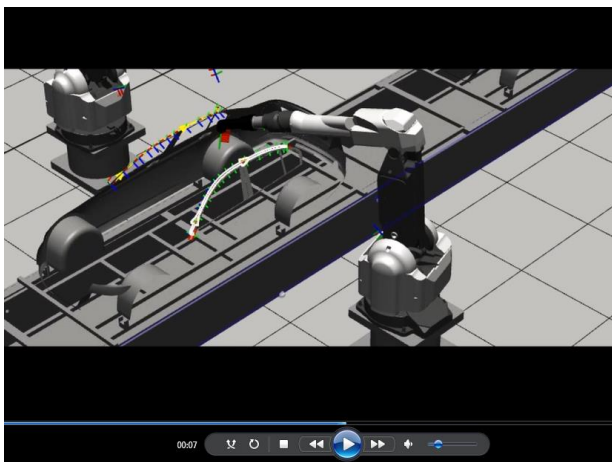
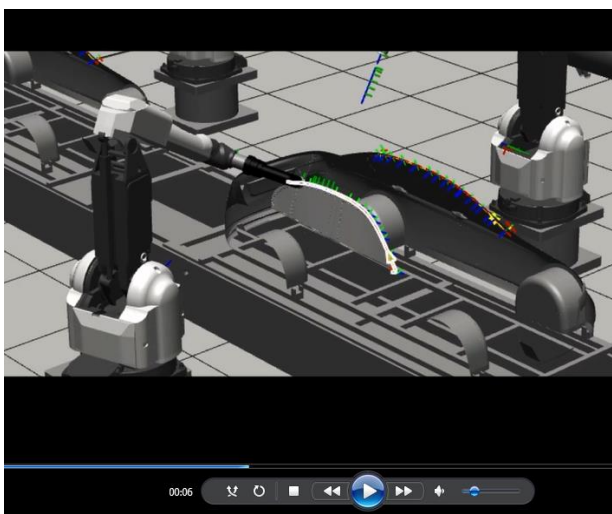
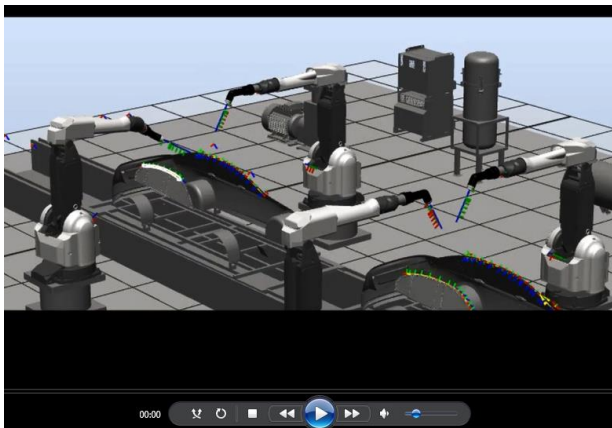


Fig. 22.Capturi succesive din cadrul simulării

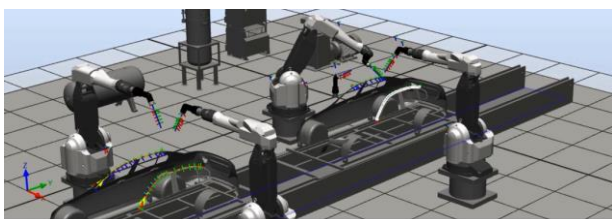


Fig. 23.Capturi succesive din cadrul simulării

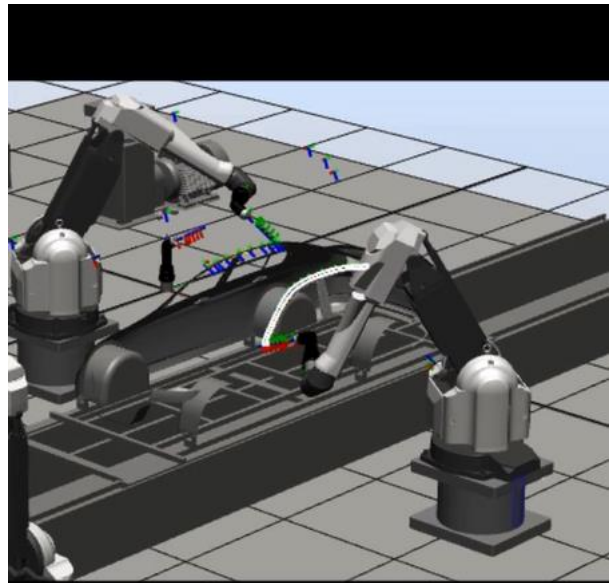


Fig. 24.Capturi succesive din cadrul simulării

În cadrul imaginilor sunt prezentate elementele majore ce alcătuiesc celula.Sunt roboți de tip braț articulat cât și restul echipamentului necesar realizării vopsirii in câmp electrostatic.Sistemul de transport al caroseriilor auto este alcătuit dintr-un mecanism de translație la sol a acestora.Mecanismul este format dintr-un suport pe care se află entitățile de tip caroserie si un sistem de acționare de tip motor-reductor si elemnt acționat.Aceste elemente au fost eliminate din simulare pentru a reduce timpii de realizare a acesteia.Odată ajunse caroseriile in dreptul postului de lucru(robot),acestia își încep programul de vopsire.

4 BIBLIOGRAFIE

- [1]. Fanuc - <http://www.fanuc.com/>.
- [2]. Abb - <http://new.abb.com/products/robotics>
- [3]. [https://www.google.ro/#q=youtube articolului](https://www.google.ro/#q=youtube+articolului)