

Sisteme robotizate adaptive pentru sudare

Ing. Alexandru CAZAN

Conducător științific: S.l.dr.ing **Corneliu RONTESCU**, S.l.dr.ing **Dumitru-Titi CICIC**

REZUMAT: Pe parcursul acestei lucrări o să prezint un sistem automat de identificare a poziției punctelor de sudare și a poziției diferențelor de mărime a lățimii rostului de sudare pentru o îmbinare cap la cap a două table. Identificarea punctelor de sudare și a diferențelor de mărime a lățimii rostului este necesară în vederea unei sudări adaptive pentru a obține o sudare de o calitate ridicată și aspectuoasă. Sudarea adaptivă permite ajustarea procesului de sudare, în vederea compensării variațiilor geometriei rostului de sudare.

CUVINTE CHEIE: robot, sudare adaptivă.

1 INTRODUCERE

Atunci când geometria pieselor, care urmează a fi sudate, nu permite realizarea unui dispozitiv pentru o fixare optimă a pieselor în vederea sudării (preluarea tuturor gradelor de libertate) și pentru a evita deformările care au loc în timpul procesului de sudare, piesele sunt prinse în puncte de sudare.

Sistemele laser de urmărire a rostului de sudare sunt realizate în vederea evitării punctelor de sudare în timpul urmăririi rostului, respectiv în timpul sudării propriu-zise. Adică în timpul sudării robotul se poate deplasa o anumită distanță (această distanță poate fi introdusă de către operator) fără să identifice rostul de sudare.

2 SCOPUL

În cazul sudurilor de mare finete este nevoie de identificarea acestor puncte de sudare, pentru schimbarea parametrilor de sudare în aceste zone (pe punctele de sudare), în vederea realizării unei îmbinări sudate de o calitate ridicată.

De asemenea, dacă tehnologia de execuție a pieselor ce urmează a fi sudate nu permite o prelucrare precisă a rostului de sudare, variațiile lățimii rostului pot fi foarte mari. În acest caz se preferă o sudare cu pendulare, dar datorită faptului că variația lățimii rostului este mare, se preferă pendulări diferite în funcție de această lățime. De aceea este importantă detecția poziției zonelor unde au loc aceste variații.

3 ECHIPAMENTE FOLOSITE

Pentru sudarea și identificarea pozițiilor punctelor de sudare și a pozițiilor zonelor în care lățimea rostului se modifică am folosit un robot industrial format dintr-un braț manipulator cu 6 axe FANUC Arc Mate-100iC/6L comandat de un controller FANUC R-30iB., o sursă de sudare digitală TIG – FRONIUS Magic Wave 2500 Job G/F și un senzor laser de urmărire a rostului Meta Vision. Brațul robot este echipat cu un sistem de avans a sârmei de sudare FRONIUS KD 1500-PAP-11m 4R/CB robot și cu un pistol de sudare FRONIUS Robacta TTW4500 PAP montat pe flanșa brațului robot prin intermediul unui senzor de coliziune Crashbox Robacta-PAP L.



Fig. 1. Braț robotic FANUC Arc Mate-100iC/6L



Fig. 2. Controller FANUC R-30iB



Fig. 3. Sursă de sudare digitală TIG – FRONIUS MagicWave 2500 Job G/F



Fig. 4. Sistem pentru urmărirea rostului Meta Vision și Pistolet de sudare – FRONIUS Robacta TTW4500 PAP

4 INTEGRAREA SISTEMULUI ADAPTIV

Pentru identificarea punctelor de sudare am folosit echipamentele prezentate în capitolul 3. Sudarea adaptivă permite ajustarea procesului de sudare, în vederea compensării variațiilor geometriei rostului de sudare. Sudarea adaptivă lucrează în paralel cu urmărirea rostului de sudare, care ajustează traiectoria TCP-ului (tool center point), pentru a compensa variațiile geometriei rostului de sudare.

Una din probele folosite în scopul acestei cercetări este prezentată în figura 5. Materialul de bază este un oțel inoxidabil austenitic 304L. Probele au o lungime de 150 mm cu o variație a rostului de la 1,6 la 2,5 mm.



Fig. 5. Probă înainte de sudare

Pentru identificarea punctelor de sudare, precum și a variației lățimii rostului de sudare, a fost necesar să despart procesul de scanare de cel de sudare. Acest lucru a fost necesar deoarece dacă scanarea și sudarea s-ar realiza concomitent, atunci când senzorul laser ar identifica punctul de sudare sau că lățimea rostului de sudare s-a modificat, electrodul de wolfram fiind poziționat în spatele senzorului laser, schimbarea parametrilor de sudare se realiza în altă zonă decât în cea dorită. Din acest motiv, în prima fază am realizat o scanare a rostului, am identificat pozițiile unde sunt variații ale rostului și pozițiile de început și sfârșit ale punctelor de sudare, am memorat aceste poziții, după care am realizat sudarea propriu-zisă.

Deoarece electrodul de sudare se află între senzorul laser și sârma de sudare (fig. 4), sudarea am realizat-o în sens invers scanării.

Realizarea unei sudări adaptive cu aceste echipamente, se realizează apelând în programul robotului instrucțiunea Track SENSOR[20] RPM[1], unde valoarea 20 face legătura între robot și senzorul laser, pentru ca senzorul laser să știe ce tip de rost urmează a fi urmărit. Pot exista până la 20 de modele diferite de rost memorate în memoria senzorului laser. Senzorul laser începe să urmărească rostul de sudare doar după ce rostul a

fost identificat. Această identificare se face prin instrucțiunea SENSOR SEARCH POINT[20].

Nu a fost de ajuns folosirea acestei metode de sudare adaptivă integrată, pentru că după cum am menționat și mai sus, prin această metodă nu se pot schimba parametrii de sudare și amplitudinea pendulării în zona punctelor de sudare și în zona unde valoarea lățimii rostului de sudare se modifică.

Deoarece scanarea și sudarea nu s-a putut realiza concomitent, am folosit funcția **RPM (Root Pass Memorization)** a robotului. RPM înregistrează datele cu privire la geometria rostului de sudare, cum ar fi lățimea rostului de sudare, aria rostului de sudare (pentru îmbinări cu rost în V, U, Y, etc.), precum și traiectoria optimă a TCP-ului, în vederea realizării unui proces de sudare de o precizie ridicată. Funcția RPM este folosită în paralel cu funcția **multi-pass (MP)**. Funcția MP folosește datele stocate în buffer-ul RPM.

4.1 Root Pass Memorization

Root pass memorization (RPM) este o funcție care înregistrează date cu privire la geometria rostului de sudare la un anumit interval (pitch), în timp ce scanează sau în timp ce sudează stratul de rădăcină. În afară de offset-urile x, y, z, care determină locația exactă a rostului, în buffer-ul RPM mai sunt stocate și alte informații: lățimea rădăcinii (gap), diferența dată de nealinierea tablelor (mismatch), aria (area), S1, S2, S3, S4 și S5. Aceste informații sunt redată în timpul trecerilor ulterioare (în cazul sudării multistrat sau în cazul în care se dorește să se realizeze o scanare înainte de sudarea propriu-zisă) și sunt folosite la sudarea adaptivă.

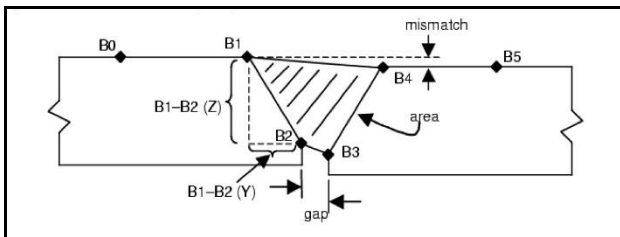


Fig. 6. Informațiile transmise robotului de către senzorul laser Meta Vision

În cele mai multe cazuri, informațiile despre lățimea rădăcinii, nealiniere și arie, sunt suficiente pentru a realiza sudarea adaptivă. În orice caz, în plus față de aceste variabile, se pot defini și folosi variabile derivate S1÷S5, pentru sudarea adaptivă. Cele 3 posibilități de folosire ale acestor variabile sunt:

- Distanța dintre două puncte (exemplu: B1-B2)
- Distanța dintre două puncte pe coordonata Y (exemplu: B1-B2Y)
- Distanța dintre două puncte pe coordonata Z (exemplu: B1-B2Z).

Program detail				
Joint data & limits				1/8
Variable	Nominal (mm)	Low (mm)	High (mm)	Error
1 Gap	0.0	0.0	0.0	NONE
2 Mismatch	0.0	0.0	0.0	NONE
3 Area (mm ²)	0.0	0.0	0.0	NONE
4 S1: **- ***	0.0	0.0	0.0	NONE
5 S2: **- ***	0.0	0.0	0.0	NONE
6 S3: **- ***	0.0	0.0	0.0	NONE
7 S4: **- ***	0.0	0.0	0.0	NONE
8 S5: **- ***	0.0	0.0	0.0	NONE

Fig. 7. Setare parametrul sudării adaptive

În funcție de limitele introduse aici, robotul poate reacționa în 3 moduri: se poate opri (va afișa un mesaj de eroare dacă a depășit limitele introduse), poate transmite un mesaj către operator că limitele au fost depășite, sau poate să nu facă nimic, cum este cazul de față.

Cum funcționează RPM

RPM înregistrează offset-urile poziției și datele cu privire la geometria rostului de sudare în memoria SRAM (Static Random Access Memory) a controller-ului. Aceasta îi permite controller-ului să rețină informațiile în cazul căderilor de tensiune. Odată înregistrate, informațiile RPM rămân în buffer, până când același buffer este rescris (sunt 10 buffer-uri care pot fi folosite). Înregistrarea informațiilor încep simultan cu mișcarea și urmărirea. Această înregistrare a informațiilor se face între instrucțiunile TRACK START și TRACK END. Începând de la rularea instrucțiunii TRACK START, offset-urile și datele despre geometria rostului sunt înregistrate, când robotul ajunge la distanța specificată (dată de pas) de-a lungul traiectoriei următoare de robot.

Pasul (PITCH) se poate introduce ca timp (senzorul laser transmite robotului informații la un anumit interval de timp) sau ca distanță (senzorul laser transmite robotului informații la o anumită distanță).

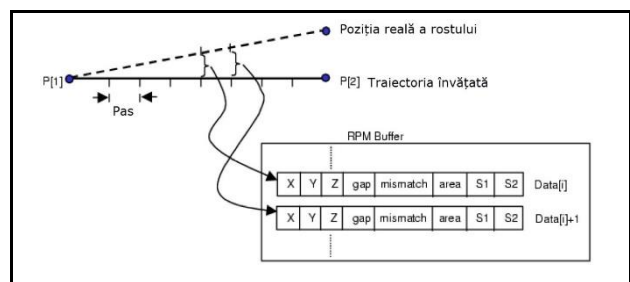


Fig. 8. Ajustarea traiectoriei de sudare

4.2 Multi-Pass

Scopul înregistrării datelor referitoare la poziția reală a rostului de sudare, precum și a geometriei rostului este pentru a folosi aceste informații la sudarea multistrat. Sudarea multistrat este folosită la sudarea tablelor groase, unde îmbinarea sudată nu se poate realiza dintr-o singură trecere. Îmbinarea sudată în acest caz se realizează sudând în straturi multiple. Practic, pentru a suda aceste tipuri de îmbinări, robotul urmărește aceeași traiectorie, dar cu diferite offset-uri. Activarea, respectiv dezactivarea funcției Multi-Pass se face cu două instrucțiuni:

- **MP OFFSET PR[...] RPM[...]**
- **MP OFFSET END**

PR[...] îți permite să schimbi traiectoria de sudare a robotului, precum și orientarea pistolului de sudare. RPM[...] îți permite să specifici ce buffer RPM folosești, din cele 10 existente.

4.3 Prezentarea sumară a programelor folosite la scanare și la sudare

Pentru a realiza scanarea și sudarea a fost nevoie să folosesc 4 programe diferite, denumite astfel:

- ✓ **TDW_CALL** - program de selecție scanare/sudare
- ✓ **TDW_SCAN_SUD** - program care conține instrucțiunile de mișcare pentru sudare/scanare și care apelează în paralel programul **TDW_HAFT_MEM** sau **TDW_JOB_CHG**
- ✓ **TDW_HAFT_MEM** - program care rulează în paralel cu **TDW_SCAN_SUD** în timpul scanării. Acest program conține logica de memorare a poziției punctelor de sudare, precum și a poziției modificării lățimii rostului de sudare.
- ✓ **TDW_JOB_CHG** - program care rulează în paralel cu **TDW_SCAN_SUD** în timpul sudării. Acest program conține logica de schimbare a job-urilor, unde sunt modificați parametrii de sudare, precum și amplitudinea pendulării.

Pentru a putea folosi informațiile înregistrate în buffer-ul RPM, programul de scanare și de sudare trebuie să fie același. Pentru a face acest lucru am folosit regiștrii argument. Atunci când în interiorul unui program este apelat un sub-program, acesta din urmă poate fi apelat ca un registru argument.

Exemplu:

Conținutul programului **TDW_CALL** este:

- **CALL TDW_SCAN_SUD(1) // apelare program TDW_SCAN_SUD. În acest caz AR[1]=1**
- **CALL TDW_SCAN_SUD(2) // apelare program TDW_SCAN_SUD. În acest caz AR[1]=2**

Conținutul programului **TDW_SCAN_SUD** este:

- ✓ **R[140]=AR[1] // Registrul 140 preia valoarea argumentului 1, care poate fi 1 sau 2**
- ✓ **IF R[140]=2, JMP LBL [1] // Dacă R[140] are valoarea 1, programul rulează instrucțiunile pentru scanarea și memorarea traiectoriei, până când ajunge la END. Dacă R[140] are valoarea 2, programul sare peste partea de scanare și memorare și rulează ce este sub LBL [1], adică instrucțiunile pentru sudare.**
- ✓ **Instrucțiuni scanare și memorare traiectorie optime de sudare în buffer-ul RPM**
- ✓ **END**

- ✓ **LBL [1]**
- ✓ **Instrucțiuni sudare folosind buffer-ul RPM**
- ✓ **END**

Pentru identificarea poziției punctelor de sudare, precum și a poziției modificării lățimii rostului de sudare, am folosit două programe care rulează în paralel. Acest lucru am făcut și în cazul schimbării parametrilor de sudare.

În primul program (**TDW_SCAN_SUD**) sunt cuprinse instrucțiunile de mișcare pentru a realiza scanarea, respectiv sudarea, precum și apelarea sub-programelor care rulează în paralel cu acesta pentru memorarea poziției punctelor de sudare, respectiv a poziției modificării lățimii rostului de sudare. Programul **TDW_SCAN_SUD** este prezentat mai jos și conține explicații referitoare la instrucțiunile folosite în interiorul acestuia.

În continuare voi prezenta un extras din programul **TDW_SCAN_SUD**. Lipsesc unele linii din program, deoarece pe consola robotului sunt niște spații libere pentru a fi urmărit mai ușor. Cu culoarea albastru sunt scrise explicațiile pentru fiecare instrucțiune.

Extras program TDW_SCAN_SUD

```

1: UFRAME_NUM[GP1]=3 ; // setarea mesei de lucru
2: UTOOL_NUM[GP1]=3 ; // setarea uneltei de lucru
5: $RPM_PG.SPITCH_MODE=1 ; // setarea modului de memorare a buffer-ului RPM
6: $RPM_PG.SPITCH=100 ; // setarea timpului de memorare a informațiilor referitoare la rostul de sudare
9: R[140:AR_SCAN_SUD]=AR[1] ; // preluare valoare AR[1]
10: IF R[140:AR_SCAN_SUD]=2,JMP LBL[1] ; // testare registru
12: F[32:HAFT_START]=(OFF) ; // dezactivare flag 32
13: F[33:ST_SCAN_SUD]=(OFF) ; // dezactivare flag 33
14: F[34:HAFT_START]=(OFF) ; // dezactivare flag 34
15: F[35:HAFT_END]=(OFF) ; // dezactivare flag 35
17: J P[1] 10% CNT100 ; // mișcare în punctul 1
18: J P[2] 10% CNT100 ; // mișcare în punctul 2
20: SENSOR ON[1] ; // pornire laser
22: L P[3] 300mm/sec FINE ; // mișcare liniară în punctul 3
24: WAIT .50(sec) ; // așteptare 0.5 secunde
26: SENSOR SEARCH START PR[1:SENSOR_REG] ; // memorare poziție început de rost în PR[1]
27: SENSOR SEARCH POINT[20] ; // început căutare rost de sudare
28: SENSOR SEARCH END ; // sfârșit căutare rost de sudare
31: F[33:ST_SCAN_SUD]=(ON) ; // activare flag 33
32: RUN TDW_HAFT_MEM ; // rulare program în paralel
34: Track SENSOR[20] RPM[1] ; // urmărirea rostului și memorarea informațiilor cu privire la acesta în buffer-ul 1
35: L PR[1:SENSOR_REG] 30cm/min FINE ; // mișcare liniară în registrul de poziție PR[1]
37: L P[4] 30cm/min FINE ; // mișcare liniară în punctul 4
39: Track End ; // oprire urmărire rost de sudare
40: SENSOR OFF ; // oprire laser
41: F[33:ST_SCAN_SUD]=(OFF) ; // dezactivare flag 33
42: F[32:HAFT_START]=(OFF) ; // dezactivare flag 32
45: L P[5] 300cm/min FINE ; // mișcare liniară în punctul 5
47: END ; // sfârșit program
48: ;
49: LBL[1] ; // eticheta 1
51: WAIT 1.00(sec) ; // așteptare 0.5 secunde
53: R[160:INCR_REG_POZ]=R[160:INCR_REG_POZ]-1 ; // decrementare R[160] cu valoarea 1
56: $AWSPCR.$ARC_OTF_ENA=1 ; // activare primire comenzi referitoare la sudare on the fly (în timp real)
57: $WV_OTF_GP[1].$OTF_ENABLE=1 ; // activare primire comenzi referitoare la pendulare on the fly (în timp real)
60: J P[5] 10% CNT100 ; // mișcare în punctul 5
62: F[33:ST_SCAN_SUD]=(ON) ; // activare flag 33
63: RUN TDW_JOB_CHG ; // rulare program în paralel
65: L P[6] 300cm/min FINE ; // mișcare liniară în punctul 6
67: Arc Start[1] ; // instrucțiune începerea procesului de sudare
68: WAIT 1.00(sec) ; // așteptare 1 secundă
69: Weave Sine[1] ; // instrucțiune începerea procesului de pendulare
72: LOCK PREG ; // blocare modificare regiștrii de poziție
73: MP Offset PR[98] RPM[1] ; // instrucțiune alocare offset la buffer-ul 1
75: L P[4] WELD_SPEED CNT100 ; // mișcare liniară în punctul 4
77: L PR[1:SENSOR_REG] WELD_SPEED CNT100 ; // mișcare liniară în registrul de poziție PR[1]
79: MP Offset End ; // instrucțiune sfârșit offset
80: F[33:ST_SCAN_SUD]=(OFF) ; // dezactivare flag 33
83: L P[7] WELD_SPEED FINE ; // mișcare liniară în punctul 7
85: Arc End[1] ; // instrucțiune oprire proces de sudare
86: Weave End ; // instrucțiune oprire proces de pendulare
87: WAIT 1.00(sec) ; // așteptare 1 secundă
89: $AWSPCR.$ARC_OTF_ENA=0 ; // dezactivare primire comenzi referitoare la sudare în timp real
90: $WV_OTF_GP[1].$OTF_ENABLE=0 ; // dezactivare primire comenzi referitoare la pendulare în timp real
91: UNLOCK PREG ; // deblocare modificare regiștrii de poziție
93: L P[2] 300cm/min FINE ; // mișcare liniară în punctul 2
/END ; // sfârșit program

```

Structura programului TDW_HAFT_MEM**1. Citire în buclă valoare STATUS SENZOR.**

Senzorul laser transmite robotului dacă “vede” rostul de sudare sau nu printr-o variabilă. Senzorul laser transmite robotului, prin această variabilă, valoarea 12 atunci când nu “vede” rostul și valoarea 3 atunci când îl “vede”.

2. Citire în buclă valoare lățime rost.

Senzorul laser transmite robotului lățimea rostului printr-o variabilă. În interiorul programului am făcut 5 interogări ale acestei variabile, am memorat aceste valori într-un registru numeric, după care am făcut media aritmetică între ele. Am făcut acest lucru pentru a ști cu o precizie mai mare dacă mărimea rostului s-a modificat sau nu. După fiecare citire, am așteptat 0.04 secunde, deoarece senzorul laser transmite informații robotului cu 25 frame-uri pe secundă. Acest timp de așteptare a fost necesar pentru a nu interoga aceeași valoare transmisă de senzorul laser de mai multe ori. Pentru probele sudate variația rostului a fost de la $1.6 \div 2.5$ mm. Pentru a suda probele cu această variație a rostului am folosit 4 intervale ($1.6 \div 1.8$; $1.81 \div 2.0$; $2.01 \div 2.3$; $2.31 \div 2.5$).

Atunci când valoarea rostului trece în alt interval, memorez poziția curentă a brațului robot într-un registru de poziție. Odată cu memorarea poziției brațului robot, memorez și valoarea numerică pentru acest interval activ, printr-un număr în același registru de poziție. În funcție de această valoare numerică, schimb parametrii de sudare, prin job-uri, prin programul TDW_JOB_CHG. Aceste valori numerice sunt de fapt numerele job-urilor folosite la sudare. Ele sunt:

51 – valoarea numerică folosită pentru intervalul 1 ($1.6 \div 1.8$);

52 – valoarea numerică folosită pentru intervalul 2 ($1.81 \div 2.0$);

53 – valoarea numerică folosită pentru intervalul 3 ($2.01 \div 2.3$);

54 – valoarea numerică folosită pentru intervalul 4 ($2.31 \div 2.5$);

55 – valoarea numerică folosită pentru puncte de sudare aflate în intervalul 1 ($1.6 \div 1.8$);

56 – valoarea numerică folosită pentru puncte de sudare aflate în intervalul 2 ($1.81 \div 2.0$);

57 – valoarea numerică folosită pentru puncte de sudare aflate în intervalul 3 ($2.01 \div 2.3$);

58 – valoarea numerică folosită pentru puncte de sudare aflate în intervalul 4 ($2.31 \div 2.5$).

Dacă valoarea rostului rămâne în același interval și dacă scanarea este activă (Verific dacă scanarea este activă sau nu printr-un semnal virtual, și anume flag-ul F[33] din programul TDW_SCAN_SUD prezentat mai sus) atunci programul rulează în buclă pasul 1 și pasul 2, până când se întâmplă unul din aceste două evenimente (fie senzorul nu mai “vede” rostul, fie se modifică intervalul rostului).

3. Memorare poziție robot - schimbare interval rost. Programul rulează acest pas atunci când valoarea lățimii rostului este cuprinsă în intervalul 1. Deoarece sunt 4 intervale, în continuarea programului mai sunt încă 3 zone unde este realizată memorarea schimbărilor de interval. După fiecare memorare, atunci când lățimea rostului este modificată, programul rulează în buclă pasul 1 și pasul 2 până când are loc următorul eveniment.

4. Memorare poziție robot - început punct de sudare. Programul rulează acest pas atunci când variabila care transmite statusul senzorului laser are valoarea 12. Aici memorez poziția robotului, după care verific de 3 ori la un interval de 0.04 secunde dacă variabila care transmite statusul senzorului are tot valoarea 12. După verificare, odată cu memorarea poziției robotului, memorez și că la această poziție este începutul unui punct de sudare, tot printr-un număr în acest registru de poziție. Deoarece sudarea se face în sens invers scanării, începutul punctului de sudare, în cazul sudării este sfârșitul punctului de sudare. Deci, valoarea numerică memorată în acest caz, este valoarea dată de intervalul activ în acel moment. Odată cu memorarea poziției și a valorii numerice, tot aici activez un semnal virtual pentru a ști că a fost detectat începutul punctului de sudare. Acest semnal este flag-ul F[32], care apare și în programul TDW_SCAN_SUD, prezentat mai sus. După această memorare programul rulează în buclă pasul 1 și pasul 2 până când are loc următorul eveniment.

5. Memorare poziție robot - sfârșit punct de sudare. Programul rulează acest pas atunci când variabila care transmite statusul senzorului laser are valoarea 3 și flag-ul F[32] este activ. Aici memorez poziția robotului, după care verific de 3 ori la un interval de 0.04 secunde dacă această variabilă are tot valoarea 3. Ca și mai sus, odată cu memorarea poziției robotului, memorez și valoarea numerică specifică începutul punctului de sudare (în cazul sudării). Această valoare

numerică este dată de intervalul activ în acel moment. După aceste memorări dezactivez semnal virtual F[32] pentru a ști că a fost detectat sfârșitul punctului de sudare, după care programul rulează în buclă pasul 1 și pasul 2 până când are loc următorul eveniment.

Acest program rulează în buclă așteptând schimbarea de evenimente, până când flag-ul F[33] este dezactivat. Activarea și dezactivarea acestui semnal se face din programul TDW_SCAN_SUD. Activarea se face după ce urmărirea rostului a început (vezi linia 31 din programul TDW_SCAN_SUD), înainte de rularea în paralel a programului TDW_HAFT_MEM. Dezactivarea acestui semnal se face după ce urmărirea rostului a fost oprită (vezi linia 41 din programul TDW_SCAN_SUD).

Structura programului TDW_JOB_CHG

1. Citire valori registru poziție. Deoarece probele au fost prevăzute cu plăcuțe de capăt, ultima poziție în care are loc un eveniment, este identificarea plăcuței de capăt de la sfârșitul scanării ca fiind început de punct de sudare, deoarece senzorul laser nu “vede” rostul de sudare, statusul senzorului laser fiind 12. Deci, ultimul registru de poziție nu se interoghează. Aici se citește valoarea coordonatei pe X a robotului unde a avut loc următorul eveniment. În cazul acestor probe, robotul se deplasează în lungul coordonatei X a mesei de lucru.

2. Citire și comparare în timp real a valorii pe axa X a robotului în buclă. După ce se citește valoarea coordonatei pe X unde va avea loc următorul eveniment (*exemplu: această valoare este 523,9*), programul citește în buclă în timp real valoarea pe axa X a robotului și o compară cu valoarea 523,9, adică cu valoarea unde va avea loc următorul eveniment. Datorită faptului că rularea programului nu se face instantaneu, pentru a evita ca brațul robot să treacă prin această coordonată și să nu ia nici o decizie datorită întârzierii, am creat un interval. La valoarea 523,9 am scăzut și am adăugat 0,1 mm. Atunci când brațul robot va avea valoarea pe axa X cuprinsă în intervalul $523,8 \div 524$ va ieși din buclă și va face următorul pas.

3. Citire valoare numerică folosită pentru schimbarea job-ului de sudare. În momentul în care robotul a ajuns la coordonata din registrul de

poziție, se citește cealaltă valoare memorată în acest registru care îmi dă de fapt numărul job-ului pe care îl voi folosi pentru a suda din acest punct până la coordonata unde va avea loc următorul eveniment. După ce modific job-ul de sudare, amplitudinea pendulării și viteza de sudare, programul rulează în buclă pasul 1 și pasul 2, până când are loc următorul eveniment.

Acest program rulează în buclă așteptând schimbarea de evenimente, până când flag-ul F[33] este dezactivat. Activarea și dezactivarea acestui semnal se face din programul TDW_SCAN_SUD la linia 62, respectiv la linia 80.

5 CONCLUZII

Cu ajutorul acestor 4 programe am reușit să identific cu succes punctele de sudare, precum și zonele unde se modifică valoarea lățimii rostului de sudare. O probă sudată este prezentată în figura de mai jos. Cu echipamentele disponibile în această celulă nu am reușit să găsim parametrii optimi de sudare pentru a topi punctele de sudare, deoarece aveam nevoie de un electrod de $\varnothing 3.2$ mm și sârmă de sudare de 1.2 mm.



Fig. 9. Probă după de sudare

Problemele avute pe parcursul acestei cercetări au fost:

✓ Sensorul laser transmitea robotului informații greșite cu privire la dimensiunea lățimii rostului. Din această cauză nu puteam identifica cu exactitate modificările lățimii rostului și începutul punctelor de sudare. Tot din această cauză uneori pe punctul de sudare sensorul laser “vedea” rost. Acest lucru se întâmpla la punctele de sudare date în intervalul 1.6-1.8 mm, care erau date sub cota superioară a probelor, deoarece distanța între marginile topite ale probelor era mai mică de 2.5 mm. Am rezolvat această problemă filtrând informațiile trimise de către senzor. Această filtrare am făcut-o restrângând foarte mult intervalele în funcție de care senzorul accepta ca fiind bună o citire a rostului. În plus față de

această filtrare, senzorul laser transmite informații robotului doar după ce face 10 citiri consecutive bune.

✓ Datorită faptului că inițial nu verificam dacă senzorul laser chiar a identificat un punct de sudare sau doar a avut o citire eronată luam decizii greșite cu privire la identificarea de început a punctului de sudare. De asemenea și identificarea sfârșitului era eronată. Am corectat acest lucru, făcând 3 citiri ale statusului senzorului laser și doar după ce toate aceste citiri sunt bune robotul ia decizii.

✓ Din cauză că nu aveam un timp de așteptare după ce făceam o citire a statusului senzorului, preluam aceeași informație de mai multe ori. Atât în cazul identificării punctelor de sudare, precum și în cazul identificării dimensiunii lățimii rostului de sudare. Am rezolvat această problemă așteptând 0,04 secunde după fiecare preluare de informație de la senzor.

Această metodă de a identifica și memora poziția punctelor de sudare și poziția modificării lățimii rostului de sudare are un dezavantaj, și anume că pentru a fi precisă, pasul pentru memorarea datelor cu privire la geometria rostului de sudare în buffer-ul RPM trebuie să fie cât mai mic. Adică precizia de memorare ale acestor date este cu atât mai mare cu cât memorarea datelor transmise de senzor se fac la o distanță cât mai mică sau la un timp cât mai scurt. Pasul folosit de mine pentru memorarea informațiilor despre geometria rostului în cadrul acestor teste a fost de 100 milisecunde. Ținând cont că viteza de scanare a fost de 30 cm/min, rezultă o precizie de identificare a rostului de 0,5 mm. După cum am spus mai sus, toate aceste informații cu privire la geometria rostului sunt stocate în SRAM (Static Random Access Memory), deci această memorie este limitată. Această metodă de identificare a poziției punctelor de sudare și a poziției modificării lățimii rostului de sudare nu se poate folosi în cazul sudurilor foarte lungi. În manual nu este precizată această limitare, dar ea sigur există. Probabil acest spațiu de memorare ar putea fi mărit, ținând cont că sunt 10 astfel de buffere. E posibil ca ele să se poată contopi și să rezulte un singur spațiu de stocare de 10 ori mai mare. Probabil controller-ul robotului se poate îmbunătăți achiziționându-se o memorie SRAM extinsă la comandă.

6 BIBLIOGRAFIE

1. Manuale tehnice FANUC
2. Manuale tehnice Meta Vision
3. Manuale tehnice FRONIUS