



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI
ȘI PROTECȚIEI SOCIALE
AMPOSDRU



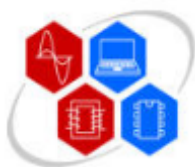
Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
CERCETĂRII
TINERETULUI
ȘI SPORTULUI
OIPOSDRU



UNIVERSITATEA “DUNĂREA DE JOS” GALAȚI

FACULTATEA DE AUTOMATICĂ, CALCULATOARE,
INGINERIE ELECTRICĂ ȘI ELECTRONICĂ



TEZĂ DE DOCTORAT

CONTRIBUȚII LA CONDUCEREA INTELIGENTĂ A ROBOȚILOR MOBILI UTILIZAȚI ÎN LINIILE FLEXIBILE DE FABRICAȚIE

Conducator Științific:

Prof. dr. ing. Adrian FILIPESCU

Doctorand:

Ing. Adrian RADASCHIN

GALAȚI
2012



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI
ȘI PROTECȚIEI SOCIALE
AMPOSDRU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
CERCETĂRII
TINERETULUI
ȘI SPORTULUI
OIPOSDRU



Investește în oameni !

FONDUL SOCIAL EUROPEAN

Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007 – 2013

Axa prioritară 1 ”Educație și formare profesională în sprijinul creșterii economice și dezvoltării societății bazate pe cunoaștere”

Domeniul major de intervenție 1.5 ”Programe doctorale și postdoctorale în sprijinul cercetării”

Titlul proiectului: Eficientizarea activității studenților din cadrul ciclului de studii doctorale - EFICIENT

Numărul de identificare al contractului: POSDRU/88/1.5/S/61445

TEZĂ DE DOCTORAT

CONTRIBUȚII LA CONDUCEREA INTELIGENTĂ A ROBOȚILOR MOBILI UTILIZAȚI ÎN LINIILE FLEXIBILE DE FABRICAȚIE

Domeniul: *Ingineria Sistemelor*

Doctorand: Ing. Adrian RADASCHIN

Componența Comisiei de doctorat:

PREȘEDINTE: Conf.dr.ing. Emilia PECHEANU
Universitatea “Dunărea de Jos” din Galați

CONDUCĂTOR ȘTIINȚIFIC: Prof.dr.ing. Adrian FILIPESCU
Universitatea “Dunărea de Jos” din Galați

REFERENT OFICIAL: Conf.dr.ing. Alina VODĂ
University JOSEPH FOURIER Grenoble 1/CNRS, Grenoble
Image Parole Signal Automatique (GIPSA-lab), Franța

REFERENT OFICIAL: Conf.dr.ing. Eugenia MINCĂ
Universitatea “Valahia” din Târgoviște

REFERENT OFICIAL: Prof.dr.ing. Viorel MÎNZU
Universitatea “Dunărea de Jos” din Galați

Prefață

Această teză constituie rezultatul activității de cercetare în perioada octombrie 2009 – septembrie 2012 în domeniul Ingineriei Sistemelor din cadrul Facultății de Automatică, Calculatoare, Inginerie Electrică și Electronică a Universității „Dunărea de Jos” din Galați. Întreaga perioadă de cercetare a fost finanțată cu sprijinul Programului Operațional Sectorial Dezvoltarea Resurselor Umane – Eficientizarea activității din cadrul ciclului de studii doctorale (EFICIENT)- POSDRU/88/1.5/S/61445 și doresc, pe această cale, să adresez mulțumiri către director și managerii acestui proiect.

Menționez că cercetarea și teza de doctorat s-a realizat cu suportul logistic și financiar din cadrul proiectului UEFISCDI IDEI, număr proiect PN-II-ID-PCE-2011-3-0641.

Doresc să-mi exprim profunda recunoștință conducătorului de doctorat, domnului Prof.dr.ing. Adrian Filipescu pentru îndrumarea acordată, implicarea și recomandările făcute și pentru suportul moral și înțelegere oferită în toată această perioadă de cercetare.

De asemenea, adresez mulțumiri deosebite doamnei Conf.dr.ing. Eugenia Mincă de la Universitatea “Valahia” din Târgoviște pentru sfaturile și ajutorul acordat de-a lungul elaborării tezei de doctorat, sfaturi fără de care nu ar fi fost posibilă concretizarea acestei lucrări.

Mulțumesc în mod special doamnei Conf.dr.ing. Alina Vodă de la Universitatea JOSEPH FOURIER Grenoble 1/CNRS, Grenoble Image Parole Signal Automatique (GIPSA-lab) din Franța pentru îndrumarea profesională și sprijinul din timpul stagiului de cercetare extern de la universitatea menționată.

Îmi exprim, în aceeași măsură, recunoștința față de domnul Șl.dr.ing. Vasiliu Grigore și domnul Șl.dr.ing. Șușnea Ioan pentru sfaturile, sprijinul și pentru formarea mea pe durata studiilor de licență și masterat și pentru ajutorul acordat pe durata doctoratului.

După șapte ani petrecuți în Facultatea de Automatică și Calculatoare, Inginerie Electrică și Electronică mulțumesc sincer tuturor membrilor facultății pentru formarea mea.

Nu în ultimul rând, mulțumesc familiei mele pentru înțelegerea, dragostea și sprijinul moral acordat pe perioada cercetărilor, oferindu-mi motivația și condițiile necesare pentru realizarea și finalizarea tezei de doctorat.

Mulțumesc tuturor celor care, direct sau indirect, m-au ajutat și m-au sprijinit de-a lungul acestor ani în care am elaborat această lucrare.

Mulțumesc mult !

*Galați, octombrie 2012
Adrian Radaschin*

Abstract

In a real flexible manufacturing systems the final products resulting from the process of assembly are tested for quality, either when the assembly is completely finished or during the assembly process.

Some products or their components contain manufacturing defects (caused by the assembly process or processing) and these items are not validated by the quality control tests so the final product is considered to be a reject. In this case, a disassembly operation is triggered in order to send the product back or on a separate assembly line where it is subjected to partial or complete disassembly in order to recover parts or subassemblies. Since disassembly locations can be different from the locations which store both faulty and good components, you can use wheeled mobile robots (WMR) equipped with manipulators that can remove the faulty components and transport them from the place where disassembly is conducted to the assigned storage space.

Considering these conditions, solving the main problems concerning assembly/disassembly operations is achieved by introducing a robot capable of both transporting and manipulation tasks either during the assembly or disassembly processes.

An example of solution treated in this research is the Pioneer P3-DX mobile robot equipped with a robotic manipulator Pioneer 5-DOF serving a flexible line assembly/ disassembly HERA & Horstmann. Robot aims to serve a flexible manufacturing line in the disassembly process, process it becomes completely reversible.

Cuprins

Prefață.....	2
Abstract.....	4
 Capitolul 1. Introducere.....	 12
1.1 Formularea Problemei	12
1.2 Scopul și obiectivele cercetării.....	13
1.3 Structura și conținutul tezei.....	14
1.4 Diseminarea rezultatelor	16
 Capitolul 2. Stadiul actual privind analiza proceselor de fabricație din cadrul sistemelor flexibile de fabricație deservite de roboți	 17
2.1 Introducere	17
2.1 Procese flexibile de fabricației	17
2.1.1 Procesul de asamblare	18
2.1.2 Procesul de dezasamblare.....	19
2.3 Concluzii	21
 Capitolul 3. Contribuții privind analiza liniilor flexibile de fabricație deservite de roboți	 22
3.1 Introducere	22
3.2 Structura unui SFF	24
3.3 Funcțiile sistemelor flexibile de fabricație	25
3.4 Conducerea sistemelor flexibile de fabricație	27
3.5 Performanțele SFF.....	28
3.6 Optimizarea sistemelor flexibile de fabricație.....	29
3.7 Sistemul flexibil de asamblare/dezasamblare HERA & Horstmann.....	30
3.8 Sistemul de prelucrare FESTO MPS-200	35
3.9 Platforma mobilă Pioneer P3-DX	39
3.10 Concluzii	41
 Capitolul 4. Contribuții privind analiza și modelarea 43proceselor de fabricație pe liniile flexibile deservite de roboți mobili.....	 43
4.1 Introducere	43
4.2 Ipoteze de analiză și modelare a liniei flexibile HERA & Horstmann	44
4.3 Planificarea taskurilor aferente procesului de dezasamblare deservit de robot mobil echipat cu manipulator	48
4.4 Modelarea procesului de asamblare/dezasamblare pe linia flexibilă deservită de robot mobil echipat cu manipulator utilizând RP netemporizate în abordare SED.....	51

4.5 Modelarea procesului de asamblare/dezasamblare pe linia flexibilă deservită de robot mobil echipat cu manipulator cu ajutorul RP temporizate în abordare SED	60
4.6 Modelarea procesului de asamblare/dezasamblare pe linia flexibilă deservită de robot mobil echipat cu manipulator cu ajutorul RPH sincronizate.....	62
4.6.1 Modelul generalizat al RPHS	62
4.7 Ipoteze de analiză și modelare a liniei flexibile FESTO MPS-200.....	71
4.8 Planificarea taskurilor aferente procesului de prelucrare al liniei flexibile FESTO MPS-200	73
4.9 Modelarea liniei flexibile de prelucrare FESTO MPS -200 cu ajutorul RP netemporizate în abordare SED.....	74
4.10 Modelarea liniei flexibile de prelucrare FESTO MPS -200 cu ajutorul RP temporizate în abordare SED	75
4.11 Concluzii	76
Capitolul 5. Contribuții privind simularea procesului de asamblare/dezasamblare deservit de un robot mobil echipat cu manipulator	78
5.1 Simularea procesului de asamblare/dezasamblare a unui produs deservit de un RM cu manipulator.....	78
5.2 Simularea structurii generalizate a modelului RPH sincronizate.....	81
5.3 Modelul cinematic al platformei mobile	83
5.4 Conducerea Sliding Mode a platformei mobile	84
5.5 Simularea conducerii în Sliding Mode.....	85
5.5.1 Simularea conducerii platformei mobile în cadrul deservirii liniei flexibile	86
HERA & Horstmann	86
5.5.2 Simularea conducerii platformei mobile în cadrul deservirii liniei flexibile	88
FESTO MPS-200	88
5.6 Concluzii	91
Capitolul 6. Contribuții privind conducerea în timp real.....	93
6.1 Implementarea conducerii în timp real.....	93
6.2 Prezentarea aplicației în mediul de programare grafică LabVIEW	94
6.3 Conducerea în timp real a procesului de dezasamblare completă.....	102
6.4 Concluzii	103
Capitolul 7. Concluzii.....	105
7.1 Sumarul contribuțiilor originale din cadrul tezei	107
7.2 Direcții de cercetare viitoare	109
Bibliografie.....	110
ANEXE.....	117

ANEXA A: Programul de conducere sliding mode al robotului Pioneer P3-DX echipat cu manipulator Pioneer 5-DOF Arm în cadrul procesului de deservire al liniei flexibile HERA & Horstmann	117
ANEXA B: Programul de conducere sliding mode al robotului Pioneer P3-DX echipat cu manipulator Pioneer 5-DOF Arm în cadrul procesului de deservire al liniei flexibile FESTO MPS-200	128
ANEXA C: Programul de conducere în timp real al liniei flexibile HERA & Horstmann din cadrul procesului de asamblare și dezasamblare totală a unui produs, dezasamblare deservită de robotul mobil Pioneer P3-DX echipat cu manipulator Pioneer 5-DOF Arm implementat în automatul programabil SIEMENS Simatic S7-300	136

Listă de figuri

Fig. 3.1 Sistem flexibil de asamblare/dezasamblare în variantă clasică, deservit de roboți

Fig. 3.2 Organizarea ierarhică din cadrul unui SFF

Fig. 3.3 Erarhizarea sistemului de conducere a unui SFF

Fig. 3.4 Structura sistemului de conducere și informațiile vehiculate

Fig. 3.5 Sistemul flexibil de asamblare/dezasamblare HERA & Horstmann

Fig. 3.6 Stații de prelucrare (depunere) și componentele aferente acestora

Fig. 3.7 Produs final

Fig. 3.8 Sistem de transport de tip bandă transportoare

Fig. 3.9 Stația de testare

Fig. 3.10 Stația de scoatere a bolțurilor în cadrul operației de dezasamblare

Fig. 3.11 Depozit de produse finite

Fig. 3.12 Sistem de automatizare stație flexibilă HERA & Horstmann

Fig. 3.13 Arhitectura sistemului de automatizare al liniei flexibile de asamblare/dezasamblare
HERA & Horstmann

Fig. 3.14 Sistem flexibil de fabricație FESTO MPS 200

Fig. 3.15 Stație de sotare

Fig. 3.16 Stație de prelucrare

Fig. 3.17 Stație de acumulare piese asamblate

Fig. 3.18 Stație de depozitare

Fig. 3.19 Arhitectura sistemului de automatizare al liniei flexibile de prelucrare FESTO
MPS-200

Fig. 3.20 Robot mobil Pioneer P3-DX

Fig. 3.21 Dimensiuni (mm) Pioneer P3-DX: a) vedere laterală; b) vedere de sus

Fig. 3.22 Schema bloc a sistemului de conducere încorporat în Pioneer P3-DX

Fig. 3.23 Manipulator robotic Pioneer 5-DOF Arm

Fig. 3.24 Sistem flexibil de asamblare/dezasamblare în variantă conceptuală, deservit de
roboți mobile echipați cu manipulator robotic

Fig. 4.1 Schema bloc a operațiilor de asamblare a unui produs

Fig. 4.2 Schema bloc a operațiilor de dezasamblare deservită de un robot mobil a unui produs

Fig. 4.3 Ciclul complet al robotului mobil echipat cu manipulator în procesul de dezasamblare

Fig. 4.4 Împărțirea pe secțiuni și zone a procesului de dezasamblare

- Fig. 4.5** Planificarea taskurilor pentru operația de dezasamblare deservită de un robot mobil echipat cu manipulator
- Fig. 4.6** Modelarea procesului de asamblare a unei componente pe linia flexibilă cu RP netemporizate în abordare discretă
- Fig. 4.7** Modelarea procesului de testare și depozitare a unui produs pe linia flexibilă cu RP netemporizate în abordare discretă
- Fig. 4.8** Modelarea procesului de asamblare pe linia flexibilă HERA & Horstman cu RP netemporizate în abordare discretă
- Fig. 4.9** Modelarea procesului de dezasamblare parțială pe linia flexibilă HERA & Horstman cu RP netemporizate în abordare discretă
- Fig. 4.10** Modelarea procesului de dezasamblare parțială pe linia flexibilă HERA & Horstman deservită de un robot mobil cu manipulator utilizând RP netemporizate în abordare discretă
- Fig. 4.11** Modelarea procesului de dezasamblare totală pe linia flexibilă HERA & Horstman deservită de un robot mobil cu manipulator utilizând RP netemporizate în abordare discretă
- Fig. 4.12** Modelarea procesului de asamblare/dezasamblare totală pe linia flexibilă HERA & Horstmann deservită de un robot mobil cu manipulator cu RP netemporizate în abordare discretă
- Fig. 4.13** Modelarea procesului de asamblare/dezasamblare totală pe liniei flexibile HERA & Horstmann deservit de un robot mobil cu manipulator cu RP temporizate în abordare Discretă
- Fig. 4.14** Reprezentarea punctelor de lucru ale procesului de asamblare/dezasamblare pentru o piesă cu N componente
- Fig. 4.15** Structura generalizată a modelului sincronizat corespunzător procesului de dezasamblare a N piese
- Fig. 4.16** Modelarea procesului de asamblare/dezasamblare totală pe linia flexibilă HERA & Horstmann deservit de un robot mobil cu manipulator cu RPH sincronizată
- Fig. 4.17** Structura generalizată a modelului sincronizat corespunzător procesului de dezasamblare a primei piese din cadrul produsului
- Fig. 4.18** Structura generalizată a modelului sincronizat corespunzător procesului de dezasamblare a ultimei piese din cadrul produsului
- Fig. 4.19** Planificarea taskurilor pentru operațiile de sortare, prelucrare și depozitare
- Fig. 4.20** Modelarea procesului de prelucrare pe linia flexibilă FESTO MPS 200 cu ajutorul RP netemporizate în abordare discretă

Fig. 4.22 Modelarea procesului de prelucrare pe linia flexibilă FESTO MPS-200 cu ajutorul RP netemporizate în abordare discretă

Fig. 4.23 Modelarea procesului de prelucrare pe linia flexibilă FESTO MPS-200 cu ajutorul RP temporizate în abordare discretă

Fig. 5.1 Evoluția stărilor corespunzătoare locațiilor aferente procesului de asamblare al primei piese de la stația S1: a) starea marcajului inițial M_0 care corespunde mărcii P_a3 ; b) starea procesului de asamblare al primei piese care corespunde mărcii P_a30 ; c) starea starea magaziei de component aferentă mărcii P_a29 ; d) starea prin care se declanșează banda transportoare aferentă mărcii P_a2 ; e) starea prin care procesul se reinițializează după efectuarea primei asamblări aferentă mărcii P_a4 ; starea prin care se transport component asamblată către stația următoare, aferentă mărcii P_a6 ;

Fig. 5.2 Evoluția stărilor corespunzătoare locațiilor aferente procesului de testare și depozitare a unui produs declarat rebut: a) starea prin care se transportă produsul asamblat către stația de testare aferentă mărcii P_a21 ; b) starea procesului de testare al produsului final care corespunde mărcii P_a22 ; c) starea magaziei de componente aferentă produselor declarate defecte, aferente mărcii P_a23 ; d) starea magaziei de componente aferentă produselor validate, aferente mărcii P_a24 ;

Fig. 5.3 Evoluția stărilor corespunzătoare locațiilor aferente procesului de dezasamblare a unei componente din cadrul unui produs declarat rebut: a) starea marcajului de inițializare a procesului de dezasamblare care corespunde mărcii P_d1 ; b) starea procesului de dezasamblare al primei piese care corespunde mărcii P_d2 ; c) starea prin care procesul se reinițializează după efectuarea primei dezasamblări aferentă mărcii P_d3 ; d) starea starea magaziei de component dezasamblate aferentă mărcii P_d4 ; e) starea prin care se transport produsul către stația următoare de dezasamblare, aferentă mărcii P_d6 ;

Fig. 5.4 Evoluția locațiilor continue și locațiilor discrete asociate robotului mobil echipat cu manipulator în cadrul procesului de dezasamblarea a primei componente

Fig. 5.5 Interfață grafică LabVIEW a RPH sincronizate

Fig. 5.6 Variabilele cinematice ale platformei mobile cu două roți motoare și o roată directoare

Fig. 5.7 Simularea parcurgerii traiectoriei în conducere Sliding Mode

Fig. 5.8 Simularea parcurgerii traiectoriei în conducere sliding mode: a) traiectoria pacursă; b) viteza platformei mobile;

Fig. 5.9 Eroare de urmărire a conducerii în sliding mode: a) eroarea pe axa x; b) eroare pe axa y;

Fig. 5.10 Comanda conducerii în sliding mode: a) comanda pe suprafața s_2 ; b) comanda pe suprafața s_1 ;

Fig. 6.1 Schema bloc a comunicației dintre echipamentele utilizate în conducerea liniei flexibile HERA & Horstmann deservită de un robot mobil Pioneer P3-DX

Fig. 6.2 Schema bloc a programului de conducere în timp real din mediul LabVIEW

Fig. 6.3 Reprezentarea schematică a conexiunii dintre linia de mecatronică și placa de achiziție

Fig. 6.4 Interfața grafică de conducere în mediul LabVIEW

Fig. 6.5 Panou de control

Fig. 6.6 Implementarea în mediul LabVIEW a acțiunii *START dezasamblare*

Fig. 6.7 Ciclul complet al robotului mobil echipat cu manipulator în procesul de dezasamblare

Fig. 6.8 Implementarea în mediul Labview a acțiunii *START (ciclu continuu)* robot mobil

Fig. 6.9 Exemplificarea operațiilor efectuate de robotul mobil: a) de apucare; b) depunere;

Fig. 6.10 Implementarea în mediul LabVIEW a acțiunii de pornire a benzii transportoare aferente stației S5

Fig. 6.11 Implementarea în mediul LabVIEW a acțiunii de pornire a benzii transportoare aferente stației S4

Fig. 6.12 Procesul de dezasamblare a piesei P3-capac: a) preluarea piesei; b) depunerea în magazie;

Fig. 6.13 Implementarea în mediul LabVIEW a acțiunii de pornire a benzii transportoare aferente stației S3

Fig. 6.14 Procesul de dezasamblare a piesei P2-corp: a) preluarea piesei; b) depunerea în magazie;

Fig. 6.15 Implementarea în mediul LabVIEW a acțiunii de pornire a benzii transportoare aferente stației S2

Fig. 6.16 Procesul de dezasamblare a piesei P1-palet: a) preluarea piesei; b) depunerea în magazie;

Fig. 6.17 Distanțe parcurse de robotul mobil echipat cu manipulator

Fig. 6.18 Operația de dezasamblare a unei component

Fig. 6.19 Operațiile de dezasamblare a unui produs

Fig. 6.20 Operațiile robotului mobil Pioneer P3-DX echipat cu manipulator Pioneer 5-DOF Arm

Capitolul 1

Introducere

-
- 1.1 Formularea problemei*
 - 1.2 Scopul și obiectivele cercetării*
 - 1.3 Structura și conținutul tezei*
 - 1.4 Diseminarea rezultatelor*
-

Dezvoltarea *sistemelor flexibile de fabricație* (SFF) și introducerea structurilor robotice reprezintă noi direcții de cercetare și de creștere a performanțelor liniilor de producție, astfel beneficiile introduse au efecte importante asupra tuturor sistemelor de fabricație.

Principalele activități care pot fi întreprinse de roboți utilizați în cadrul sistemelor flexibile de fabricație sunt legate de transportul și manipularea obiectelor (componente sau subansamble) sau și uneori, de realizarea unor diverse procese (asamblare, dezasamblare, prelucrare, transport etc.).

1.1 Formularea Problemei

În cadrul sistemelor flexibile de fabricație reală dintr-o întreprindere produsul final rezultat în urma procesului de asamblare este supus unui test de calitate, fie la terminarea completă a asamblării, fie pe parcursul acesteia. În cazul anumitor produse sau ale componentelor ce intră în componența acestora apar anumite defecte de fabricație (defecte de asamblare sau prelucrare), urmând ca acest produs să nu fie validat la trecerea prin testul de calitate, astfel produsul final se consideră a fi un rebut. În acest caz, este declanșată operația de dezasamblare, astfel produsul va fi trimis înapoi pe linia de fabricație fie pe cea care s-a efectuat asamblarea fie pe o altă linie de dezasamblare, fiind supus unor operații de dezasamblare parțială sau totală pentru recuperarea părților componente sau subansamblelor. Având în vedere că locațiile unde se fac dezasamblări pot fi diferite de locațiile unde se stochează componentele, atât cele defecte cât și cele bune, se pot utiliza *roboți mobili* (RM) echipați cu manipuloare care preiau părțile componente ale produselor declarate rebut din punctele unde se face dezasamblarea și le transportă la magazii de stocare aferente componentelor.

În scopul de a optimiza execuția acestor operații de asamblare/dezasamblare, multe studii actuale din domeniu conduc spre utilizarea structurilor robotice. Utilitatea acestora duce la creșterea flexibilității SFF, a fiabilității sistemului automatizat cât și la eliminarea nesiguranței și a erorilor specifice factorului uman, ceea ce va permite un control eficient a SFF și implicit va permite trecerea la conducerea în timp real a producției.

În aceste condiții, soluționarea principalelor probleme legate de operațiile de asamblare/dezasamblare ținând cont de tendința de dezvoltare rapidă a sistemelor flexibile de fabricație o reprezintă introducerea robotului, structură capabilă să efectueze atât sarcini de transport cât și operații de manipulare atât în cadrul procesului de asamblare cât și dezasamblare sau prelucrare.

Un exemplu de soluție tratată în această cercetare îl reprezintă robotul mobil Pioneer P3-DX echipat cu un manipulator robotic Pioneer 5-DOF care deservește o linie flexibilă de asamblare și dezasamblare (parțială) HERA & Horstmann. Obiectivele robotului fiind de a deservi linia flexibilă de fabricație în cadrul procesului de dezasamblare, proces în care acesta devine complet reversibilă.

1.2 Scopul și obiectivele cercetării

Tratarea sistemelor flexibile de fabricație deservite de roboți într-o concepție pe cât posibil unitară, în cadrul operațiilor de asamblare, dezasamblare și prelucrare, urmărind atât aspecte de modelare și de conducere necesare în aplicațiile practice industriale a constituit scopul principal al acestei cercetări. Ca urmare a creșterii complexității proceselor de asamblare/dezasamblare și a sporirii cerințelor privitoare la performanțele acestor procese a căpătat o importanță deosebită nevoia de optimizare a liniilor flexibile de asamblare astfel încât să permită și operații de dezasamblare ale componentelor.

În această lucrare unul dintre obiective propune optimizarea unui sistem mecatronic didactic format din o linie flexibilă de asamblare care permite și o dezasamblare parțială a componentelor. Linia flexibilă va fi deservită de un RM echipat cu manipulator. Scopul optimizării este de a face linia flexibiă de asamblare complet reversibilă, adică să permită dezasamblarea completă a unui produs. În final sistemul mecatronic va funcționa complet automatizat fără intervenția operatorului uman. Asamblarea va fi un proces periodic iar dezasamblarea va fi un proces accidental care se declanșează atunci când produsul final nu este validat la testul de calitate.

Un alt obiectiv îl reprezintă elaborarea modelelor corespunzătoare proceselor de asamblare și dezasamblare a componentelor pe linia mecatronică. Deoarece linia flexibilă de fabricație este un sistem didactic dedicat efectuării asamblării, propunem ca efectuarea dezasamblării să

se realizeze în cazul detectării unei piese rebut, prin deservirea/asistarea operațiilor elementare de dezasamblare de către un RM echipat cu manipulator.

Pentru elaborarea acestor modele se vor utiliza ca instrument de modelare *Rețelele Petri* (RP) cunoscute în literatură de specialitate ca fiind instrumentul specific modelării *sistemelor cu evenimente discrete* (SED). Deoarece ne propunem ulterior implementarea conducerii în timp real a liniei, aceste modele vor trebuie să fie coerente cu evoluția procesului real de asamblare sau dezasamblare. Din acest motiv se vor utiliza diferite tipologii de RP dedicate modelării duratelor operațiilor de asamblare și dezasamblare precum și a deplasării robotului în diferite puncte ale liniei flexibile de fabricație în scopul preluării/transportării și depozitării componentelor dezasamblate în magazii.

Pentru linia flexibilă de asamblare și dezasamblare se vor elabora modele cu RP care vor descrie stările și tranzițiile asociate procesului de asamblare, dezasamblare și ale robotului mobil echipat cu manipulator care va transporta piesele dezasamblate la magazinele de depozitare. Se va implementa un sistem de conducere în timp real a liniei flexibile reversibile sub platforma LabVIEW.

Din obiectivele generale ale cercerării prezentate rezultă următoarele obiective specifice:

- analiza, modelare și simularea procesului de asamblare din cadrul SFF;
- analiza, modelarea, simularea și optimizarea procesului de dezasamblare din cadrul SFF, utilizând roboți mobili echipați cu manipulator;
- analiza și modelarea procesului de prelucrare din cadrul SFF;
- elaborarea, dezvoltarea, implementarea și testarea procesului de dezasamblare completă a SFF;
- optimizarea unei linii flexibile de fabricație în cadrul procesului de asamblare și dezasamblare deservită de un robot mobil echipat cu manipulator care să permită și o dezasamblare totală a unui produs;
- implementarea unui program de conducere în timp real a procesului de asamblare/dezasamblare deservită de un robot mobil echipat cu manipulator în mediul LabVIEW;

1.3 Structura și conținutul tezei

Raportul este împărțit în 7 capitole după cum urmează:

În *Capitolul 1* sunt prezentate aspecte generale legate de formularea problemei în cadrul acestei cercetări. În cea de-a doua parte sunt prezentate scopul și obiectivele raportului de cercetare științifică, pornindu-se de la ideea utilizării liniilor flexibile de fabricație deservite

de roboți mobili echipați cu manipulator necesari operației de dezasamblare pentru a putea face sistemul mecatronic HERA & Horstmann complet reversibil.

În **Capitolul 2** sunt prezentate cele mai cunoscute și utilizate sisteme flexibile de fabricație flexibilă. Sunt descrise și analizate procesele flexibile din cadrul sistemelor flexibile de fabricație.

În **Capitolul 3** se prezintă o analiză dintre sistemele flexibile de fabricație și locul unde se pot aduce contribuții. Se prezintă aspecte privind integrarea roboților mobili în sisteme flexibile de fabricație. Se evidențiază faptul că dintre cele mai multe operații posibile a fi executate de roboți mobili echipați cu manipulator, în mediile industriale echipate cu linii flexibile de fabricație, sunt cele de transport și manipulare, urmărindu-se că în ultima parte a acestui capitol ideea de reversibilitate din cadrul liniilor flexibile de fabricație prin integrarea roboților mobili, încheindu-se cu prezentarea sistemului mecatronic didactic HERA & Horstmann, a robotului Pioneer 3-DX echipat cu manipulator Pioneer 5-DOF Arm și a sistemului de prelucrare FESTO MPS-200.

În **Capitolul 4** se prezintă contribuțiile privind analiza și modelarea dinamicii liniei flexibile didactice HERA & Horstmann cu rețele Petri netemporizate a procesului de asamblare și dezasamblare parțială, dezasamblare parțială deservit de un robot mobil echipat cu manipulator, a procesului de asamblare/dezasamblare completă a unui produs deservit de un robot mobil echipat cu manipulator cât și a procesului de asamblare/dezasamblare a unui produs deservit de un robot mobil echipat cu un manipulator utilizând rețele Petri temporizate în abordare SED. Se mai prezintă și contribuțiile privind analiza și modelarea dinamicii unui sistem didactic FESTO MPS-200 de prelucrare a unui produs, utilizând rețele Petri netemporizate și temporizate în abordare SED. Se prezintă formalismul de descriere a modelului RPH generalizate și a RPH sincronizate pentru conducerea liniei flexibile HERA & Horstmann deservită de un robot mobil echipat cu manipulator.

În **Capitolul 5** se prezintă contribuțiile privind simularea modelelor prezentate în Capitolul 4 utilizând programele VisualObject Net ++ și Sirphyco. Se prezintă contribuțiile privind simularea în timp real a liniei flexibile HERA & Horstmann în cadrul procesului de asamblare/dezasamblare deservit de un robot mobil echipat cu manipulator utilizând programul LabVIEW.

În **Capitolul 6** este prezentat programul și aplicația de conducere în timp real a procesului de asamblare și dezasamblare deservită de un robot mobil echipat cu manipulator.

În **Capitolul 7** sunt prezentate concluziile.

1.4 Diseminarea rezultatelor

Diseminarea rezultatelor s-a facut prin publicarea unui număr de 10 articole, din care 3 ca prim autor și 7 ca coautor după cum urmează:

1. **Radaschin A.**, *Sliding mode control of an autonomous welding robot*, *Automation Quality and Testing Robotics*, 2010 IEEE International Conference, Automation Quality and Testing Robotics (AQTR 2010), 2010 IEEE International Conference, Cluj-Napoca, Romania , 28-30 May 2010, volume 1, pp.1-5, ISBN: 978-1-4244-6724-2.
2. **Radaschin A.**, Filipescu A., Manzu V., Minca E. and Filipescu Jr. A. (2011). *Adaptive disassembly sequence control by using mobile robots and system information*, Proceeding of 15th IEEE International Conference in System Theory, Control and Computing (ICSTCC 2011), 14-16 October 2011, Sinaia, Romania, ISBN 978-973-621-323-6, 499-505.
3. **Radaschin A.**, A. Voda, E. Minca, A. Filipescu, *Task Planning Algorithm in Hybrid Assembly/Disassembly Process*, 14 th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2012), 23-25 May 2012, IFAC Proceedings Vol.14, Part 1, ISSN: 1474-6670, ISBN: 978-3-902661-98-2.
4. Minca E., Dragomir O.E., Dragomir F., Enache M.A., and **Radaschin A.**, (2011). *Assembly-Disassembly Flexible Lines and Collaborative Robots Considered as Hierarchical System in Temporal Recurrent Modeling*. 9th World Congress on Intelligent Control and Automation, IEEE Catalog No. CFP11496-DVD, ISBN: 978-1-61284, pp.69-74.
5. Dumitrascu B., A. Filipescu, **A. Radaschin**, Minca E., A.Filipescu Jr., *Discrete-Time Sliding Mode Control Of Wheeled Mobile Robots*, 8th Asian Control Conference ASCC2011, 15-18 May 2011, IEEE ISBN: 978-1-61284-487-9, pp. 771-776.
6. Susnea I. Vasiliu G, Filipescu A, Coman G., **Radaschin A.**, *Real-Time Control of Autonomous Mobile Robots Using Virtual Pheromones*, Proceedings of the 7th Asian Control Conference, Hong Kong, China, August 27-29, 2009, IEEE Catalog Number CFP09832, ISBN:978-89-956056-9-1, pp.1450-1455.
7. Susnea I., Filipescu A., Serbencu A, **Radaschin A.**, *Virtual Pheromones to Control Mobile Robots . A Neural Network Approach*, Proceedings of the IEEE International Conference on Automation and Logistics, Shenyang, China, ISBN: 978-1-4244-4795-4/09, 2009 IEEE, CD-ROM Proceedings IEEE, Catalog: CFP09CAL, August 5 - 7, 2009, Shenyang, China, pp1962-1967.
8. Susnea I., Vasiliu G., Filipescu A., **Radaschin A.**, *Virtual Pheromones for Real-Time Control of Autonomous Mobile Robots*, STUDIES IN INFORMATICS AND CONTROL, Volume: 18 Issue: 3 Pages: 233-240 Published: SEP 2009, IDS Number: 499SC, ISSN: 1220-1766.
9. Susnea I., Adrian Filipescu, Grigore Vasiliu, **A. Radaschin**, *The Bubble Rebound Obstacle Avoidance Algorithm for Mobile Robots*, Control and Automation (ICCA), 2010 8th IEEE International Conference in Control and Automation, 9-11 June 2010, Xiamen, China, ISBN: 978-1-4244-5195-1, ISSN: 1948-3449, pp.540-545.
10. Alina V., **A. Radaschin**, E. Minca, A. Filipescu, *Control of Automatic Robot with Guided Manipulator Integrated into Flexible Manufacturing System Using Hybrid Petri Nets*, 16 th International Conference on System Theory, Control and Computing Joint Conference (ICSTCC 2012), 12-14 October, Sinaia, Romania, ISBN 978-606-834-848-3, IEEE Catalog Number CFP1236P-CDR.

Capitolul 2

Stadiul actual privind analiza proceselor de fabricație din cadrul sistemelor flexibile de fabricație deservite de roboți

2.1 Introducere

2.2 Procese flexibile de fabricație

2.3 Concluzii

2.1 Introducere

În ultimul deceniu industria cunoaște o nouă evoluție, una globală, antrenată de creșterea progresului tehnologic, cunoscută sub numele de sisteme și procese flexibile de fabricație. Acest progres se propagă în toate domeniile industriale și declanșează formarea de noi generații de sisteme avansate de producție flexibilă, de noi sisteme și metode de conducere centralizată, distribuită sau supervizată, de noi roboți și mașini-unelte de prelucrare cât și de introducerea sistemelor eficiente de transport și manipulare, toate acestea conducând spre globalizarea întregii industrii [1], [2], [3].

Fabricarea flexibilă generează un nou trend, cel al produselor diversificate și personalizate, cu calități superioare, costuri competitive și cu cicluri de fabricație reduse.

În prezent, în evoluția proceselor flexibile de fabricație, roboții preiau o foarte mare parte din operațiile de fabricație, transport și manipulare, acest aspect constituind o puternică flexibilizare și automatizare atât a echipamentelor cât și a programului de conducere [35]. Din acest motiv un mare procent al utilajelor și echipamentelor care intră în componența unui SFF îl reprezintă roboți.

2.1 Procese flexibile de fabricației

Pe parcursul ultimelor două zeci de ani, flexibilitatea și optimizarea proceselor de fabricație a atras atenția centrelor de cercetare din acest domeniu. Cele mai multe studii abordate au la bază creșterea numărului de operații de fabricație cu aceleași utilaje și creșterea productivității, ambele având un impact asupra calității produsului final [4], [5]. Recent, s-a demonstrat că, calitatea produsului și procesul de fabricație sunt strâns cuplate [6]-[13]. Se

analizează în [14] caracteristica de flexibilitate și impactul acesteia asupra creșterii performanțelor sistemelor flexibile de fabricație și mai exact al proceselor flexibile din cadrul acestora.

Recent, tehnologia flexibilă de fabricație dezvoltă un nou proces, cel a dezasamblării produselor sau componentelor. Noile tendințe presupun planificarea producției prin metode de cooperare dintre procesul de asamblare/dezasamblare și roboți.

2.1.1 Procesul de asamblare

Procesul de asamblare, unul dintre cele mai vechi forme de producție flexibilă prezintă o extraordinară modernizare datorate evoluției tehnologiei. Procesul de asamblare este definit în literatură ca fiind o funcție productivă de a construi anumite piese individuale, subansamble de piese sau structuri, într-o anumită cantitate dată pe o perioadă de timp bine determinată.

Reversibilitatea procesului de asamblare, dezasamblarea, se definește ca totalitatea proceselor prin care o structură se poate descompune într-o anumită perioadă de timp. Cercetările actuale urmăresc optimizarea timpului de dezasamblare a unei structuri prin creșterea flexibilității, dezvoltarea echipamentelor și optimizarea algoritmilor de planificare [18]. *Proiectarea procesului de asamblare* (PPA) presupune luarea în considerare a aspectelor de proiectare ale unui produs, de procedurile și etapele de asamblare care au efect și asupra ciclului de viață al produsului [15], [16]. În ultimul deceniu, *proiectarea procesului de dezasamblare* (PPD) a devenit o operație adițională PPA. Ambele procese au rolul de a integra cunoștințele specifice domeniului din procesul de fabricație flexibil, proiectare și planificare precum și de luare a deciziilor [19]. Obiectivul principal urmărit este de a reduce costul total de fabricație și de a îmbunătăți calitatea acestuia [17]. În [20], [21] asamblarea este prezentată ca un proces tradițional de fabricație, în care una din cele mai importante etape în asamblarea unui produs îl reprezintă în general costurile totale de producție și calitatea acestuia. Experiența a demonstrat că *proiectarea procesului de asamblare și/sau dezasamblare* (PPAD) reprezintă una dintre cele mai eficiente tehnici de reducere semnificativă a costurilor de fabricație încă din stadiul de proiectare și de reducere a costurilor de dezasamblare printr-o proiectare cât mai simplă a produselor în vederea scăderii numărului de operații. Numeroase exemple disponibile arată că simplitatea unui produs duce la o analiză a PPAD prin care sunt îmbunătățite costurile de fabricație, creșterea fluxului de producției, impunerea de proiectare a utilajlor care trebuie să realizeze mai multe funcții cât și alte beneficii, cum este indicat și în [22], [23], [24].

Abordări mai recente tratează procesele de asamblare și dezasamblare prin integrarea acestora în medii de realitate virtuală pentru a optimiza metodele de proiectare, planificare a

proceselor, de deservire și colaborarea cu unități robotice așa cum sunt descrise și în [25], [26].

2.1.2 Procesul de dezasamblare

Dezasamblarea, ca primă etapă în procesul de reciclare va avea o rată mare de expansiune pe viitor în mediul industrial [26], [33]. În prezent, dezasamblarea de produse se execută în principal manual, iar uneori printr-un proces mecanizat într-o oarecare măsură. O dată cu creșterea cantității de produse necesare reciclării, va crește prin urmare și numărul de sisteme de dezasamblat. Acestea vor prezenta un grad ridicat de flexibilitate și un cost redus pe produs dezasamblat. Puține componente din echipamente (sau produse) sunt reciclate după dezasamblare, cu toate acestea gradul de automatizare este încă foarte mic, doar câteva proiecte pilot sau demonstrative sunt realizate în principal în institute de cercetare. Pentru masa mare de produse ce vor fi nevoite a fi reciclate și care vor beneficia de procesul de dezasamblare, marile companii producătoare de sisteme flexibile de fabricație, fac cercetări privind proiectarea noilor produse ce vor ieși pe piață, cercetări privind modalități noi de fabricare (sau asamblare) cât și mărirea gradului de flexibilitate al echipamentelor, proces foarte costisitor [27]. La momentul actual există foarte puține sisteme flexibile care execută dezasamblarea componentelor produselor pe care le-au fabricat (sau ansamblat). În acest moment se dezasamblează în special piesele produselor de înaltă calitate, sau componente, în scopul de a reutiliza anumite componente [28], [29], [30]. Majoritatea produselor actuale sunt distruse pentru reciclare fără a trece printr-un proces de demontare [31], [32].

Conceptele existente de sisteme flexibile de dezasamblat prezintă caracteristici de inflexibilitate și sunt dezvoltate doar pentru o anumită sarcină sau produs, efectuarea operației de dezasamblare completă de către același sistem de asamblare este încă în stadiu de cercetare. În acest caz, domeniul dezasamblării flexibile prezintă o foarte mare atracție, actual dezvoltându-se conceptul de dezasamblare a unei familii sau grupuri similare sau diferite de produse, care necesită aproape aceleași operații de dezasamblare, realizate cu aceleași instrumente de montare/asamblare [34].

2.2 Sisteme flexibile de fabricație: aspecte specifice

Nucleul tehnologiei flexibile de fabricație se împarte între Japonia și Europa. În ciuda faptului că primele linii flexibile de fabricație deservite de roboți au fost fabricate și puse în funcțiune în SUA, marile companii producătoare de sisteme flexibile de fabricație care realizează diferite operații de fabricație ca: asamblare, dezasamblare, prelucrare, transport și

manipulare sunt reprezentate de General Motors, Cincinnati Milacron, Westinghouse și General Electric acestea dezvoltă acest domeniu, rămânând ca lideri importanți pe piața de echipamente industriale. Cu toate acestea există un număr mare de întreprinderi mici care dezvoltă categorii de roboți care deservește procesele de fabricație, în special roboți de serviciu, în plină dezvoltare fiind companiile iRobot, Mobile Robotic și Evolution Robotics.

Cele două mari companii producătoare de SFF și roboți care le deservește, din Europa sunt ABB și Kuka. Peste 50% din nivelul cercetării, dezvoltării și producției, al companiei ABB, este reprezentat de dezvoltarea structurilor robotice, al echipamentelor și sistemelor de automatizare necesare roboților și liniilor flexibile de fabricație.

Un al producător important din domeniul roboților care deservește procesele de fabricație flexibilă este compania FANUC, acesta produce roboți și echipamente ce intră în alcătuirea acestora, sisteme flexibile de fabricației pentru diferite procese, celule de fabricație și mașini cu CNC. FANUC reprezintă 17% din piața roboților industriali din Japonia, 16% din Europa și 20% din America de Nord. Următoarele companii după FANUC sunt Kawasaki și Yaskawa, este menționată și compania SIEMENS din Europa care reprezintă cel mai puternic competitor al acestora. Spre deosebire de aceste companii care deservește mediul industrial, sunt prezente și companiile care dezvoltă servicii robotizate, companii precum Sony, Fujitsu și Honda. În Europa cel mai mare procent de procese flexibile din cadrul fabricație flexibilă îl reprezintă industria auto, industria din Japonia este favorizată de producția de echipamente și componente electronice cât și de integrarea de sisteme flexibile de fabricație.

Printre producători de sisteme robotizate, echipamente și utilaje flexibile se numără și fabricanți de celule flexibile de fabricație. În Europa reprezentant este compania FASTEMS din Finlanda. Aceasta produce celule robotizate modulare pentru deservirea mașinilor-unelte, celule de prelucrare prin aschiere și celule personalizate cu diferite scopuri, acestea permit o creștere a gradului de utilizare al mașinilor-unelte pentru strunjire cu CNC, centrelor de prelucrare sau a altor mașini unelte.

Alte companii cu prestigiu ce realizează celule flexibile robotizate pentru alte operații de: debavurare, polizare, rectificare, șlefuire, lustruire, curățire sau netezire, se enumeră: compania MAPE cu sediul în Grenoble, Franța, produce celule flexibile de fabricație robotizate pentru realizarea de procese de polizare, debavurare și de superfinisare. Celulele flexibile, sunt dotate cu roboți antropomorfi cu 6 grade de libertate și sarcina utilă cuprinsă între 5 și 150 Kg. Pentru a lucra cu piese/componente portabile, MAPE a dezvoltat utilaje în vederea construirii de celule flexibile robotizate modulare, acestea fiind dotate cu sisteme de vedere și captare cu laser care permit recunoașterea pieselor înainte de șlefuire-lustruire sau rectificare. Compania KUKA din Germania este recunoscută prin roboții industriali ce

operează în diverse celule flexibile de fabricație, dar și prin echipamentele de prelucrare în celule și softurile adecvate acestor operații, acesta devenind foarte prestigioasă în acest domeniu. Compania STARMATIK cu sediul în Treviso, Italia, produce celule flexibile de fabricație, modulare, pentru rectificarea și lustruirea pieselor din oțel, cu roboți antropomorfi cu 6 axe sau cu roboți cartezieni.

2.3 Concluzii

În prezent, concurența și cerințele ridicate privind calitatea pentru o mare varietate de produse a dus la creșterea rolului sistemelor de producție și mai ales ale proceselor pe care acestea le îndeplinesc. Sistemele flexibile de fabricație sunt compuse în prezent din elemente dinamice și adaptive care impun utilizarea eficientă a tuturor resurselor, cu toate acestea anumite procese de fabricație, cum ar fi procesul de dezasamblare mai necesită modificări majore privind adaptabilitatea și flexibilitatea acestuia. Introducerea structurilor robotice, a sistemelor de conducere distribuite, optimizarea utilajelor, etc. joacă un rol important în optimizarea întregului ansamblu de fabricație. În conceptul producției actuale nu s-a modificat esențial rolul și locul sistemelor de fabricație, optimizarea proceselor de fabricației existente fac ca un proces să devină competitiv.

Pornind de la necesitatea îmbunătățirii performanțelor proceselor tehnologice de asamblare, prin modernizarea tehnologiilor de montaj, se pot asigura creșteri importante ale productivității, în condițiile îmbunătățirii substanțiale ale calității. Consecința directă se reflectă asupra ciclului de viață al produselor sau de apariția accidentală a unui defect al produsului sau piesei ce intră în componența sa, care impune reutilizarea acestora, acest ciclu presupune introducerea unor sisteme de dezasamblare sau optimizarea unei structuri de asamblare ale aceleași game de produse în vederea dezasamblării cu aceleași utilaje.

Detalierea teoretică și practică în vederea optimizării unui sistem flexibil de fabricație care poate să execute atât procese de asamblare și dezasamblare cu aceleași utilaje impune parcurgerea unor etape de concepere, realizare și implementare a unei structuri capabile să execute două tipuri de sarcini cu aceleași utilaje. În acest context subiectul tezei tratează soluții noi aplicabile sistemelor flexibile de asamblare, în vederea optimizării acestora în cadrul procesului de dezasamblare precum și integrarea structurilor robotice în vederea creșterii flexibilității și extinderii anumitor sarcini din cadrul procesului tehnologic.

Capitolul 3

Contribuții privind analiza liniilor flexibile de fabricație deservite de roboți

-
- 3.1 Introducere*
 - 3.2 Structura unui SFF*
 - 3.3 Funcțiile sistemelor flexibile de fabricație*
 - 3.4 Conducerea sistemelor flexibile de fabricație*
 - 3.5 Performanțele SFF*
 - 3.6 Optimizarea sistemelor flexibile de fabricație*
 - 3.7 Sistemul flexibil de asamblare/dezasamblare HERA & Horstmann*
 - 3.8 Sistemul flexibil de prelucrare FESTO MPS-200*
 - 3.9 Platforma mobilă Pioneer P3-DX*
 - 3.10 Concluzii*
-

Acest capitol este dedicat analizei și descrierii aspectelor importante privind componența și caracteristicile liniilor flexibile de fabricație [39], [42], [43], [44], [45]. Detalierea lor se realizează în cadrul a trei secțiuni principale. În prima secțiune se analizează o structură clasică de sistem flexibil de fabricație în conformitate cu literatura de specialitate, urmărind ca, în secțiunea următoare să se particularizeze structura clasică pe un proces de asamblare, dezasamblare și prelucrare, procese care pot fi deservite de un robot mobil echipat cu manipulator. Ultima secțiune este dedicată concluziilor prin care se pun în evidență contribuțiile privind analiza unei structuri flexibile de asamblare și deservirea acesteia în cadrul procesului de dezasamblare cu ajutorul un robot mobil echipat cu manipulator.

3.1 Introducere

Un sistem flexibil de fabricație reprezintă totalitatea mijloacelor, utilajelor, echipamentelor și a relațiilor dintre acestea, capabile să rezolve o sarcină sau un ansamblu de sarcini de fabricație. Conceptul de SFF a fost proiectat și dezvoltat pentru fabricarea de produse diferite, în loturi mici sau medii.

În **Fig. 3.1** este prezentat schema bloc a unui SFF (asamblare/dezasamblare) clasic deservit de roboți, acesta este alcătuit din următoarele blocuri funcționale:

- robot industrial necesar operațiilor de manipulare (necesită precizie, sistem de control al traiectoriei și sistem de senzori și traductoare).
- robot de transport (necesită sistem de control al traiectoriei sau sistem de ghidare, sistem de senzori de poziție și navigație)
- scule (instrumente) de montare/demontare/prelucrare necesare roboților industriali sau celulelor de fabricație.
- sistem de stocare componente și/sau subansamble necesare asigurării unui flux continuu de asamblare/dezasamblare. Sunt incluse și magaziiile de stocare componente dezasamblate.

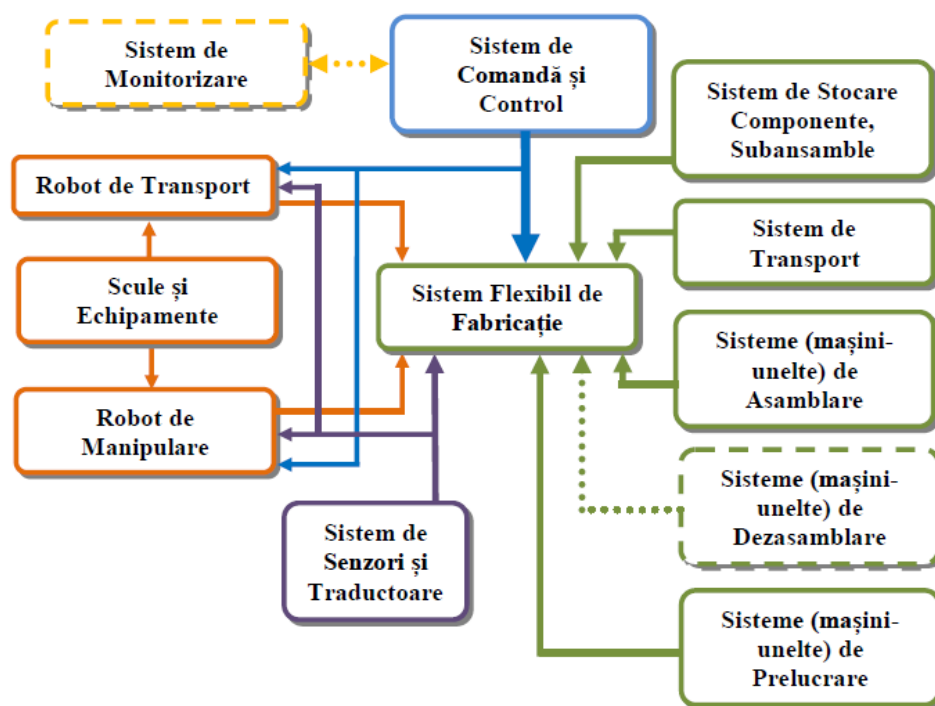


Fig. 3.1 Sistem flexibil de asamblare/dezasamblare în variantă clasică, deservit de roboți

- sistem de transport (benzi transportoare) necesare transportului de la o celulă flexibilă la alta a componentelor sau subansamblelor;
- sisteme (celule) flexibile echipate cu mașini-unelte necesare operației de asamblare/dezasamblare sau prelucrare;
- sistem de senzori și traductoare necesar SFF cât și roboților de transport și manipulare;
- sistem de comandă și control local sau distribuit interconectat cu un sistem de monitorizare;

Pornind de la obiectivele impuse și de analiză a SFF privind realizările teoretice și experimentale din domeniul industrial se prezintă în acest capitol sistemul flexibil de asamblare/dezasamblare HERA & Horstmann deservit de un robot mobil Pioneer P3-DX echipat cu un manipulator robotic Pioneer 5-DOF și un sistem flexibil de prelucrare FESTO MPS-200.

3.2 Structura unui SFF

Structurile sistemelor flexibile de fabricație sunt direct legate de stabilitatea ierarhizării pe nivele a acestor unități. Până în prezent nu apare un accord unanim în această privință, în continuare fiind prezentate structurile corespunzătoare unui număr de patru nivele (**Fig. 3.2**):

a. *Nivelul 1* (SFF_1): la acest nivel se găsește cea mai mică unitate cu funcții de fabricație autonome, aceasta fiind o mașină-unealtă flexibilă, multifuncțională, respectiv un echipament care concentrează un număr mare de operații cu prelucrări diferite, caracterizată de:

- procentul de dotare cu comandă numerică;
- posibilități de prelucrare multiple;
- prezența unui dispozitiv de înmagazinare scule, dispozitiv care nu influențează procesul tehnologic și la care depozitarea temporară a sculelor este efectuată codificat;
- schimbarea și transferul automat al sculelor;

b. *Nivelul 2* (SFF_2): intervine *celula de fabricație flexibilă* (CFF) în care sunt concentrate mai multe mașini-unelte cu comandă numerică, de regulă 2, ..., 4 mașini, deservite de un robot industrial, tot ansamblu fiind controlat de un calculator (sau *automat programabil* (AP)) care asigură întreaga funcționare a celulei. Celula flexibilă poate asigura prelucrarea integrală automată a unor produse sau/și piese diferite (componente ale unei clase stabilite în prealabil) având un grad de flexibilitate ridicat.

c. *Nivelul 3* (SFF_3): se găsesc sistemele flexibile, compuse de regulă din mai multe celule flexibile, legate prin dispozitive de transport și manipulare. În cadrul acestor sisteme de la acest nivel se pot deosebi mai multe tipuri de subnivele (sau subsisteme). Astfel pentru sistemul de transport pot fi prevăzute transportoare cu deplasarea semifabricatelor într-o singură direcție în cadrul sistemului, acestea fiind comandate din calculatorul central de comandă și ca urmare semifabricatele sau piesele sunt paletizate urmând a fi deplasate la orice stație de prelucrare și în orice ordine ceea ce deschide largi posibilități de optimizare a funcționării întregului ansamblu. Transportul pieselor poate fi efectuat și prin intermediul carucioarelor autopropulsate, în multe cazuri acestea sunt comandate prin diverse metode (cabluri pilot amplasate în pardoseală etc.).

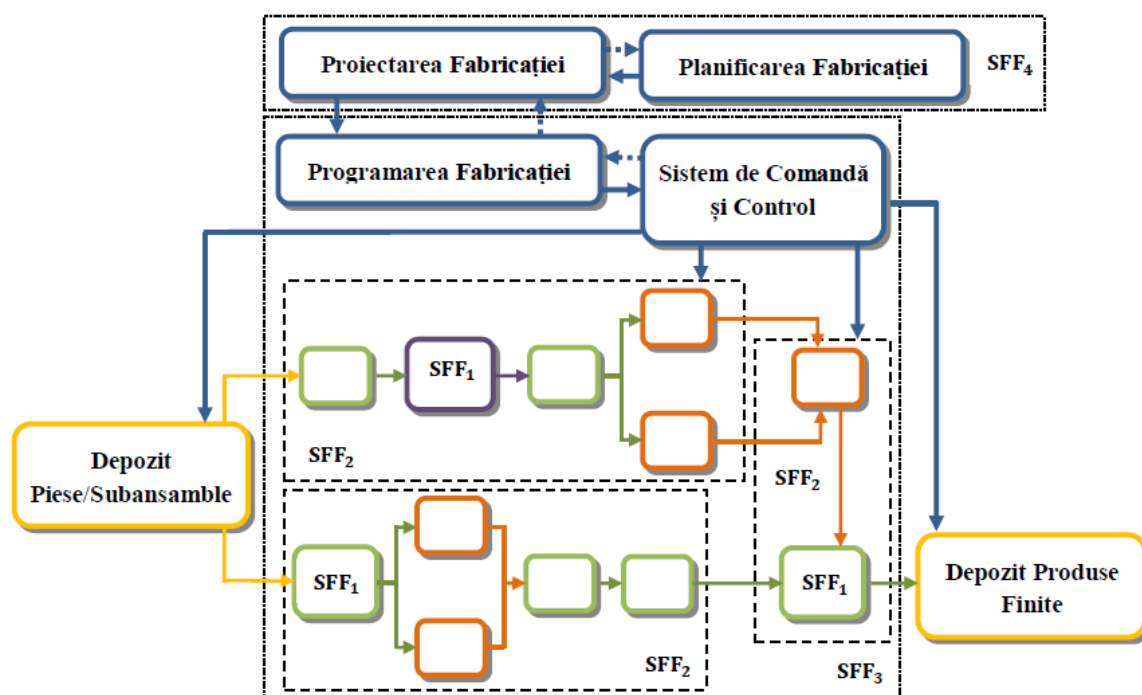


Fig. 3.2 Organizarea ierarhică din cadrul unui SFF

Preluarea semifabricatelor și pieselor de pe dispozitivele de transport, alimentarea mașinilor-unelte și readucerea pieselor (după prelucrare) pe dispozitivele respective sunt de regulă realizate de manipolatoare, roboți industriali și de mecanisme de paletizare (așezare și fixare pe palete) sau de depaletizare (desprinderea de pe palete). În cazul sistemelor de la nivelul 3 este prevăzută posibilitatea unor activități suplimentare (în raport cu cele executate la nivelele 1 și 2: manipulare, transport, alimentare, prelucrare, evacuare piese prelucrate, schimbarea sculelor, supravegherea instalației etc.) cum sunt cele de testări automate de pregătirea fabricației sau de comandă a aprovizionării cu materiale.

d. *Nivelul 4* (SFF₄): La acest nivel intervin totalitatea mijloacelor tehnice și persoanele necesare pentru realizarea aprovizionării, depozitării, planificării de lungă durată, proiectării constructive și tehnologice a produselor și fabricației propriu-zise.

Prezentarea structurilor și nivelelor de organizare are în vedere fluxurile de obiecte (semifabricate, piese, scule) și utilajele necesare prelucrării, manipulării și transportului, în cadrul acestor structuri intervenind și depozitele, dispozitivele de control, de întreținere și reparare etc., care pot fi realizate cu funcționare automată.

3.3 Funcțiile sistemelor flexibile de fabricație

Structura generală a unui SFF, prezentată sub forma schemei bloc (Fig. 3.1) permite evidențierea funcțiilor generale ale sistemului:

- Funcția de prelucrare automată a pieselor sau subansamblelor;
- Funcția de depozitare, transport și manipulare automată;
- Funcția de comandă automată a tuturor componentelor sistemului și de supraveghere, control și diagnostic automat;

Funcția de prelucrare automată se realizează în cadrul subsistemului tehnologic al SFF, având în componență stațiile (celule) de lucru, mijloacele de manipulare a pieselor și sculelor. Realizarea acestei funcții presupune alimentarea automată cu piese și scule a mașinii-unelte, prelucrarea propriu-zisă în comandă numerică și eventual optimizarea procesului de comandă pe mașina-uneltă. Pot fi incluse aici și dispozitivele de asamblare/dezasamblare, unele dintre acestea având funcții speciale.

Funcția de depozitare, transport și manipulare automată se referă la fluxul automat al sculelor, pieselor, componentelor și subansamblelor necesare SFF și care includ mai multe funcții parțiale:

- a. înmagazinarea automată a pieselor, sculelor, dispozitivelor și materialelor auxiliare;
- b. identificarea și livrarea în sistem a piesei sau subansamblelor în mod automat;
 - transportul automat al pieselor, sculelor, dispozitivelor și materialelor auxiliare între depozite și stațiile de lucru. Condiția principală în funcționarea subsistemului de depozitare și transport este ca transferul materialelor să se efectueze totdeauna la locul și momentul potrivit;
 - manipularea pieselor, subansamblelor, sculelor și dispozitivelor în depozite și între stațiile de lucru;

Funcția de comandă, monitorizare, control și diagnostic dintr-un SFF este realizată de subsistemul informațional prin fluxul informațional care se transmite în 2 sensuri: sensul direct, al informațiilor de comandă și sensul invers, al informațiilor de monitorizare, control și diagnostic.

Funcția de comandă automată se realizează cu ajutorul unuia sau mai multor AP în diverse configurații, centralizat sau distribuit, sau calculatoare de proces ce lucrează în timp real sau unități locale de comandă (echipamente CNC, AP la sistemele de manipulare și transport, microcalculatoare pentru comanda depozitelor automate etc.). Programele de calculator, furnizează întregului sistem informațiile necesare pentru comanda procesului de prelucrare și pentru comanda producției (comanda depozitelor de piese și scule, comanda sistemului de transport, etc.).

Informațiile pentru realizarea acestor subfuncții sunt obținute din sistem cu ajutorul unor traductoare, senzori, aparate de măsură etc. și se transmit în sens invers, către calculatorul de proces, AP sau microcalculatorului local.

3.4 Conducerea sistemelor flexibile de fabricație

Structurile și nivelele de organizare sunt puternic corelate cu nivelele de comandă și control, acestea se realizează sub forma unei rețele de echipamente (centralizate sau distribuite) de conducere care permit legarea într-un singur sistem a tuturor echipamentelor (AP, calculatoare etc.) care comandă mașinile-unelte, roboți industriali, sistemul de manipulare, transport și depozitare a pieselor, subansamblelor etc.

Structura generală a subsistemului de comandă al SFF se distribuie pe nivele ierarhice, numărul acestora depinzând de mărimea SFF, de domeniul de aplicare și de numărul funcțiilor de fabricație flexibilă integrate în sistem.

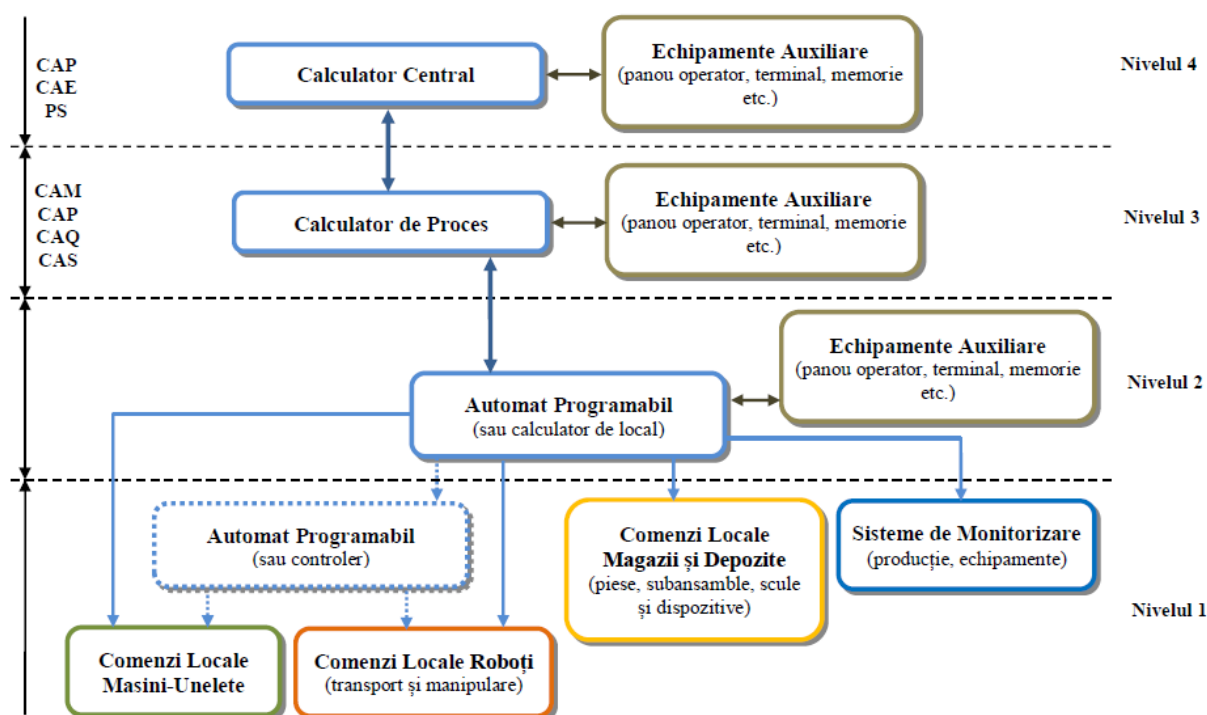


Fig. 3.3 Erarhizarea sistemului de conducere a unui SFF

Corespunzător SFF, la care toate funcțiile ilustrate în **Fig. 3.1** sunt automatizate, structura generală a subsistemului de comandă se prezintă ca o structură distribuită pe patru nivele (**Fig. 3.3**). La partea inferioară a structurii de comandă (Nivelul 1) se află echipamentele industriale de comandă a mașinilor-unelte, roboților industriali, precum și echipamentele de comandă locală a depozitelor și sistemelor de transport. La nivelul ierarhic 2 se află AP (sau calculatorul de conducere locală) a fabricației care realizează conducerea echipamentelor din nivelul inferior și transmiterea informațiilor către nivelul superior. La acest nivel se realizează diagnosticarea instalațiilor și echipamentelor de lucru dar în unele cazuri și planificarea producției la nivel de celulă de fabricație.

La nivelele ierarhice 3 și 4 se realizează proiectarea produselor, pregătirea și planificarea fabricației, aceste nivele ierarhice putând funcționa și independent. Pentru realizarea unui concept de produs finit, sistemele de la nivelele inferioare sunt conectate la nivelele superioare, acestea fiind conectate la o structura de procesare de capacitate mare, care realizează automat funcțiile de proiectare a tehnologiei de prelucrare și elaborare a programelor (CAM – Computer Aided Manufacturing), de planificare a prelucrărilor și a producției (CAP – Computer Aided Planning), de control și supraveghere a proceselor și subsistemelor (CAQ – Computer Aided Quality) și de întreținere (CAS – Computer Aided Service).

La nivelul 4 se află un calculator care realizează funcțiile de concepție și de proiectare constructivă a produselor (CAD – Computer Aided Design), de analiză a formei și structurii produselor și de rentabilizare (CAE – Computer Aided Engineering) și de planificare strategică (PS) etc.).

Structura generală a subsistemelor de comandă în SFF prezentată în **Fig. 3.4** este una generală, care să fie aplicabilă pentru orice sistem flexibil de fabricație. În funcție de gradul de complexitate al funcțiilor ce se realizează în cadrul SFF și în funcție de modul cum se organizează activitățile pe diferite nivele ierarhice apar modificări, unele dintre nivele putând chiar să nu existe.

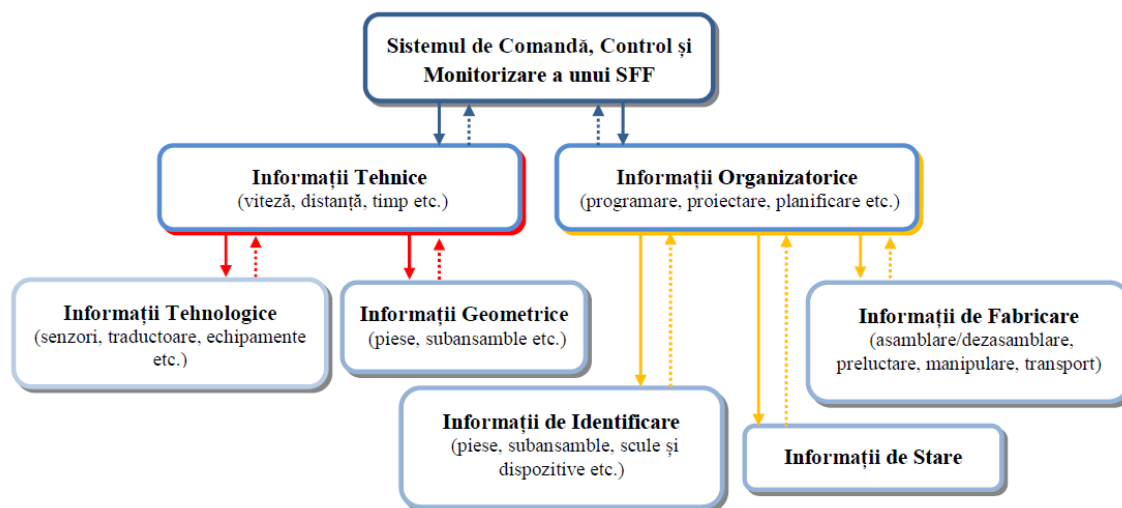


Fig. 3.4 Structura sistemului de conducere și informațiile vehiculate

3.5 Performanțele SFF

În cazul planificării producției, cu deplasări ale componentelor/pieselor sau a echipamentelor de transport, operații de manipulare sau fabricație și durate ale proceselor tehnologice cu timpi de așteptare la stocuri (de componente sau subansamble) intermediare se obțin indici de performanță, relativ reduși în cadrul SFF. În acest sens sunt edificatoare două

cifre, reprezentand valori medii, publicate într-un raport din anul 1975, referitor la productivitatea din SUA, în comparație cu cea din alte state industrializate:

- din tot timpul consumat pentru a produce o piesa numai 5% este cheltuit de celula de fabricație;
- din tot timpul consumat de celula de fabricație, numai 1,5% este folosit pentru prelucrarea efectivă;

Prima cifra arată că 95% din timpul consumat pentru fabricarea sau prelucrarea unei piese este cheltuit pentru manipulare și transport semifabricate sau piese, pentru așteptări alimentare stocuri (magazii), pentru prelucrare și procesare. În cazul sistemelor flexibile de fabricație, timpul efectiv de lucru ajunge la 50-85% din totalul timpului de lucru concomitent cu o creștere a gradului de utilizare a capacității de producție. Din această constatare rezultă că indicele de performanță aferent eficienței utilizării celulei de fabricație este foarte scăzut, existând astfel posibilități de îmbunătățire a performanțelor prin planificarea task-urilor procesului de fabricație și mărirea flexibilității SFF prin introducerea de echipamente care îndeplinesc mai multe sarcini. Pentru aprecierea eficienței unui mod de organizare a fabricației pot fi considerate valorile a două tipuri de variabile care caracterizează procesele de producție flexibilă în dinamica lor: variabile referitoare la debite ale produselor care se găsesc în curs de prelucrare (variabile denumite “fluxuri”) și variabile referitoare la acumulări intermediare de produse parțial prelucrate, aflate în stare de stagnare (variabile denumite uneori “nivele”). Cu cât raportul dintre valorile medii ale fluxurilor și nivelurilor este mai mare cu atât este mai ridicată eficiența planificării de fabricație, de aceea unul dintre obiectivele principale ale perfecționării organizării fabricației este legat de raportul menționat și de metodele abordate de planificare a task-urilor.

Dacă fabricația este astfel proiectată și organizată încât prin calcule corespunzătoare să se determine cantitățile maxime de material, semifabricate și piese care se pot găsi un timp cât mai mare în stadiul de prelucrare, deplasare sau transport cu stabilirea succesiunilor și traseelor optime din punct de vedere al micșorării cantităților stocate intermediare și a duratelor de stagnare, atunci rezultă o creștere semnificativă a raportului dintre fluxuri și niveluri și a eficienței planificării fabricației. Introducerea de roboți cu funcții multiple, duce la îmbunătățirea timpului de prelucrare și devine semnificativ în raport cu durata ciclului total de fabricație.

3.6 Optimizarea sistemelor flexibile de fabricație

O prima etapă în optimizarea SFF o constituie conducerea și optimizarea fluxurilor de activități (operații) care implică două faze: planificare fluxurilor și execuția acestora.

Pentru sistemele flexibile de fabricație trebuie găsite metode adecvate de planificare; acestea trebuie să contribuie la evitarea acțiunilor de blocare și conflictuale. Planificarea în SFF se poate descrie ca o abordare a coordonării, în care din faza de proiectare a secvenței de acțiuni pe care le execută un utilaj care realizează o anumită operație trebuie să se țină seama de interacțiunile dintre utilajele din componența SFF. Această abordare trebuie să permită sistemului de planificare a operațiilor să construiască un plan care să conțină detalii ale tuturor operațiilor și interacțiunilor viitoare. În acest fel se realizează propriile scopuri și se întrepătrunde execuția operațiilor din SFF cu mai multe etape de planificare și re-planificare. Varianta cea mai des utilizată de optimizare a SFF în cazul sistemelor centralizate, o reprezintă planificarea în spațiul stărilor, pentru cazul sistemelor descentralizate în care o parte din operații se împart unor utilaje care deserve SFF, se poate utiliza, planificarea în spațiul planurilor care poate furniza soluții adecvate.

Dezvoltarea unui algoritm de planificare a operațiilor într-un mediu de fabricație flexibil prezintă o mare dificultate cauzată de anumite aspecte de natură practică. De exemplu, un produs are deja un plan de fabricație dezvoltat de proiectant și deci putem vorbi de o planificare offline a operațiilor de fabricație deja efectuată și cunoscută în prealabil. Planificarea offline a operațiilor va continua cu o planificare online în care planul dezvoltat este completat astfel încât să fie pregătit pentru fabricație.

O alta etapă de optimizare a SFF o reprezintă creșterea gradului de flexibilitate a echipamentelor, dezvoltarea și perfecționarea utilajelor pentru a putea executa mai multe operații. Prin mărirea gradului de flexibilitate se reduc timpi în care produsul pe linia de fabricație execută diverse operații care necesită timp (transport, manipulare, etc.).

3.7 Sistemul flexibil de asamblare/dezasamblare HERA & Horstmann

Structura sistemului flexibil de asamblare/dezasamblare HERA Horstmann este alcătuită din mai multe module (**Fig. 3.5**):

a. Structura hardware:

- 5 stații (celule) de prelucrare și un depozit de produse finite
- sistem de transport dintre stațiile de prelucrare de tip benzi transportoare
- sistem de manipulare și depozitare de tip lift

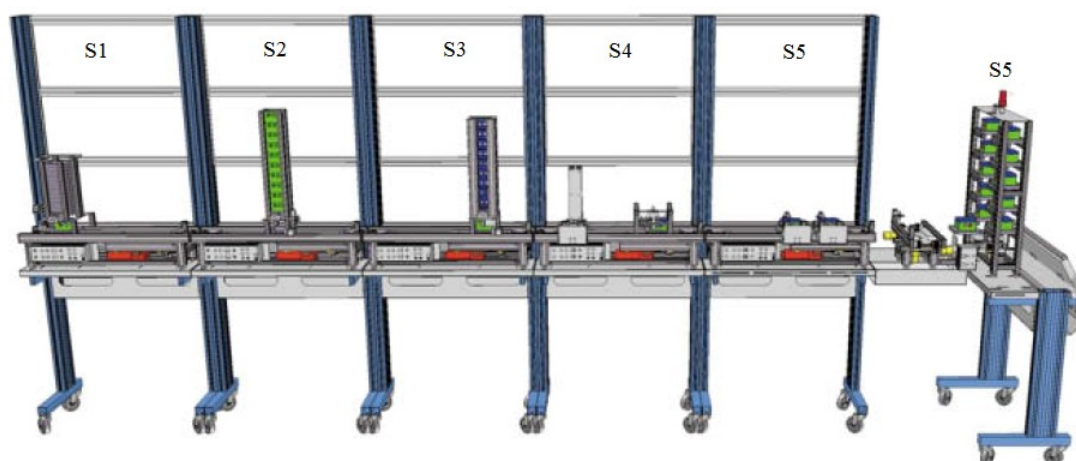


Fig. 3.5 Sistemul flexibil de asamblare/dezasamblare HERA & Horstmann

Stațiile de prelucrare S1, S2, S3 și S4 conform **Fig. 3.5** și **Fig. 3.6**, prezintă câte o magazie de piese, în fiecare magazie găsiindu-se un alt tip de componentă care intră în alcătuirea produsului final (**Fig. 3.7**). Fiecare stație depune (prelucrează) câte o piesă.

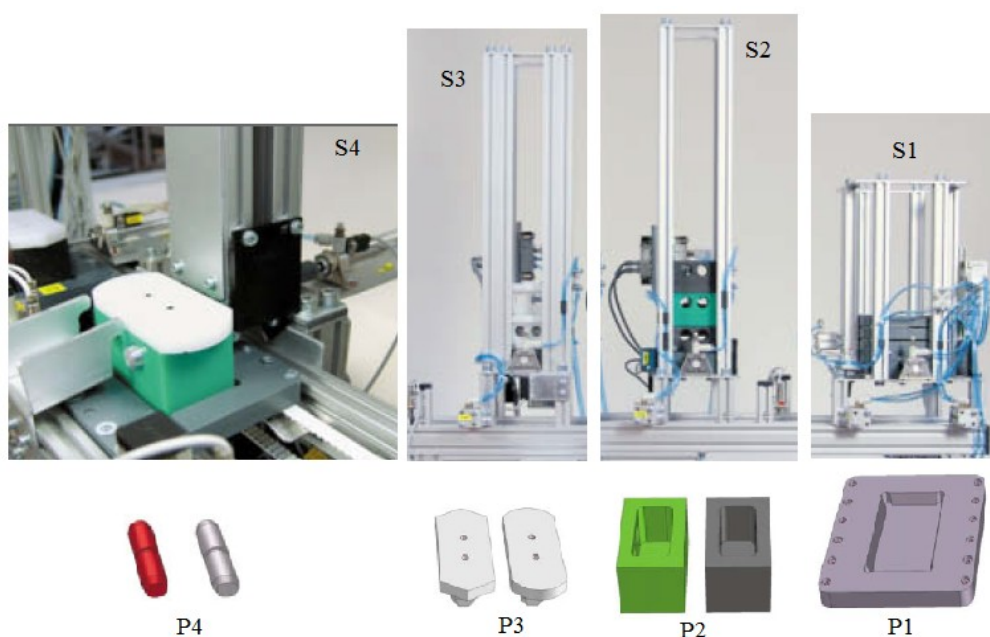


Fig. 3.6 Stații de prelucrare (depunere) și componentele aferente acestora

Depunerea unei piese pe banda transportoare (**Fig. 3.8**) se realizează prin intermediul unor elemente de execuție de tip piston pneumatic, actionat de un sistem pneumatic. Fiecare stație este echipată cu traductori de poziție, pentru o poziționare precisă în dreptul fiecărei magazii. Stația S1 conține în magazie piesa P1, denumită și palet, acesta are rolul de a transporta celelalte piese pe banda transportoare.



Fig. 3.7 Produs final

Paletul are în componența sa șase discuri metalice dispuse în ambele părți laterale, acestea au rolul de a transmite poziția paletului pe banda transportoare prin identificarea numărului de discuri de către traductoarele de poziție. Stația S2 conține un corp dreptunghiular prevăzut cu o deschizătură în partea superioară și 2 deschizături în părțile laterale. În partea superioară se montează la stația S3, piesa P3, denumită și capac, aceasta se prezintă în două forme constructive, cu margină ascuțită sau cu margine rotundă. La stația S4 se montează piesa P4, denumită și cilindru, în partile laterale. Această stație mai conține și o altă magazie de depozitare a pieselor P4, în momentul în care, la asamblare s-a produs o eroare sau aceasta nu a fost corect executată, cilindrul cade automat în această magazie.

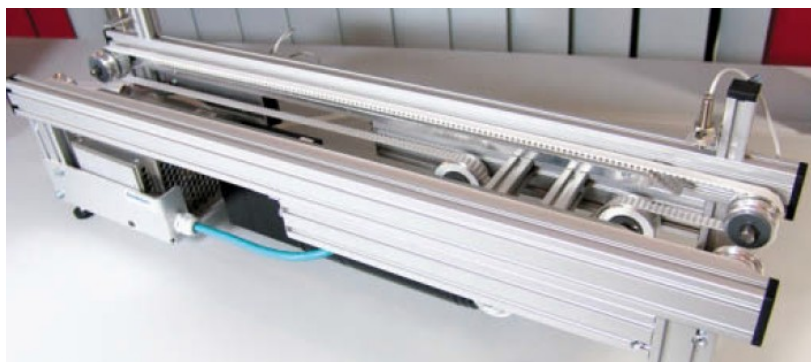


Fig. 3.8 Sistem de transport de tip bandă transportoare

Stația S4 mai conține un sistem de testare a produsului final (**Fig. 3.9**), înainte ca acesta să fie transportat și depozitat în depozitul D. În funcție de testarea efectuată, stația S4 transmite date despre produs sistemului de conducere, acesta selectează locația unde va fi depozitat noul produs. Testarea este efectuată cu ajutorul a trei traductoare, două dintre ele verifică dacă piesa P4, este din material metalic sau plastic, iar cel de-al treilea traductor verifica piesa P3.

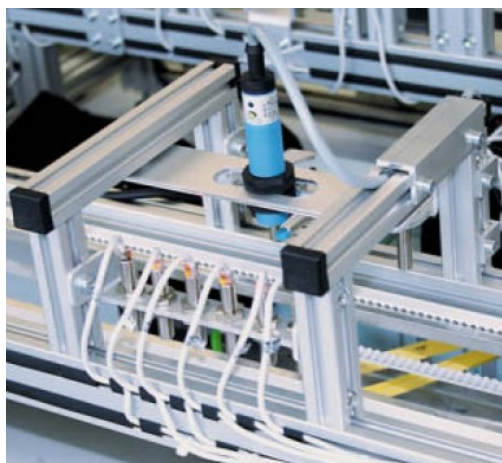


Fig. 3.9 Stația de testare

Stația S5 (**Fig. 3.9**) din cadrul sistemului flexibil de asamblare/dezasamblare, are rolul de a efectua o dezasamblare parțială a unui produs din depozit, mai exact efectuează dezasamblarea piesei P4. Dezasamblarea este realizată cu ajutorul a două pistoane pneumatice, piesa dezasamblată cade automat în magaziile aferente fiecărui piston.

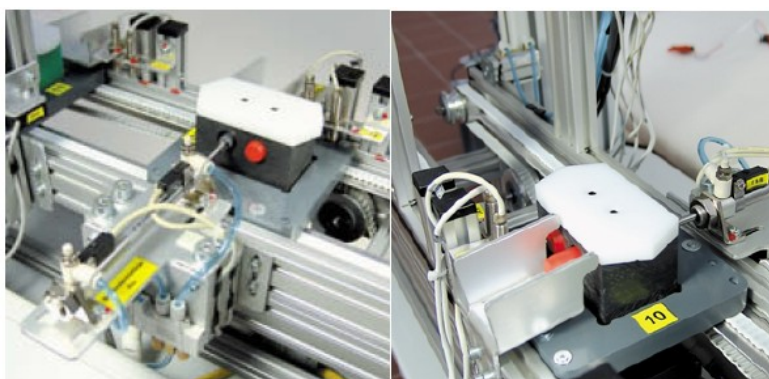


Fig. 3.10 Stația de scoatere a bolțurilor în cadrul operației de dezasamblare

Depozitul D are rolul de a stoca produsele finite, în 8 locații. Depozitul este compus dintr-un sistem de manipulare de tip lift, care ridică produsul de pe banda transportoare și îl poziționează în locația corespunzătoare. Liftul efectuează atât operații de stocare în depozit cât și operații de scoatere din depozit.



Fig. 3.11 Depozit de produse finite

- b. Structura software: este împărțită în echipamente de automatizare și program de conducere.

Structura sistemului de automatizare (**Fig. 3.13**) este de tip distribuită și este formată dintr-un AP SIEMENS Simatic S7-300 cu procesor din seria CP 314C-2 DP și modul de comunicație SIEMENS CP 343-2. Acesta se conectează pe magistrala PROFIBUS DP care conectează *modulele auxiliare* (MA) de interfațare I/O de tip SIEMENS ET200S-IM 151-1 distribuite pe fiecare dintre stațiile sistemului flexibil de asamblare/dezasamblare. Fiecare din cele 6 module SIEMENS ET200S-IM 151-1 prezintă module de I/O digitale și analogice, acestea preluând semnale provenite de la traductoare și transmițând comenzi elementelor de execuție.



Fig. 3.12 Sistem de automatizare stație flexibilă HERA & Horstmann

Pe magistrala PROFIBUS DP este conectat un terminal de tip panou operator SIEMENS Simatic HMI TP 177, prin intermediul căruia se poate vizualiza starea sistemului flexibil și se poate pune în execuție un proces de asamblare sau dezasamblare.

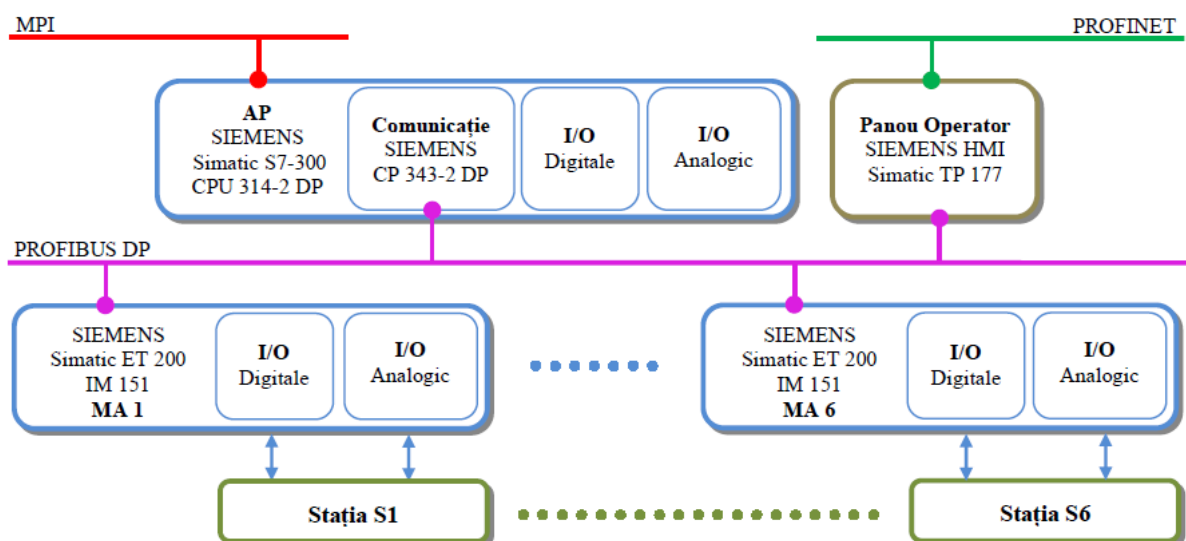


Fig. 3.13 Arhitectura sistemului de automatizare al liniei flexibile de asamblare/dezasamblare
HERA & Horstmann

3.8 Sistemul de prelucrare FESTO MPS-200

Sistemul mecatronic FESTO MPS-200 reprezintă o linie flexibilă didactică de prelucrare, sortare și depozitare. Aceasta este compusă din 4 stații (celule), fiecare realizând operații diferite.

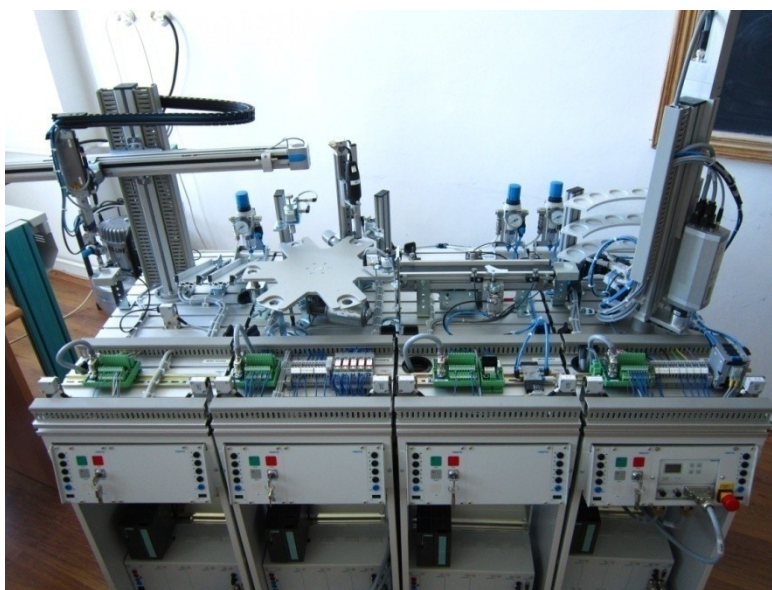


Fig. 3.14 Sistem flexibil de fabricație FESTO MPS-200

Structura sistemului flexibil FESTO MPS-200 este alcătuită din următoarele stații:

– Celulă de sortare (**Fig. 3.15**): preia componenta prelucrată de la stația anterioară cu ajutorul unui manipulator axial echipat cu gripper pneumatic, după ce a fost supusă unui test de culoare. Dacă culoarea diferă față de cea corectă, sistemul pneumatic preia piesa și o depune într-o magazie alăturată. Magazia de componente este compusă din două părți. Fiecarei părți i se atribuie o piesă de o anumită culoare. Dacă piesa a trecut testul de culoare, aceasta nu va mai fi stocată în magazie, sistemul de manipulare axial o depune într-o locație a stației următoare.



Fig. 3.15 Stație de sortare

– Celulă de prelucrare (**Fig. 3.16**): aceasta execută două operații distincte, una de găurire și cealaltă de șlefuire a componentei provenite de la stația de testare și sortare. Celula de prelucrare prezintă un sistem de acumulare de tip masă rotativă a șase piese. Prin fiecare rotație a sistemului rotativ se poziționează câte două piese simultan pentru a fi prelucrate. Operațiile de găurire și alezare se execută simultan de către 2 mașini-unelte. După terminarea ambelor prelucrări a unei piese sistemul de acumulare rotativ se poziționează în dreptul următoarei stații. Celula de prelucrare este echipată cu un sistem de senzori de proximitate pentru o poziționare precisă în dreptul celor două mașini-unelte.



Fig. 3.16 Stație de prelucrare

– Celulă de acumulare (de tip buffer) (**Fig. 3.17**): are rolul de a prelua și stoca piesele provenite de la stația de asamblare. Poate stoca un număr de 5 piese, care vor fi trimise individual către stația următoare la anumite intervale de timp. Această celulă este echipată cu un sistem de senzori pentru monitorizarea a nivelului de încărcare, un sistem pneumatic de oprire/eliberare piese.



Fig. 3.17 Stație de acumulare piese asamblate

– Celulă de depozitare (**Fig. 3.18**): în urma procesului de sortare sunt selectate ordinea și locația unde vor fi depozitate produsele finale. Sortarea produselor este realizată cu ajutorul unui sensor de culoare. Sistemul de depozitare este compus din 3 magazii în care fiecărei magazii îi sunt atribuite produse de o anumită culoare. Depozitarea produselor se face cu

ajutorul unui manipulator axial echipat cu un gripper pneumatic, care preia produsul de pe banda transportoare a stației anterioare și o depune în depozit. Manipulatorul axial este controlat cu ajutorul unui controller MTR-DCI-42S. Celula de depozitare reprezintă ultima stație din cadrul sistemului flexibil de fabricație FESTO MPS-200 având o capacitate de stocare de 18 produse prelucrate.

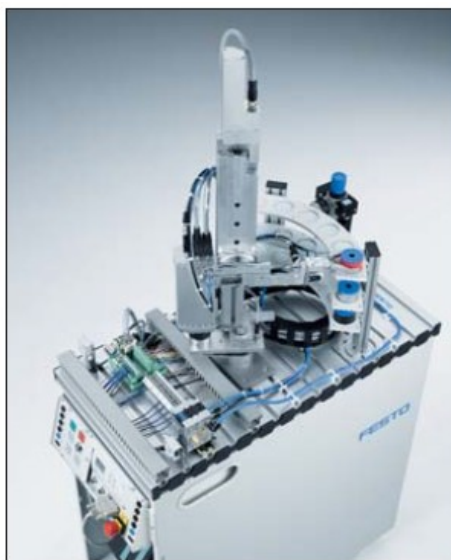


Fig. 3.18 Stație de depozitare

Structura sistemului de automatizare (**Fig. 3.19**) este de tip locală pe fiecare stație și este compusă dintr-un AP SIEMENS Simatic S7-300 cu procesor din seria CP 312C-2 DP și module auxiliare de interfațare I/O distribuite pe fiecare dintre stațiile sistemului flexibil. Fiecare din cele 4 AP prezintă module de I/O digitale și analogice, acestea preluând semnale provenite de la traductoare și transmițând comenzi elementelor de execuție.

Stația de depozitare este echipată cu un controller necesar manipulatorului axial și un modul de comunicație care realizează interfața dintre controler și AP. Sistemul de automatizare controlează atât sistemul de senzori și traductoare cât și elementele de execuție de tip electric și pneumatic.

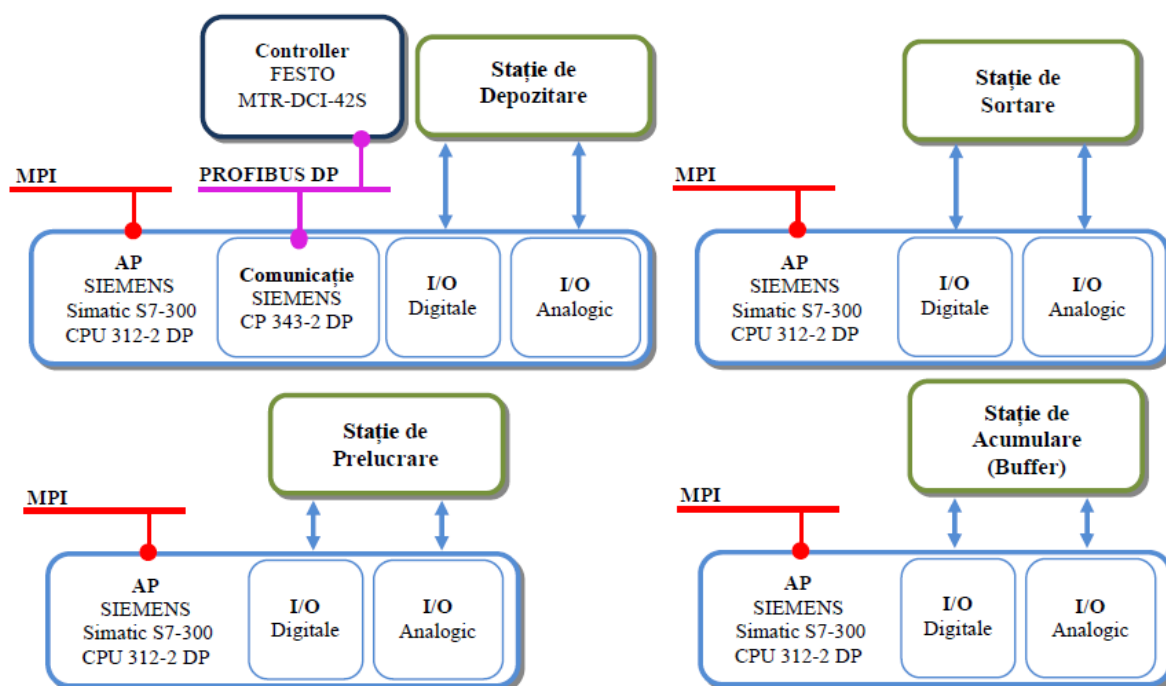


Fig. 3.19 Arhitectura sistemului de automatizare al liniei flexibile de prelucrare
FESTO MPS-200

3.9 Platforma mobilă Pioneer 3-DX

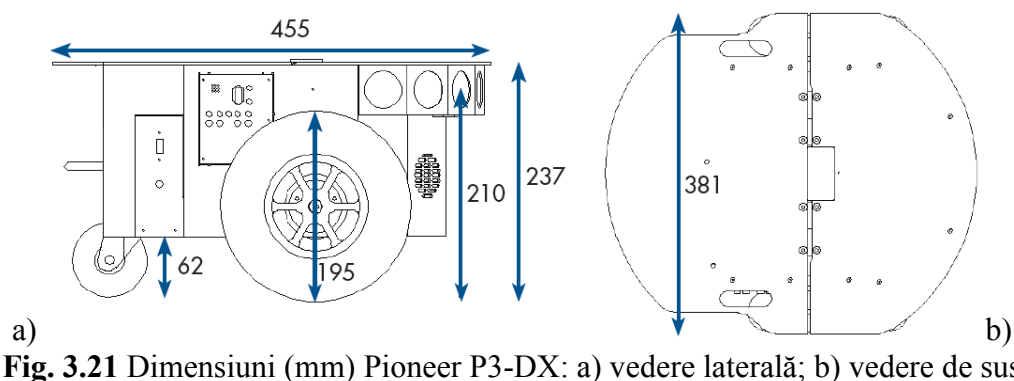
Platforma mobilă Pioneer 3-DX (**Fig. 3.20**) face parte dintr-o familie de roboți mobili produși de compania MOBILE ROBOTS. În aceasta familie intrând roboți Pioneer 1, Pioneer AT, Pioneer 2-DX, și multi alți. Aceste platforme mobile de dezvoltare și cercetare au în comun arhitectura și software-ul, fiind echipate cu sistem de conducere pe 2 sau 4 roți.



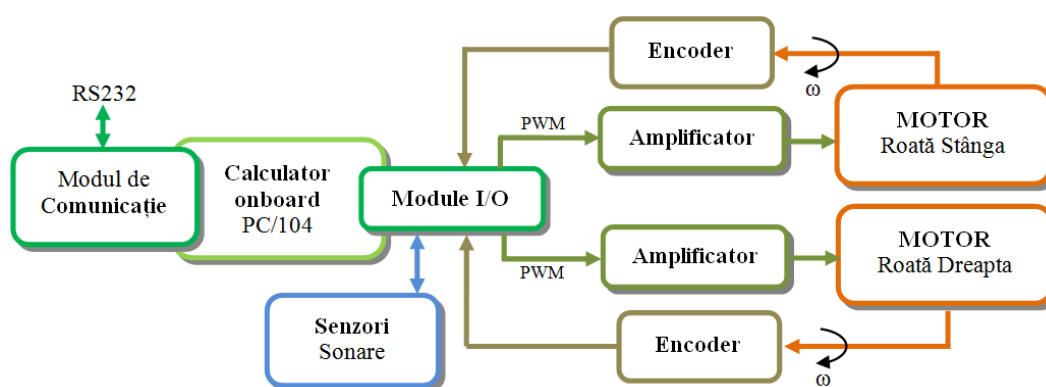
Fig. 3.20 Robot mobil Pioneer 3-DX

Pioneer 3-DX este dotat cu sistem de conducere onboard, devenind astfel un robot mobil autonom. Spre deosebire de alți roboți, dimensiunea mică a platformei mobile Pioneer 3-DX permite navigarea pe culoare înguste și spații aglomerate. Sistemul de conducere al robotului Pioneer 3-DX folosește două motoare de curent continuu, fiecare fiind echipat cu un encoder

optic de înaltă rezoluție pentru o poziționare precisă și o sensibilitate ridicată la determinarea vitezei.



Pioneer 3-DX poate urca în rampă cu o înclinație de maximum 25%, pe teren plan viteza robotului mobil poate ajunge până la 1,6 m/s (5,76 km/h). Cântărește 9 kg cu un număr minim de baterii. Aceste caracteristici îi permit să transporte o sarcină de pînă la 23kg. Robotul mobil este echipat în parte frontală cu un sistem de senzori de tip de sonare. Cele 8 sonare au rază de acțiune cuprinsă între 15cm și 5m. Poziționarea sonarelor este realizată prin dispunerea a câte un sonar pe fiecare parte a robotului din cele 8 iar celelalte 6 în fața, situate la un interval de 15° unul de celălalt. Opțional poate dispune și de un inel de sonare în partea din spate, cu aceeași configurație. Pe partea de control dispune de un calculator embedded PC/104 și module de I/O prin care se pot conecta diverse dispozitive externe.



Manipulatorul robotic Pioneer 5-DOF Arm (Fig. 3.23) poate manipula obiecte pînă la 150g. Pioneer 5-DOF Arm este alcătuit din:

- bază rotativă;
- 2 articulații mobile;
- o articulație pivotantă și rotativă;
- un gripper pivotant;



Fig. 3.23 Manipulator robotic Pioneer 5-DOF Arm

Toate articulațiile cu excepția gripperului pot pivota și roti cel puțin 180° . El se poate monta pe platforma mobilă Pioneer P3-DX și poate efectua o gamă largă de operații, având o rază de acțiune de 50 cm de la baza rotativă până la vârful gripperului.

3.10 Concluzii

Principalele contribuții rezultate în urma cercetărilor efectuate în cadrul prezentului capitol au vizat, în mod deosebit, analiza sistemelor flexibile de fabricație din cadrul proceselor de asamblare, dezasamblare și prelucrare pentru elaborarea unei noi structuri flexibile de fabricație capabilă să îndeplinească sarcini multiple cu aceleași echipamente în cadrul diferitelor procese de fabricație. Rezultatele analizei obținute au urmărit deservirea proceselor flexibile de fabricație de o platformă mobilă echipată cu manipulator robotic capabilă să îndeplinească două sarcini diferite, transport și manipulare. Ca rezultat al optimizării gradului de flexibilitate al echipamentelor și introducerea structurii robotice care deservește linia flexibilă de asamblare/dezasamblare, se poate evidenția în **Fig. 3.24** noua structură care permite posibilitatea dezasamblării complete a unui produs care s-a asamblat cu aceleași utilaje, dezasamblare deservită de o platformă mobilă echipată cu manipulator. Se evidențiază subsistemele flexibile de fabricație (prelucrare, transport, manipulare) care pot executa operații de asamblare/dezasamblare și prelucrare distincte cu aceleași utilaje.

Cele mai importante contribuții din cadrul acestui capitol sunt:

1. obținerea unei structuri optimizate de fabricație flexibilă care permite cu aceleași utilaje efectuarea a două procese distincte: asamblare și dezasamblare;

Capitolul 4

Contribuții privind analiza și modelarea proceselor de fabricație pe liniile flexibile deservite de roboți mobili

4.1 Introducere

4.2 Ipoteze de analiză și modelare a liniei flexibile HERA & Horstmann

4.3 Planificarea taskurilor aferente procesului de dezasamblare deservit de robot mobil echipat cu manipulator

4.4 Modelarea conducerii liniei de asamblare/dezasamblare deservită de robot mobil echipat cu manipulator cu ajutorul RP netemporizate în abordare SED

4.5 Modelarea conducerii liniei de asamblare/dezasamblare deservită de robot mobil echipat cu manipulator cu ajutorul RP temporizate în abordare SED

4.6 Modelarea conducerii liniei de asamblare/dezasamblare deservită de robot mobil echipat cu manipulator cu ajutorul RPH sincronizate

4.7 Ipoteze de analiză și modelare a liniei flexibile FESTO MP-200

4.8 Planificarea taskurilor aferente procesului de prelucrare al liniei flexibile FESTO MPS-200

4.9 Modelarea conducerii liniei de prelucrare FESTO MPS-200 cu ajutorul RP netemporizate în abordare SED

4.10 Concluzii

Principalul obiectiv urmărit în acest capitol este stabilirea de noi rezultate în modelarea proceselor de fabricație pe liniile flexibile deservite de roboți mobili echipați cu manipolatoare. Având în vedere structura particulară a liniei flexibile de asamblare/dezasamblare deservită de un robot mobil echipat cu manipulator, prezentată în Capitolul 3, au fost utilizate mai multe abordări de modelare pentru a evidenția cu acuratețe procesul real de fabricație. Diferite abordări de modelare au fost tratate în literatură în [40], [41], [47], [48].

4.1 Introducere

În SFF se desfășoară evenimente asincrone paralele (paralelism în timp), denumite și evenimente asincrone concurente. Întrucât aceste evenimente nesincronizate tind la atingerea unui scop comun, cel al funcționării întregului ansamblu, aceste evenimente coincid la intervalele de timp în care au loc [48], [49]. Asemenea evenimente asincrone paralele din SFF

pot fi prelucrările simultane ale unor piese pe diverse stații (mașini-unelte), efectuate în acelaș timp cu transportul și/sau manipularea unor semifabricate (sau piese) pe (sau de pe) alte stații, cu executarea unor diferite operații (de prelucrare, asamblare, dezasamblare etc.). Evenimentele menționate sunt asincrone, întrucât nu există un sistem de sincronizare a acestui număr mare de evenimente. Un astfel de sistem nu ar putea fi realizat, ținându-se seama de complexitatea SFF și de faptul ca evenimentele nu se desfasoară conform unor secvențe prestabilite cu precizie, ci ca rezultat al unor condiționări succesive care nu au caracter de invariabilitate, iar vitezele relative de desfasurare a evenimentelor nu pot fi dinainte cunoscute.

Existenta evenimentelor asincrone paralele impune tehnici complexe de modelare adoptate pentru conducerea unui SFF iar numărul relativ redus de tehnici utilizate până în prezent a fost determinat de acest inconvenient, precum și de cerintele impuse modelului: generalitatea aplicării, simplitatea utilizării și fidelitatea reprezentării. Principalele tehnici de analiză și modelare folosite în prezent și descrise pentru a evidentia SFF deservite de roboți prin evenimentele asincrone paralele sunt *rețelele Petri* (RP).

Principalele obiective ale acestui capitol sunt:

- Analiza și modelarea dinamicii sistemului de asamblare și dezasamblare deservită de roboți mobili a liniei flexibile HERA & Horstmann la apariția evenimentelor utilizând RP clasice;
- Analiza și modelarea dinamicii sistemului de asamblare și dezasamblare deservită de roboți mobili a liniei flexibile HERA & Horstmann la apariția evenimentelor a căror durată este cuantificată și considerată în model utilizând RP temporizate;
- Analiza și modelarea dinamicii sistemului de asamblare și dezasamblare deservită de roboți mobili a liniei flexibile HERA & Horstmann în care se consideră atât aspectul de SED dar și cel al variației continue a stării utilizând *RP Hibride* (RPH);
- Analiza și modelarea dinamicii sistemului de prelucrare FESTO MPS-200 la apariția evenimentelor utilizând RP clasice;

4.2 Ipoteze de analiză și modelare a liniei flexibile HERA & Horstmann

În cadrul analizei și modelării liniei flexibile de asamblare/dezasamblare HERA & Horstmann se introduc inițial ipoteze de funcționare a procesului de asamblare și dezasamblare.

- a. Analiza procesului de asamblare (**Fig. 4.1**): la pornirea liniei flexibile HERA & Horstmann, se declanșează procesul de asamblare.
 - inițial toate cele patru magazii aferente celor patru stații de asamblare sunt încărcate cu piese;

- la stația S1 se declanșează primul proces de asamblare, prin plasarea unui palet P1 pe banda transportoare în dreptul primei magazii. După plasarea paletului pe banda transportoare acesta este transportat până în dreptul stației S2. După eliberarea paletului stația S1 revine la starea inițială;
- când paletul P1 ajunge în dreptul stației S2 se pornește banda transportoare și paletul este adus în dreptul magaziei de piese a stației S2 unde se montează corpul P2 pe paletul P1. După operația de asamblare se transportă paletul P1 până la stația S3, moment în care stația 2 revine la starea inițială;
- Când paletul P1 ajunge în dreptul stației S3 se pornește banda transportoare și paletul este adus în dreptul magaziei de piese a stației S3 unde se montează capacul P3 pe corpul P2. După operația de asamblare se transportă paletul P1 până la stația S4, moment în care stația S3 revine la starea inițială;
- Când paletul P1 ajunge în dreptul stației S4 se pornește banda transportoare și paletul este adus în dreptul magaziei de piese a stației S4 unde se montează cei doi cilindri P4, cel de metal și respectiv cel de plastic în corpul P2. În cadrul stației S4 după montarea celor doi cilindri P4 în corpul P2, paletul P1 trece printr-un control de calitate în care se verifică cei doi cilindri, dacă sunt din material plastic sau metal. După operația de testare se consideră că s-a terminat procesul de asamblare și a rezultat un produs care este transportat până la stația următoare, moment în care stația 4 revine la starea inițială;
- după efectuarea controlului de calitate la stația S4, produsul final este transportat către magazia de stocare a produsului final;

b. Analiza procesului de dezasamblare deservit de robot mobil (**Fig. 4.2**): dacă se constată că produsul final este rebut se declanșează procesul de dezasamblare. Dezasamblarea este un proces accidental care se declanșează atunci când produsul nu trece testul de calitate. Un produs este considerat rebut dacă în componența sa are un cilindru de metal și unul de plastic sau ambele cilindri sunt de metal.

- din magazia de produse este readus pe banda transportoare a stației S5 produsul considerat rebut;

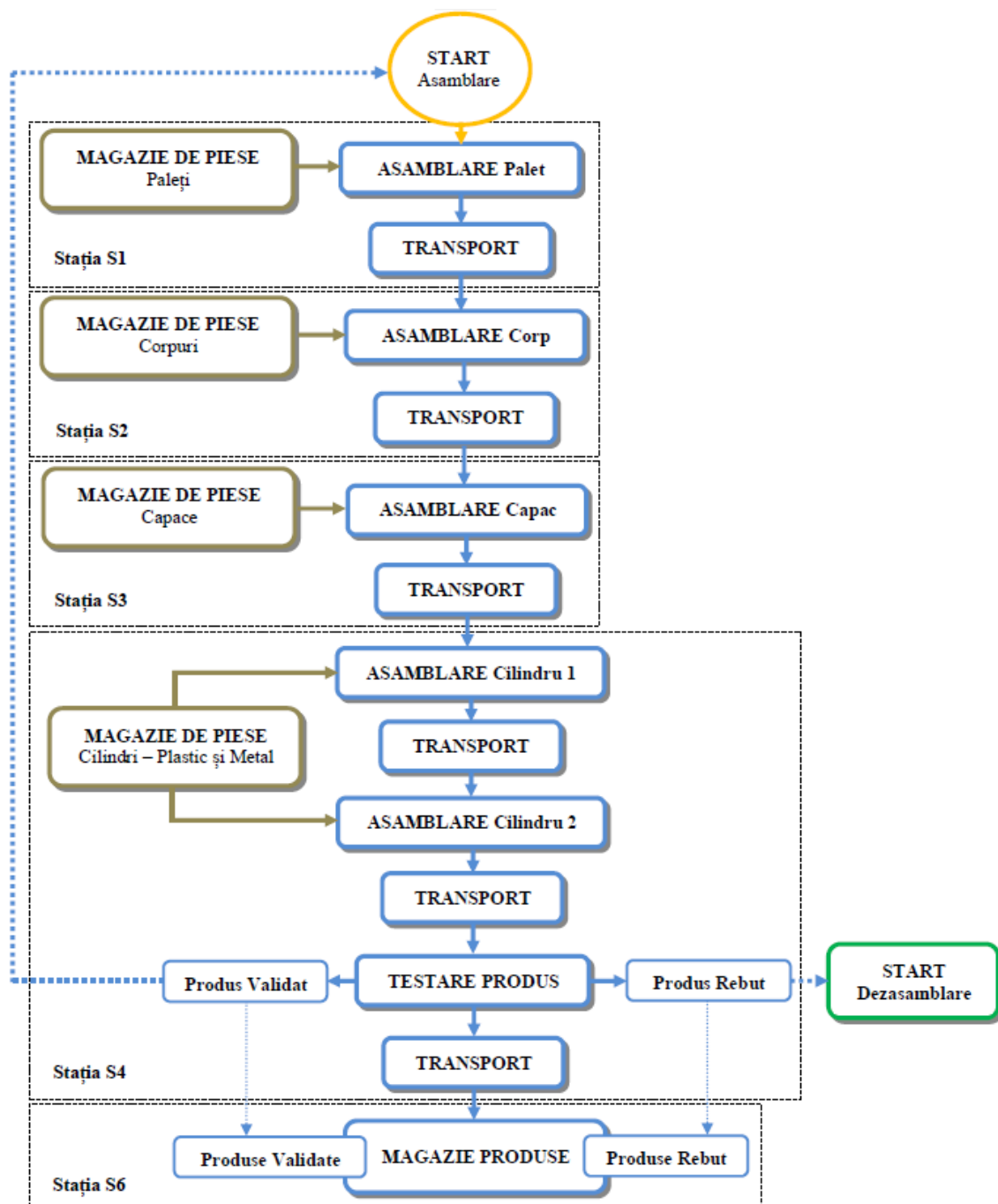


Fig. 4.1 Schema bloc a operațiilor de asamblare a unui produs

- când produsul de pe banda transportoare ajunge în dreptul stației S5 în care se găsesc două pistoane pneumatice se efectuează operația de dezasamblare. Scoaterea celor doi cilindri de metal sau plastic și metal se efectuează separat de către cele două pistoane pneumatice. Cilindri scoși sunt depozitați în cele două magazine aferente pistoanelor pneumatice. După scoaterea primului cilindru robotul mobil echipat cu manipulator preia primul cilindru din magazie și îl transportă la magazia de cilindri aferentă stației S4, după această operație se întoarce la magazia aferentă celui de-al doilea cilindru și

efectuează aceeași operație, moment în care produsul este transportat pe banda transportoare către stația S3 pentru următorul proces de dezasamblare;

- când produsul ajunge în dreptul stației S3 se efectuează operația de dezasamblare a capacului P3 cu ajutorul robotului mobil echipat cu manipulator. După preluarea piesei P3 de către manipulator, aceasta este transportată la magazia de capace aferentă stației S3, moment în care produsul este transportat pe banda transportoare către stația S2 pentru următorul proces de dezasamblare;
- când produsul ajunge în dreptul stației S2 se efectuează operația de dezasamblare a corpului P2 cu ajutorul robotului mobil echipat cu manipulator. După preluarea piesei P2 de către manipulator, aceasta este transportată la magazia de corpuri aferentă stației S2, moment în care produsul este transportat pe banda transportoare către stația S1 pentru următorul proces de dezasamblare;
- când produsul ajunge în dreptul stației S1 se efectuează operația de preluare a paletului P1 cu ajutorul robotului mobil echipat cu manipulator. După preluarea piesei P1 de către manipulator, aceasta este transportată la magazia de paleți aferentă stației S1, moment în care se încheie procesul de dezasamblare iar linia flexibilă devine complet reversibilă. După terminarea operației de dezasamblare robotul mobil se reîntoarce la poziția inițială, fiind inițializat pentru următorul proces de dezasamblare iar linia flexibilă este pregătită pentru o nouă asamblare;

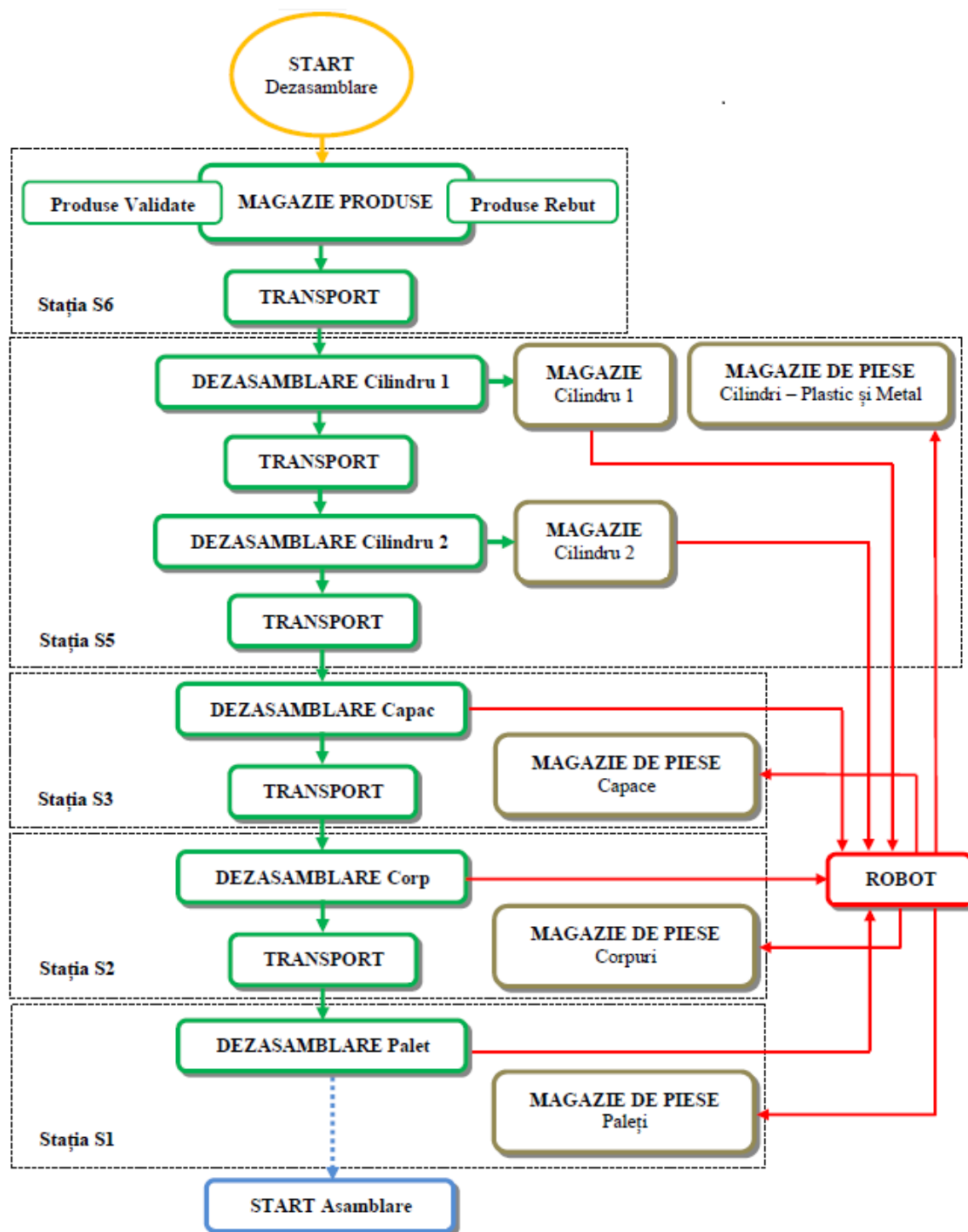


Fig. 4.2 Schema bloc a operațiilor de dezasamblare deservită de un robot mobil a unui produs

4.3 Planificarea taskurilor aferente procesului de dezasamblare deservit de robot mobil echipat cu manipulator

Conform ipotezelor descrise în subcapitolul 4.2 în **Fig. 4.3** și **Fig. 4.4** sunt prezentate zonele și traiectoriile pe care trebuie să le parcurgă platforma mobilă în decursul procesului de dezasamblare. Traiectoriile sunt linii drepte pe care le parcurge robotul mobil echipat cu

manipulator pentru a transporta la magaziile de depozitare piesele rezultate în urma operațiilor de dezasamblare.

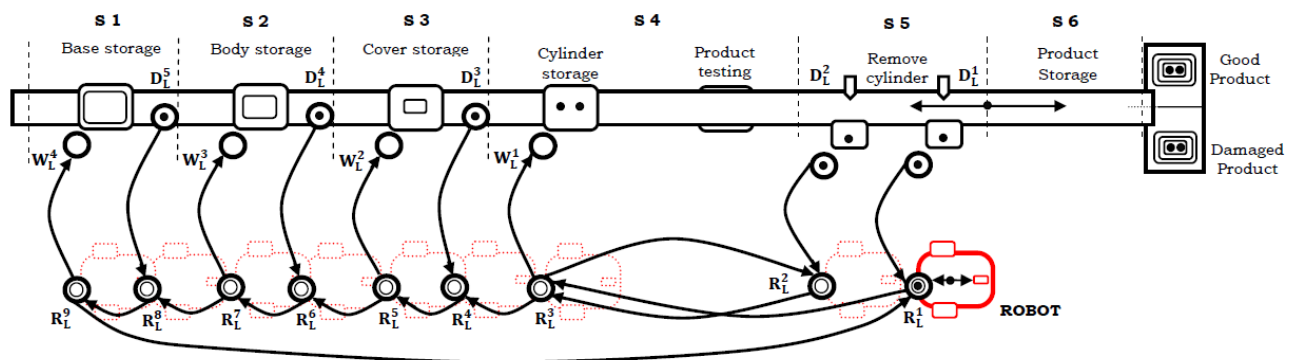


Fig. 4.3 Ciclul complet al robotului mobil echipat cu manipulator în procesul de dezasamblare

- R_L^1, \dots, R_L^9 : reprezintă numărul și poziția locațiilor în care se găsește robotul mobil echipat cu manipulator în cadrul procesului de dezasamblare;
- D_L^1 : reprezintă numărul și locația în care se dezasamblează cilindrul numărul 1;
- D_L^2 : reprezintă numărul și locația în care se dezasamblează cilindrul numărul 2;
- D_L^3 : reprezintă numărul și locația în care se dezasamblează capacul P3;
- D_L^4 : reprezintă numărul și locația în care se dezasamblează corpul P2;
- D_L^5 : reprezintă numărul și locația în care se dezasamblează paletul P1;
- W_L^1 : reprezintă numărul și locația în care se depozitează cilindrul numărul 1;
- W_L^2 : reprezintă numărul și locația în care se depozitează cilindrul numărul 2;
- W_L^3 : reprezintă numărul și locația în care se depozitează capacul P3;
- W_L^4 : reprezintă numărul și locația în care se depozitează corpul P2;
- W_L^5 : reprezintă numărul și locația în care se depozitează paletul P1;

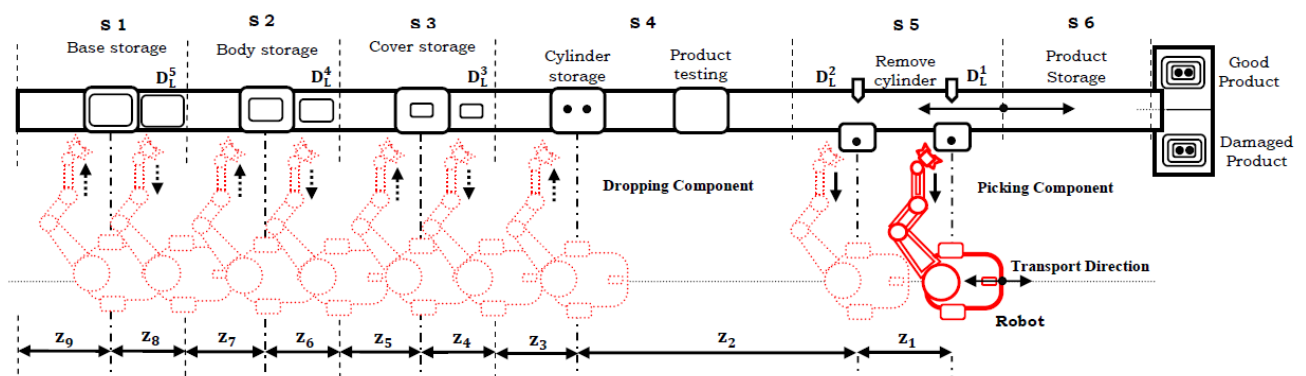


Fig. 4.4 Împărțirea pe secțiuni și zone a procesului de dezasamblare

- Z_1, \dots, Z_9 : reprezintă numărul și zonele aferente fiecărei stații a liniei flexibile HERA & Horstman pe care robotul mobil echipat cu manipulator trebuie să le parcurgă în cadrul procesului de dezasamblare;

Strategia de dezasamblare este bazată pe un graf de reprezentare a produsului asamblat în care relațiile dintre componente sunt exprimate prin săgeți. Utilizând acest graf se elaborează o planificare a taskurilor prin care se determină secvența în care componentele sunt dezasamblate și transportate la magazinele de depozitare [36], [37], [38], [51], [52], .

Dacă un produs final asamblat este declarat rebut, planificarea taskurilor furnizează cea mai bună secvență pentru dezasamblarea produsului. Se propune planificarea taskurilor din **Fig. 4.5**.

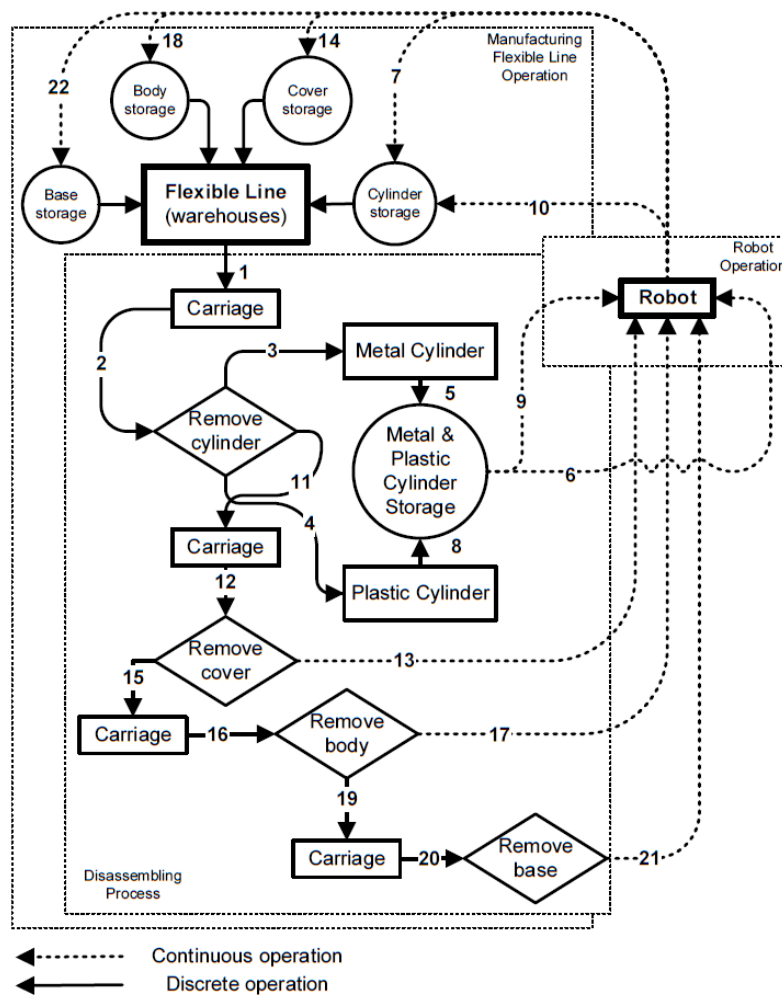


Fig. 4.5 Planificarea taskurilor pentru operația de dezasamblare deservită de un robot mobil echipat cu manipulator

4.4 Modelarea procesului de asamblare/dezasamblare pe linia flexibilă deservită de robot mobil echipat cu manipulator utilizând RP netemporizate în abordare SED

O rețea Petri este [53], [54] un cvintuplu, $PN = (P, T, F, W, M0)$ unde:

- $P = \{P_1, P_2, \dots, P_n\}$ este o mulțime finită de locații, unde:

$$P = \{P_1, P_2, \dots, P_n\} = P^A \cup P^D \quad (4.1)$$

Unde,

$$P^D = \{Pa_i\}_{i=\overline{1,n}} \cup \{Pd_j\}_{j=\overline{1,n}} \cup \{Pr_k\}_{k=\overline{1,n}} \quad (4.2)$$

- $\{Pa_i\}_{i=\overline{1,n}}$ este mulțimea locațiilor aferente procesului de asamblare;
- $\{Pd_j\}_{j=\overline{1,n}}$ este mulțimea locațiilor aferente procesului de dezasamblare;
- $\{Pr_k\}_{k=\overline{1,n}}$ este mulțimea locațiilor pentru stările platformei mobile în timpul procesului de dezasamblare;
- $T = \{Ta_1, Ta_2, \dots, Ta_n\}$ este o mulțime de tranziții, unde:

$$T = \{T_1, T_2, \dots, T_m\} = T^A \cup T^D \quad (4.4)$$

Unde,

$$T^D = \{Ta_i\}_{i=\overline{1,n}} \cup \{Td_j\}_{j=\overline{1,n}} \cup \{Tr_k\}_{k=\overline{1,n}} \quad (4.5)$$

- $\{Ta_i\}_{i=\overline{1,n}}$ sunt tranzițiile pentru operațiile de asamblare;
- $\{Td_j\}_{j=\overline{1,n}}$ sunt tranzițiile pentru operațiile de dezasamblare;
- $\{Tr_k\}_{k=\overline{1,n}}$ sunt tranzițiile discrete asociate stărilor platformei mobile care deservește procesul de dezasamblare;
- $F \subseteq (P \times T) \cup (T \times P)$ este o mulțime de arce;
- $W: F \rightarrow \{1, 2, 3, \dots\}$ este o funcție de pondere a arcelor;
- $M0: P \rightarrow \{1, 2, 3, \dots\}$ este o funcție de marcaj inițial;
- n reprezintă numărul maxim de locații sau de tranziții asociate operațiilor de asamblare, dezasamblare și operațiilor de deservire a liniei flexibile de un robot mobil;

Se poate spune:

- Mulțimile P și T sunt disjuncte $P \cap T = \{\emptyset\}$;
- Pentru a asigura obiectivul definiției de mai sus, mulțimile P și T satisfac condiția $P \cup T = \{\emptyset\}$;

- c. O structură de rețea Petri $N = (P, T, F, W)$ fără nici o specificație referitoare la marcaj se poate nota cu N , notație care desemnează topologia rețelei;
- d. O rețea Petri cu un marcaj inițial M_0 se va nota prin (N, M_0) ;
- e. O rețea Petri cu un marcaj oarecare M se va nota prin (N, M) ;

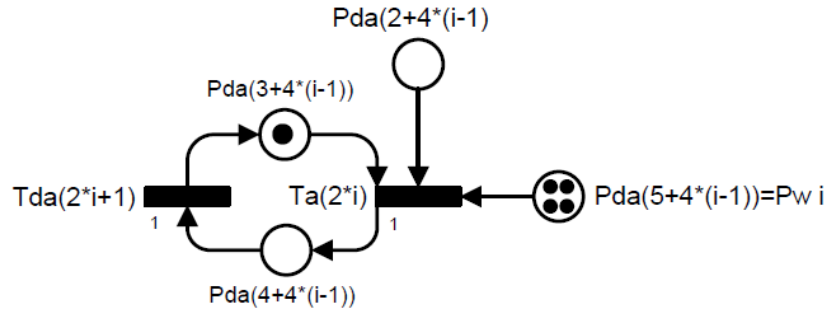


Fig. 4.6 Modelarea procesului de asamblare a unei componente pe linia flexibilă cu RP netemporizate în abordare discretă

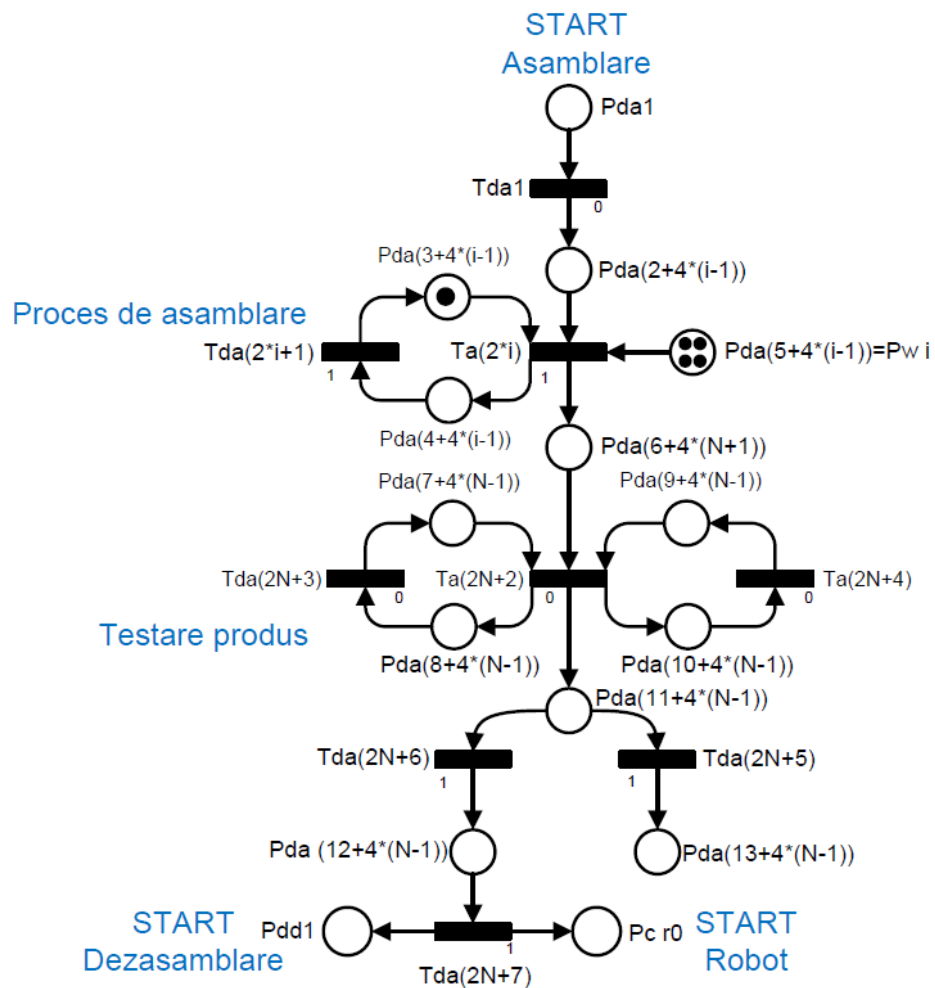


Fig. 4.7 Modelarea procesului de testare și depozitare a unui produs pe linia flexibilă cu RP netemporizate în abordare discretă

Fie F mulțimea tuturor arcelor rețelei Petri N . Se definește mulțimea predecessor și mulțimea successor a tranziției T două mulțimi de poziție definite prin:

- $*T = \{P | (P, T) \in F\}$ este mulțimea tuturor pozițiilor de intrare ale lui T ;
- $T* = \{P | (T, P) \in F\}$ este mulțimea tuturor pozițiilor de ieșire ale lui T ;

Se definește mulțimea predecesor și mulțimea succesor a poziției P două mulțimi de tranziții definite prin:

- $*P = \{T | (T, P) \in F\}$ este mulțimea tuturor tranzițiilor de intrare ale lui P ;
- $P* = \{T | (P, T) \in F\}$ este mulțimea tuturor tranzițiilor de ieșire ale lui P ;

În **Fig. 4.6** se prezintă modelarea unui singur proces de asamblare, în care, din magazia de piese este preluată o componentă și prin operația de asamblare aceasta intră în componența produsului. Astfel, se pot modela utilizând rețele Petri netemporizate un număr de n stații de asamblare, în abordare discretă. În **Fig. 4.7** se prezintă modelarea unui proces de asamblare a unei componente care este supusă unui test de calitate în vederea stocării produsului final pe două criterii: produse bune (care au fost validate la testul de calitate) și produse cu defect (care au fost invalidate la testul de calitate). La apariția unui produs declarat defect este inițializat procesul de dezasamblare iar produsul este întors din magazia aferentă pe linia de asamblare pentru a fi dezasamblat în vederea recuperării părților componente.

Considerând planificarea taskurilor din **Fig. 4.5** și descrierea operațiilor din **Fig 4.1** în **Fig. 4.8** este prezentată RP netemporizată ce reprezintă modelul liniei flexibile HERA & Horstmann în cadrul procesului de asamblare în abordare SED [54].

Unde,

$$P = \{Pa_1, Pa_2, \dots, Pa_{29}\} = P^A \quad (4.6)$$

Unde,

$$P^A = \{Pa_i\}_{i=\overline{1,29}} \quad (4.7)$$

- $\{Pd_i\}_{i=\overline{1,29}}$ este mulțimea locațiilor aferente procesului de asamblare;

Unde,

$$T = \{Ta_1, Ta_2, \dots, Ta_{17}\} = T^A \quad (4.8)$$

Unde,

$$T^A = \{Ta_i\}_{i=\overline{1,17}} \quad (4.9)$$

- $\{Ta_i\}_{i=\overline{1,n}}$ sunt tranzițiile pentru operațiile de asamblare;
- $M0: Pa_1$ (reprezintă START proces de asamblare) este o funcție de marcaj inițial;

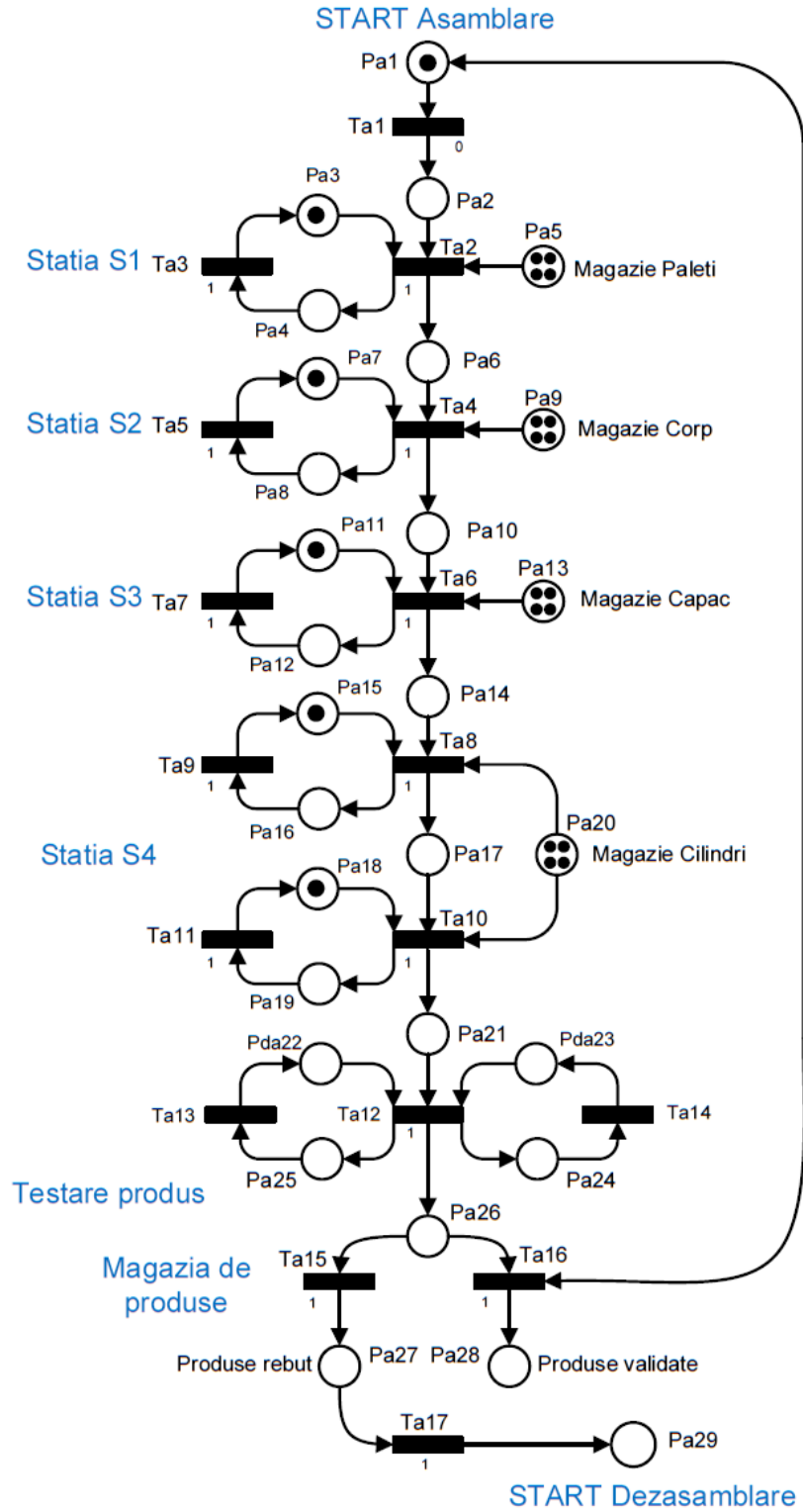


Fig. 4.8 Modelarea procesului de asamblare pe linia flexibilă HERA & Horstmann cu RP netemporizate în abordare discretă

Considerând planificarea taskurilor din **Fig. 4.5** și descrierea operațiilor din **Fig. 4.1** în **Fig. 4.9** este prezentată RP netemporizată ce reprezintă modelul liniei flexibile HERA & Horstmann în cadrul procesului de dezasamblare parțială în abordare SED.

Unde,

$$P = \{Pd_1, Pd_2, \dots, Pd_{18}\} = P^D \quad (4.10)$$

Unde,

$$P^D = \{Pd_j\}_{j=\overline{1,18}} \quad (4.11)$$

- $\{Pd_j\}_{j=\overline{1,18}}$ este mulțimea locațiilor aferente procesului de dezasamblare parțială;

Unde,

$$T = \{Td_1, Td_2, \dots, Td_{11}\} = T^D \quad (4.12)$$

Unde,

$$T^D = \{Td_j\}_{j=\overline{1,11}} \quad (4.13)$$

- $\{Td_j\}_{j=\overline{1,11}}$ sunt tranzițiile pentru operațiile de asamblare;
- $M0: Pd_1$ (reprezintă START proces de dezasamblare parțială) este o funcție de marcaj inițial;

Prin modelarea procesului de dezasamblare parțială prezentată în **Fig. 4.9** se pun în evidență operațiile prin care sunt dezasamblate două componente din cadrul produsului și depozitate în magaziile aferente celor două locații de dezasamblare. Dezasamblarea parțială a celor două componente face parte din cadrul procesului de asamblare și dezasamblare (parțială) pe linia flexibilă HERA & Horstmann. În urma optimizării procesului de dezasamblare și introducerii robotului mobil echipat cu manipulator care deservește linia flexibilă se obține o dezasamblare totală a produsului și astfel linia flexibilă devine complet reversibilă, modelarea acestor operații este prezentată mai jos.

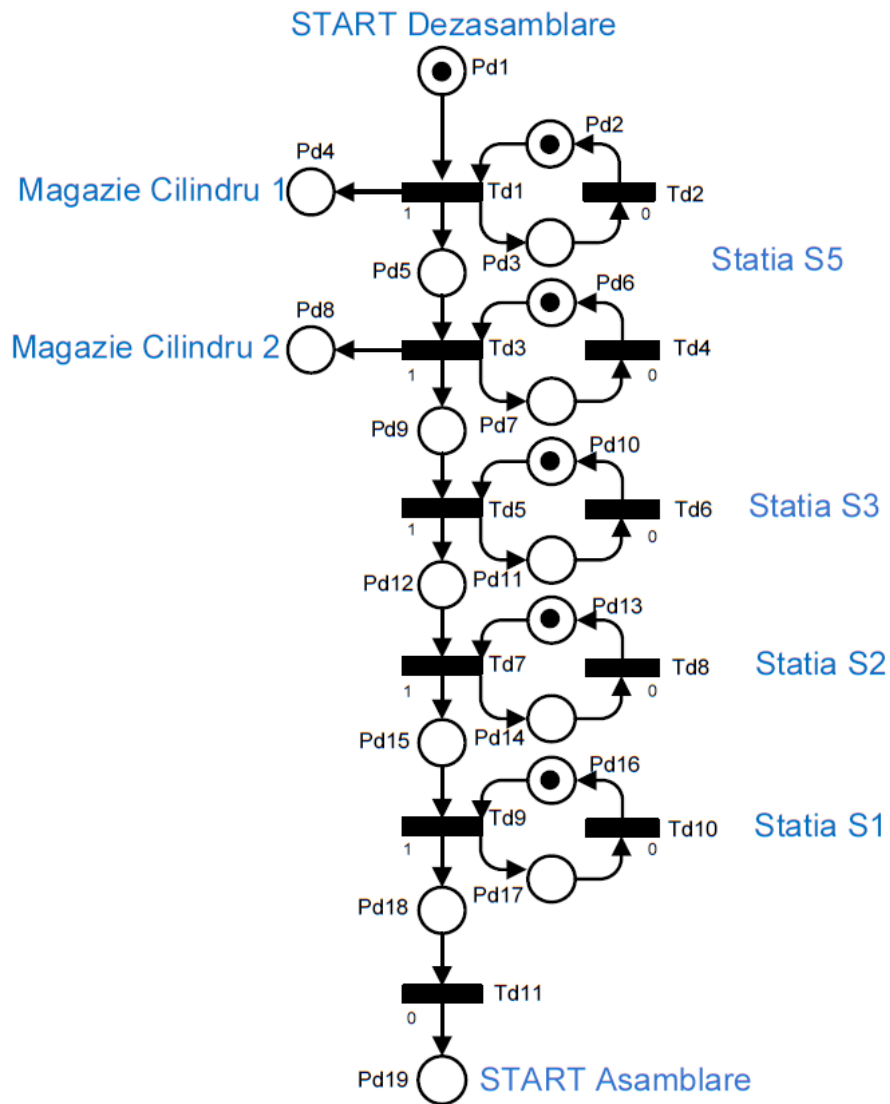


Fig. 4.9 Modelarea procesului de dezasamblare parțială pe linia flexibilă HERA & Horstmann cu RP netemporizate în abordare discretă

Considerând planificarea taskurilor din **Fig. 4.5** și descrierea operațiilor din **Fig. 4.1** în **Fig. 4.10** este prezentată RP netemporizată ce reprezintă modelul liniei flexibile HERA & Horstman în cadrul procesului de dezasamblare parțială deservit de un robot mobil cu manipulator în abordare SED.

Unde,

$$P = \{Pd_1, Pd_2, \dots, Pd_{18}\} = P^D \quad (4.14)$$

Unde,

$$P^D = \{Pd_j\}_{j=\overline{1,18}} \cup \{Pr_k\}_{k=\overline{1,4}} \quad (4.15)$$

- $\{Pd_j\}_{j=\overline{1,18}}$ este mulțimea locațiilor aferente procesului de dezasamblare parțială;
- $\{Pr_k\}_{k=\overline{1,4}}$ este mulțimea locațiilor pentru stările platformei mobile în timpul procesului de dezasamblare parțială;

Unde,

$$T = \{Td_1, Td_2, \dots, Td_{11}\} = T^D \quad (4.16)$$

Unde,

$$T^D = \{Td_j\}_{j=\overline{1,11}} \cup \{Tr_k\}_{k=\overline{1,3}} \quad (4.17)$$

- $\{Td_j\}_{j=\overline{1,11}}$ sunt tranzițiile pentru operațiile de dezasamblare parțială;
- $\{Tr_k\}_{k=\overline{1,3}}$ sunt tranzițiile discrete asociate stărilor platformei mobile care deservește procesul de dezasamblare parțială;

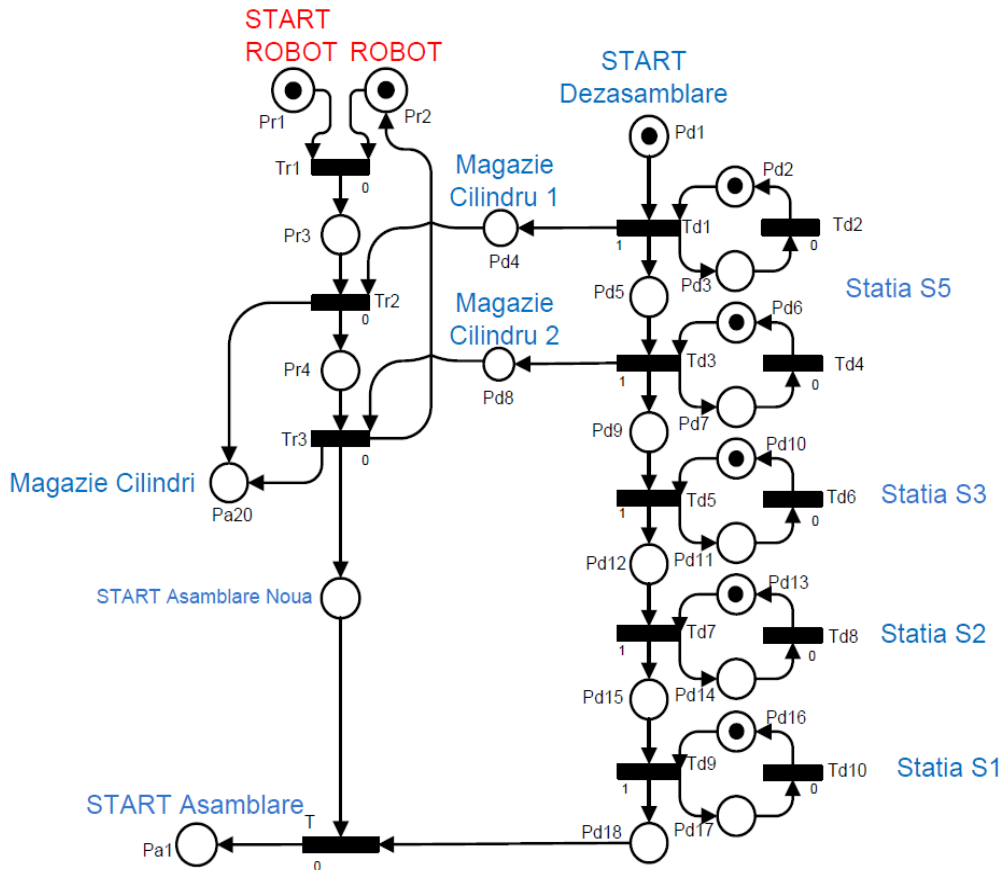


Fig. 4.10 Modelarea procesului de dezasamblare parțială pe linia flexibilă HERA & Horstmann deservită de un robot mobil cu manipulator utilizând RP netemporizate în abordare discretă

Considerând planificarea taskurilor din **Fig. 4.5** și descrierea operațiilor din **Fig. 4.1** în **Fig. 4.11** este prezentată RP netemporizată ce reprezintă modelul liniei flexibile HERA & Horstmann în cadrul procesului de dezasamblare totală deservit de un robot mobil cu manipulator în abordare SED.

Unde,

$$P = \{Pd_1, Pd_2, \dots, Pd_{18}\} = P^D \quad (4.18)$$

Unde,

$$P^D = \{Pd_j\}_{j=\overline{1,18}} \cup \{Pr_k\}_{k=\overline{1,7}} \quad (4.19)$$

- $\{Pd_j\}_{j=\overline{1,18}}$ este mulțimea locațiilor aferente procesului de dezasamblare totală;
- $\{Pr_k\}_{k=\overline{1,7}}$ este mulțimea locațiilor pentru stările platformei mobile în timpul procesului de dezasamblare totală;

Unde,

$$T = \{Td_1, Td_2, \dots, Td_{11}\} = T^D \quad (4.20)$$

Unde,

$$T^D = \{Td_j\}_{j=\overline{1,11}} \cup \{Tr_k\}_{k=6} \quad (4.21)$$

- $\{Td_j\}_{j=\overline{1,11}}$ sunt tranzițiile pentru operațiile de dezasamblare parțială;
- $\{Tr_k\}_{k=\overline{1,6}}$ sunt tranzițiile discrete asociate stărilor platformei mobile care deservește procesul de dezasamblare parțială;
- $M0: Pd_1$ (reprezintă START proces de dezasamblare) este o funcție de marcaj inițial;

Considerând planificarea taskurilor din **Fig. 4.5** și descrierea operațiilor din **Fig. 4.1** în **Fig. 4.12** este prezentată RP netemporizată ce reprezintă modelul liniei flexibile HERA & Horstmann în cadrul procesului de asamblare și dezasamblare totală deservit de un robot mobil cu manipulator în abordare SED.

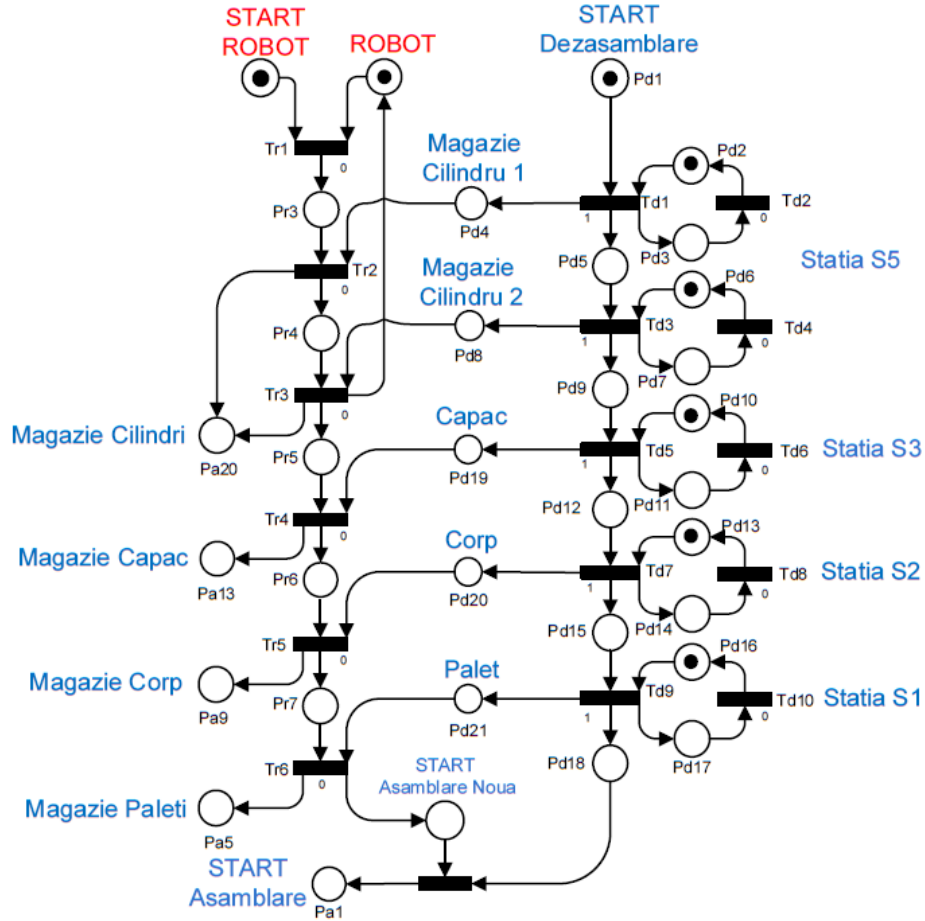


Fig. 4.11 Modelarea procesului de dezasamblare totală pe linia flexibilă HERA & Horstmann deservită de un robot mobil cu manipulator utilizând RP netemporizate în abordare discretă

Unde,

$$P = \{Pa_1, Pa_2, \dots, Pa_{28}\} \cup \{Pd_1, Pd_2, \dots, Pd_{18}\} = P^A \cup P^D \quad (4.22)$$

Unde,

$$P = \{Pa_i\}_{i=\overline{1,28}} \cup \{Pd_j\}_{j=\overline{1,18}} \cup \{Pr_k\}_{k=\overline{1,n}} \quad (4.23)$$

- $\{Pa_i\}_{i=\overline{1,28}}$ este mulțimea locațiilor aferente procesului de asamblare;
- $\{Pd_j\}_{j=\overline{1,18}}$ este mulțimea locațiilor aferente procesului de dezasamblare totală;
- $\{Pr_k\}_{k=\overline{1,7}}$ este mulțimea locațiilor pentru stările platformei mobile în timpul procesului de dezasamblare totală;

Unde,

$$T = \{Ta_1, Ta_2, \dots, Ta_{17}\} \cup \{Td_1, Td_2, \dots, Td_{11}\} = T^A \cup T^D \quad (4.24)$$

Unde,

$$T = \{Ta_i\}_{i=\overline{1,17}} \cup \{Td_j\}_{j=\overline{1,11}} \cup \{Tr_k\}_{k=\overline{1,6}} \quad (4.25)$$

- $\{Ta_i\}_{i=\overline{1,17}}$ sunt tranzițiile pentru operațiile de asamblare;
- $\{Td_j\}_{j=\overline{1,11}}$ sunt tranzițiile pentru operațiile de dezasamblare totală;
- $\{Tr_k\}_{k=\overline{1,6}}$ sunt tranzițiile discrete asociate stărilor platformei mobile care deservește procesul de dezasamblare totală;
- $M0: Pa_1$ (reprezintă START proces de asamblare) este o funcție de marcaj inițial;

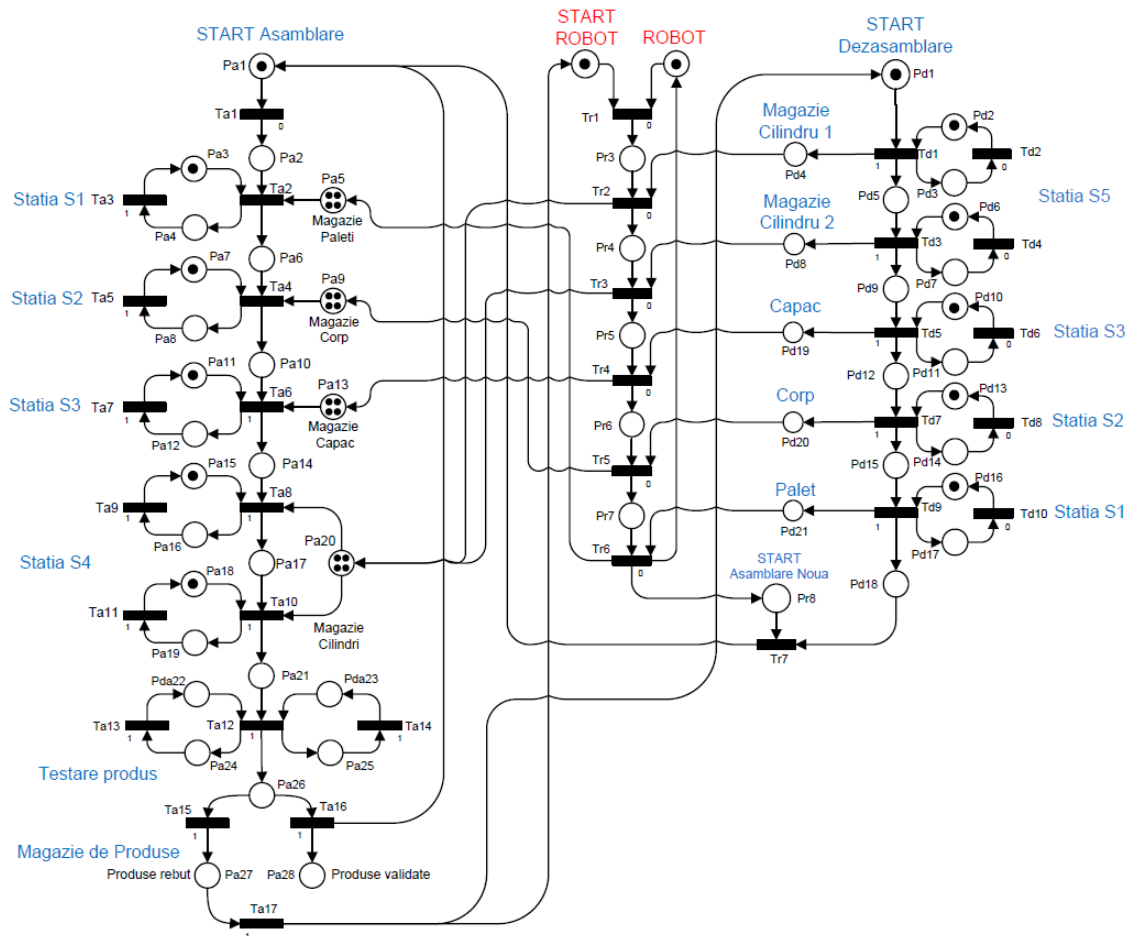


Fig. 4.12 Modelarea procesului de asamblare/dezasamblare totală pe linia flexibilă HERA & Horstmann deservită de un robot mobil cu manipulator cu RP netemporizate în abordare discretă

4.5 Modelarea procesului de asamblare/dezasamblare pe linia flexibilă deservită de robot mobil echipat cu manipulator cu ajutorul RP temporizate în abordare SED

Modelarea procesului de asamblare și dezasamblare pe linia flexibilă HERA & Horstmann deservită de un robot mobil echipat cu manipulator utilizând rețele Petri netemporizate în

abordare discretă se realizează la nivel de stare: în care se determină ce acțiuni se produc în sistem, care stări preced acestor acțiuni și în ce stări va trece sistemul după producerea acțiunilor precedente, astfel este pus în evidență întregul comportament.

Prin analiza rețelei Petri netemporizată se cunosc stările în care s-a aflat sau nu sistemul, care sunt în principiu, stările necontrolabile, etc., însă o astfel de analiză nu poate oferi informații suficiente despre caracteristicile numerice care determină stările sistemului și nu poate pune în evidență cât mai real comportamentul sistemului, de aceea am utilizat în modelare și rețele Petri temporizate în abordare discretă.

Rețelele Petri temporizate se împart în două categorii:

- Rețele Petri cu tranziții temporizate sau se poate spune temporizată T ;
- Rețele Petri cu poziții temporizate sau se poate spune temporizată P ;

Se poate define o rețea cu tranziții temporizate, dacă fiecărei tranziții $T_i, i = 1, \dots, n$, i se asociază un interval de timp $d_i \geq 0$, prin intermediul unei funcții de temporizare de tip T (tranziție).

În funcționarea rețelei Petri temporizate T , intervalele de timp $d_i \geq 0$ au rolul de întârzieri, care reprezintă durate din procesul real, acestea se comportă după cum urmează: din momentul când tranziția T_i este validată, un număr de $a_{i,j}$ jetoane vor fi ocupate (acestea nefiind disponibile din anumite puncte de vedere logic sau al aplicării de reguli asupra tranziției) în poziția P_j care precede T_i pentru d_i unități de timp, înainte de deplasarea lor prin validarea tranziției T_i .

În cazul în care două sau mai multe tranziții aflate în conflict, selectarea tranziției care se va executa se realizează pe baza unui mecanism de priorități asignate acelor tranziții.

Se poate define o rețea cu poziții temporizate, dacă fiecărei poziții $P_j, j = 1, \dots, n$, i se asociază un interval de timp $d_j \geq 0$, prin intermediul unei funcții de temporizare de tip P (poziție). În funcționarea rețelei Petri temporizate P , intervalele de timp $d_j \geq 0$ joacă rolul unei întârzieri ce se manifestă după cum urmează: din momentul când tranziția T_i care precede P_j este executată, cu un număr de a_{ij}^+ jetoane vor rămâne nedisponibile în poziția P_j pentru d_j unități de timp, înainte de a putea fi utilizate pentru a valida tranzițiile ce succed P_j . În cazul temporizării P , se presupune că executarea oricărei tranziții are loc instantaneu.

Astfel rețeaua Petri temporizată P se folosește pentru a introduce în modelul matematic informațiile privitoare la durata activităților, elaborând astfel, un model cantitativ [50], [54], [55]. Cu ajutorul acestui model se pot lua în discuție toate caracteristicile temporale specifice începerii unui proces de asamblare a unui nou produs, urmat de un proces de dezasamblare la apariția unui produs declarat defect (nevalidat), urmat de anumite succesiuni de operații pe

care le execută robotul mobil echipat cu manipulator care deserveste întregul proces de dezasamblare.

Considerând planificarea taskurilor din **Fig. 4.5**, descrierea operațiilor din **Fig. 4.1** și ecuațiile 4.22, 4.23, 4.24, 4.25 în **Fig. 4.13** este prezentată RP temporizată ce reprezintă modelul liniei flexibile HERA & Horstmann în cadrul procesului de asamblare și dezasamblare totală deservit de un robot mobil cu manipulator în abordare SED.

În cadrul procesului de asamblare, tranzițiile temporizate sunt:

- $(Ta_1, Ta_3, Ta_5, Ta_7, Ta_9, Ta_{11}) = 0$, acestor tranziții li se atribuie valoarea zero, deoarece fiecarei tranziții îi corespunde o stare din proces care se întâmplă instantaneu;
- $Ta_2 = 9.5$ s; $Ta_4 = 9.3$ s; $Ta_6 = 8.5$ s; $Ta_8 = 0.5$ s; $Ta_{10} = 4.5$ s; $Ta_{12} = 27.2$ s; $Ta_{13} = Ta_{14} = 19.3$ s;

În cadrul procesului de dezasamblare, tranzițiile temporizate sunt:

- $(Td_2, Td_4, Td_6, Td_8, Td_{10}) = 0$, acestor tranziții li se atribuie valoarea zero, deoarece fiecarei tranziții îi corespunde o stare din proces care se întâmplă instantaneu;
- $Td_1 = 6.1$ s; $Td_3 = 22.3$ s; $Td_5 = 7.9$ s; $Td_7 = 8.8$ s; $Td_9 = 1$ s;

4.6 Modelarea procesului de asamblare/dezasamblare pe linia flexibilă deservită de robot mobil echipat cu manipulator cu ajutorul RPH sincronizate

Pentru elaborarea modelului global al asamblării și/sau dezasamblării, se va considera aspectul hibrid al procesului de asamblare/dezasamblare deservit de platforma mobilă. Instrumentul dedicat modelării sunt rețele Petri hibride care întregesc atât aspectul discret (în modelarea proceselor de asamblare/dezasamblare) dar și cel continuu (deplasarea robotului mobil și acțiunile lui de manipulare a componentelor). Modelul final este de tip *RPH sincronizate* (RPHS) deoarece este interfațat cu evenimente externe de sincronizare, într-o abordare a modelării/simulării în timp real.

4.6.1 Modelul generalizat al RPHS

Modelarea structurii de RPHS aferentă procesului de asamblare și dezasamblare a unui produs cu N componente este prezentată în **Fig. 4.14**, unde:

- N reprezintă componentele produsului supuse operațiilor de asamblare/dezasamblare;
- $n_{a_i}, i = \overline{1, N}$ reprezintă localizarea pe axa Ox a sistemului de coordonate xOy, în sensul crescător al axei Ox, corespunzător etapei curente de asamblare a unui produs;

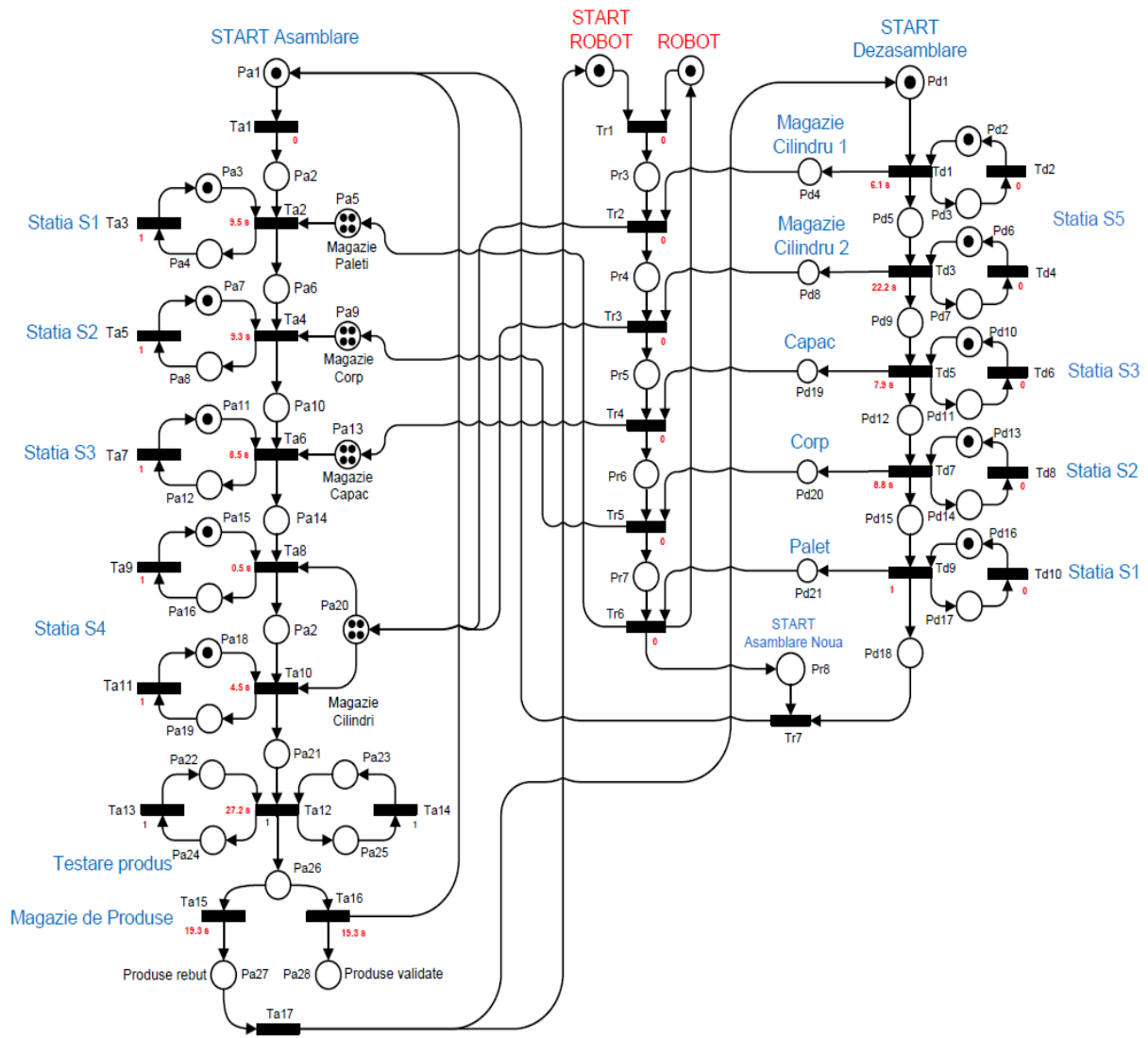


Fig. 4.13 Modelarea procesului de asamblare/dezasamblare totală pe liniei flexibile HERA & Horstmann deservit de un robot mobil cu manipulator cu RP temporizate în abordare discretă

- $n_{d_j}, j = \overline{1, N-1}$ reprezintă localizarea pe axa Ox a sistemului de coordonate xOy, în sensul descrescător al axei Ox, corespunzător etapei curente de dezasamblare;
- $W_i, i = \overline{1, N}$ reprezintă localizarea axa Ox a sistemului de coordonate xOy a magaziiilor de stocare a componentelor;

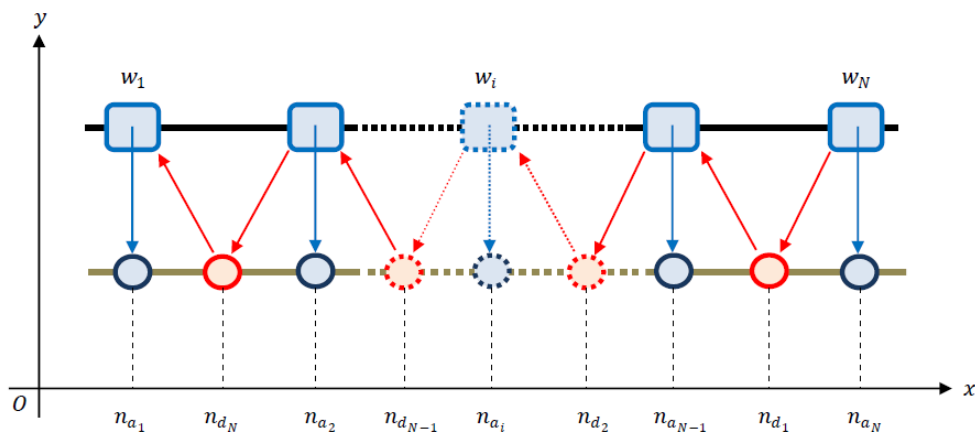


Fig. 4.14 Reprezentarea punctelor de lucru ale procesului de asamblare/dezasamblare pentru o piesă cu N componente

Conform **Fig. 4.14** notăm următoarele:

- $D(n_{d_j}, w_{N+1-j})$ reprezintă distanța dintre punctul de lucru curent al dezasamblării n_{d_j} și magazia aferentă componentei dezasamblate w_{N+1-j} ;
- $D(w_{N+1-j}, n_{d_{j+1}})$ reprezintă distanța dintre magazia aferentă componentei dezasamblate w_{N+1-j} și punctul de lucru curent $n_{d_{j+1}}$, corespunzător următoarei etape de dezasamblare;
- $D_{r_j} = D(n_{d_j}, w_{N+1-j}) + D(w_{N+1-j}, n_{d_{j+1}})$ reprezintă distanța parcursă de robotul mobil în cadrul etapei j de dezasamblare a unei componente;

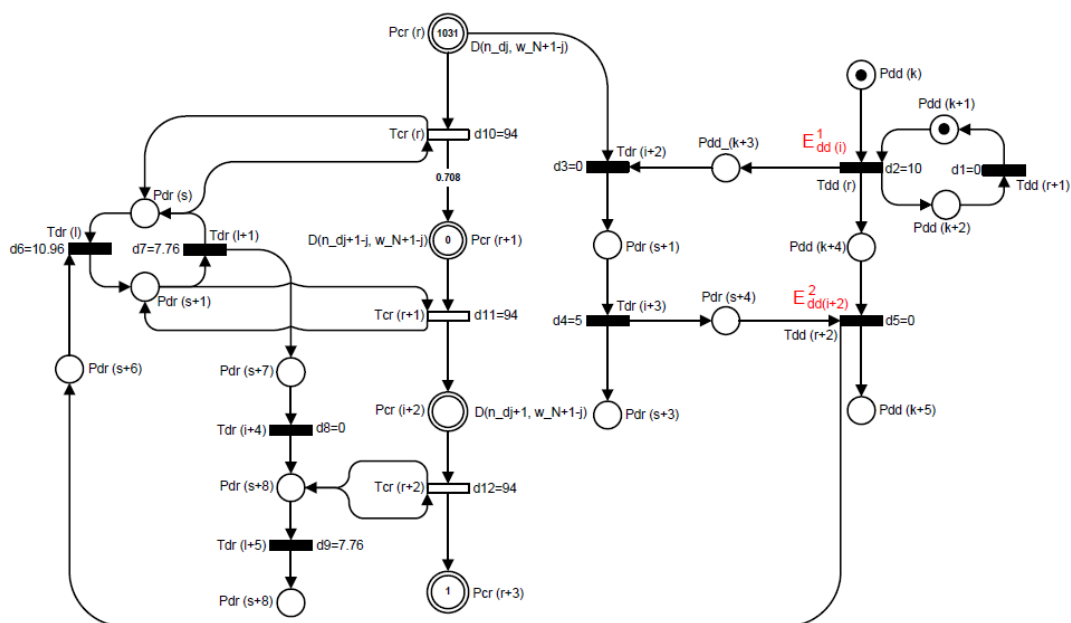


Fig. 4.15 Structura generalizată a modelului sincronizat corespunzător procesului de dezasamblare a N piese

Unde:

- $j = \overline{1, N-1}$ corespunde etapelor de dezasamblare;
- $i = N - j + 1$ corespunde etapelor de asamblare;
- $w_{N+1-j} = w_i, j = \overline{1, N-1}$ reprezintă localizarea pe axa Ox a sistemului de coordonate xOy a magaziiilor de stocare a componentelor la operația de dezasamblare, $w_i | i = \overline{1, N} = w_{N+1-j} |_{j=\overline{1, N}}$;
- $s = 1 + (j - 1) * 8$ se referă la Pdr - este mulțimea locațiilor discrete pentru stările platformei mobile în timpul procesului de dezasamblare;
- $r = 1 + (j - 1) * 3$ referă la Pcr - sunt locațiile continue pentru stările asociate deplasării pe traiectorie a platformei mobile corespunzătoare fiecărei operații de dezasamblare; Tcr - sunt tranzițiile continue pentru stările asociate deplasării pe traiectorie a platformei mobile corespunzătoare fiecărei operații de dezasamblare; Tdd - sunt tranzițiile discrete pentru operațiile de dezasamblare;
- $k = 1 + (j - 1) * 5$ se referă la Pdd - este mulțimea locațiilor discrete aferente procesului de dezasamblare;
- $l = 1 + (j - 1) * 4$ se referă la Tdr - sunt tranzițiile discrete asociate stărilor platformei mobile care deservește procesul de dezasamblare;

În cadrul analizării procesului de asamblare/dezasamblare se introduc ipoteze de modelare:

- Locațiile magaziiilor cu piese coincid cu locațiile operațiilor unde se efectuează asamblarea;
- În urma operației de dezasamblare rezultă o singură componentă;
- Locațiile operațiilor unde se desfășoară dezasamblarea sunt diferite de locațiile magaziiilor;
- Robotul mobil echipat cu manipulator execută câte o singură procedură de dezasamblare a unei componente din cadrul produsului;
- Robotul mobil echipat cu manipulator transportă câte o singură componentă la magazia aferentă;

Pentru a descrie modelul asociat liniei de asamblare/dezasamblare deservită de robotul mobil echipat cu manipulator utilizat în conducere, se apelează la formalismul RPHS [56]-[59], [61]-[64]. În **Fig. 4.16** este dat acest model.

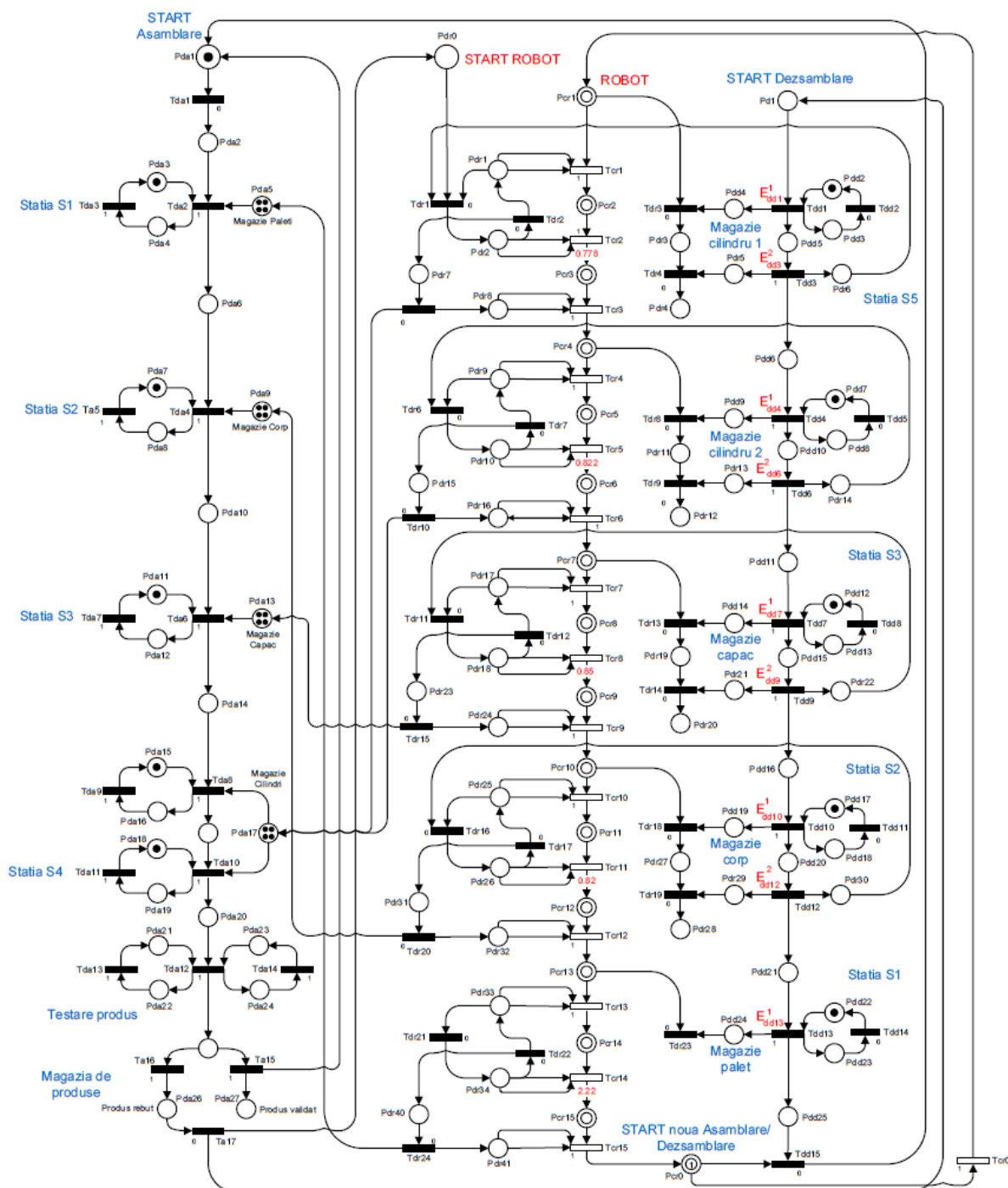


Fig. 4.16 Modelarea procesului de asamblare/dezasamblare totală pe linia flexibilă HERA & Horstmann deservit de un robot mobil cu manipulator cu RPH sincronizată

O RPHS este un triplet

$$RPHS = \langle RPHT, E, Sync \rangle \quad (4.26)$$

unde: RPHT este un 7-uplu

$$\text{RPHT} = \langle P, T, Pre, Post, m_0, h, tempo \rangle \quad (4.27)$$

- E este o mulțime de evenimente externe

$$E = \{E_i^1, E_j^2\}_{\substack{i=\overline{1,13} \\ j=\overline{1,15}}} \cup \{e\} \quad (4.28)$$

- $Sync$ este o funcție care asociază fiecărei tranziții din mulțimea tranzițiilor T evenimente de tipul E^1, E^2 sau e ,

$$Sync: T \rightarrow \{E^1, E^2\} \cup \{e\} \quad (4.29)$$

- e este evenimentul care se întâmplă întotdeauna (este elementul neutru al monoidului E^*).

$$P = \{P_1, P_2, \dots, P_n\} = P^D \cup P^C \quad (4.30)$$

este o mulțime finită de locații cu

$$P^D = \{Pda_i\}_{i=\overline{1,22}} \cup \{Pdd_j\}_{j=\overline{1,26}} \cup \{Pdr_k\}_{k=\overline{1,16}} \quad (4.31)$$

$$P^C = \{Pcr_k\}_{k=\overline{1,15}} \quad (4.32)$$

- $\{Pda_i\}_{i=\overline{1,22}}$ este mulțimea locațiilor discrete aferente procesului de asamblare;
- $\{Pdd_j\}_{j=\overline{1,26}}$ este mulțimea locațiilor discrete aferente procesului de dezasamblare;
- $\{Pdr_k\}_{k=\overline{1,16}}$ este mulțimea locațiilor discrete pentru stările platformei mobile în timpul procesului de dezasamblare;
- Pcr sunt locațiile continue pentru stările asociate deplasării pe traiectorie a platformei mobile corespunzătoare fiecărei operații de dezasamblare;
- $T = \{T_1, T_2, \dots, T_m\} = T^D \cup T^C$ este o mulțime finită de tranziții;

$$T^D = \{Tda_i\}_{i=\overline{1,14}} \cup \{Tdd_j\}_{j=\overline{1,16}} \cup \{Tdr_k\}_{k=\overline{1,21}} \quad (4.33)$$

$$T^C = \{Tcr_k\}_{k=\overline{1,10}} \quad (4.34)$$

- Tda sunt tranzițiile discrete pentru operațiile de asamblare;

- Tdd sunt tranzițiile discrete pentru operațiile de dezasamblare;
- Tdr sunt tranzițiile discrete asociate stărilor platformei mobile care deservește procesul de dezasamblare;
- Tcr sunt tranzițiile continue pentru stările asociate deplasării pe traiectorie a platformei mobile corespunzătoare fiecărei operații de dezasamblare;
- $P \cap T = \emptyset$ mulțimile P și T sunt disjuncte;
- $m_0: P \rightarrow R_+ \text{ or } N$ este marca inițială.
- $h: P \cup T \rightarrow \{D, C\}$ se numește "*hybrid function*", care indică pentru fiecare nod dacă este nod discret (mulțimile P^D și T^D) sau nod continuu (mulțimile P^C și T^C).

Pentru modelul liniei reversibile avem ecuațiile 4.31, 4.32, 4.33, 4.34 unde:

- $h: P^D \cup T^D \rightarrow \{D\}$
- $h: P^C \cup T^C \rightarrow \{C\}$
- $Pre: P \times T \rightarrow Q_+ \text{ sau } N$ este funcția de incidență la intrare;
- $Post: P \times T \rightarrow Q_+ \text{ sau } N$ este funcția de incidență la ieșire;

Observație: În definițiile lui $Pre, Post$ și m_0, N corespunde cazului când $P_i \in P^D$, iar $Q_+ \text{ or } R_+$ corespunde cazului când $P_i \in P^C$.

- $tempo$ este o funcție de la mulțimea tranzițiilor T , la mulțimea numerelor raționale, pozitive și diferite de zero;
- $tempo: T \rightarrow Q_+ \cup \{0\}$

Dacă $T_j \in T^D \Rightarrow d_j = tempo(T_j)$ este temporizarea asociată lui T_j ;

- pentru $T^D = \{Tda_i\}_{i=\overline{1,14}} \Rightarrow d_{da_i} = 1$, unde d_{da_i} este întârzierea asociată operației de asamblare i ;
- pentru $T^D = \{Tdd_j\}_{j=\overline{1,16}} \Rightarrow d_{dd_j} = 1$, unde d_{dd_j} reprezintă întârzierea asociată operației de dezasamblare i și întârzierea datorată transportului piesei desasamblate la magazia de depozitare;
- pentru $T^D = \{Tdr_k\}_{k=\overline{1,21}} \Rightarrow d_{dr_k} = \{0,1\}$, unde d_{dr_k} reprezintă întârzierea datorată prinderii piesei dezasamblate și depozitării ei în magazie.

Dacă $T_j \in T^C \Rightarrow U_j = \frac{1}{tempo(T_j)}$ reprezintă rata de flux asociată lui T_j .

$$T^C = \{Tcr_k\}_{k=\overline{1,10}} \Rightarrow U_{cr_k} = \{\{v_k\}_{k=i+4} \cup \{v_k\}_{k=i}\}_{i=\overline{1,6}} \quad (4.35)$$

unde U_{cr} este fluxul variabil asociat deplasării robotului mobil între locațiile unde se efectuează dezasamblări.

Definitie: Viteza maximă de declanșare a unei tranziții T_j este produsul între propria rata de flux U_j și propriul grad de activare, D : $V_j = U_j \cdot D(T_j, m)$.

Definitie: Gradul de activare D , a unei C-tranziții, T_j asociat unei marci m , notată cu $D(T_j, m)$, este gradul de activare T_j , după ce toate arcele de la C-locății la C-tranziții au fost șterse:

$$D(T_j, m) = \min_{P_i \in {}^0 T_j \cap P^D} \left\lfloor \frac{m_i}{Pre(P_i, T_j)} \right\rfloor \quad (4.36)$$

Particularizând pentru RPHS: arcele $(P_i \times T_j)$ unde,

$$P_i = \{ROBOT\ state\ 1, ROBOT\ state\ 2\} \in {}^0 \{Tcr_k\}_{k=\overline{1,10}} \cap P^D \quad (4.37)$$

au pondere 1. În acest caz $Vcr_k = Ucr_k = 2|_{k=\overline{1,10}}$ ilustrat în **Fig. 4.17**, aceasta reprezintă modelarea structurii generalizate corespunzătoare procesului de dezasamblare a primei piese din cadrul produsului. Cele patru stații de dezasamblare deservite de robotul mobil echipat cu manipulator sunt identice, mai puțin Stația 1.

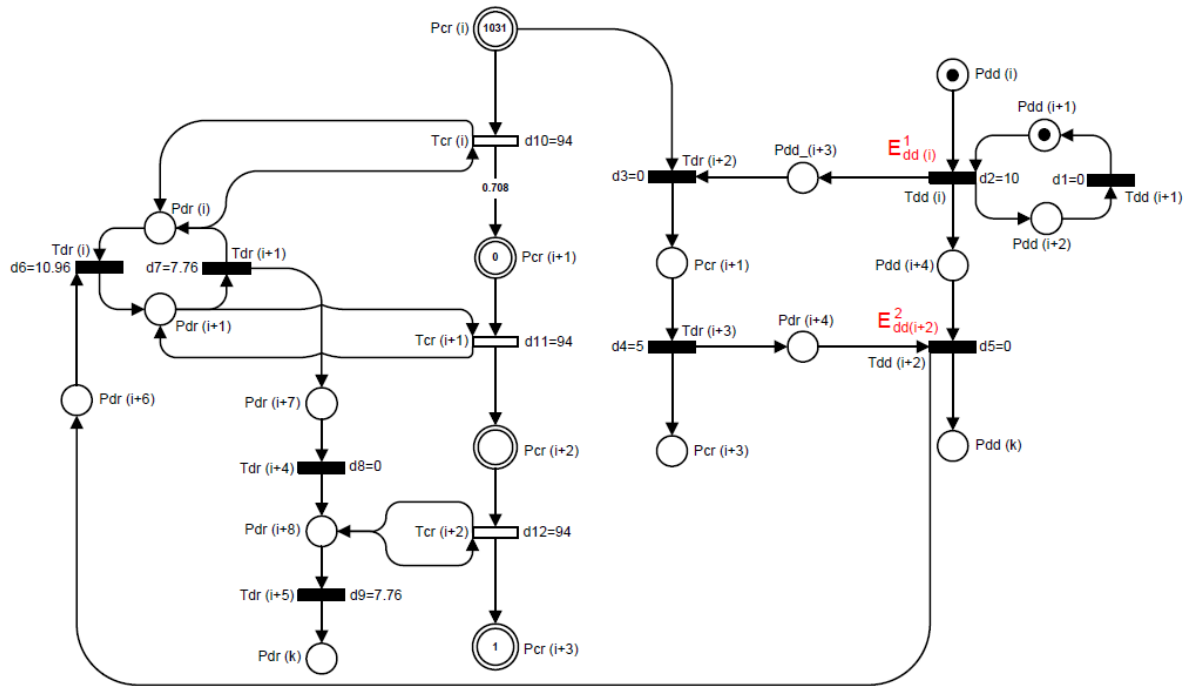


Fig. 4.17 Structura generalizată a modelului sincronizat corespunzător procesului de dezasamblare a primei piese din cadrul produsului

În **Fig. 4.18** se prezintă modelarea structurii generalizate corespunzătoare ultimului proces de dezasamblare aferent ultimei componente din cadrul produsului, proces deservit de un robot mobil echipat cu manipulator.

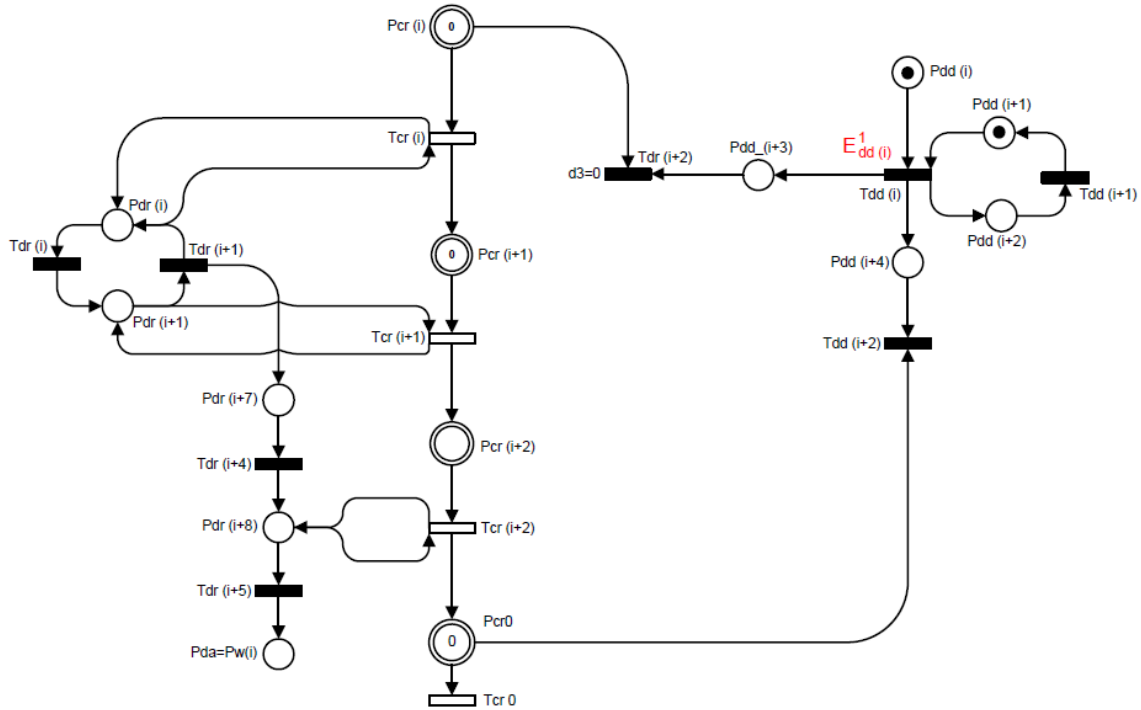


Fig. 4.18 Structura generalizată a modelului sincronizat corespunzător procesului de dezasamblare a ultimei piese din cadrul produsului

- E^1 este un semnal de sincronizare extern asociat procesului de dezasamblare și corepunde la *STOP* linie de mecatronică și *START* operație de dezasamblare;
- E^2 este un semnal de sincronizare extern asociat procesului de dezasamblare și corepunde prinderii piesei dezasamblate și *START* linie de mecatronică;
- e este un eveniment intern care apare în permanență și nu corespunde nici unuia dintre evenimentele externe utilizate pentru sincronizare;

Observatie: Pentru o RPHS o tranziție este activată când fiecare din locațiile ei de intrare conține un număr de jetoane mai mare sau egal cu ponderea arcului/arcelor care unesc tranziția/tranzițiile cu starea/stările din amonte. Dacă tranziția este activată atunci ea se declanșează atunci când apare evenimentul.

Aplicația,

$$Sync: T \rightarrow \{E^1, E^2\} \cup \{e\} \quad (4.38)$$

poate fi descrisă astfel:

$$Sync: \{Tdd_j\}_{j=\overline{1,16}} \rightarrow \{E^1, E^2\} \quad (4.39)$$

unde,

$$Sync: \{Tdd_i\}_{i=\{1,4,7,13\}} \rightarrow \{E_i^1\}_{i=\{1,4,7,13\}} \quad (4.40)$$

$$Sync: \{Tdd_i\}_{i=\{3,6,9,12,15\}} \rightarrow \{E_i^2\}_{i=\{3,6,9,12,15\}} \quad (4.41)$$

$$Sync: \{Tdd_j\}_{j=\overline{1,16}} \cup \{Tdr_k\}_{k=\overline{1,21}} \cup \{Tcr_k\}_{k=\overline{1,10}} \rightarrow e \quad (4.42)$$

Aspectul hibrid al unei RPHS este determinat de variabilele asociate distanțelor între locațiile de dezasamblare și magaziile de depozitare, distanțe pe care robotul mobil trebuie să le parcurgă.

- $\{D_L^i\}_{i=\overline{1,5}}$ este mulțimea distanțelor între punctele de prindere a pieselor dezasamblate și magaziile de depozitare: $\{D_L^1, D_L^2\}$ pentru cilindri iar $\{D_L^3, D_L^4, D_L^5\}$ pentru capac, corp și palet. Variația acestor variabile se poate face cu viteza constantă sau variabilă. Această variație reprezintă viteza platformei între locațiile $\{D_L^i\}_{i=\overline{1,5}}$ ale linei de mecatronică.

4.7 Ipoteze de analiză și modelare a liniei flexibile FESTO MPS-200

În cadrul analizei și modelării liniei flexibile de prelucrare FESTO MPS-200 (Fig. 4.19) se introduc inițial ipoteze de funcționare a procesului de prelucrare.

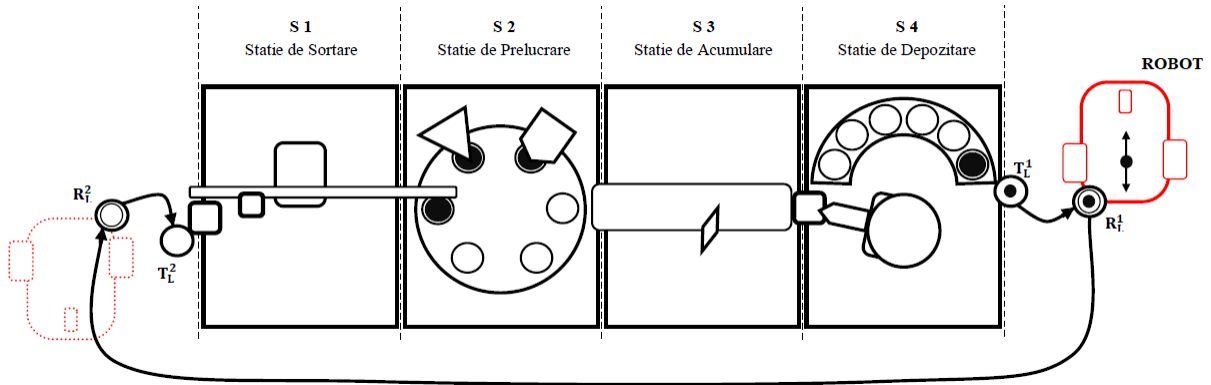


Fig. 4.19 Ciclul complet al robotului mobil echipat cu manipulator deservind linia flexibilă FESTO MPS-200

Analiza procesului de prelucrare al liniei flexibile FESTO MPS-200:

- la pornirea liniei flexibile FESTO MPS-200 se declanșează procesul de prelucrare;

- inițial la prima stație de sortare există o componentă care urmează a fi prelucrată, acesta este trecut printr-un test de validare;
- în urma testului de validare sunt selectate componentele declarate rebut. Cele care au fost validate sunt trimise către stația următoare. Validarea sau invalidarea unei piese se face pe baza culorii pieselor;
- la stația de prelucrare sunt efectuate două tipuri de prelucrări alezare și gărire a componentelor provenite de la stația anterioară, urmând ca piesa prelucrată să fie transferată către stația următoare;
- la stația buffer sunt acumulate piese în vederea depozitării în magazia de piese. Stația de acumulare transferă piesele către stația următoare cu ajutorul unei benzi transportoare.
- în urma acumulării pieselor la stația buffer, acestea sunt eliberate una câte una pentru a fi preluate cu ajutorul unui manipulator axial în vederea stocării în magazie. Înainte de preluarea piesei cu ajutorul unui sistem de tip gripper pneumatic cu care este echipat manipulatorul axial se efectuează un test de validare a culorii. Acumulându-se piese la stația buffer de diferite culori, la operația de depozitare acestea sunt sortate după culoare. Fiecărei culori îi corespunde o anumită locație în magazia de piese, magazia fiind dispusă pe două nivele;
- piesele stocate pe nivelul superior sunt considerate piese rebut și vor fi readuse pe linia flexibilă pentru o nouă operație de prelucrare. Readucerea pieselor declarate rebut se face cu ajutorul robotului mobil echipat cu manipulator;

Conform ipotezelor descrise, în **Fig. 4.20** sunt prezentate secțiunile și traiectoriile pe care le parcurge platforma mobilă echipată cu manipulator în cadrul procesului de deservire a liniei flexibile de prelucrare FESTO MPS-200.

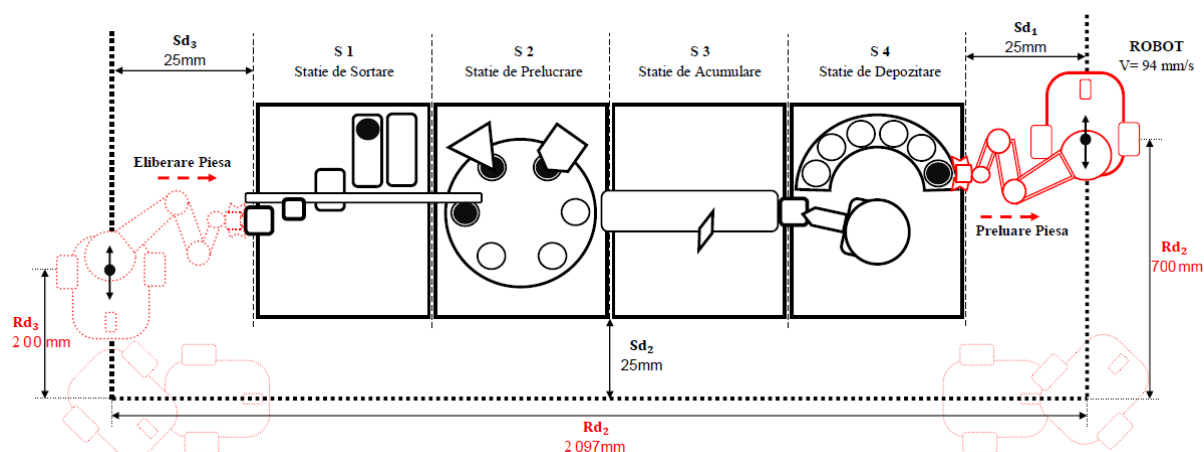


Fig. 4.20 Împărțirea pe secțiuni și traiectoria pe care o parcurge platforma mobilă

Unde:

- R_L^1 : reprezintă numărul și poziția locației în care se găsește robotul mobil echipat cu manipulator în cadrul procesului de preluare a piesei;
- R_L^2 : reprezintă numărul și poziția locației în care se găsește robotul mobil echipat cu manipulator în cadrul procesului de depunere a piesei;
- T_L^1 : reprezintă numărul locației în care se stochează piesa în vederea preluării acesteia de către robotul mobil echipat cu manipulator;
- T_L^2 : reprezintă numărul locației în care se depune piesa de către robotul mobil echipat cu manipulator în vederea unei noi operații de prelucrare;

4.8 Planificarea taskurilor aferente procesului de prelucrare al liniei flexibile FESTO MPS-200

Conform ipotezelor descrise în subcapitolul 4.8 în **Fig. 4.21** se propune planificarea taskurilor aferente liniei flexibile FESTO MPS-200.

Strategia de sortare, prelucrare și depozitare este bazată pe un graf de reprezentare a produsului prelucrat în care relațiile dintre stații sunt exprimate prin săgeți. Utilizând acest graf se elaborează o planificare a taskurilor prin care se determină secvența în care componentele sunt prelucrate.

Dacă o componentă nu este validată la testul de culoare sau este validată, planificarea taskurilor furnizează cea mai bună secvență pentru efectuarea prelucrării și depozitării acesteia în depozit.

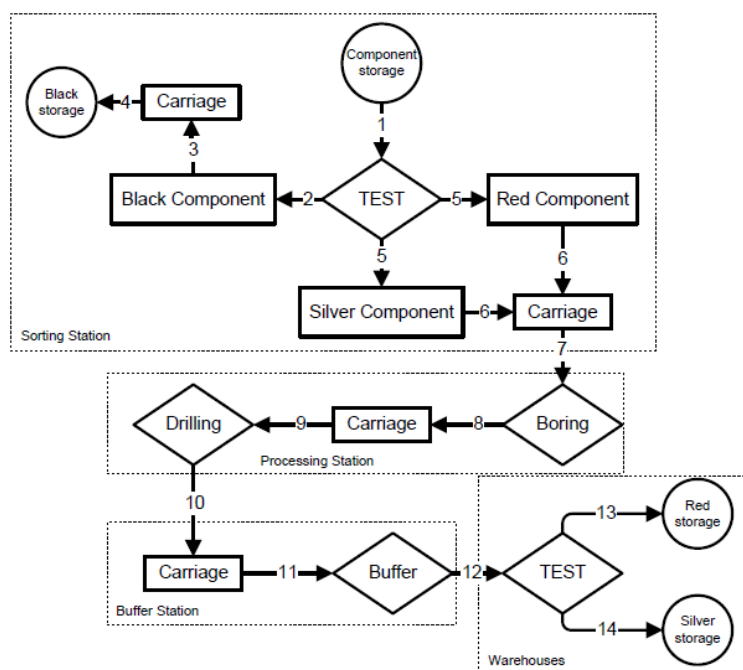


Fig. 4.21 Planificarea taskurilor pentru operațiile de sortare, prelucrare și depozitare

4.9 Modelarea liniei flexibile de prelucrare FESTO MPS-200 cu ajutorul RP netemporizate în abordare SED

Considerând planificarea taskurilor din **Fig. 4.21** și descrierea operațiilor din subcapitolul 4.8 împreună cu ecuațiile din subcapitolul 4.4 în **Fig. 4.22** este prezentată RP netemporizată ce reprezintă modelul liniei flexibile FESTO MPS-200 în cadrul procesului de prelucrare în abordare SED.

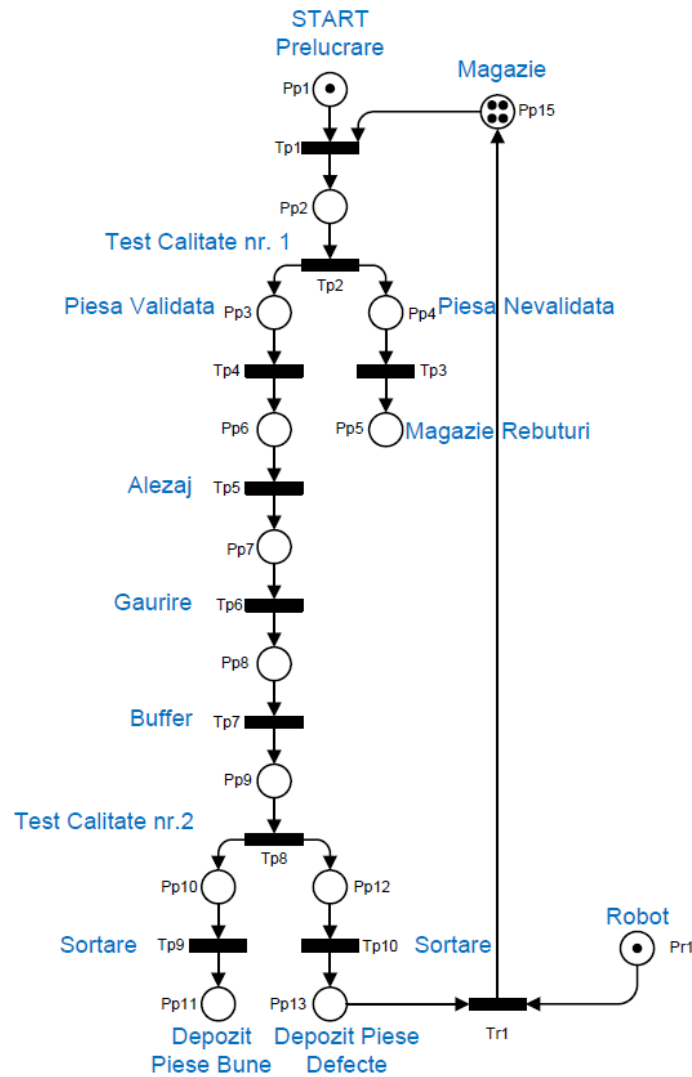


Fig. 4.22 Modelarea procesului de prelucrare pe linia flexibilă FESTO MPS-200 cu ajutorul RP netemporizate în abordare discretă

Unde,

$$P = \{Pp_1, Pp_2, \dots, Pp_n\} = P^P \quad (4.43)$$

$$P^P = \{Pp_i\}_{i=\overline{1,13}} \quad (4.44)$$

Unde,

- $\{Pp_i\}_{i=\overline{1,13}}$ este mulțimea locațiilor aferente procesului de prelucrare;

$$T = \{Tp_1, Tp_2, \dots, Tp_m\} = T^P \quad (4.45)$$

$$T^P = \{Tp_i\}_{i=\overline{1,10}} \quad (4.46)$$

Unde,

- $\{Tp_i\}_{i=\overline{1,10}}$ este mulțimea tranzițiilor aferente procesului de prelucrare;
- $M0: Pp_1$ este o funcție de marcaj inițial;

4.10 Modelarea liniei flexibile de prelucrare FESTO MPS -200 cu ajutorul RP temporizate în abordare SED

Modelarea procesului de prelucrare pe linia flexibilă FESTO MPS-200 deservită de un robot mobil echipat cu manipulator utilizând rețele Petri temporizate este prezentată în **Fig. 4.23**. Considerând planificarea taskurilor din **Fig. 4.21**, descrierea operațiilor din Capitolul 4.7 RP temporizată reprezintă modelul liniei flexibile FESTO MPS-200, model prin care se evidențiază procesul real de prelucrare deservită de un robot mobil cu manipulator în abordare SED.

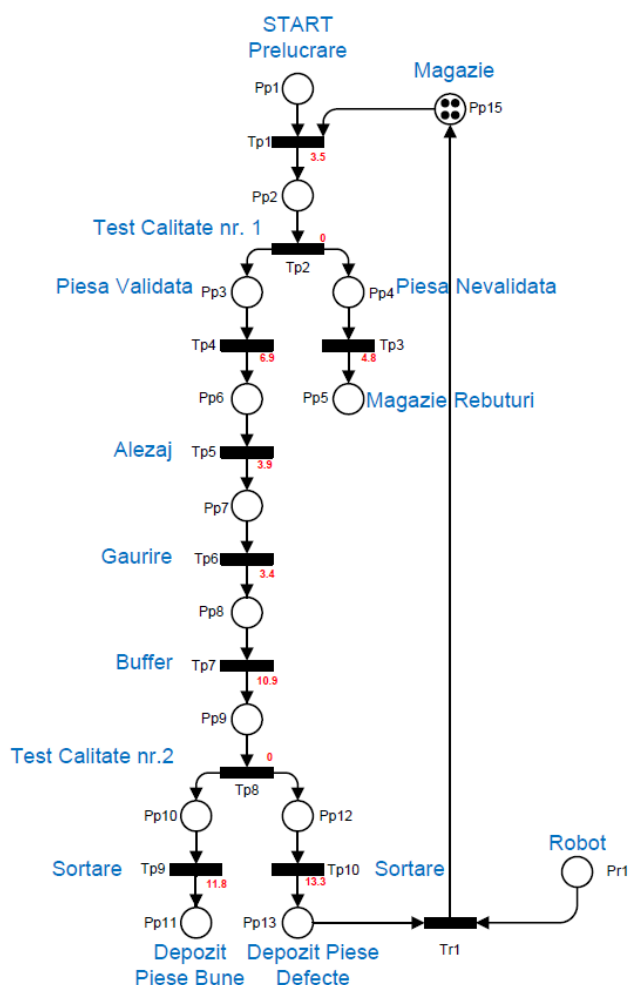


Fig. 4.23 Modelarea procesului de prelucrare pe linia flexibilă FESTO MPS-200 cu ajutorul RP temporizate în abordare discretă

În cadrul procesului de prelucrare, tranzițiile temporizate sunt:

- $(Tp_2, Tp_8) = 0$, acestor tranziții li se atribuie valoarea zero, deoarece fiecărei tranziții îi corespunde o stare din proces care se întâmplă instantaneu;
- $Tp_1 = 3.5$ s; $Tp_3 = 4.8$ s; $Tp_4 = 6.9$ s; $Tp_5 = 3.9$ s; $Tp_6 = 3.4$ s; $Tp_7 = 10.9$ s; $Tp_9 = 11.8$ s; $Tp_{10} = 13.3$ s;
- $Tr_1 = 121.7$ s;

4.10 Concluzii

Principalele contribuții rezultate în urma cercetărilor efectuate în cadrul prezentului capitol au vizat, în mod deosebit, modelarea proceselor de asamblare și dezasamblare deservită de un robot mobil echipat cu un manipulator și a unui proces de prelucrare. Cele mai importante contribuții din cadrul acestui capitol sunt:

1. obținerea unui instrument de simulare a procesului de asamblare, dezasamblare și prelucrare deservită de un robot mobil echipat cu manipulator, considerând un model cât mai complex al sistemului real care conține componentele: stații de asamblare care pot efectua și o dezasamblare parțială, dezasamblarea totală a unui produs fiind realizată prin deservirea de către o platformă mobilă echipată cu manipulator și stații de prelucrare;

2. obținerea unei structuri complete de modelare a unui proces de flexibil de asamblare, dezasamblare și prelucrare deservită de un robot mobil echipat cu manipulator. Menționăm că informațiile disponibile din literatură se referă la modele care nu descriu în totalitate un proces de fabricație care poate realiza diverse funcții.

3. deducerea modelului proceselor flexibile de asamblare, dezasamblare și prelucrare deservită de un robot mobil echipat cu manipulator s-a realizat printr-o procedură în două etape. În prima etapă se construiește un model de planificare al task-urilor, definit printr-un graf de activități. Modelul grafului de activități nu este explicit, dar pune în evidență principalele task-uri care se desfășoară și ordinea acestora. În a doua etapă, se realizează modelarea procesului cu ajutorul grafului de activități din prima etapă.

4. modelarea procesului de asamblare pe linii flexibile utilizând rețele Petri netemporizate în abordare SED, particularizat pe linia flexibilă HERA & Horstmann;

5. modelarea procesului de dezasamblare parțială pe linii flexibile utilizând rețele Petri netemporizate în abordare SED, particularizat pe linia flexibilă HERA & Horstmann;

6. modelarea procesului de asamblare și dezasamblare parțială pe linii flexibile utilizând rețele Petri netemporizate în abordare SED, particularizat pe linia flexibilă HERA & Horstmann;

7. modelarea procesului de asamblare și dezasamblare totală a unui produs pe linii flexibile utilizând rețele Petri temporizate și netemporizate în abordare SED, particularizat pe linia flexibilă HERA & Horstmann;

8. modelarea procesului de asamblare și dezasamblare totală a unui produs pe linii flexibile deservite de un robot mobil echipat cu manipulator, utilizând rețele Petri Hibrice generalizate, particularizat pe linia flexibilă HERA & Horstmann;

9. modelarea procesului de asamblare și dezasamblare totală a unui produs pe linii flexibile deservite de un robot mobil echipat cu manipulator, utilizând rețele Petri Hibrice sincronizate, particularizat pe linia flexibilă HERA & Horstmann.

10. modelarea procesului de prelucrare pe linii flexibile utilizând rețele Petri netemporizate în abordare SED, particularizat pe linia flexibilă FESTO MPS-200;

11. modelarea procesului de prelucrare pe linii flexibile utilizând rețele Petri temporizate în abordare SED, particularizat pe linia flexibilă FESTO MPS-200;

Capitolul 5

Contribuții privind simularea procesului de asamblare/dezasamblare deservit de un robot mobil echipat cu manipulator

-
- 5.1 Simularea procesului de asamblare/dezasamblare a unui produs deservit de un RM cu manipulator*
 - 5.2 Simularea structurii generalizate a modelului RPH sincronizate*
 - 5.3 Modelul cinematic al platformei mobile*
 - 5.4 Conducerea Sliding Mode a platformei mobile*
 - 5.5 Simularea conducerii în Sliding Mode*
 - 5.6 Concluzii*
-

Modele rețelelor Petri netemporizate și temporizate permit studierea proprietăților calitative ale funcționării sistemelor dinamice cu evenimente discrete. O altă clasă importantă de astfel de modele o constituie RPH sincronizată.

Simularea și analiza permit evaluarea a unor indici (criterii) de performanță ce caracterizează funcționarea procesului fizic modelat prin rețea Petri netemporizată, temporizată, și RPH sincronizată, exprimând eficiența atât în derularea preceselor de asamblare și dezasamblare deservită de un robot mobil echipat cu manipulator, cât și în utilizarea resurselor întregului sistem flexibil. În faza de simulare, evaluarea criteriilor de performanță permite introducerea unor strategii de planificare în utilizarea resurselor liniei flexibile de fabricație cât și a RM cu manipulator, care adaugă reguli suplimentare pentru operarea întregului proces de dezasamblare a unui produs sau le modifică pe cele rezultate numai din considerente logice, calitative [65]-[69].

5.1 Simularea procesului de asamblare/dezasamblare a unui produs deservit de un RM cu manipulator

În cadrul simulării s-a utilizat programul Visual Object Net ++. Valorile mărcilor și ale tranzițiilor corespund procesului real, acestea au fost utilizate în simulare pentru a evidenția evoluția corectă a etapelor de asamblare, dezasamblare și deservirea robotului mobil echipat cu manipulator în cadrul procesului de dezasamblare completă a unui produs. S-au utilizat în simulare modelele de rețele Petri descrise în Capitolul 4.

Proprietățile comportamentale ale rețelei Petri temporizate sunt dependente atât de topologia cât și de introducerea tranzițiilor temporizate al rețelei. Operația de asamblare fiind un proces continuu care se desfășoară până când în mod accidental apare o componentă sau un produs declarat defect la testul de calitate, inițializându-se astfel procesul de dezasamblare, proces care este deservit de un robot mobil echipat cu manipulator. În **Fig. 5.1** se poate observa rezultatul procesului de asamblare al primei componente, proces care durează 9.5 s, conform valorii temporizării tranziției Ta_2 și a tranziției $Ta_3 = 0$.

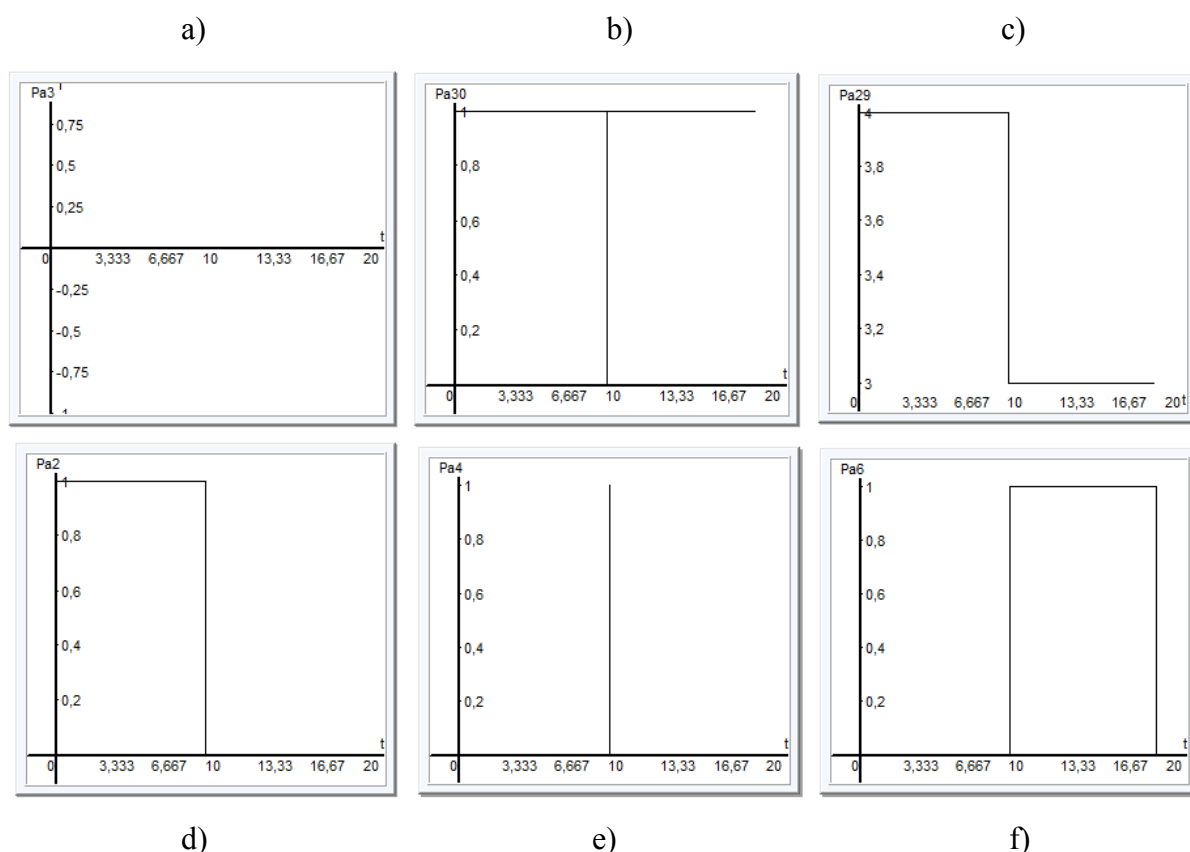


Fig. 5.1 Evoluția stărilor corespunzătoare locațiilor aferente procesului de asamblare al primei piese de la stația S1: a) starea marcajului inițial M_0 care corespunde mărcii P_a3 ; b) starea procesului de asamblare al primei piese care corespunde mărcii P_a30 ; c) starea starea magaziei de component aferentă mărcii P_a29 ; d) starea prin care se declanșează banda transportoare aferentă mărcii P_a2 ; e) starea prin care procesul se reinițializează după efectuarea primei asamblări aferentă mărcii P_a4 ; starea prin care se transport component asamblată către stația următoare, aferentă mărcii P_a6 ;

Rezultatul procesului de testare și depune produs declarat defect la testul de calitate în magazia aferentă de produse defecte se poate observa în **Fig. 5.2**. Procesul de testare durează 27.2 s, conform valorii temporizării tranziției Ta_{12} iar procesul de depune în magazine a produsului durează 19.3 s conform temporizării aferente tranziției $Ta_{13} = Ta_{14}$.

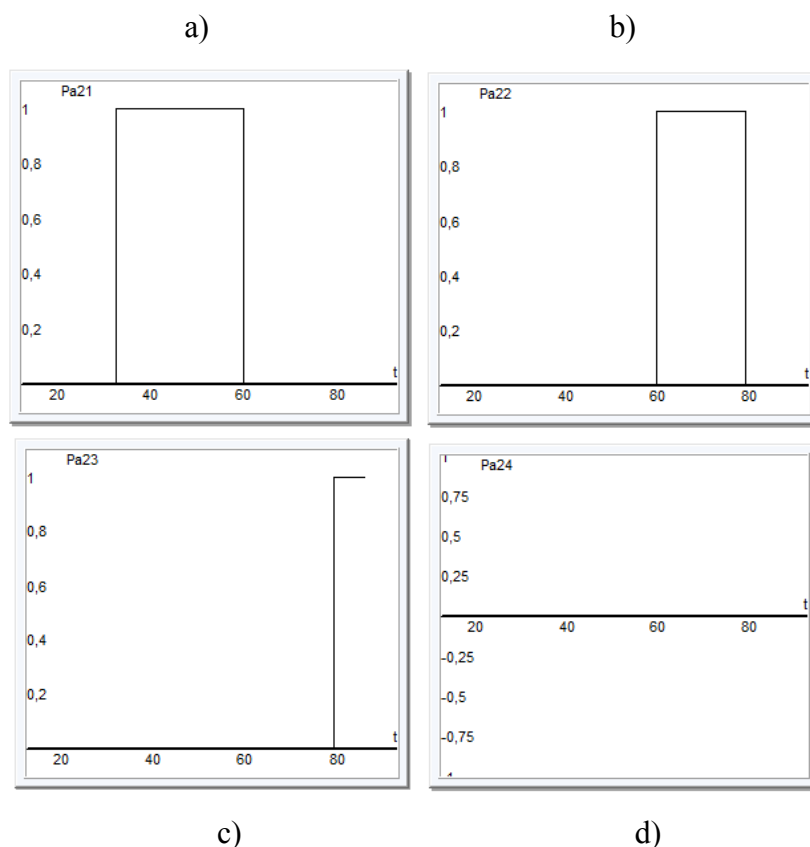


Fig. 5.2 Evoluția stărilor corespunzătoare locațiilor aferente procesului de testare și depozitare a unui produs declarat rebut: a) starea prin care se transportă produsul asamblat către stația de testare aferentă mărcii $P_a 21$; b) starea procesului de testare al produsului final care corespunde mărcii $P_a 22$; c) starea magaziei de componente aferentă produselor declarate defecte, aferente mărcii $P_a 23$; d) starea magaziei de componente aferentă produselor validate, aferente mărcii $P_a 24$;

Valoarea timpului total al ciclului de asamblare a unui produs este $Ta_{ciclu} = 78.8$ s, această valoare corespunde atât depozitării produsului declarat defect cât și a produsului validat la testul de calitate.

În **Fig. 5.3** se poate observa rezultatul procesului de dezasamblare al primei componente din cadrul produsului, proces care durează 6.1 s, conform valorii temporizării tranziției Td_1 și la care se adaugă timpul de la inițializarea operației de dezasamblare și timpul de transport al produsului rebut din magazine în dreptul primei stații de dezasamblare, aferent tranziției Ta_{15} care are o durată de 32.8 s.

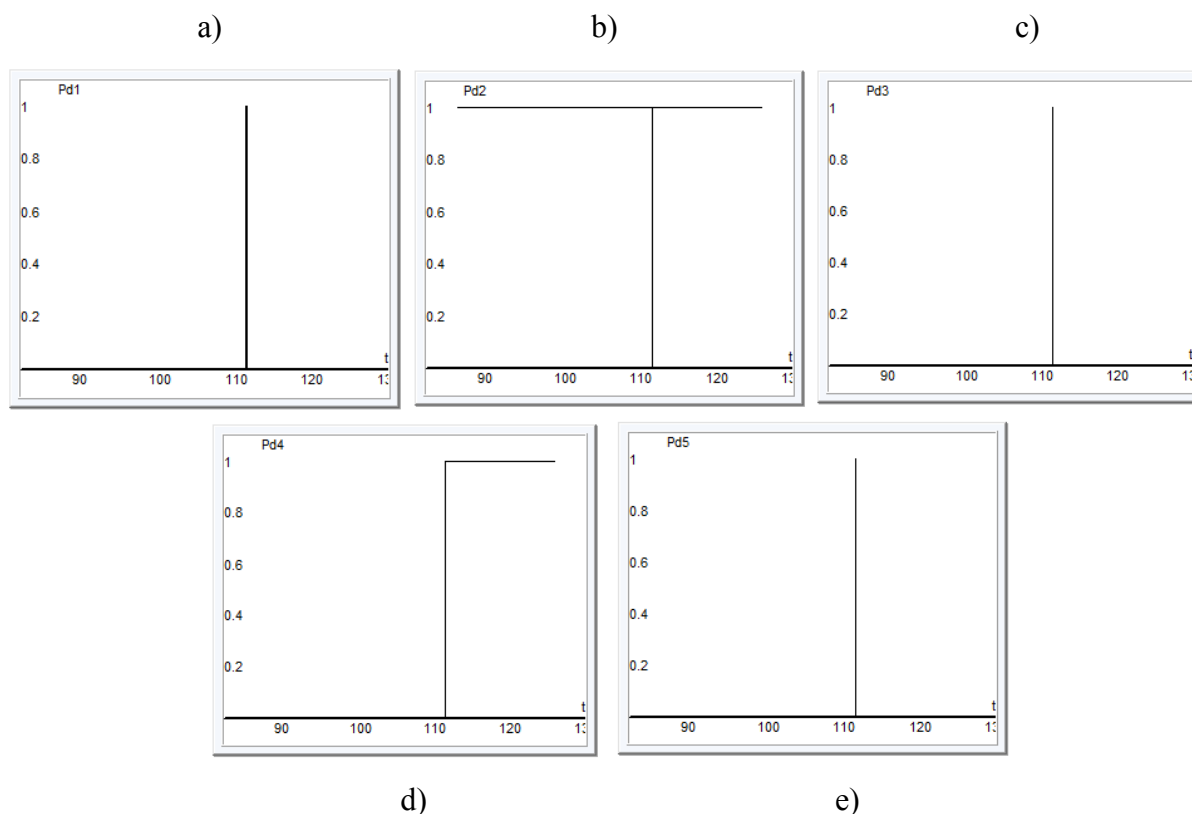


Fig. 5.3 Evoluția stărilor corespunzătoare locațiilor aferente procesului de dezasamblare a unei componente din cadrul unui produs declarat rebut: a) starea marcajului de inițializare a procesului de dezasamblare care corespunde mărcii $P_d 1$; b) starea procesului de dezasamblare al primei piese care corespunde mărcii $P_d 2$; c) starea prin care procesul se reinițializează după efectuarea primei dezasamblări aferentă mărcii $P_d 3$; d) starea starea magaziei de component dezasamblate aferentă mărcii $P_d 4$; e) starea prin care se transport produsul către stația următoare de dezasamblare, aferentă mărcii $P_a 6$;

Valoarea timpului total al ciclului de dezasamblare a unui produs este $Td_{ciclu} = 53.2$ s, în cadrul acestui timp total al ciclului de dezasamblare se adaugă timpul ciclului în care robotul deservește linia flexibilă în cadrul procesului de dezasamblare Tr_{ciclu} . Valoarea lui Tr_{ciclu} este rezultată din simularea RPH, prezentată în subcapitolul următor [70].

5.2 Simularea structurii generalizate a modelului RPH sincronizate

În cadrul simulării structurii generalizate a modelului RPH sincronizată s-au utilizat două programe Visual Object Net ++ și Sirphyco. Utilizarea a două medii de simulare a fost necesară din motive de limitare a ambelor programe, atât a numărului de elemente folosite cât și a posibilităților restrânse de elaborare a RPH. Valorile mărcilor și ale tranzițiilor corespund procesului real, acestea au fost utilizate în simulare pentru a evidenția evoluția corectă a etapelor de asamblare, dezasamblare și deservirea robotului mobil echipat cu manipulator în

cadrul procesului de dezasamblare completă a unui produs. S-au utilizat în simulare modelele de rețele Petri descrise în Capitolul 4. Pentru a evidenția evoluția unei singure etape de dezasamblare deservită de un robot mobil echipat cu manipulator, în figura de mai jos este prezentată evoluția marcajelor RPH generalizate din **Fig. 5.4**.

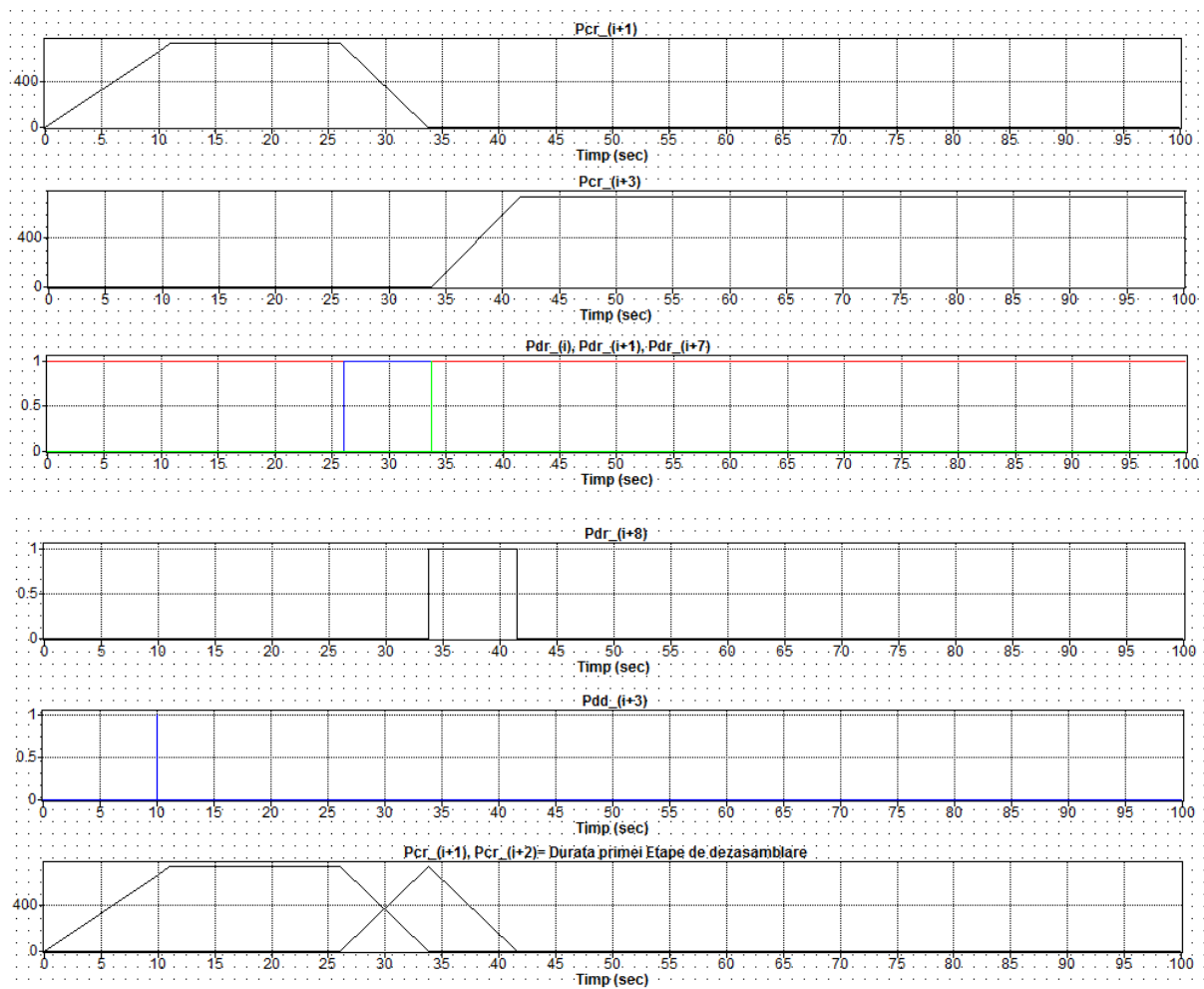


Fig. 5.4 Evoluția locațiilor continue și locațiilor discrete asociate robotului mobil echipat cu manipulator în cadrul procesului de dezasamblarea a primei componente

Pentru a evidenția conducerea în timp real și evoluția RPH s-a utilizat programul LabVIEW în care s-a implementat RPH printr-o interfață grafică. Astfel se poate pune în evidență evoluția procesului de dezasamblare, deplasarea robotului mobil echipat cu manipulator și deservirea acestuia în cadrul procesului de dezasamblare completă a unui produs la apariția unui rebut. În figura de mai jos se poate observa interfața grafică.

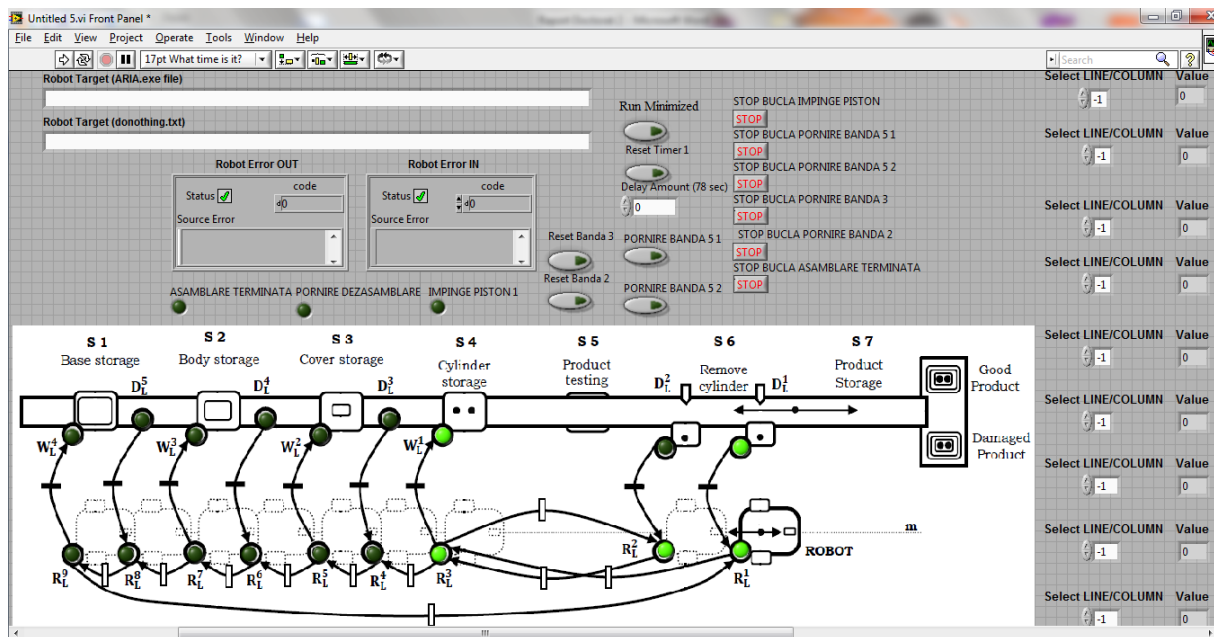


Fig. 5.5 Interfață grafică LabVIEW a RPH sincronizate

5.3 Modelul cinematic al platformei mobile

Platforma mobilă descrisă în subcapitolul 3.9, prezintă două roți motoare cu tracțiune diferențială și o roată directoare după cum se poate observa în Fig. 5.6.

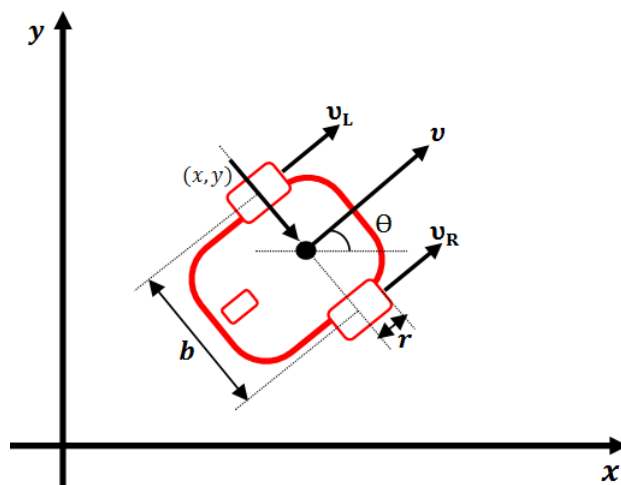


Fig. 5.6 Variabilele cinematice ale platformei mobile cu două roți motoare și o roată directoare

Problema modelării cinematice a platformelor mobile a fost tratată explicit în literatură [83], [84].

$$\begin{cases} \dot{x}_r(t) = v_r(t) \cdot \cos \theta_r(t) \\ \dot{y}_r(t) = v_r(t) \cdot \sin \theta_r(t) \\ \dot{\theta}_r(t) = \omega_r(t) \end{cases} \quad (5.1)$$

unde, x_r și y_r reprezintă coordonatele Carteziene ale centrului geometric al platformei mobile, v_r este viteza liniară a platformei mobile, Θ_r reprezintă unghiul direcției, ω_r reprezintă viteza unghiulară a robotului și b reprezintă distanța între planele roților motoare.

5.4 Conducerea Sliding Mode a platformei mobile

Erorile de urmărire a traiectoriei pot fi caracterizate prin (x_e, y_e, Θ_e) . Scopul acestei secțiuni este de a proiecta un controler stabil care generează un vector comandă (v_c, ω_c) . Erorile de urmărire a traiectoriei pot fi caracterizate prin (x_e, y_e, Θ_e) . Vectorul eroare pentru urmărirea traiectoriei este,

$$\begin{bmatrix} x_e \\ y_e \\ \Theta_e \end{bmatrix} = \begin{bmatrix} \cos \Theta_d & \sin \Theta_d & 0 \\ -\sin \Theta_d & \cos \Theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x_d \\ y_r - y_d \\ \Theta_r - \Theta_d \end{bmatrix} \quad (5.2)$$

Unde, (x_d, y_d, Θ_d) reprezintă poziția virtuală a platformei mobile.

Derivata erorilor de urmărire se poate scrie,

$$\begin{cases} \dot{x}_e = -v_d + v_r \cdot \cos \Theta_e + y_e \cdot \omega_d \\ \dot{y}_e = v_r \cdot \sin \Theta_e - x_e \cdot \omega_d \\ \dot{\Theta}_e = \omega_r - \omega_d \end{cases} \quad (5.3)$$

unde, v_d și ω_d reprezintă vitezele liniare și unghiulare dorite.

În cadrul acestei lucrări am presupus că $|\Theta_e| < \frac{\pi}{2}$, care presupune ca orientarea platformei mobile nu trebuie să fie perpendiculară pe direcția dorită. S-a introdus o nouă suprafață de alunecare astfel încât eroarea laterală, y_e și variabila unghiulară, Θ_e sunt cuplate intern în aceeași suprafață și ambele variabile converg spre zero. În acest scop sau definit următoarele suprafețe:

$$s_1 = \dot{x}_e + k_1 \cdot x_e \quad (5.4)$$

$$s_2 = \dot{y}_e + k_2 \cdot y_e + k_0 \cdot \text{sgn}(y_e) \cdot \Theta_e \quad (5.5)$$

unde, k_0, k_1, k_2 reprezintă parametri constanți pozitivi, x_e, y_e și Θ_e reprezintă erorile de urmărire definite în ecuația 5.2.

Dacă suprafața s_1 converge la zero, x_e converge la zero. Dacă s_2 converge la zero, atunci y_2 devine $\dot{y}_e = -k_2 \cdot y_e - k_0 \cdot \text{sgn}(y_e) \cdot \Theta_e$. Dacă $y_e < 0 \Rightarrow \dot{y}_e > 0$ atunci $k_0 < k_2 \cdot$

$|y_e|/|\Theta_e|$. Dacă $y_e > 0 \Rightarrow \dot{y}_e < 0$ atunci $k_0 < k_2 \cdot |y_e|/|\Theta_e|$. În final se poate cunoaște din s_2 că convergența lui y_e și \dot{y}_e conduce la convergența lui Θ_e la zero.

Forma practică generală a legii de conducere este,

$$\dot{s} = -Q \cdot s - P \cdot \text{sgn}(s) \quad (5.6)$$

unde Q și P sunt constante pozitive. Adăugând termenul $-Q \cdot s$, starea este forțată să se apropie rapid de zona de comutație când s este mare.

Din derivarea în domeniul timp a ecuațiilor 5.4 și 5.5 se poate obține și știind că,

$$\dot{\Theta}_e = \dot{\Theta}_r - \dot{\Theta}_d = \omega_r - \omega_d \quad (5.7)$$

Se obține,

$$\dot{v}_c = \frac{-Q_1 \cdot s_1 - P_1 \cdot \text{sgn}(s_1) - k_1 \cdot \dot{x}_e - \dot{\omega}_d \cdot y_e - \omega_d \cdot \dot{y}_e + v_r \cdot \dot{\Theta}_e \cdot \sin \Theta_e + \dot{v}_d}{\cos \Theta_e} \quad (5.8)$$

$$\omega_c = \frac{-Q_2 \cdot s_2 - P_2 \cdot \text{sgn}(s_2) - k_2 \cdot \dot{y}_e - \dot{v}_r \cdot \sin \Theta_e + \dot{\omega}_d \cdot x_e + \omega_d \cdot \dot{x}_e}{v_r \cdot \cos \Theta_e + k_0 \cdot \text{sgn}(y_e)} + \omega_d \quad (5.9)$$

Se definește funcția Lyapunov $V = \frac{1}{2} \cdot s^T \cdot s$, astfel derivata în domeniul timp este,

$$\begin{aligned} \dot{V} &= s_1 \cdot \dot{s}_1 + s_2 \cdot \dot{s}_2 = s_1 \cdot (-Q_1 \cdot s_1 - P_1 \cdot \text{sgn}(s_1)) + s_2 \cdot (-Q_2 \cdot s_2 - P_2 \cdot \text{sgn}(s_2)) = \\ &= -s^T \cdot Q \cdot s - P_1 \cdot |s_1| - P_2 \cdot |s_2| \end{aligned} \quad (5.10)$$

Pentru ca \dot{V} să fie negativă și semidefinită este suficient să alegem Q_i și P_i astfel încât $Q_i, P_i \geq 0$.

5.5 Simularea conducerii în Sliding Mode

Considerând robotul mobil Pioneer P3-DX un robot mobil autonom, modelat ca un solid rigid sau articulat, cu cinematică și geometrie cunoscută, situat într-un punct $P_n = [x, y, \Theta]^T$ unde $n = \{n_d, n_p\}$, care poate corespunde unei locații $P_{n_d} = \{1, \dots, 9\}$ de deservire a liniei flexibile HERA & Horstmann în procesul de dezasamblare a unei componente sau a unei locații $P_{n_p} = \{1, 2\}$ de deservire a liniei FESTO MPS-200 de prelucrare. S-a simulat în programul MobileSim parcurgerea traiectoriilor ambelor procese de către platforma mobilă.

5.5.1 Simularea conducerii platformei mobile în cadrul deservirii liniei flexibile HERA & Horstmann

În cadrul simulării s-au impus anumite condiții de parcurgere a traiectoriei de către platforma mobilă Pioneer 3-DX în conducere sliding mode [71]-[77]. Traiectoria simulată corespunde procesului real de deservire a liniei flexibile HERA & Horstmann în cadrul procesului de dezasamblare deservit de o platformă mobilă echipată cu manipulator.

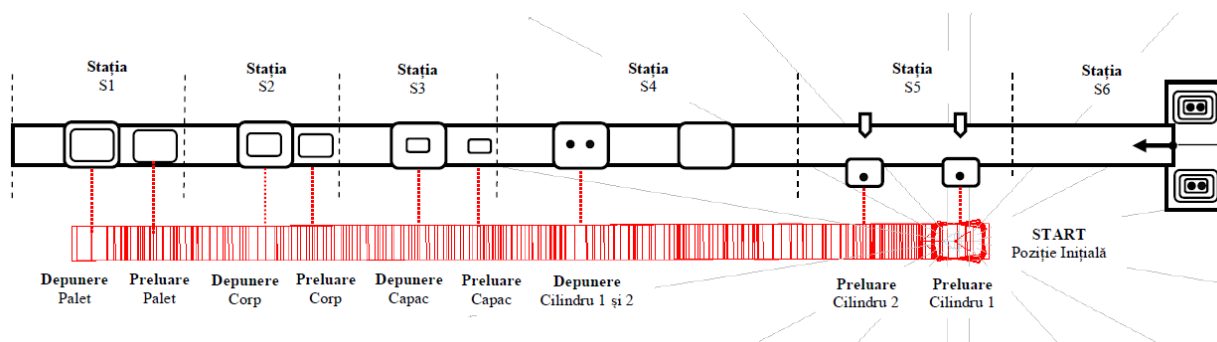


Fig. 5.7 Simularea parcurgerii traiectoriei în conducere Sliding Mode din cadrul procesului de deservire al liniei flexibile HERA & Horstmann

Astfel s-au impus următoarele condiții de simulare:

- Distanța totală parcursă de platforma mobilă este de 7960 mm;
- Viteza platformei mobile a fost de 94 mm/s;
- În cadrul simulării nu s-au luat în considerare timpi de staționare a platformei mobile pentru efectuarea operațiilor de preluare sau depunere de către manipulator a pieselor;
- Traiectoria parcursă a fost în linie dreaptă, deoarece în procesul real prin construcția liniei flexibile nu au existat locuri în care platforma mobilă era nevoită să efectueze o curbă sau o întoarcere;

Rezultatele simulării se pot observa în **Fig. 5.8**, unde în a) se poate observa traiectoria reală a platformei mobile cât și traiectoria impusă. Ambele traiectorii coincid pe toată distanța parcursă. În b) se observă variații ale vitezei liniare reale față de viteza liniară impusă. Din simulare a rezultat că timpul de parcurgere a distanței de 7960 mm a fost de 84.6 s.

În **Fig. 5.9** a) se prezintă eroarea de urmărire pe axa x, valoarea maximă a erorii se încadrează într-o deviație a platformei mobile față de distanța de parcurs cu aproximativ 4 mm. În b) eroarea de urmărire pe axa y este zero, deoarece platforma mobilă nu execută nici o curbă sau o întoarcere.

În Fig. 5.10 a) se prezintă eroarea direcției platformei mobile, se observă că valoarea erorii este zero la fel și valoarea vitezei unghiulare din b), datorită faptului că platforma mobilă parcurge o traiectorie dreaptă fără a face curbe sau întoarceri.

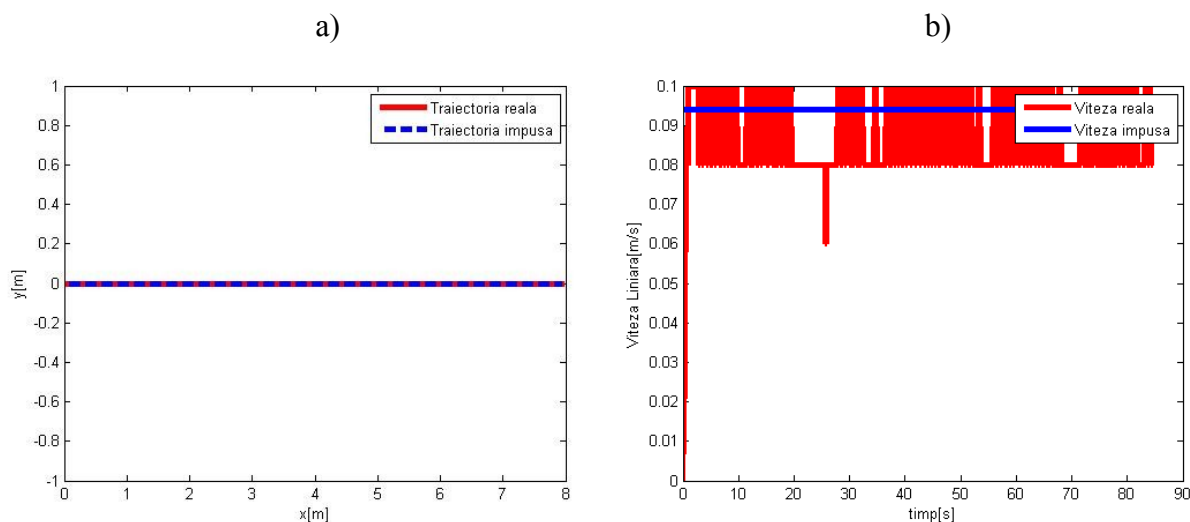


Fig. 5.8 Simularea parcurgerii traiectoriei în conducere sliding mode: a) traiectoria parcursă; b) viteza platformei mobile;

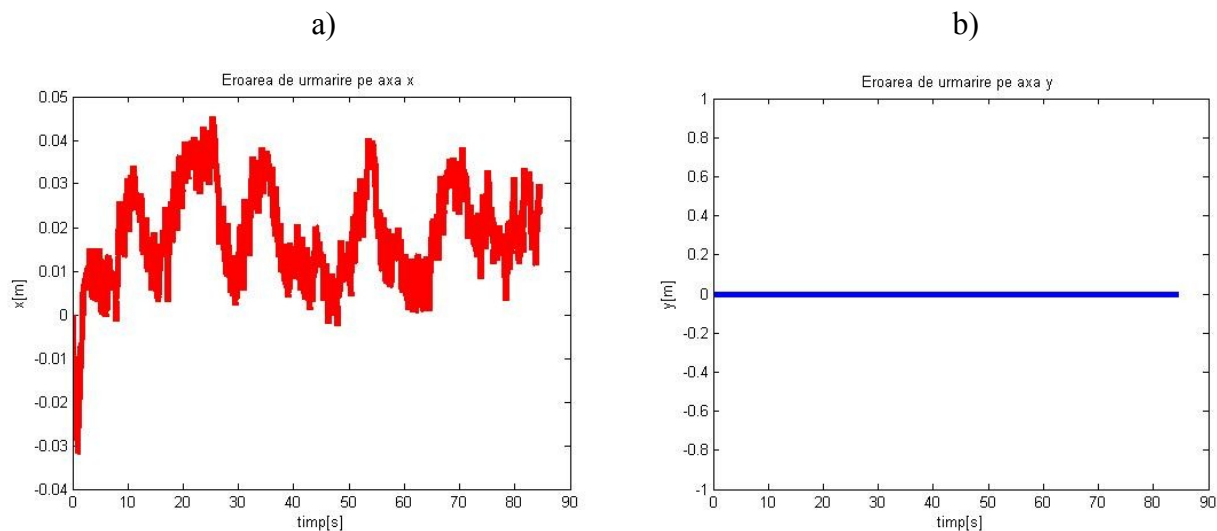


Fig. 5.9 Eroare de urmărire a conducerii în sliding mode: a) eroarea pe axa x; b) eroare pe axa y;

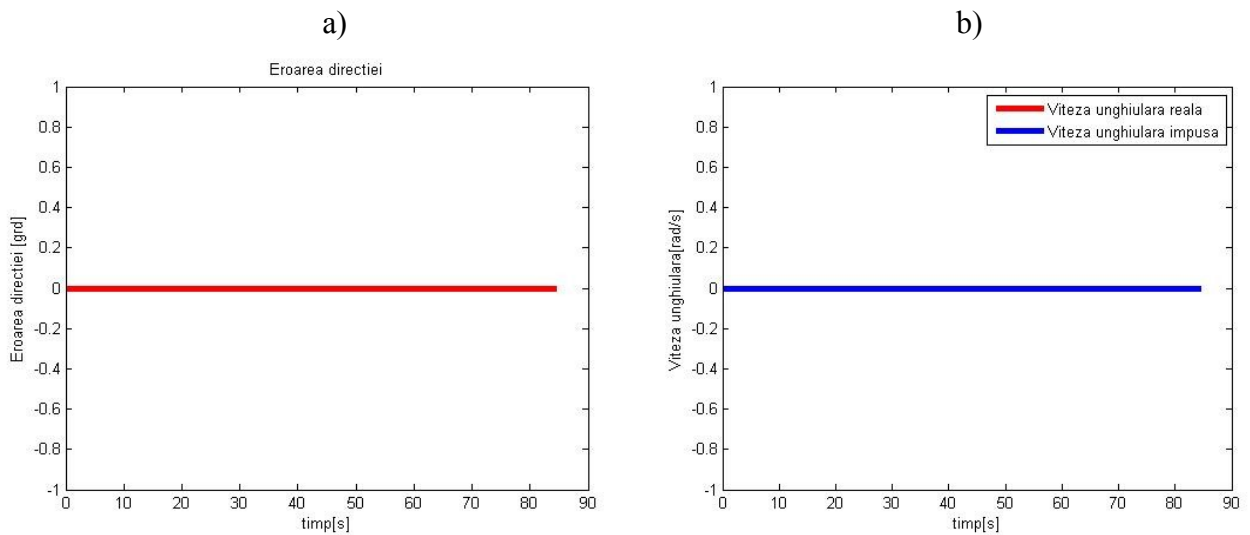


Fig. 5.10 a) Eroarea direcției; b) Viteza unghiulară;

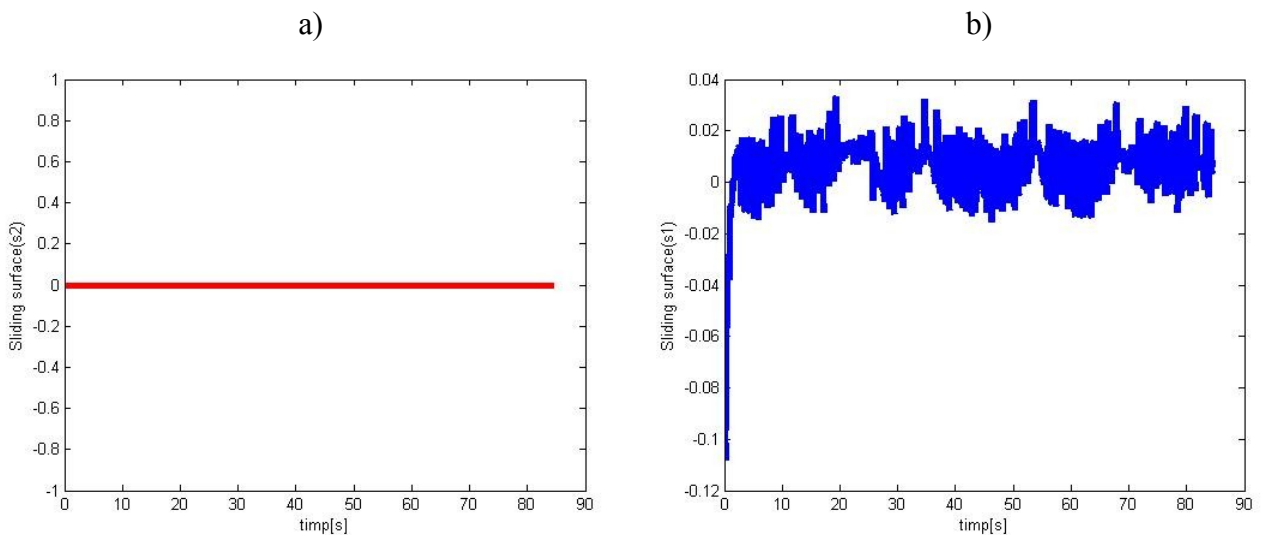


Fig. 5.11 Comanda conducerii în sliding mode: a) comanda pe suprafața s_2 ;
a) comanda pe suprafața s_1 ;

Se poate observa în **Fig. 5.11** a) și b) comanda conducerii în sliding mode, în cadrul suprafeței s_1 se pot observa variații ale comenzii pentru menținerea traiectoriei în linie dreaptă, pe când în cadrul suprafeței s_2 comanda are valoare zero.

5.5.2 Simularea conducerii platformei mobile în cadrul deservirii liniei flexibile FESTO MPS-200

Utilizând ecuațiile din Capitolul 5.3 și 5.4 în cadrul simulării conducerii și parcurgerii traiectoriei de către platforma mobilă Pioneer P3-DX în conducere sliding mode din cadrul procesului de deservire a liniei flexibile FESTO MPS-200 traiectoria simulată corespunde procesului real și este prezentată în **Fig. 5.12**.

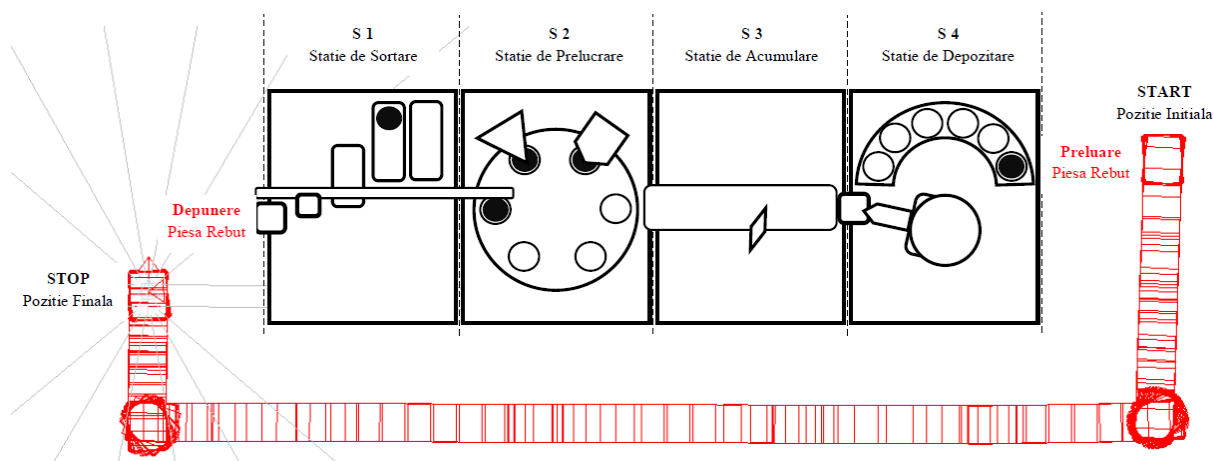


Fig. 5.12 Simularea parcurgerii traiectoriei în conducere Sliding Mode din cadrul procesului de prelucrare al liniei flexibile FESTO MPS-200

În cadrul simulării s-au impus următoarele condiții:

- Distanța totală parcursă de platforma mobilă este de 5634 mm; Această traiectorie este parcursă în două etape de preluare - depunere și întoarcere la poziția inițială, fiecare etapă poate fi împărțită în 3 secțiuni (fiecare etapă are o distanță de 2817 mm): în prima secțiune se parcurge distanța de 700 mm în linie dreaptă (din poziția de *START*) după care platforma mobilă execută o întoarcere de 90° apoi parcurge în linie dreaptă distanța de 2097 mm care corespunde celei de-a doua secțiuni, execută cea de-a doua întoarcere de 90° și parcurge cea de-a treia secțiune (până la poziția de *STOP*) cu o distanță de 200 mm; După parcurgerea celor 3 secțiuni în care se transportă o componentă, platforma mobilă revine în poziția inițială parcurgând aceeași traiectorie;
- Viteza platformei mobile a fost de 94 mm/s;
- În cadrul simulării nu s-au luat în considerare timpi de staționare a platformei mobile pentru efectuarea operațiilor de preluare sau depunere de către manipulator a pieselor;
- În cadrul simulării platforma mobilă echipată cu manipulator preia o singură piesă declarată rebut și o readuce pe linia de prelucrare după care platforma mobilă se reîntoarce în poziția inițială;

Rezultatele simulării se pot observa în **Fig. 5.13**, unde în a) se poate observa traiectoria reală a platformei mobile cât și traiectoria impusă. Ambele traiectorii coincid pe toată distanța parcursă. Se poate observa la momentele în care platforma mobilă execută o întoarcere de 90° o mică deviația a traiectoriei. În b) se observă variații ale vitezei liniare reale față de viteza liniară impusă. Din simulare a rezultat că timpul de parcurgere a distanței de 5634 mm a fost de 121.7 s.

În **Fig. 5.14** a) se prezintă eroarea de urmărire pe axa x, valoarea maximă a erorii se încadrează într-o deviație a platformei mobile față de distanța de parcurs cu aproximativ 4 mm. În b) eroarea de urmărire pe axa y este de aproximativ 2 mm, se poate observa că la momentul de revenire în poziția inițială în care platforma mobilă execută ultima întoarcere de 90° , eroarea de urmărire atinge valoarea maximă de 2.37 mm.

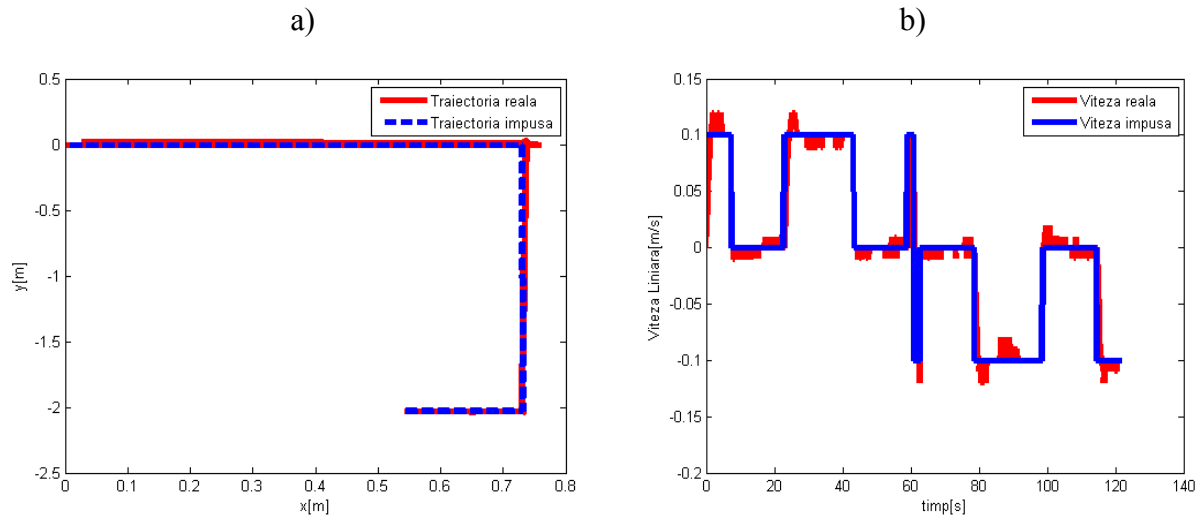


Fig. 5.13 Simularea parcurgerii traiectoriei în conducere sliding mode: a) traiectoria parcursă; b) viteza platformei mobile;

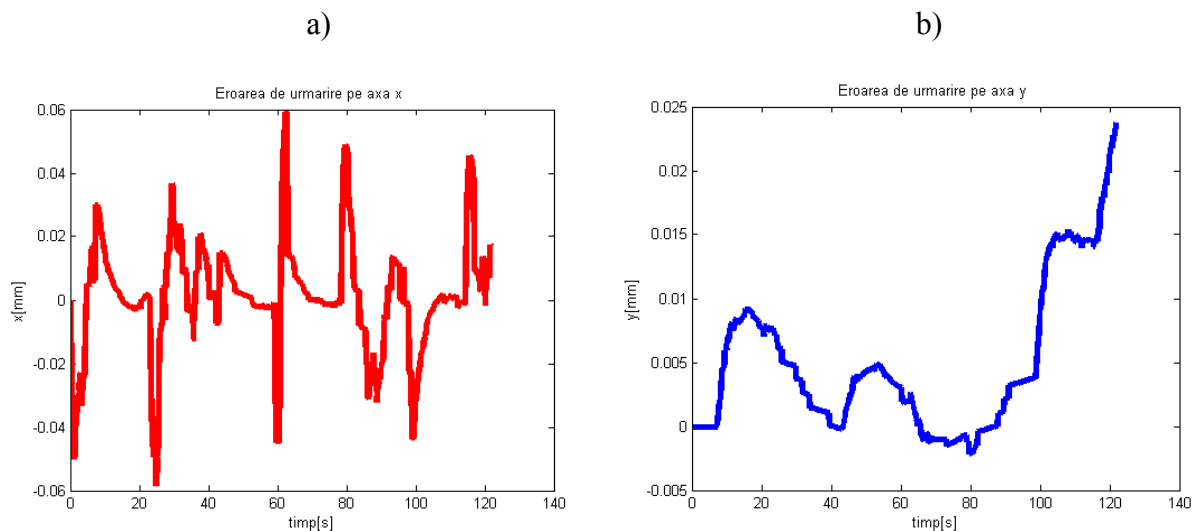


Fig. 5.14 Eroare de urmărire a conducerii în sliding mode: a) eroarea pe axa x; b) eroare pe axa y;

În **Fig. 5.15** a) se prezintă eroarea direcției platformei mobile, se observă că valoarea erorii este de aproximativ 2° și valoarea vitezei unghiulare reale din **Fig. 5.15** b) oscilează în jurul valorii impuse cu aproximativ 0.5 rad/s.

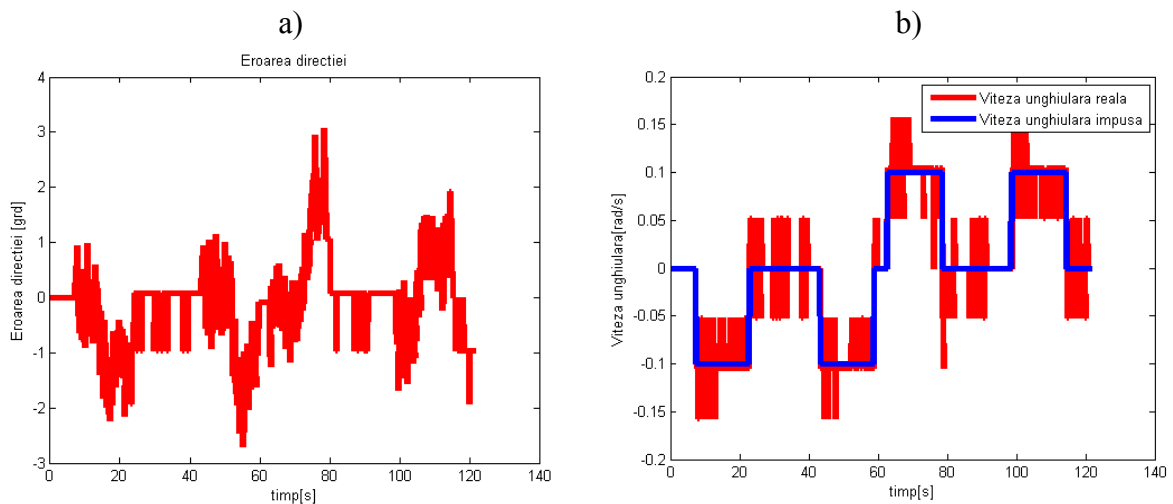


Fig. 5.15 a) Eroarea direcției; b) Viteza unghiulară;

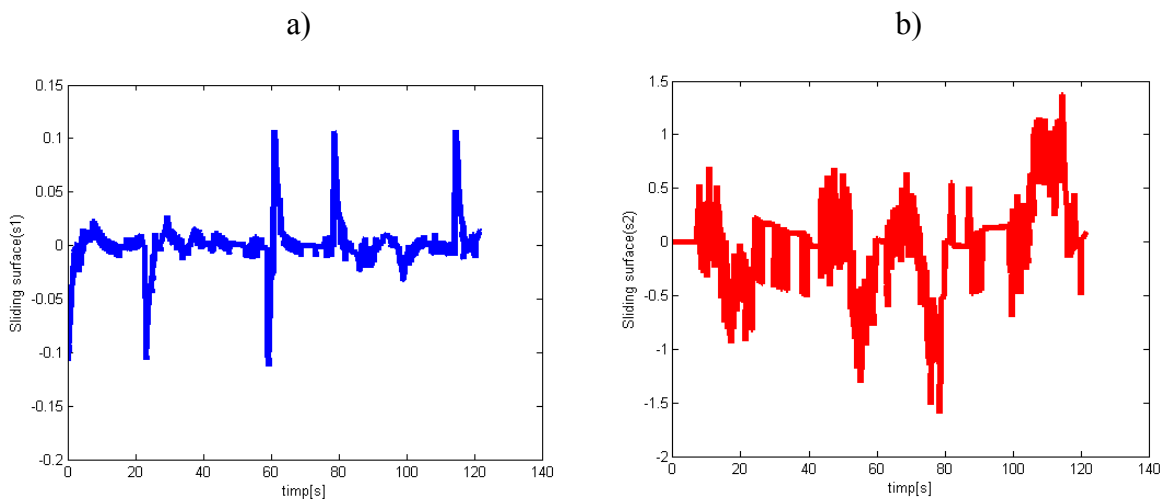


Fig. 5.16 Comanda conducerii în sliding mode: a) comanda pe suprafața s_2 ;
a) comanda pe suprafața s_1 ;

Se poate observa în **Fig. 5.16** a) și b) comanda conducerii în sliding mode, în cadrul suprafeței s_1 și a suprafeței s_2 se pot observa variații ale comenzii pentru menținerea traiectoriei.

5.6 Concluzii

Ca unealtă de modelare, un model de rețea Petri poate fi descris de un set de ecuații lineare algebrice sau de alte modele care să reflecte comportamentul sistemului analizat. Acest lucru permite o verificare formală a proprietăților asociate comportamentului derulării procesului de fabricație flexibilă vizat (relații de precedență între evenimente, operații concurente, sincronizările necesare, eliminarea situațiilor de blocare (*deadlock*), activitățile repetitive și excluderile mutuale ale resurselor partajate).

Validarea modelului prin simulare poate produce un set limitat de stări ale sistemului modelat și astfel poate arăta doar prezența (nu și absența) erorilor din model și specificațiile sale de bază. În cadrul simulării procesului de asamblare/dezasamblare deservit de un robot mobil echipat cu manipulator aferent liniei flexibile de fabricație HERA & Horstmann și a liniei flexibile de prelucrare FESTO MPS-200, s-au eliminat erorile, situațiile și evenimentele nedorite. În simulare s-au utilizat valori ale tranzițiilor aferente stărilor din procesul real pentru ca simularea să reflecte cât mai bine comportamentul liniilor flexibile. Pentru o mai bună evidențiere a procesului real modelat s-au simulat următoarele modele, care reprezintă și principalele contribuții ale acestui capitol:

- simularea modelului cu rețele Petri temporizate a procesului de asamblare și dezasamblare a unui produs, dezasamblare deservită de un robot mobil echipat cu manipulator;
- simularea modelului cu rețele Petri temporizate a procesului de prelucrare a unui produs, proces deservit de un robot mobil echipat cu manipulator;
- simularea modelului cu RPH sincronizate a procesului de asamblare și dezasamblare a unui produs, dezasamblare deservită de un robot mobil echipat cu manipulator;
- testarea modelului cu RPH sincronizate a procesului de asamblare și dezasamblare a unui produs, dezasamblare deservită de un robot mobil echipat cu manipulator utilizând programul LabVIEW;
- simularea conducerii robotului mobil echipat cu manipulator în cadrul deservirii liniei flexibile de asamblare/dezasamblare HERA & Horstmann;
- simularea conducerii robotului mobil echipat cu manipulator în cadrul deservirii liniei flexibile de prelucrare FESTO MPS-200;

Capitolul 6

Contribuții privind conducerea în timp real

6.1 Implementarea conducerii în timp real

6.2 Prezentarea aplicației în mediul de programare grafică LabVIEW

6.3 Conucerea în timp real a procesului de dezasamblare completă

6.4 Concluzii

6.1 Implementarea conducerii în timp real

Implementarea are la bază utilizarea programului LabVIEW 2010 [85]-[87] pentru dezvoltarea unei aplicații grafice de control și conducere a liniei flexibile HERA & Horstmann deservită de robotul mobil Pioneer P3-DX echipat cu manipulator Pioneer 5-DOF Arm în procesul de asamblare și dezasamblare completă a unui produs. Prin această aplicație se v-a controla și conduce în timp real procesul de asamblare/dezasamblare și se v-a sincroniza robotul mobil astfel încât să se execute operații de dezasamblare concomitent cu operațiile robotului mobil [78], [79], [80].

Programarea proceselor automatizate de asamblare/dezasamblare pe linia mecatronică se realizează prin programarea modulelor AP SIEMENS în limbajul de programare dedicat Simatic STEP 7. Comunicarea dintre AP al liniei flexibile HERA & Horstmann și stația de lucru care asigură sincronizarea cu platforma robotică se face prin intermediul plăcii de achiziție DAQ NI USB-6008.

Deoarece ieșirile și intrările digitale ale AP SIEMENS cu procesor CPU 314C-2 DP funcționează cu tensiune de 0÷24 V iar placa de achiziție lucrează cu tensiuni cuprinse între intervalul de 0÷5 V s-a utilizat o placă cu relee destinată interfațării plăcii de achiziție. În **Fig. 6.1** este prezentată schema bloc de conectare a echipamentelor în vederea realizării aplicației de control în timp real [81], [82].

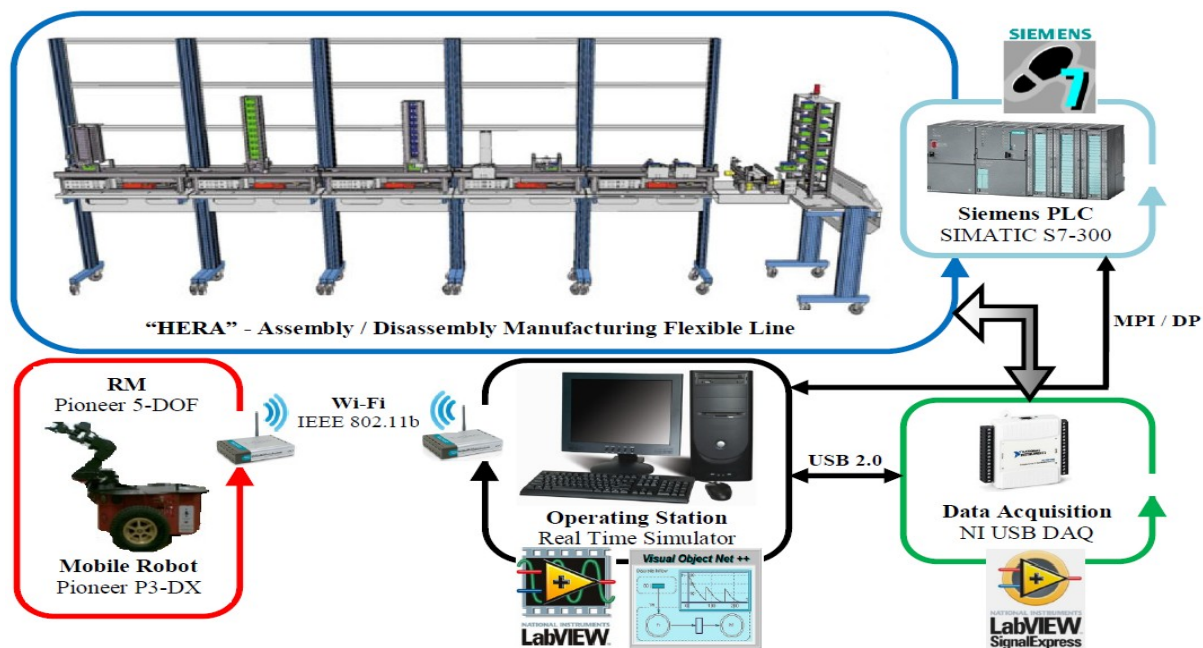


Fig. 6.1 Schema bloc a comunicației dintre echipamentele utilizate în conducerea liniei flexibile HERA & Horstmann deservită de un robot mobil Pioneer P3-DX

6.2 Prezentarea aplicației în mediul de programare grafică LabVIEW

Programul de conducere a întregului proces de asamblare și dezasamblare este împărțit în două bucle locale de conducere (**Fig. 6.2**) după cum urmează:

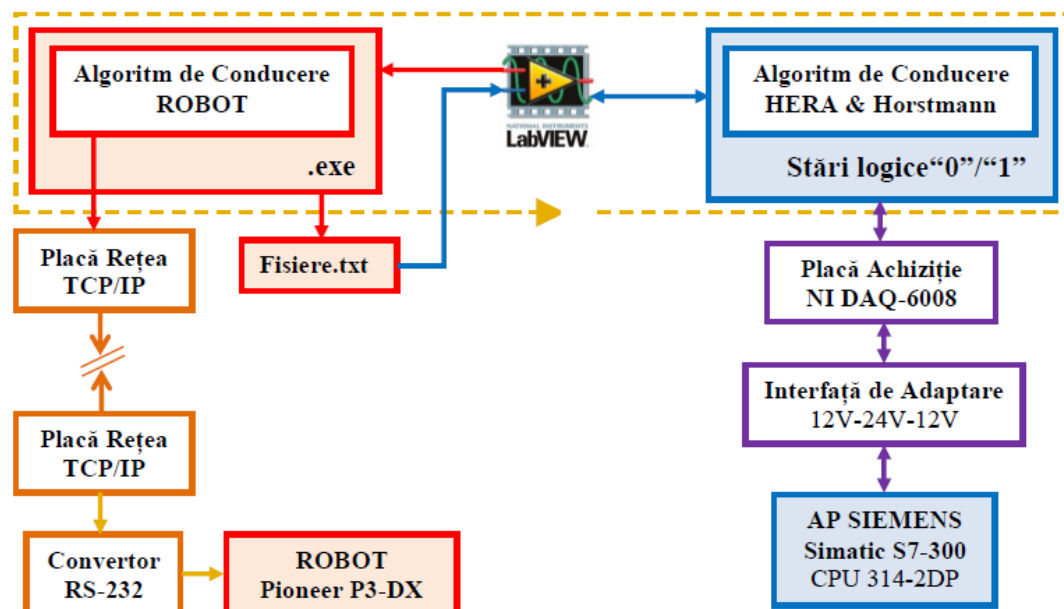


Fig. 6.2 Schema bloc a programului de conducere în timp real din mediul LabVIEW

- prima buclă de conducere are ca obiectiv controlul procesului de asamblare/dezasamblare a liniei flexibile HERA & Horstmann. Această buclă de conducere este implementată în automatul programabil SIEMENS S7-300 cu procesor 314-2 DP și programată în mediul de programare Simatic STEP 7.
- a doua buclă de conducere locală o reprezintă algoritmul de conducere al robotului mobil Pioneer P3-DX echipat cu manipulator. Algoritmul de conducere este implementat într-un fisier de tip executabil care este lansat în aplicație de programul LabVIEW. În urma derulării fisierului executabil sunt scrise în fișiere de tip .txt variabile care reprezintă poziția robotului mobil echipat cu manipulator din cadrul procesului de dezasamblare. Programul LabVIEW citește variabilele din fișierele .txt, actualizează interfața grafică cu noua poziție a robotului și intervine în algoritmul de sincronizare al operațiilor de dezasamblare deservită de robotul mobil echipat cu manipulator.

Comunicația dintre aceste două bucle locale de conducere se realizează printr-o interfațare dintre linia flexibilă și un calculator de proces, printr-o placa de achiziție (**Fig. 6.3**) care culege și transmite date din procesul de asamblare și dezasamblare și o comunicație wireless pe un protocol TCP/IP dintre robotul mobil și calculatorul de proces. Pe acest calculator de proces se găsește programul de conducere implementat în mediul LabVIEW care are rolul de a sincroniza în timp real cele două bucle de conducere în cadrul procesului de dezasamblare.

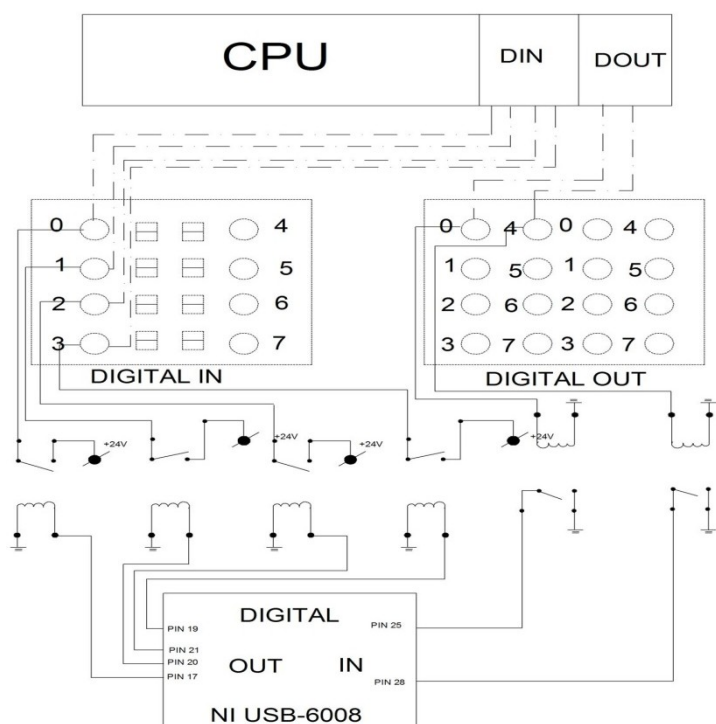


Fig. 6.3 Reprezentarea schematică a conexiunii dintre linia de mecatronică și placa de achiziție

Interfața grafică de control și conducere realizată în mediul LabVIEW este prezentată în Fig. 6.4 iar în Fig. 6.5 este prezentat panoul de control.

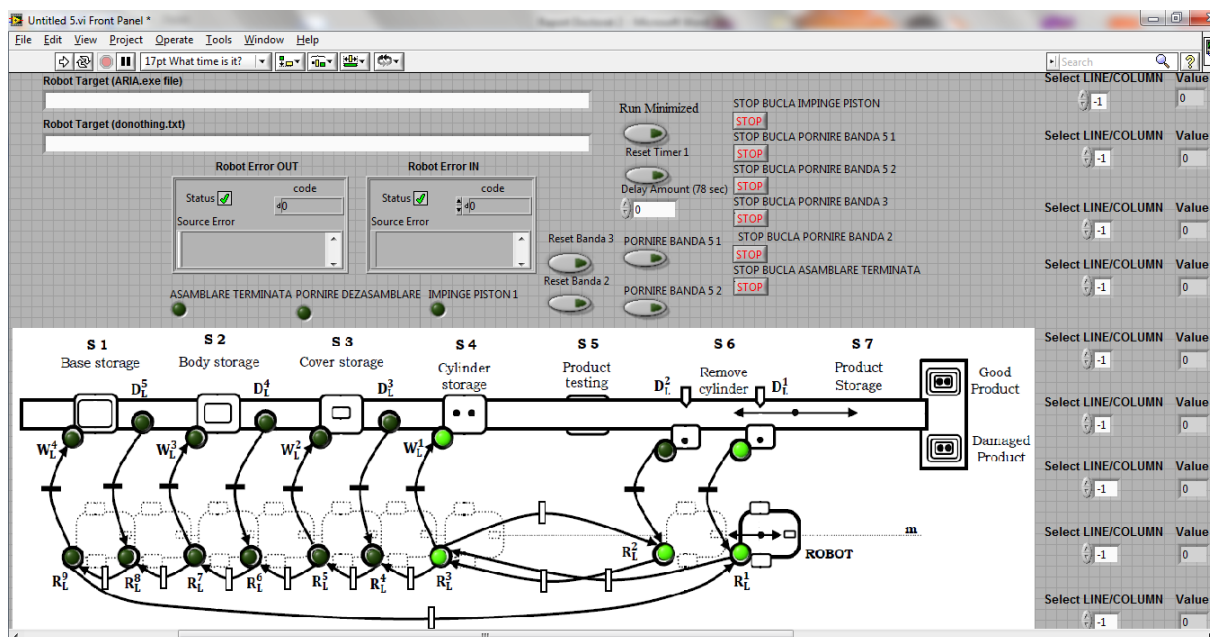


Fig. 6.4 Interfața grafică de conducere în mediul LabVIEW

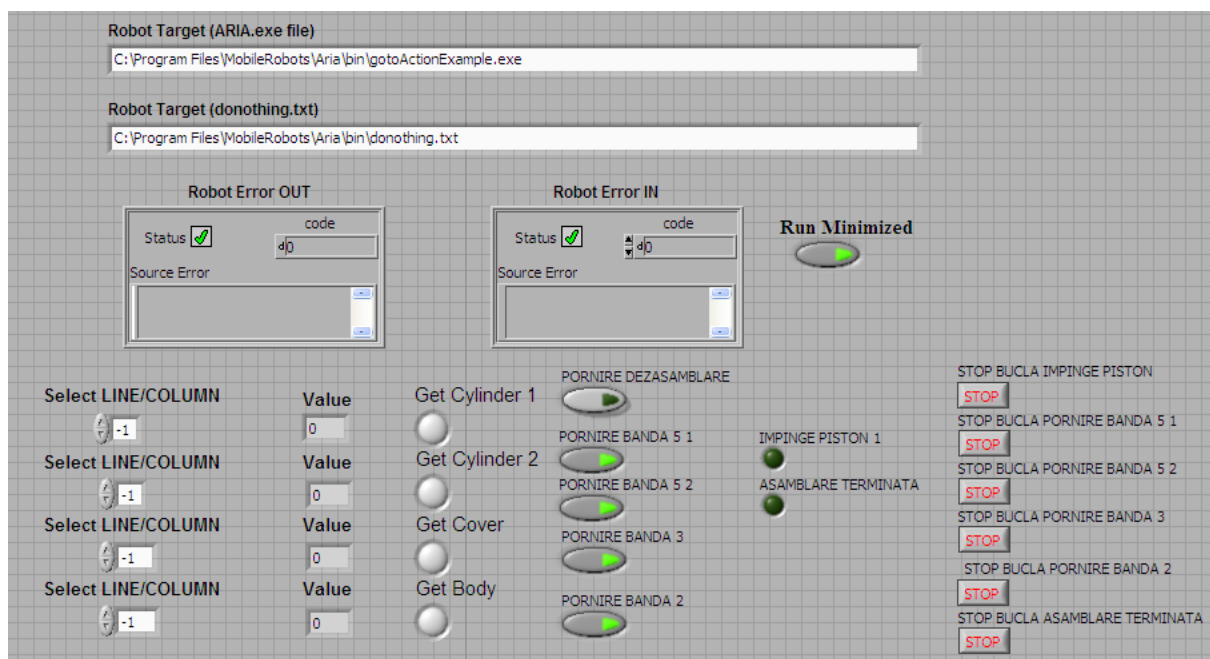


Fig. 6.5 Panou de control

Funcționarea aplicației:

- atunci când se primește un semnal de stare la intrarea digitală a plăcii de achiziție pe

pinul 28 (**Fig. 6.3**), se va trimite un semnal la ieșirea digitală a plăcii de achiziție pe pinul 17. Semnalul va fi trimis apoi la intrarea digitală I4.0 a modului de I/O al automatului programabil. Acest semnal de stare declanșează procesul de *START dezasamblare*.

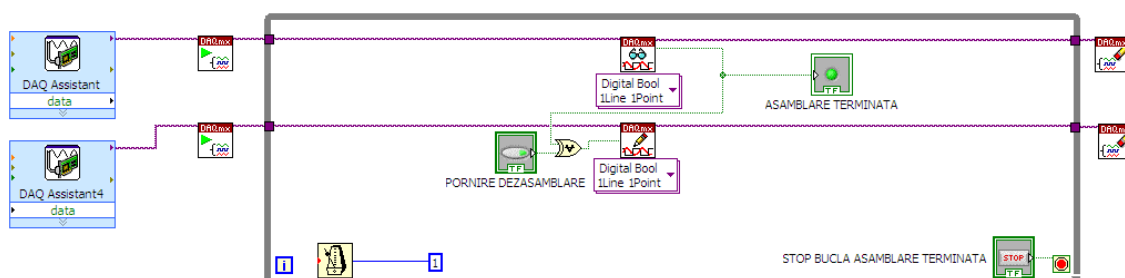


Fig. 6.6 Implementarea în mediul LabVIEW a acțiunii *START dezasamblare*

- prima operație a procesului de dezasamblare constă în deplasarea pe bandă a piesei rebut până în dreptul pistonului D_L^1 (**Fig. 6.7**).

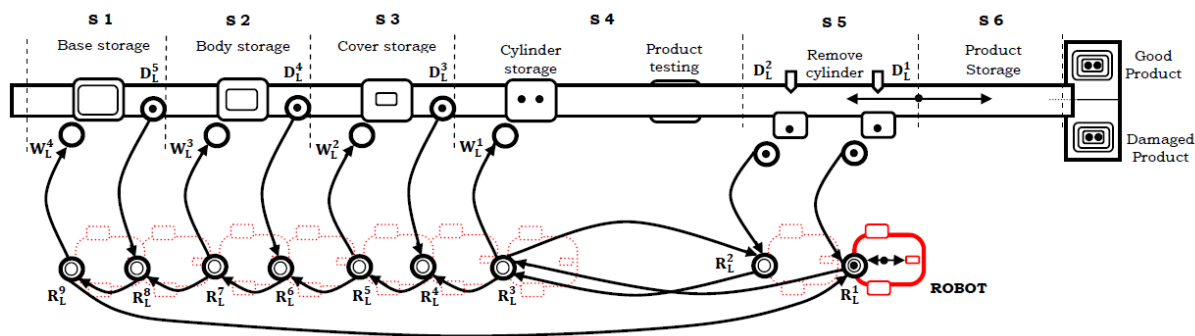


Fig. 6.7 Ciclul complet al robotului mobil echipat cu manipulator în procesul de dezasamblare

Aici se va scoate primul cilindru simultan cu scrierea unui semnal de stare pe ieșirea digitală Q0.0 a automatului programabil. Semnalul este achiziționat de placă de achiziție pe intrarea digitală aferentă pinului 28. Acest semnal de stare declanșează acțiunea de *START (ciclu continuu) robot mobil*.

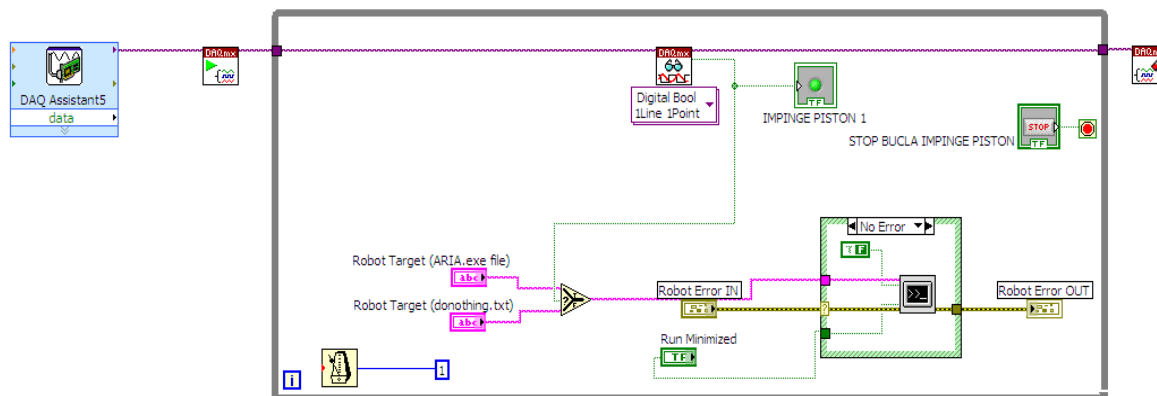


Fig. 6.8 Implementarea în mediul Labview a acțiunii *START (ciclu continuu) robot mobil*

- la declanșarea procesului de dezasamblare este înlăturat (dezasamblat) primul cilindru. După această operație se inițializează ciclul robotului mobil care atunci când preia cu ajutorul manipulatorului cilindru dezasamblat și gripperul este închis programul scrie într-un fișier text un caracter numeric care reprezintă o variabilă de stare, fișier text care apoi este deschis de către programul LabVIEW pentru a scrie valoarea variabilei la ieșirea digitală a plăcii de achiziție de la pinul 21. Semnalul ajunge la intrarea digitală I2.0 a automatului programabil care declanșează pornirea benzii și implicit deplasarea piesei la următorul piston D_L^2 care este acționat și astfel este scos cilindru numărul doi. Înainte de această acțiune de dezasamblare a celui de-al doilea cilindru este oprită banda transportoare. În **Fig. 6.9** se pot observa operațiile de apucare/depunere ale manipulatorului din cadrul procesului de dezasamblare.

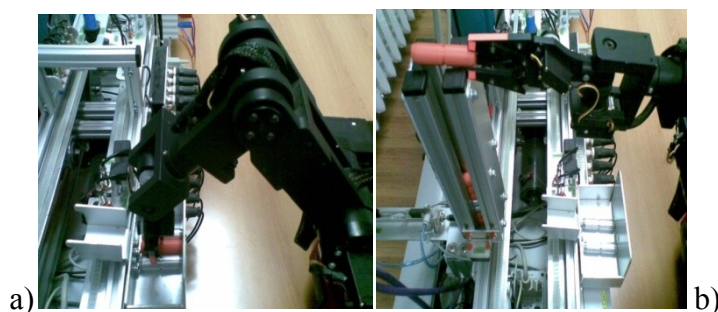


Fig. 6.9 Exemplificarea operațiilor efectuate de robotul mobil: a) de apucare; b) depunere;

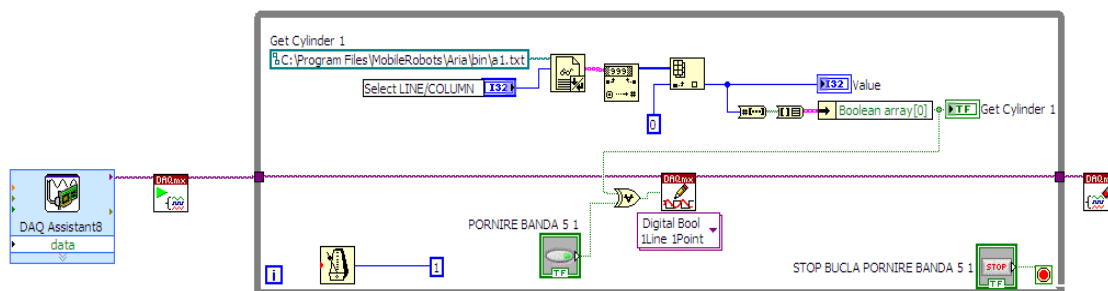


Fig. 6.10 Implementarea în mediul LabVIEW a acțiunii de pornire a benzii transportoare aferente stației S5

În continuarea secvenței de sincronizare dintre linia flexibilă HERA & Horstmann și robotul Pioneer P3-DX, ca și până acum, programul de conducere a robotului are specificat să creeze un fișier text în care se scrie o variabilă de stare, cifra 1 în cazul de față, după ce se efectuează de către manipulator mișcarea de închidere a gripperului. După închiderea gripperul, programul LabVIEW trimite la ieșirea digitală a plăcii de achiziție de la pinul

numărul 19 un semnal care va porni banda transportoare a liniei flexibile pentru a deplasa produsul la următoarea stație unde se efectuează un alt proces de dezasamblare.

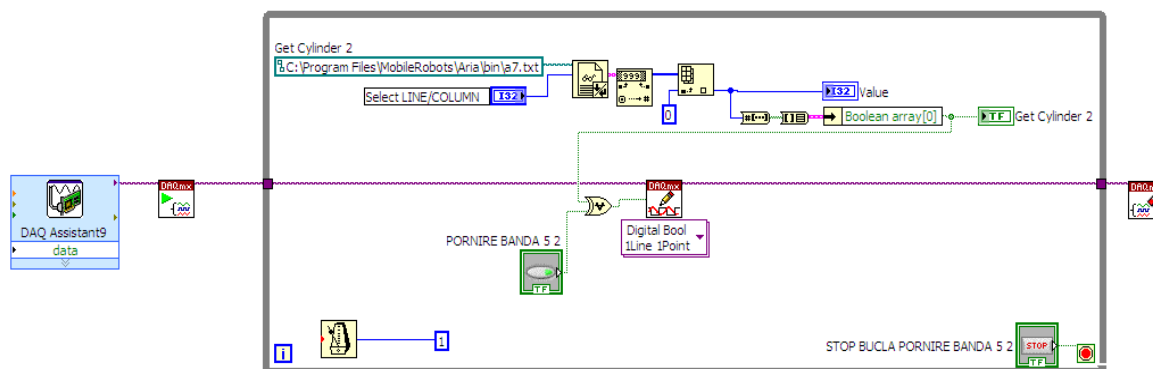


Fig. 6.11 Implementarea în mediul LabVIEW a acțiunii de pornire a benzii transportoare aferente stației S4

După ce robotul mobil s-a deplasat de la stația S4 aferentă magaziei de cilindri, la stația S3 în dreptul senzorului E15.5, începe secvența de dezasamblare a următoarei componente, cea a capacului. După ce gripperul manipulatorului a simulat că a apucat capacul, se scrie în fișierul text o altă variabilă de stare. Programul preia valoarea variabilei din fișier și activează ieșirea digitală numărul 17 a plăcii de achiziție, care va trimite un semnal la intrarea digitală I0.0 a automatului programabil. Acest semnal pornește banda transportoare și transportă produsul până la senzorul E13.5.

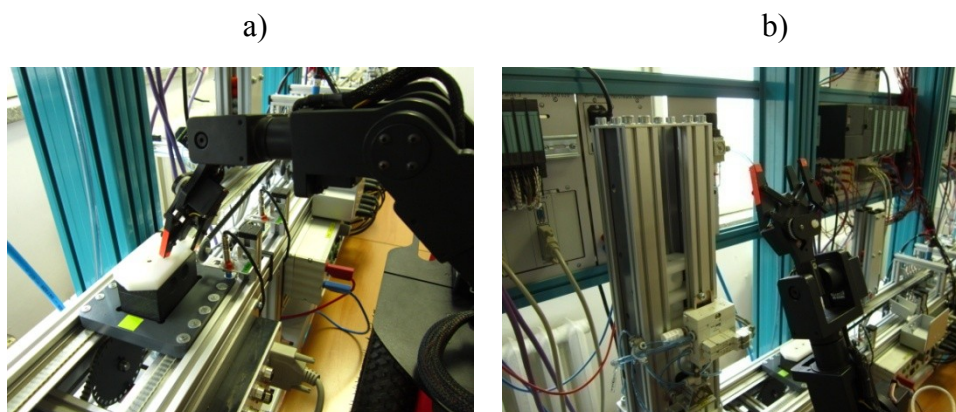


Fig. 6.12 Procesul de dezasamblare a piesei P3-capac: a) preluarea piesei; b) depunerea în magazie;

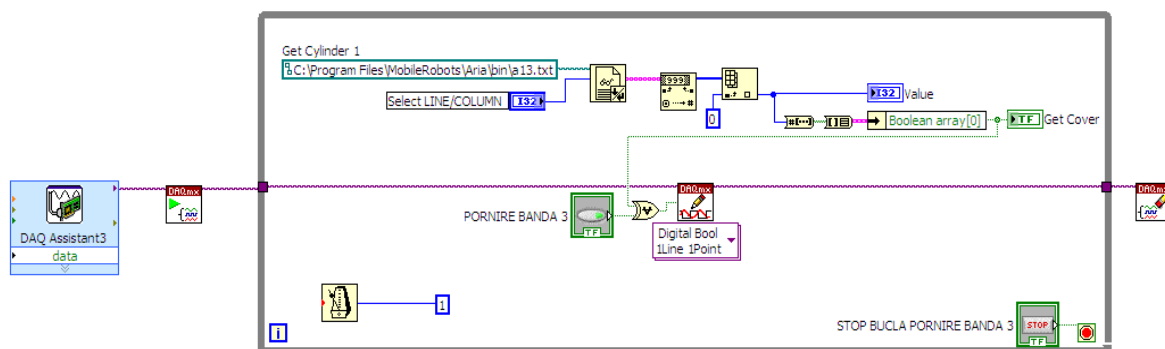


Fig. 6.13 Implementarea în mediul LabVIEW a acțiunii de pornire a benzii transportoare aferente stației S3

În cadrul penultimului proces de dezasamblare al componentei corp se efectuează aceeași procedură de sincronizare a deplasării robotului mobil până la senzorul E13.5, se preia corpul, după care paletul este deplasat de banda transportoare până în dreptul senzorului E11.5 a stației următoare unde este efectuată ultima operație de dezasamblare.

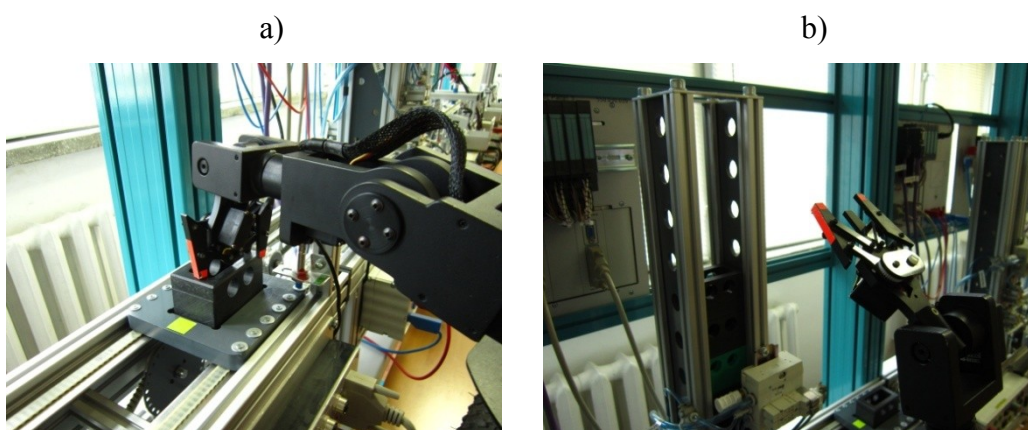


Fig. 6.14 Procesul de dezasamblare a piesei P2-corp: a) preluarea piesei; b) depunerea în magazie;

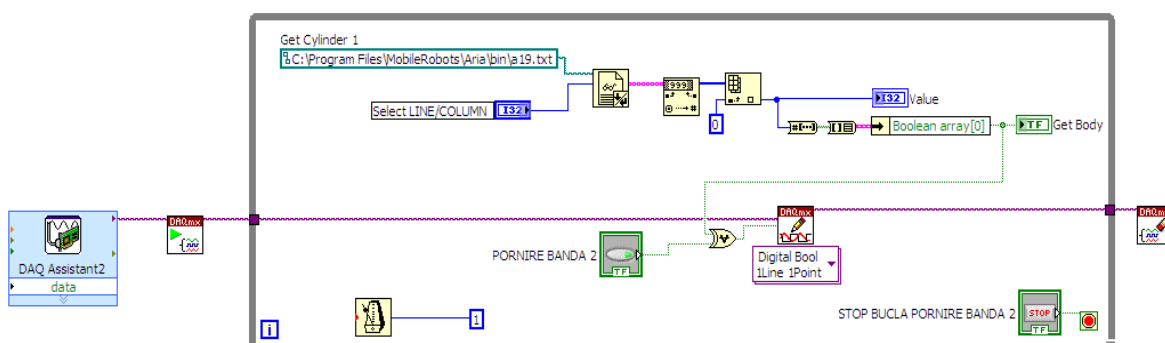


Fig. 6.15 Implementarea în mediul LabVIEW a acțiunii de pornire a benzii transportoare aferente stației S2

În ultima etapă a dezasamblării se preia paletul din dreptul sensorului E11.5 de către manipulator și se depozitează în magazia de paleți. După care robotul revine în poziția inițială pentru a fi pregătit în cadrul următorului proces de dezasamblare.

S-au prevăzut două elemente de control destinate preluării de la utilizator a path-urilor: elemente de tip executabil și elemente de tip text. Prin lansarea aplicației de conducere se vor lansa în mod automat aceste două elemente. Elementul de tip executabil este aferent conducerii robotului mobil, acesta este inițializat și executat în mod automat de către programul LabVIEW la fiecare declanșare a operației de dezasamblare. În cadrul elementului de tip text sunt salvate variabile din procesul de dezasamblare. Variabile care sunt citite prin intermediul plăcii de achiziție și salvate în LabVIEW. Prin aceste două tipuri de variabile se execută sincronizarea procesului de dezasamblare a liniei flexibile HERA & Horstmann cu robotul mobil Pioneer P3-DX echipat cu manipulator.

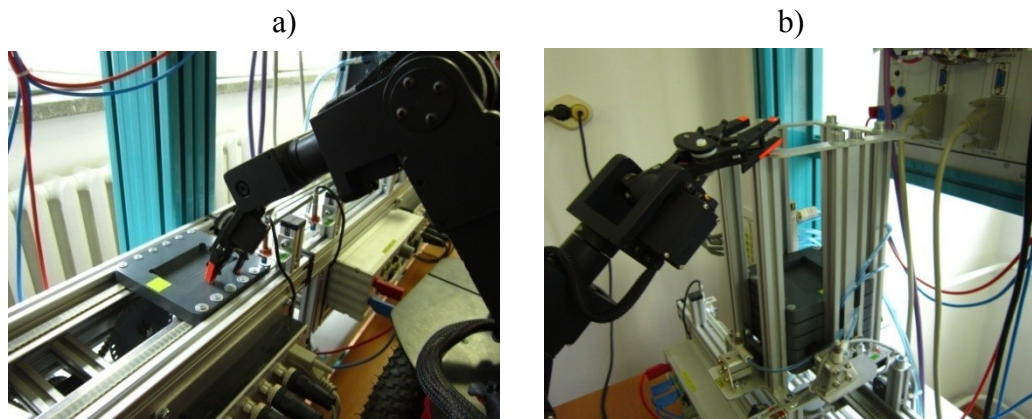


Fig. 6.16 Procesul de dezasamblare a piesei P1-palet: a) preluarea piesei; b) depunerea în magazie;

În **Fig. 6.17** sunt date distanțele ($Rd_1 \dots Rd_8$) pe care le parcurge robotul mobil echipat cu manipulator în cadrul procesului de dezasamblare completă a unui produs.

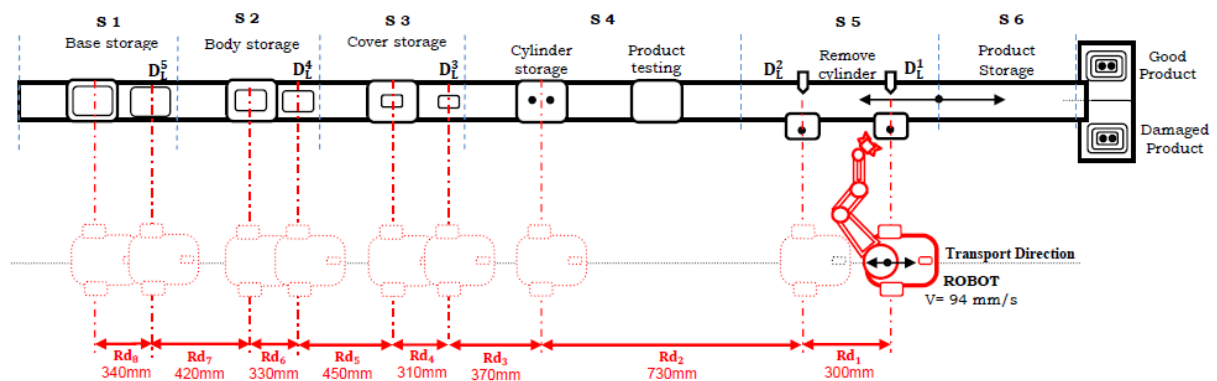


Fig. 6.17 Distanțe parcurse de robotul mobil echipat cu manipulator

- $Rd_1 + Rd_2 + Rd_2 = 300 + 730 + 730 = 1760 \text{ mm}$: reprezintă distanța parcursă de robotul mobil echipat cu manipulator în cadrul procesului de dezasamblare al primului cilindru;
- $Rd_2 + Rd_3 = 730 + 370 = 1100 \text{ mm}$: reprezintă distanța parcursă de robotul mobil echipat cu manipulator în cadrul procesului de dezasamblare al celui de-al doilea cilindru;
- $Rd_4 + Rd_5 = 310 + 450 = 760 \text{ mm}$: reprezintă distanța parcursă de robotul mobil echipat cu manipulator în cadrul procesului de dezasamblare al componentei capac;
- $Rd_6 + Rd_7 = 330 + 420 = 750 \text{ mm}$: reprezintă distanța parcursă de robotul mobil echipat cu manipulator în cadrul procesului de dezasamblare al componentei corp;
- $Rd_8 + Rd_5 = 340 \text{ mm}$: reprezintă distanța parcursă de robotul mobil echipat cu manipulator în cadrul procesului de dezasamblare al componentei palet;
- $Rd_8 + Rd_7 + Rd_6 + Rd_5 + Rd_4 + Rd_3 + Rd_2 + Rd_1 = 340 + 420 + 330 + 450 + 310 + 370 + 730 + 300 = 3250 \text{ mm}$: reprezintă distanța parcursă de robotul mobil echipat cu manipulator în cadrul procesului de dezasamblare la întoarcerea în poziția inițială;
- viteza robotului mobil echipat cu manipulator este de 94 mm/s ;

6.3 Conducerea în timp real a procesului de dezasamblare completă.

În cadrul procesului de testare în timp real din mediul de programare grafică LabVIEW se poate observa în **Fig 6.18** evoluția operației de dezasamblare a unei componente, prin preluarea de către manipulator a primului cilindru, transportul acestuia cu ajutorul platformei mobile până la deposit și eliberarea acestuia.

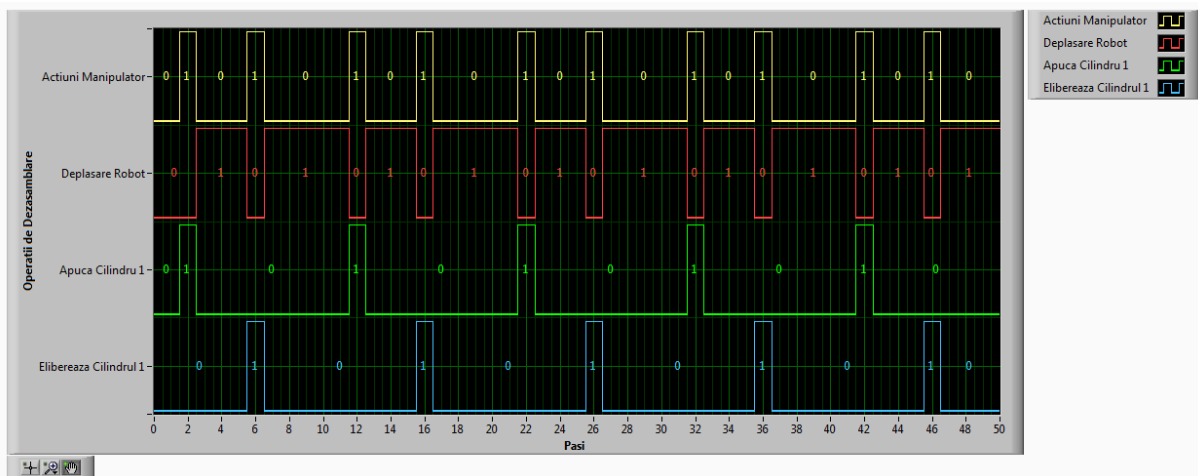


Fig. 6.18 Operația de dezasamblare a unei componente

În Fig. 6.19 este prezentată evoluția întregului proces de dezasamblare a unui produs și momentele în care manipulatorul deservește linia flexibilă.

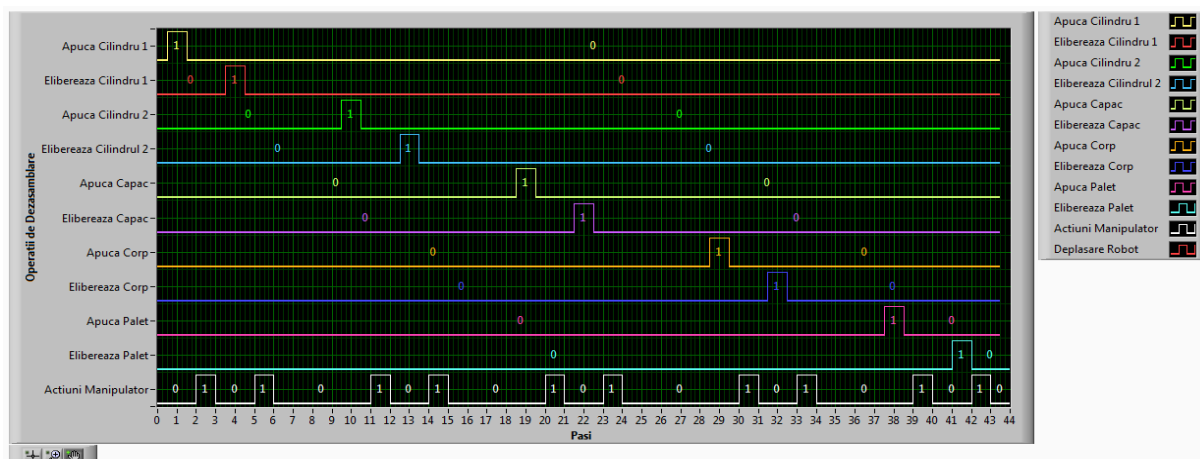


Fig. 6.19 Operațiile de dezasamblare a unui produs

În Fig. 6.20 se observă acțiunile de deplasare a robotului mobil în cadrul procesului de dezasamblare și operațiile manipulatorului.

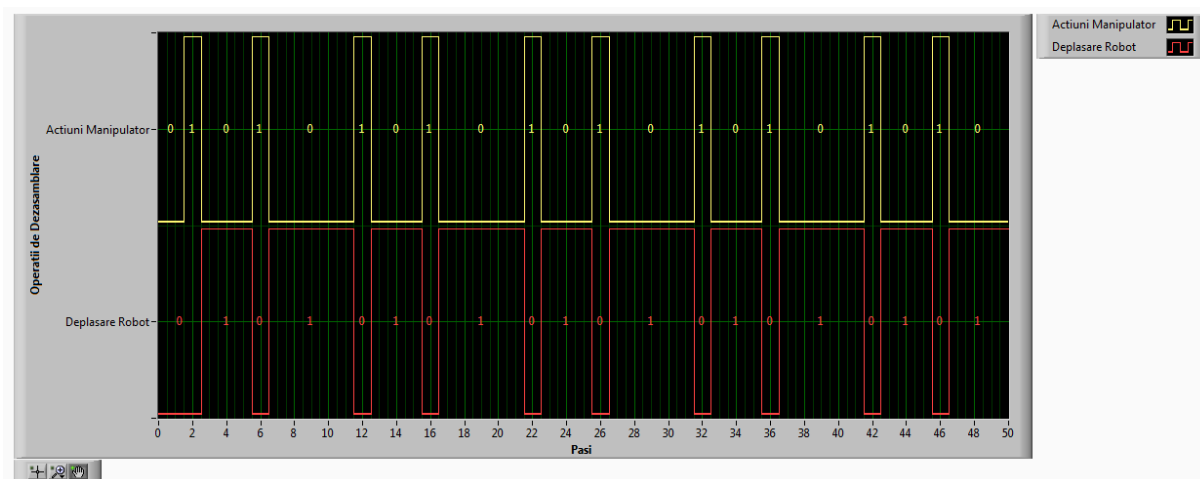


Fig. 6.20 Operațiile robotului mobil Pioneer P3-DX echipat cu manipulator Pioneer 5-DOF Arm

6.4 Concluzii

Principalele contribuții rezultate în urma implementării aplicației de conducere în timp real în cadrul prezentului capitol au vizat, în mod deosebit, elaborarea unei noi structuri flexibile de fabricație capabilă să îndeplinească sarcini multiple cu aceleași echipamente în cadrul diferitelor procese de fabricație. Rezultatele implementării au urmărit deservirea proceselor

flexibile de fabricație de o platformă mobilă echipată cu manipulator robotic capabilă să îndeplinească două sarcini diferite, transport și manipulare. Principalele contribuții sunt:

- dezvoltarea și implementarea unei aplicații grafice de control și conducere în timp real a sistemului flexibil de asamblare și dezasamblare a unui produs, derivat de un robot mobil echipat cu manipulator;
- dezvoltarea și implementarea unei metode de sincronizare a echipamentelor prin introducerea sistemului distribuit de conducere;

Capitolul 7

Concluzii

Dezvoltarea proceselor din cadrul sistemelor flexibile de fabricație deservite de o entitate robotică precum și dezvoltarea echipamentelor dedicate proceselor industriale a dus la dezvoltarea sistemelor de fabricație flexibilă, la conducerea proceselor de fabricație în timp real, la monitorizarea cu sisteme de tip calculator de proces și conducerea roboților utilizând sisteme de sincronizare ale echipamentelor pentru scăderea timpului de ciclu de fabricație.

Principala contribuție a acestei teze o reprezintă optimizarea și conducerea unor astfel de sisteme flexibile, prin realizarea programelor de conducere în timp real a unei linii flexibile de fabricație deservită de un robot mobil echipat cu manipulator care deservește un proces de asamblare și dezasamblare a unui produs, proces în urma căruia linia flexibilă devine complet reversibilă.

Principalele contribuții ale cercetării din cadrul acestei lucrări gravitează în jurul următoarelor domenii:

- Sisteme flexibile de fabricație
- Roboți industriali
- Conducerea liniilor flexibile de fabricație deservite de roboți

În **Capitolul 2** au fost abordate, dintr-o perspectivă tehnologică procesele de fabricației flexibile din cadrul sistemelor de fabricație flexibilă. O mare parte a acestor perspective sunt trecute în acest capitol introductiv dedicat stadiului actual.

În literatura de specialitate există multe abordări ce pot fi utilizate pentru o dezvoltare și optimizare a liniilor flexibile de fabricație. O mare parte a principalelor aspecte au fost prezentate în **Capitolul 3**, principala contribuție adusă o reprezintă optimizarea unei linii flexibile de fabricație care execută un proces de asamblare și dezasamblare parțială. Obiectivul urmărit a fost obținerea de contribuții privind optimizarea liniei flexibile în cadrul procesului de dezasamblare prin introducerea unei platforme mobile echipată cu manipulator. Astfel procesul de dezasamblare parțială este deservit de o entitate robotică capabilă să execute operații de dezasamblare, operații în urma cărora linia flexibila de asamblare și

dezasamblare parțială, devine complet reversibilă în urma dezamblării complete a unui produs. De asemenea în urma optimizării procesului de asamblare și dezamblare deservită de un robot mobil echipat cu manipulator au rezultat contribuții prin creșterea gradului de flexibilitate a liniei flexibile de fabricației în urma căruia se pot executa operații de asamblare/dezasamblare și prelucrare cu aceleași utilaje.

Pentru un sistem flexibil de fabricație complex, care execută operații de asamblare și dezamblare deservită de un robot mobil echipat cu manipulator o modelare riguroasă presupune modelarea cu acuratețe a tuturor etapelor din cadrul proceselor de asamblare și dezamblare cât și modelarea acțiunilor de transport și manipulare a robotului mobil echipat cu manipulator. Modelarea acestora poate fi realizată utilizând instrumente specifice modelării sistemelor cu evenimente discrete cât și utilizarea de diverse tipologii de rețele Petri dedicate modelării duratelor operațiilor aferente platformei mobile. În **Capitolul 4** se prezintă rezultate în urma modelării proceselor de asamblare, dezamblare parțială, dezamblare parțială deservită de un robot mobil echipat cu manipulator, asamblare și dezamblare parțială deservită de un robot mobil echipat cu manipulator, asamblare și dezamblare totală deservită de un robot mobil echipat cu manipulator. Concluziile acestei modelări evidențiază că modelarea proceselor flexibile din cadrul sistemelor flexibile de fabricație utilizează ca instrumente rețelele Petri netemporizate și netemporizate, rețele Petri hibride, și rețele Petri hibride sincronizate reprezintă o soluție pentru a evidenția cu acuratețe procesul real și de a scoate în evidență necesitatea optimizării operațiilor în vederea îmbunătățirii timpului total de ciclu de fabricație. În cadrul acestui capitol s-au elaborat și modelele unui proces de prelucrare din cadrul unui sistem flexibil de fabricație utilizând rețele Petri netemporizate și temporizate în abordare discretă. Modelarea se realizează printr-o procedură în două etape. În prima etapă se analizează procesele din cadrul liniei flexibile de fabricație și ordinea executării acestora rezultând un graf al procesului de asamblare și dezamblare deservită de un robot mobil echipat cu manipulator. În urma grafului rezultat și a descrierii individuale amănunțite se implementează modelele aferente fiecărui proces. Aceste modele sunt utilizate, de asemenea, în simulările din capitolul următor.

Abordările de modelare a proceselor din cadrul liniei flexibile de fabricație utilizând diverse tipologii de rețele Petri permit analiza și studierea diferitelor proprietăți ale sistemelor cu evenimente discrete. Toate modele, prezentate în Capitolul 3, au fost utilizate în **Capitolul 5**, în care pe baza instrumentelor de simulare a rezultat o analiză detaliată a proceselor de asamblare, dezamblare și prelucrare din care rezultă anumite criterii de performanță ce caracterizează funcționarea procesului fizic. Simularea a permis evidențierea proprietăților comportamentale ale liniei flexibile și a evoluției conform procesului real, a etapelor de

asamblare, dezasamblare deservită de robot mobil echipat cu manipulator și prelucrare. Rezultatele simulării arată că procesul de asamblare este un proces continuu care se desfășoară până când în mod accidental apare o componentă sau un produs defect, inițializându-se astfel procesul de dezasamblare al produsului.

Validarea modelui prin simulare reflectă comportamentul sistemului analizat. Acest lucru permite o verificare a proprietăților operațiilor derulate din cadrul liniei flexibile, operații care sunt controlate printr-un algoritm de conducere în cadrul **Capitolului 6**. Implementarea are la bază utilizarea unui mediu de programare pentru dezvoltarea unei aplicații grafice de control și conducere în timp real a liniei flexibile deservită de un robot mobil echipat cu manipulator în cadrul procesului de asamblare și dezasamblare a unui produs. S-a dezvoltat o metodă de sincronizare a echipamentelor și s-a implementat un sistem de conducere a cărui structură poate fi adaptată la modificări importante ale intrărilor/ieșirilor (scopurile de fabricație flexibilă în cazul nostru) și respectiv să asigure continuarea fluxului de asamblare/dezasamblare.

7.1 Sumarul contribuțiilor originale din cadrul tezei

În Capitolul 3

- optimizarea unei structuri clasice de sistem flexibil de fabricație care să permită efectuarea cu aceleași utilaje a două procese de fabricație: asamblare și dezasamblare a unui produs;
- introducerea unui robot mobil echipat cu manipulator pentru deservirea liniei flexibile de asamblare și dezasamblare în cadrul procesului de dezasamblare a unui produs;
- implementarea și optimizarea sistemului de control prin introducerea sistemului distribuit de conducere a liniei flexibile de asamblare și dezasamblare deservită de un robot mobil echipat cu manipulator;

În Capitolul 4

- modelarea procesului de asamblare pe linii flexibile utilizând rețele Petri netemporizate în abordare SED, particularizat pe linia flexibilă HERA & Horstmann;
- modelarea procesului de dezasamblare parțială pe linii flexibile utilizând rețele Petri netemporizate în abordare SED, particularizat pe linia flexibilă HERA & Horstmann;
- modelarea procesului de asamblare și dezasamblare parțială pe linii flexibile utilizând rețele Petri netemporizate în abordare SED, particularizat pe linia flexibilă HERA & Horstmann;

- modelarea procesului de asamblare și dezasamblare totală a unui produs pe linii flexibile utilizând rețele Petri temporizate și netemporizate în abordare SED, particularizat pe linia flexibilă HERA & Horstmann;
- modelarea procesului de asamblare și dezasamblare totală a unui produs pe linii flexibile deservite de un robot mobil echipat cu manipulator, utilizând rețele Petri Hibride generalizate, particularizat pe linia flexibilă HERA & Horstmann;
- modelarea procesului de asamblare și dezasamblare totală a unui produs pe linii flexibile deservite de un robot mobil echipat cu manipulator, utilizând rețele Petri Hibride sincronizate, particularizat pe linia flexibilă HERA & Horstmann.
- modelarea procesului de prelucrare pe linii flexibile utilizând rețele Petri netemporizate în abordare SED, particularizat pe linia flexibilă FESTO MPS-200;
- modelarea procesului de prelucrare pe linii flexibile utilizând rețele Petri temporizate în abordare SED, particularizat pe linia flexibilă FESTO MPS-200;

În Capitolul 5

- simularea modelului cu rețele Petri temporizate a procesului de asamblare și dezasamblare a unui produs, dezasamblare deservită de un robot mobil echipat cu manipulator;
- simularea modelului cu rețele Petri temporizate a procesului de prelucrare a unui produs;
- simularea modelului cu RPH sincronizate a procesului de asamblare și dezasamblare a unui produs, dezasamblare deservită de un robot mobil echipat cu manipulator;
- validarea modelelor simulate, analiza comportamentului sistemului flexibil deservit de un robot mobil echipat cu manipulator;
- testarea modelului cu RPH sincronizate a procesului de asamblare și dezasamblare a unui produs, dezasamblare deservită de un robot mobil echipat cu manipulator utilizând programul LabVIEW;
- simularea conducerii în timp real a robotului mobil echipat cu manipulator în cadrul deservirii liniei flexibile de asamblare și dezasamblare HERA & Horstmann;

În Capitolul 6

- dezvoltarea și implementarea unei aplicații grafice de control și conducere în timp real a sistemului flexibil de asamblare și dezasamblare a unui produs, deservit de un robot mobil echipat cu manipulator;
- dezvoltarea și implementarea unei metode de sincronizare a echipamentelor prin introducerea sistemului distribuit de conducere;

7.2 Direcții de cercetare viitoare

Având la bază rezultatele prezentei teze de doctorat, se pot identifica următoarele direcții de cercetare viitoare:

- utilizarea protocolului de comunicație PROFIBUS al liniei flexibile de asamblare și dezasamblare HERA & Horstmann pentru a transmite și citi variabile de stare ale procesului prin intermediul calculatorului de proces;
- implementarea unei aplicații în mediul de programare LabVIEW de tip client-server pentru a citi variabilele de stare la o viteză net superioară achiziției datelor prin intermediul placilor de achiziție. Aplicația poate utiliza programele OPC NI Server și OPC Siemens Server;
- implementarea unei bucle de conducere al manipulatorului utilizând diferite tehnici de procesare și recunoaștere a secvențelor video;
- implementarea unui algoritm de conducere al roboților mobili echipați cu manipulator pentru deservirea liniilor flexibile de prelucrare în vederea optimizării și deservirii acestora în procesul de prelucrare;

Bibliografie

- [1] Toni A. and Tonchia S. *Manufacturing Flexibility: a literature review*. International Journal of Production Research, 1998, vol. 36, no. 6, 1587-617.
- [2] Chryssolouris G. *Manufacturing Systems – Theory and Practice*. New York, NY: Springer Verlag, 2005. 2nd edition.
- [3] Tolio D., *Design of Flexible Production Systems – Methodologies and Tools*. Berlin: Springer, 2009. ISBN 978-3-540-85413-5.
- [4] F.F. Chen and E.E. Adams, “*The Impact of Flexible Manufacturing Systems on Productivity and Quality*”, IEEE Transactions of Engineering Management, vol. 38, pp. 33-45, 1991.
- [5] G. Da Silverira, D. Borenstein and F.S. Fogliatto, “*Mass Customization: Literature Review and Research Directions*”, International Journal of Production Economics, vol. 72, pp. 1-13, 2001.
- [6] R.R. Inman, D.E. Blumenfeld, N. Huang and J. Li, “*Production System Design for Quality: Research Opportunities from Automotive Industry Prospective*”, International Journal of Production Research, vol. 41, pp. 1953-1971, 2003.
- [7] J.E. Brady and T.T. Allen, “*Six Sigma Literature: A Review and Agenda for Future Research*”, Quality and Reliability Engineering International, vol. 22, pp. 335-367, 2006.
- [8] F. Tsung, Y. Li, and M. Jin, “*Statistical Process Control for Multistage Manufacturing and Service Operation: A Review and Some Extensions*”, International Journal of Services Operations and Informatics, vol. 3, pp. 191-204, 2008.
- [9] T. Lager, “*The Industrial Usability of Quality Function Deployment: A Literature Review and Synthesis on a Meta-level*”, R&D Management, vol. 35, pp.409-426, 2005.
- [10] J.R. Sharma, A.M. Rawani, and M. Barahate, “*Quality Function Deployment: A Comprehensive Literature Review*”, International Journal of Data Analysis Techniques and Strategies, vol. 1, pp. 78-103, 2008.
- [11] J. Shi and S. Zhou, “*Quality Control and Improvement for Multistage Systems: A Survey*”, IIE Transactions, vol. 41, pp. 744-753, 2009.
- [12] J. Li and N. Huang, “*Quality Evaluation in Flexible Manufacturing Systems: A Markovian Approach*”, Mathematical Problems in Engineering, vol. 2007, article ID 57128, 2007.
- [13] J. Li, D.E. Blumenfeld and S.P. Marin, “*Production System Design for Quality Robustness*”, IEE Transactions, vol. 40, pp. 162-176, 2008.

- [14] C. Galitsky and E. Worrell, "*Energy Efficiency Improvement and Cost Saving Opportunities for the Vehicle Assembly Industry*", Lawrence Berkeley National Laboratory (LBNL-50939-Revision), 2008.
- [15] Boothroyd, G. and Althing, L. "*Design for assembly and disassembly*", *Annals of CIRP* 41(2), 625–636, 1992.
- [16] Feddema, J. T. and Simon, R. W., "*Visual servoing and CAD-driven microassembly*", *IEEE Robotics Automat. Mag.* 5(4), 18–24, 1998.
- [17] Fisher, E. L. and Nof, S. Y., "*Knowledge-based economic analysis of manufacturing systems*", *J. Manufacturing Systems* 6(2), 138–150, 1987.
- [18] Huang, Y. F. and Lee, C. S. G., "*An automatic assembly planning system*", *Proc. of the IEEE Internat. Conf. on Robotics and Automation*, Cincinnati, OH, pp. 1594–1599, 1990.
- [19] O'Grady, P. and Oh, J. S., "*A review of approaches to design for assembly*", *Concurrent Engineering* 1(3), 5–11, 1991.
- [20] Lotter, B., *Manufacturing Assembly Handbook*, Butterworths, London, 1989.
- [21] Nof, S. Y., Wilhelm, W. E., and Warnecke, H. J., *Industrial Assembly*, Chapman & Hall, London, 1997.
- [22] Warnecke, H. J., Schweizer, M., Tamaki, K., and Nof, S. Y., Assembly, in: G. Salvendy (ed.), *Handbook of Industrial Engineering*, 2nd edn, Wiley, New York, Chapter 19, 1992.
- [23] H. Tempelmeier and H. Kuhn, "*Flexible Manufacturing Systems: Decision Support for Design and Operation*", Wiley-Interscience, 1993.
- [24] A.K. Sethi and P.S. Sethi, "*Flexibility in Manufacturing: A Survey*", *International Journal of Flexible Manufacturing Systems*, vol. 2, pp. 289–328, 1990.
- [25] M. Barad and S.Y. Nof, "*CIM Flexibility Measures: A Review and a Framework for Analysis and Applicability Assessment*", *International Journal of Computer Integrated manufacturing*, vol. 10, pp. 296–308, 1997.
- [26] Rajan, V. N. and Nof, S. Y., "*Logic and communication issues in cooperation planning for multi-machine workstations*", *Internat. J. Systems Automat. Res. Appl.* 2, 193–212, 1992.
- [26] Zussman, E. M. and Seliger, G., "Production remanufacturing", in: S. Y. Nof (ed.), *Handbook of Industrial Robotics*, 2nd edn, Wiley, New York, Chapter 54, pp. 1037–1044, 1999.
- [27] Boothroyd, G. and Althing, L., "*Design for assembly and disassembly*", *Annals of CIRP* 41(2), 625–636, 1992.
- [28] B. Kopacek and P. Kopacek, "*Intelligent Disassembly of Electronic Equipment*", in *Proceedings of the 1st IFAC Workshop on „Intelligent Assembly and Disassembly – IAD'98“*, pp. 87–92, Oxford, UK, 1998.

- [29] Kopacek and P.Kopacek, *"Intelligent Disassembly of Electronic Products"*, in Proceedings of the 2nd Symposium „Eco-Efficient Concepts for the Electronics Industry Towards Sustainability - CARE INNOVATION'98“, pp. 130-136, Vienna, Austria, 1998.
- [30] Miyata M., *"Development of Robot System for Automatic Disassembling and Assembling of Brake Equipment Parts"*, Japanese Railway Engineering no.119, 1992.
- [31] Noller, R.M., *"Design for Disassembly Tactics"*, Assembly, Jan 1992.
- [32] Reynders, R.J.M., *"Design of a disassembly system"*, T.U. Eindhoven, 1995.
- [33] Selinger, G., *"Design for Disassembly in the Framework of Life-Cycle-Design"*, CIRP Copenhagen, June 1993.
- [34] Subramani, A.K. and P. Dewhurst, *"Automatic generation of Product Disassembly Sequences"*, Annals of the CIRP, Vol. 40, no.1, 1991.
- [35] Warnecke, J., Schweizer, M. and M. Kahmeyer, *"Flexible disassembly with industrial robots"*, IPA, Stuttgart, 1993.
- [36] Zhang GW, Zhang SC, Xu YS. *"Research on flexible transfer line schematic design using hierarchical process planning"*. Journal of Material Processing Technology 2002; 129: 629-633.
- [37] Contini P, Tolio T. *"Computer-aided set-up planning for machining centers configuration"*, International Journal of production research 2004; vol 42, n 17: 3473-3491.
- [38] Dolgui A, Guschinsky N, Levin G. *"A special case of transfer lines balancing by graph approach"*, European Journal of Operational Research 2006, 168: 732-746.
- [39] Matta A, Semeraro Q. *"Design of advanced manufacturing systems"*. Springer 2005.
- [40] Z. Banaszak (Ed.), *"Modelling and Control of FMS: Petri Net Approach"*, Wroclaw Technical University Press, Wroclaw, Poland, 1991.
- [41] F. Dicesare, G. Harhalakis, J.M. Proth, M. Silva, F.B. Vernadat, *"Practice of Petri Nets in Manufacturing"*, Chapman and Hall, London, 1993.
- [42] P. KOŠTÁL, K. KRAJČOVÁ, and R. RUŽAROVSKÝ, *"Material flow description in flexible manufacturing"*, Advanced logistic systems, vol. 4, pp. 104-108, 2010.
- [43] P. KOŠTÁL, A. MUDRIKOVÁ, and D. R. DELGADO SOBRINO, *"Material Flow in Flexible Production Systems"*, Proceedings in Manufacturing Systems, vol. 5, no. 4 (2010), pp. 213-216, 2010.
- [44] Radaschin A., Filipescu A., Manzu V., Minca E. and Filipescu Jr. A. *"Adaptive disassembly sequence control by using mobile robots and system information"*, Proceeding of 15th IEEE International Conference in System Theory, Control and Computing (ICSTCC 2011), 14-16 October 2011, Sinaia, Romania, ISBN 978-973-621-323-6, 499-505, (2011).

- [45] R. RUZAROVSKY, S. HORVATH, and K. VELISEK, "*Designing of automated manufacturing and assembly systems*", in Annals Of DAAAM For 2008 & Proceedings Of The 19th International DAAAM Symposium, Vienna Univ Technology, Karlsplatz 13, Wien, A-1040, Austria, 2008, pp. 1201-1202.
- [46] David, R. and Alla, H., "*Petri Net and Grafcet: Tools for Modeling Discrete Event Systems*", Prentice Hall.
- [47] Lee, D. and DiCesare, F. "*Scheduling flexible manufacturing systems using Petri nets and heuristic search*", IEEE Transaction on Robotics and Automation, 10, 123-132, 1994.
- [48] Critan, I., "*Sisteme Flexibile*", TCMM, vol. I si vol. II. Editura Tehnică, Bucuresti, 1987.
- [49] Popescu, D., Dumitru, C., "*Sisteme flexibile de prelucrare*". Reprografia Universității din Craiova, 1996
- [50] Shih, H. and Sekiguchi, T., "*A timed Petri net and beam search based on-line fms scheduling system with routing flexibility*", Proceedings of the 1991 IEEE International Conference on Robotics and Automation, 1991, pp. 2548-2553.
- [51] Stecke, K. E. and Raman, N., "*Production planning decisions in flexible manufacturing systems with random material flows*", IIE Transactions, 26(5), 2-17, 1996).
- [52] Radaschin A., A. Voda, E. Minca, A. Filipescu, "*Task Planning Algorithm in Hybrid Assembly/Disassembly Process*", 14th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2012), 23-25 May 2012, IFAC Proceedings Vol.14, Part 1, ISSN: 1474-6670, ISBN: 978-3-902661-98-2.
- [53] F. Balduzzi, A. Giua, and G. Menga., "*First-order hybrid Petri nets: A model for optimization and control*", IEEE Trans. on Robotics and Automation, 16(4):382-399, 2000.
- [54] C. Chaouiya and Y. Dallery. "*Petri net models of pull control systems for assembly manufacturing systems*", In Procs. of the 2nd Int. Workshop on Manufacturing and Petri Nets, ICATPN, pages 85-103, Toulouse, France, 1997.
- [55] Moore, K.E. and Gupta, S.M., "*Petri net models of flexible and automated manufacturing systems: A survey*", International Journal of Production Research, 34(11), pp. 3001-3035, 1996,.
- [56] Alla, H., Cavaill'e, J.-B., Le Bail, J. and Bel, G., "*Les syst`emes de production par lot: une approche discret-continu utilisant les r`eseaux de Petri hybrids*". Symposium ADPM '92. Paris 1992..
- [57] Alla, H. and David, R., "Continuous and Hybrid Petri Nets". Journal of Circuits, Systems & Computers 8(1): 159-188, 1998.
- [58] Allam, M. and Alla, H., "Modelling production systems by hybrid automata and hybrid Petri nets". Conf. on Control of Industrial Systems Belfort, 1997.

- [59] Alur, R., Courcoubetis, C., Halwachs, N., Henzinger, T.A. , Ho, P.H., Nicollin, X. , Olivero, A., Sifakis, J. and Yovine, S., The algorithmic analysis of hybrid systems. *Theoretical Computer Science* 138: 3–34, 1995.
- [60] Brinkman, P. L. and Blaauboer, W. A. , “Timed continuous Petri nets: A tool for analysis and simulation of discrete event systems”. *European Simulation Symposium* Ghent, 1990.
- [61] David, R. , “Modeling of hybrid systems using continuous and hybrid Petri nets”. *Petri Nets & Performance Models (PNPM’97)*. Saint Malo, France, 1997, pp. 47–58.
- [62] Le Bail, J., Alla, H. and David, R., “Hybrid Petri nets”. *European Control Conference* Grenoble, 1991, pp. 1472–1477.
- [63] Weiting, R., Hybrid high-level Nets. *Proceedings of the Winter Simulation Conference*, Coronado, USA, 1996, pp. 848–855.
- [64] Weiting, R., Modeling and Simulation of Hybrid Systems Using Hybrid High-Level Nets. *Proceedings of the 8th European Simulation Symposium*, Genova, 1996, pp. 158–162.
- [65] Balduzzi, F., Giua, A. & Seatzu, C., “Modelling and Simulation of Manufacturing Systems Using First-Order Hybrid Petri Nets”, *Int. J. of Production Research*, Vol. 39, No. 2, pp. 255-282, (2001).
- [66] Balduzzi, F., Giua, A. & Menga, G., “First-Order Hybrid Petri Nets: a Model for Optimization and Control”, *IEEE Trans. Robotics and Automation*, Vol. 16, pp. 382-399, (2000).
- [67] Champagnat, R., Esteban, P., Pingaud, H. & Valette, R., “Modeling and Simulation of a Hybrid System Through PR/TR PN-DAE Model, *Proc. 3rd Int. Conf. on Automation of Mixed Processes*, Reims, France, (1998)..
- [68] Chen, H. & Hanisch, H.-M., “Hybrid net condition/event systems for modeling and analysis of batch processes“, *Proc. 3rd Int. Conf. on Automation of Mixed Processes*, Reims, France, (1998).
- [69] Chen, H., Amodeo, L., Chu, F. & Labadi, K., “Modeling and performance evaluation of supply chains using batch deterministic and stochastic Petri nets”, *IEEE Transactions on Automation Science and Engineering*, vol. 2, no. 2, pp. 132-144, (2005).
- [70] Demongodin, I. & Giua, A., “Some analysis methods for continuous and hybrid Petri nets”, *Proc. IFAC World Congress*, Barcelona, Spain, (2002).
- [71] Alina V., A. Radaschin, E. Minca, A. Filipescu, “Control of Automatic Robot with Guided Manipulator Integrated into Flexible Manufacturing System Using Hybrid Petri Nets”, 16 th International Conference on System Theory, Control and Computing Joint Conference (ICSTCC 2012), 12-14 October, Sinaia, Romania.
- [72] Dumitrascu B., A. Filipescu, A. Radaschin, Minca E., A.Filipescu Jr., *Discrete-Time Sliding Mode Control Of Wheeled Mobile Robots*, 8th Asian Control Conference ASCC2011, 15-18 May 2011, IEEE ISBN: 978-1-61284-487-9, pp. 771-776.

- [73] Susnea I. Vasiliu G, Filipescu A, Coman G., Radaschin A., *Real-Time Control of Autonomous Mobile Robots Using Virtual Pheromones*, Proceedings of the 7th Asian Control Conference, Hong Kong, China, August 27-29, 2009, IEEE Catalog Number CFP09832, ISBN:978-89-956056-9-1, pp.1450-1455.
- [74] Susnea I., Filipescu A., Serbencu A, Radaschin A., *Virtual Pheromones to Control Mobile Robots . A Neural Network Approach*, Proceedings of the IEEE International Conference on Automation and Logistics, Shenyang, China, ISBN: 978-1-4244-4795-4/09, 2009 IEEE, CD-ROM Proceedings IEEE, Catalog: CFP09CAL, August 5 - 7, 2009, Shenyang, China, pp1962-1967.
- [75] Susnea I., Vasiliu G., Filipescu A., Radaschin A., *Virtual Pheromones for Real-Time Control of Autonomous Mobile Robots*, STUDIES IN INFORMATICS AND CONTROL, Volume: 18 Issue: 3 Pages: 233-240 Published: SEP 2009, IDS Number: 499SC, ISSN: 1220-1766.
- [76] Susnea I., Adrian Filipescu, Grigore Vasiliu, A. Radaschin, *The Bubble Rebound Obstacle Avoidance Algorithm for Mobile Robots*, Control and Automation (ICCA), 2010 8th IEEE International Conference in Control and Automation, 9-11 June 2010, Xiamen, China, ISBN: 978-1-4244-5195-1, ISSN:1948-3449, pp.540-545.
- [77] Radaschin A., *Sliding mode control of an autonomous welding robot*, *Automation Quality and Testing Robotics*, 2010 IEEE International Conference, Automation Quality and Testing Robotics (AQTR 2010), 2010 IEEE International Conference, Cluj-Napoca, Romania , 28-30 May 2010, volume 1, pp.1-5, ISBN: 978-1-4244-6724-2.
- [78] Wiendahl, H.-P., and Breithaupt, J.-W., "Automatic Production Control Applying Control Theory," *Int. J. Prod. Econ.*, 63(1), pp. 33–46, 2000.
- [79] Ratering, A. M., and Duffie, N. A., "Design and Analysis of a Closed- Loop Single Workstation PPC System," *CIRP Ann.*, 52(1), pp. 355–358, 2003.
- [80] Tom Pherson, 'Overall Equipment Effectiveness & Real-Time Visual Management Critical Lean Tools' *Intelligent Manufacturing Solutions*, June 2006, Page(s):1-18.
- [81] S. K. Subramaniam, S. H. Husin, Y. Yusop, A. H. Hamidon, "Data collection using a real time production monitoring system for factors affecting production lines ", *Journal of Science and Technology in the Tropics* (2008) 4 Page(s):129-134.
- [82] Siva Kumar a/l Subramaniam, Siti Huzaimah binti Husin, Yusmarnita binti Yusop and Abdul Hamid bin Hamidon, "*Real Time Production Performance Monitoring System a Production Aid for all Industries*". The 6th WSEAS International Conference on Circuits, Systems, Electronics, Control & Signal Processing (CSECS'07) World Scientific and Engineering Academy and Society, 29-31 December 2007, Page(s):181-184.
- [83] Fukao T., Nakagawa H., Adachi N., *Adaptive tracking control of a nonholonomic mobile robot*, *IEEE Transactions on Robotics and Automation*, vol. 16, nr.5, pag. 609-615, 2000.
- [84] De Luca A. and Oriolo G. (1995). *Modelling and Control of Nonholonomic Mechanical*

Systems, in: J. Angeles, A. Kecskemethy (Editors), *Kinematics and Dynamics of Multi-Body Systems*, Springer-Verlag, 277–342.

[85] LabVIEW User Manual, 2011. <http://www.ni.com/pdf/manuals/320999e.pdf>

[86] LabVIEW Function and VI Reference Manual, 1998.
<http://www.ni.com/pdf/manuals/321526b.pdf>

[87] LabVIEW Data Acquisition Basic Manual, 1996.
<http://www.ni.com/pdf/manuals/320997a.pdf>

ANEXE

ANEXA A: Programul de conducere sliding mode al robotului Pioneer P3-DX echipat cu manipulator Pioneer 5-DOF Arm în cadrul procesului de deservire al liniei flexibile HERA & Horstmann

```
#include "Aria.h"
#include "ArActionDesired.h"
#include <iostream>
#include <math.h>
#include <conio.h>
#include <stdio.h>
//#include "clock.cpp"
double v, av, w, aw; /* velocity to drive left wheel*/
double theta_des_next; /* theta desired */
double ref_xr, ref_yr, ref_Th, ref_w, ref_v, x_des_next, y_des_next; /*
desired position */
double x_e, y_e, theta_e, x_e_der, y_e_der, theta_e_der;
/* errors */
double s_1, s_2, Q1, Q2, P1, P2, gama0, gama_x, gama_y;
double v_c, v_c_der, w_c, temp1, temp2, V_R;
double Ts, v_l, v_r, timp, alpha, epsilon1, epsilon2, q1,q2;
double x,y, theta;
int i, n, j;
ArRobot robot;
ArSonarDevice sonar;
ArP2Arm arm;
void Delay() {
    // Make the changes visible
    long lngDelay;
    for(lngDelay = 0; lngDelay < 100000; lngDelay++) {
    }
}
static double angleLimit(double angle)
{
    if (angle > 3.14)
        angle = angle - 2*3.14;
    else
        if (angle < -3.14)
            angle = angle + 2*3.14;
    return(angle);
}
/* ----- SIGN & SATURATION FUNCTION ----- */
double sign(double value){
    if (value<0)
        return(-1.00);
    else
        if (value>0)
            return(1.00);
        else
            return(1.00);
}
double satur(double value){
    if (value<-1)
        return (-1.00);
    else if (value > 1)
        return (1.00);
    else
```

```

    return (value);
}
FILE *date6;
FILE *date5;
FILE *date4;
FILE *date3;
void sliding(int value, double vel)
{
    date6=fopen("C:/Program
Files/MobileRobots/Aria/bin/viteza_ciclu_complet.txt","a+");
    date5=fopen("C:/Program Files/MobileRobots/Aria/bin/eroare_x_e.txt","w+");
    date4=fopen("C:/Program Files/MobileRobots/Aria/bin/eroare_y_e.txt","w+");
    date3=fopen("C:/Program
Files/MobileRobots/Aria/bin/eroare_theta_e.txt","w+");
    ref_v = 0.0;
    ref_w = 0.0;
    robot.lock();
    ArUtil::sleep(100);
    robot.unlock();
    ArUtil::sleep(1000);
    for (i=1; i<value; i++){
        robot.lock();
        v=vel;
        av=0;
        w=0;
        aw=0;
        theta_des_next = angleLimit(Ts*w+ref_Th);
        x_des_next = Ts*v*cos(ref_Th) + ref_xr;
        y_des_next = Ts*v*sin(ref_Th) + ref_yr;
        x_e = ((robot.getX()/1000)-
ref_xr)*cos(ref_Th)+((robot.getY()/1000)-ref_yr)*sin(ref_Th);
        y_e = -((robot.getX()/1000)-
ref_xr)*sin(ref_Th)+((robot.getY()/1000)-ref_yr)*cos(ref_Th);
        theta_e = angleLimit((robot.getTh()*(3.14/180))-ref_Th);
        if (y_e < 0)
        {
            gama_y = -abs(gama_y);
        }
        else if( y_e > 0 )
        {
            gama_y = abs( gama_y );
        }
        x_e_der = -v + (robot.getVel()/1000)*cos(theta_e)+y_e*w;
        y_e_der = (robot.getVel()/1000)*sin(theta_e)-x_e*w;
        theta_e_der = (robot.getRotVel()*3.14/180)-w;
        /* SLIDING SURFACE */
        s_1 = x_e_der + gama_x*x_e;
        s_2 = y_e_der + gama_y*y_e + gama0*sign(y_e)*(theta_e);
        // CAZUL II (B)
        temp1 = -Q1*satur(s_1/0.5) - (P1*s_1) - gama_x*x_e_der -
        (aw*y_e) - w*y_e_der + av;
        temp2 = -Q2*satur(s_2/0.5) - (P2*s_2) - gama_y*y_e_der +
        (aw*x_e) + w*x_e_der;
        v_c_der = (temp1 +
        (robot.getVel()/1000)*theta_e_der*sin(theta_e))/(cos(theta_e));
        v_c = (Ts*v_c_der) + ref_v;
        w_c = (temp2-
        v_c_der*sin(theta_e))/((robot.getVel()/1000)*cos(theta_e)+gama0*sign(y_e))+
        w;

        // ROBOT
        v_l = v_c + 0.19*w_c;    //0.19 //0.28 //0.24
        v_r = v_c - 0.19*w_c;
    }
}

```

```

        //      Simulator
        //      v_r = v_c + 0.19*w_c;
        //      v_l = v_c - 0.19*w_c;

        // setVel2 (double leftVelocity, double rightVelocity)
        robot.setVel2(1000*v_r, 1000*v_l);
        printf("\t %4.5f \t %4.5f \t %4.5f \t %4.5f \t %4.5f \n", x_e,
y_e, theta_e, v, w);
        fprintf(date6, "\n");
        fclose(date6);
        date6=fopen("C:/Program
Files/MobileRobots/Aria/bin/viteza_ciclu_complet.txt", "a+");
        fprintf(date6, "%4.4f \n", robot.getVel()/1000);
        fclose(date6);
        fprintf(date5, "%4.4f \n", x_e);
        fprintf(date4, "%4.4f \n", y_e);
        fprintf(date3, "%4.4f \n", theta_e);
        ref_xr = x_des_next;
        ref_yr = y_des_next;
        ref_Th = theta_des_next;
        ref_w = w_c;
        ref_v = v_c;
        timp=timp+Ts;
        robot.unlock();
        ArUtil::sleep(100);
    }
}
void ridicare()
{
        date6=fopen("C:/Program
Files/MobileRobots/Aria/bin/viteza_ciclu_complet.txt", "a+");
        fprintf(date6, "\n");
        fclose(date6);
        date6=fopen("C:/Program
Files/MobileRobots/Aria/bin/viteza_ciclu_complet.txt", "a+");
        fprintf(date6, "%4.4f \n", robot.getVel()/1000);
        fclose(date6);
        arm.moveTo(1, 65, 20);
        arm.moveTo(2, 105, 20);
        arm.moveTo(3, 0, 20);
        arm.moveTo(5, 0, 20);
        ArUtil::sleep (1000); //4000
    }
    void apucareunu()
    {
        date6=fopen("C:/Program
Files/MobileRobots/Aria/bin/viteza_ciclu_complet.txt", "a+");
        fprintf(date6, "\n");
        fclose(date6);
        date6=fopen("C:/Program
Files/MobileRobots/Aria/bin/viteza_ciclu_complet.txt", "a+");
        fprintf(date6, "%4.4f \n", robot.getVel()/1000);
        fclose(date6);
        arm.moveTo(1, 90, 20);
        arm.moveTo(2, 65, 20);
        arm.moveTo(3, -140, 20);
        arm.moveTo(5, 45, 20);
        ArUtil::sleep (5000); //5000
        arm.moveTo(6, 0, 20);
        ArUtil::sleep (2000); //2000
    }
}

```

```

void apucaretrei()
{
    date6=fopen("C:/Program
Files/MobileRobots/Aria/bin/viteza_ciclu_complet.txt","a+");
    fprintf(date6,"\n");
    fclose(date6);
    date6=fopen("C:/Program
Files/MobileRobots/Aria/bin/viteza_ciclu_complet.txt","a+");
    fprintf(date6,"%4.4f \n",robot.getVel()/1000);
    fclose(date6);
    arm.moveTo(1,70,20);
    arm.moveTo(2,30,20);
    arm.moveTo(3,-35,20);
    arm.moveTo(5,100,20);

    ArUtil::sleep (5000);//5000
    arm.moveTo(6, 0, 20);
    ArUtil::sleep (2000);//2000
}
void eliberare()
{
    date6=fopen("C:/Program
Files/MobileRobots/Aria/bin/viteza_ciclu_complet.txt","a+");
    fprintf(date6,"\n");
    fclose(date6);
    date6=fopen("C:/Program
Files/MobileRobots/Aria/bin/viteza_ciclu_complet.txt","a+");
    fprintf(date6,"%4.4f \n",robot.getVel()/1000);
    fclose(date6);
    arm.moveTo(1,80,20);
    arm.moveTo(2,55,20);
    arm.moveTo(3,0,20);
    arm.moveTo(5,45,20);
    ArUtil::sleep (3000);//4000
    arm.moveTo(6, 100, 20);
    ArUtil::sleep (1000);//2000
}
void eliberaredoi()
{
    date6=fopen("C:/Program
Files/MobileRobots/Aria/bin/viteza_ciclu_complet.txt","a+");
    fprintf(date6,"\n");
    fclose(date6);
    date6=fopen("C:/Program
Files/MobileRobots/Aria/bin/viteza_ciclu_complet.txt","a+");
    fprintf(date6,"%4.4f \n",robot.getVel()/1000);
    fclose(date6);
    arm.moveTo(1,85,20);
    arm.moveTo(2,55,20);
    arm.moveTo(3,0,20);
    arm.moveTo(5,45,20);
    ArUtil::sleep (3000);//4000
    arm.moveTo(6, 100, 20);
    ArUtil::sleep (1000);//2000
}
void eliberaretrei()
{
    date6=fopen("C:/Program
Files/MobileRobots/Aria/bin/viteza_ciclu_complet.txt","a+");
    fprintf(date6,"\n");
    fclose(date6);
    date6=fopen("C:/Program
Files/MobileRobots/Aria/bin/viteza_ciclu_complet.txt","a+");

```



```

        fprintf(date6,"%4.4f \n",robot.getVel()/1000);
        fclose(date6);
arm.moveTo(1,80,20);
arm.moveTo(2,60,20);
arm.moveTo(3,0,20);
arm.moveTo(5,45,20);
ArUtil::sleep (3000);//4000
arm.moveTo(6, 100, 20);
ArUtil::sleep (1000);//2000
}
int main(int argc, char **argv )
{
FILE *date1;
FILE *date2;
int start;
Aria::init();
ArArgumentParser parser(&argc, argv);
parser.loadDefaultArguments();
ArSimpleConnector simpleConnector(&parser);
ArAnalogGyro gyro(&robot);
robot.addRangeDevice(&sonar);
// Make a key handler, so that escape will shut down the program
// cleanly
ArKeyHandler keyHandler;
Aria::setKeyHandler(&keyHandler);
robot.attachKeyHandler(&keyHandler);
printf("You may press escape to exit\n");
// Collision avoidance actions at higher priority
ArActionLimiterForwards limiterAction("speed limiter near", 300, 600,
250);
ArActionLimiterForwards limiterFarAction("speed limiter far", 300, 11000,
400);
//ArActionLimiterForwards limiterFarAction("speed limiter far", 600,
2200, 800);
ArActionLimiterTableSensor tableLimiterAction;
robot.addAction(&tableLimiterAction, 100);
robot.addAction(&limiterAction, 95);
robot.addAction(&limiterFarAction, 90);
// Goto action at lower priority
ArActionGoto gotoPoseAction("goto");
robot.addAction(&gotoPoseAction, 50);
ArSimpleConnector con(&argc, argv);
if(!Aria::parseArgs())
{
    Aria::logOptions();
    Aria::shutdown();
    return 1;
}
//trebuie necomentata cand se lucreaza cu bratul
ArLog::log(ArLog::Normal, "armExample: Connecting to the robot.");
if(!con.connectRobot(&robot))
{
    ArLog::log(ArLog::Terse, "armExample: Could not connect to the robot.
Exiting.");
    Aria::shutdown();
    return 1;
}
robot.runAsync(true);
// turn on the motors, turn off amigobot sounds
robot.enableMotors();
// turn off sonar
//robot.comInt(28, 0);
// Set up and initialize the arm

```

```

arm.setRobot(&robot);
if (arm.init() != ArP2Arm::SUCCESS)
{
    ArLog::log(ArLog::Terse, "armExample: Error initializing the P2 Arm!");
    return 1;
}
// Print out some of the settings
P2ArmJoint *joint;
printf("Current joint info:\nJoint    Vel    Home    Center\n");
for (int i=1; i<=ArP2Arm::NumJoints; i++)
{
    joint = arm.getJoint(i);
    printf("  %2i:  %5i %5i  %5i\n", i, joint->myVel, joint->myHome,
joint->myCenter);
}
printf("\n");
// Put the arm to work
printf("Powering on (takes a couple seconds to stabilize)\n");
arm.powerOn();
// Request one status packet and print out the arm's status
printf("Current arm status:\n");
arm.requestStatus(ArP2Arm::StatusSingle);
ArUtil::sleep(200); // Give time to get the packet
printf("Arm Status: ");
if (arm.getStatus() & ArP2Arm::ArmGood)
    printf("Good=1 ");
else
    printf("Good=0 ");
if (arm.getStatus() & ArP2Arm::ArmInited)
    printf("Inited=1 ");
else
    printf("Inited=0 ");
if (arm.getStatus() & ArP2Arm::ArmPower)
    printf("Power=1 ");
else
    printf("Power=0 ");
if (arm.getStatus() & ArP2Arm::ArmHoming)
    printf("Homing=1 ");
else
    printf("Homing=0 ");
printf("\n\n");
    arm.moveTo(1,65,20);
    arm.moveTo(2,95,20);
    arm.moveTo(3,0,20);
    arm.moveTo(5,0,20);
    ArUtil::sleep (20500); //3000
timp=0.0;
Ts=0.1;
Q1 = 0.05;           //0.05           0.5
Q2 = 0.5;             //0.5           0.75
P1 = 0.5;             //0.5           1.75
P2 = 0.75;            //1.75           0.75
alpha = 0.5;
gama0 = 30;           //20
gama_x = 0.75;         //1.25
gama_y = 25;           //15
ref_xr = 0.0;
ref_yr = 0.0;
ref_Th = 0.0;
ArUtil::sleep (200);
    Delay();
{ int p,m;
{datel=fopen("C:/Program Files/MobileRobots/Aria/bin/a1.txt","w+");

```

```

date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","w+");
apucareunu();
p=1;m=1;
fprintf(date1,"%d \n",p);
fprintf(date2,"%d \n",m);
fclose(date1);
fclose(date2);
}
ArUtil::sleep (250);//1000
{
    date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a2.txt","w+");
    date2=fopen("C:/Program
Files/MobileRobots/Aria/bin/b1.txt","a+");
    ridicare();
    p=0;m=0;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}
ArUtil::sleep (250);//1000
{
    date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a3.txt","w+");
    date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
    sliding(100,0.1);
    p=0;m=0;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}
robot.setVel2(0,0);
ArUtil::sleep (100);
{
    date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a4.txt","w+");
    date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
    eliberare();
    p=1;m=1;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}
{
    date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a5.txt","w+");
    date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
    ridicare();
    p=0;m=0;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}
ArUtil::sleep (250);//2000
{
    date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a6.txt","w+");
    date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
    sliding(65,-0.1);
    p=0;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}
robot.setVel2(0,0);
ArUtil::sleep (100);
{ date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a7.txt","w+");

```

```

    date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
    apucareunu();
    p=1;m=1;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}
{
    date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a8.txt","w+");
    date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
    ridicare();
    p=0;m=0;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}
ArUtil::sleep (250); //2000
{
    date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a9.txt","w+");
    date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
    sliding(68,0.1);
    p=0;m=0;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}
robot.setVel2(0,0);
ArUtil::sleep (100);
{
    date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a10.txt","w+");
    date2=fopen("C:/Program
Files/MobileRobots/Aria/bin/b1.txt","a+");
    eliberaredoi();
    p=1;m=1;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}
{
    date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a11.txt","w+");
    date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
    ridicare();
    p=0;m=0;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}
ArUtil::sleep (250); //2000
{
    date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a12.txt","w+");
    date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
    sliding(33,0.1);
    p=0;m=0;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}
robot.setVel2(0,0);
{
    date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a13.txt","w+");
    date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
    apucaretrei();
    p=1;m=1;

```

```

        fprintf(date1,"%d \n",p);
        fprintf(date2,"%d \n",m);
        fclose(date1);
        fclose(date2);
    }
    {
        date1=fopen("C:/Program
Files/MobileRobots/Aria/bin/a14.txt","w+");
        date2=fopen("C:/Program
Files/MobileRobots/Aria/bin/b1.txt","a+");
        ridicare();
        p=0;m=0;
        fprintf(date1,"%d \n",p);
        fprintf(date2,"%d \n",m);
        fclose(date1);
        fclose(date2);
    }
    ArUtil::sleep(250); //2000
    {
        date1=fopen("C:/Program
Files/MobileRobots/Aria/bin/a15.txt","w+");
        date2=fopen("C:/Program
Files/MobileRobots/Aria/bin/b1.txt","a+");
        sliding(36,0.1);
        p=0;m=0;
        fprintf(date1,"%d \n",p);
        fprintf(date2,"%d \n",m);
        fclose(date1);
        fclose(date2);
    }
    robot.setVel2(0,0);
    {
        date1=fopen("C:/Program
Files/MobileRobots/Aria/bin/a16.txt","w+");
        date2=fopen("C:/Program
Files/MobileRobots/Aria/bin/b1.txt","a+");
        eliberaretrei();
        p=1;m=1;
        fprintf(date1,"%d \n",p);
        fprintf(date2,"%d \n",m);
        fclose(date1);
        fclose(date2);
    }
    {
        date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a17.txt","w+");
        date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
        ridicare();
        p=0;m=0;
        fprintf(date1,"%d \n",p);
        fprintf(date2,"%d \n",m);
        fclose(date1);
        fclose(date2);
    }
    ArUtil::sleep(250); //2000
    {
        date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a18.txt","w+");
        date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
        sliding(43,0.1);
        p=0;m=0;
        fprintf(date1,"%d \n",p);
        fprintf(date2,"%d \n",m);
        fclose(date1);
        fclose(date2);
    }
    robot.setVel2(0,0);
    {
        date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a19.txt","w+");
        date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
        apucaretrei();

```

```

        p=1;m=1;
        fprintf(date1,"%d \n",p);
        fprintf(date2,"%d \n",p);
        fclose(date1);
        fclose(date2);
    }
    { date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a20.txt","w+");
      date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
      ridicare();
      p=0;m=0;
      fprintf(date1,"%d \n",p);
      fprintf(date2,"%d \n",m);
      fclose(date1);
      fclose(date2);
    }
    ArUtil::sleep(250); //2000
    { date1=fopen("C:/Program
Files/MobileRobots/Aria/bin/a21.txt","w+");
      date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
      sliding(39,0.1);
      p=0;m=0;
      fprintf(date1,"%d \n",p);
      fprintf(date2,"%d \n",m);
      fclose(date1);
      fclose(date2);
    }
    robot.setVel2(0,0);
    { date1=fopen("C:/Program
Files/MobileRobots/Aria/bin/a22.txt","w+");
      date2=fopen("C:/Program
Files/MobileRobots/Aria/bin/b1.txt","a+");
      eliberaretrei();
      p=1;m=1;
      fprintf(date1,"%d \n",p);
      fprintf(date2,"%d \n",m);
      fclose(date1);
      fclose(date2);
    }
    { date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a23.txt","w+");
      date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
      ridicare();
      p=0;m=0;
      fprintf(date1,"%d \n",p);
      fprintf(date2,"%d \n",m);
      fclose(date1);
      fclose(date2);
    }
    ArUtil::sleep(250); //2000
    { date1=fopen("C:/Program
Files/MobileRobots/Aria/bin/a24.txt","w+");
      date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
      sliding(36,0.1);
      p=0;m=0;
      fprintf(date1,"%d \n",p);
      fprintf(date2,"%d \n",m);
      fclose(date1);
      fclose(date2);
    }
    robot.setVel2(0,0);
    { date1=fopen("C:/Program
Files/MobileRobots/Aria/bin/a25.txt","w+");
      date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
      apucaretrei();

```

```

        p=1;m=1;
        fprintf(date1,"%d \n",p);
        fprintf(date2,"%d \n",m);
        fclose(date1);
        fclose(date2);
    }
    { date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a26.txt","w+");
      date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
      ridicare();
      p=0;m=0;
      fprintf(date1,"%d \n",p);
      fprintf(date2,"%d \n",m);
      fclose(date1);
      fclose(date2);
    }
    ArUtil::sleep(250);//2000
    { date1=fopen("C:/Program
Files/MobileRobots/Aria/bin/a27.txt","w+");
      date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
      sliding(35,0.1);
      p=0;m=0;
      fprintf(date1,"%d \n",p);
      fprintf(date2,"%d \n",m);
      fclose(date1);
      fclose(date2);
    }
    robot.setVel2(0,0);
    { date1=fopen("C:/Program
Files/MobileRobots/Aria/bin/a28.txt","w+");
      date2=fopen("C:/Program
Files/MobileRobots/Aria/bin/b1.txt","a+");
      eliberaretrei();
      p=1;m=1;
      fprintf(date1,"%d \n",p);
      fprintf(date2,"%d \n",m);
      fclose(date1);
      fclose(date2);
    }
    { date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a29.txt","w+");
      date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
      ridicare();
      p=0;m=0;
      fprintf(date1,"%d \n",p);
      fprintf(date2,"%d \n",m);
      fclose(date1);
      fclose(date2);
    }
    ArUtil::sleep(250);//2000
    { date1=fopen("C:/Program Files/MobileRobots/Aria/bin/a30.txt","w+");
      date2=fopen("C:/Program Files/MobileRobots/Aria/bin/b1.txt","a+");
      sliding(314,-0.1);
      p=0;m=0;
      fprintf(date1,"%d \n",p);
      fprintf(date2,"%d \n",m);
      fclose(date1);
      fclose(date2);
    }
    robot.setVel2(0,0);
    ArUtil::sleep (100);
    ArUtil::sleep (750);//3000
}

Aria::shutdown();

```

```

    return 0;
}

```

ANEXA B: Programul de conducere sliding mode al robotului Pioneer P3-DX echipat cu manipulator Pioneer 5-DOF Arm în cadrul procesului de deservire al liniei flexibile FESTO MPS-200

```

#include "Aria.h"
#include "ArActionDesired.h"
#include <iostream>
#include <math.h>
#include <conio.h>
#include <stdio.h>
// #include "clock.cpp"
double v, av, w, aw;           /* velocity to drive left wheel*/
double theta_des_next;         /* theta desired */
double ref_xr, ref_yr, ref_Th, ref_w, ref_v, x_des_next, y_des_next; /*
desired position */
double x_e, y_e, theta_e, x_e_der, y_e_der, theta_e_der;
/* errors */
double s_1, s_2, Q1, Q2, P1, P2, gama0, gama_x, gama_y;
double v_c, v_c_der, w_c, temp1, temp2, V_R;
double Ts, v_l, v_r, timp, alpha, epsilon1, epsilon2, q1, q2;
double x, y, theta;
int i, n, j;
FILE *dateplot;
ArRobot robot;
ArSonarDevice sonar;
ArP2Arm arm;
void Delay() {
    // Make the changes visible
    long lngDelay;
    for(lngDelay = 0; lngDelay < 100000; lngDelay++) {
    }
}
static double angleLimit(double angle)
{
    if (angle > 3.14)
        angle = angle - 2*3.14;
    else
        if (angle < -3.14)
            angle = angle + 2*3.14;
    return(angle);
}
/* ----- SIGN & SATURATION FUNCTION ----- */
double sign(double value){
    if (value<0)
        return(-1.00);
    else
        if (value>0)
            return(1.00);
        else
            return(1.00);
}
double satur(double value){
    if (value<-1)
        return (-1.00);
    else if (value > 1)
        return (1.00);
    else
        return (value);
}

```



```

}
FILE *date6;
FILE *date5;
FILE *date4;
FILE *date3;
void sliding(int value, double vel, double omega)
{
date6=fopen("D:/Pioneerviteza_ciclu_complet.txt","a+");
date5=fopen("D:/Pioneereroare_x_e.txt","w+");
date4=fopen("D:/Pioneereroare_y_e.txt","w+");
date3=fopen("D:/Pioneereroare_theta_e.txt","w+");
ref_v = 0.0;
ref_w = 0.0;
robot.lock();
ArUtil::sleep(100);
robot.unlock();
ArUtil::sleep(1000);
for (i=1; i<value; i++){
robot.lock();
v=vel;
av=0;
w=omega;
aw=0;
theta_des_next = angleLimit(Ts*w+ref_Th);
x_des_next = Ts*v*cos(ref_Th) + ref_xr;
y_des_next = Ts*v*sin(ref_Th) + ref_yr;
x_e = ((robot.getX()/1000)-
ref_xr)*cos(ref_Th)+((robot.getY()/1000)-ref_yr)*sin(ref_Th);
y_e = -((robot.getX()/1000)-
ref_xr)*sin(ref_Th)+((robot.getY()/1000)-ref_yr)*cos(ref_Th);
theta_e = angleLimit((robot.getTh()*(3.14/180))-ref_Th);
if (y_e < 0)
{
gama_y = -abs(gama_y);
}
else if( y_e > 0 )
{
gama_y = abs( gama_y );
}
x_e_der = -v + (robot.getVel()/1000)*cos(theta_e)+y_e*w;
y_e_der = (robot.getVel()/1000)*sin(theta_e)-x_e*w;
theta_e_der = (robot.getRotVel()*3.14/180)-w;
/* SLIDING SURFACE */
s_1 = x_e_der + gama_x*x_e;
s_2 = y_e_der + gama_y*y_e + gama0*sign(y_e)*(theta_e);
// CAZUL II (B)
temp1 = -Q1*satur(s_1/0.5) - (P1*s_1) - gama_x*x_e_der -
(aw*y_e) - w*y_e_der + av;
temp2 = -Q2*satur(s_2/0.5) - (P2*s_2) - gama_y*y_e_der +
(aw*x_e) + w*x_e_der;
v_c_der = (temp1 +
(robot.getVel()/1000)*theta_e_der*sin(theta_e))/(cos(theta_e));
v_c = (Ts*v_c_der) + ref_v;
w_c = (temp2-
v_c_der*sin(theta_e))/( (robot.getVel()/1000)*cos(theta_e)+gama0*sign(y_e))+
w;

// ROBOT
v_l = v_c + 0.19*w_c; //0.19 //0.28 //0.24
v_r = v_c - 0.19*w_c;
// Simulator
// v_r = v_c + 0.19*w_c;
// v_l = v_c - 0.19*w_c;

```

```

        // setVel2 (double leftVelocity, double rightVelocity)
        robot.setVel2(1000*v_r, 1000*v_l);
        fprintf(dateplot, " %4.5f \t %4.5f \t %4.5f \t %4.5f \t %4.5f \t %4.5f \t
        %4.5f \t %4.5f \t %4.5f \t %4.5f \t %4.5f \t %4.5f \t %4.5f \t %4.5f \t
        %4.5f \t %4.5f \t %4.5f \t %4.5f \t %4.5f \t %4.5f \t %4.5f \n",
        robot.getX()/1000, robot.getY()/1000, robot.getTh()*(3.14/180), ref_xr,
        ref_yr, ref_Th, x_e, y_e, theta_e, x_e_der, y_e_der, theta_e_der,
        (robot.getVel()/100), (robot.getRotVel()*3.14/180), v, w, v_c, w_c, s_1,
        s_2, timp);
        printf("\t %4.5f \t %4.5f \t %4.5f \t %4.5f \t %4.5f \n", x_e,
        y_e, theta_e, v, w);
        fprintf(date6, "\n");
        fclose(date6);
        date6=fopen("D:/Pioneerviteza_ciclu_complet.txt", "a+");
        fprintf(date6, "%4.4f \n", robot.getVel()/1000);
        fclose(date6);
        fprintf(date5, "%4.4f \n", x_e);
        fprintf(date4, "%4.4f \n", y_e);
        fprintf(date3, "%4.4f \n", theta_e);
        ref_xr = x_des_next;
        ref_yr = y_des_next;
        ref_Th = theta_des_next;
        ref_w = w_c;
        ref_v = v_c;
        timp=timp+Ts;
        robot.unlock();
        ArUtil::sleep(100);
    }
}
void ridicare()
{
        date6=fopen("D:/Pioneerviteza_ciclu_complet.txt", "a+");
        fprintf(date6, "\n");
        fclose(date6);
        date6=fopen("D:/Pioneerviteza_ciclu_complet.txt", "a+");
        fprintf(date6, "%4.4f \n", robot.getVel()/1000);
        fclose(date6);
        // arm.moveTo(1,65,20);
        arm.moveTo(2,105,20);
        arm.moveTo(3,0,20);
        arm.moveTo(5,0,20);
        ArUtil::sleep (1000);//4000
}
void apucareunu()
{
        date6=fopen("D:/Pioneerviteza_ciclu_complet.txt", "a+");
        fprintf(date6, "\n");
        fclose(date6);
        date6=fopen("D:/Pioneerviteza_ciclu_complet.txt", "a+");
        fprintf(date6, "%4.4f \n", robot.getVel()/1000);
        fclose(date6);
        arm.moveTo(1,75,20);
        arm.moveTo(2,30,20);
        arm.moveTo(3,120,20);
        //arm.moveTo(3,-125,20);

        arm.moveTo(5,65,20);
        ArUtil::sleep (5000); //5000
        arm.moveTo(6, 0, 20);
        ArUtil::sleep (2000); //2000
}
void apucaretrei()
{

```

```

        date6=fopen("D:/Pioneerviteza_ciclu_complet.txt","a+");
        fprintf(date6,"\n");
        fclose(date6);
        date6=fopen("D:/Pioneerviteza_ciclu_complet.txt","a+");
        fprintf(date6,"%4.4f \n",robot.getVel()/1000);
        fclose(date6);
//    arm.moveTo(1,70,20);
        arm.moveTo(2,30,20);
        arm.moveTo(3,-35,20);
        arm.moveTo(5,100,20);

        ArUtil::sleep (5000);//5000
        arm.moveTo(6, 0, 20);
        ArUtil::sleep (2000);//2000
    }
void eliberare()
{
    date6=fopen("D:/Pioneerviteza_ciclu_complet.txt","a+");
    fprintf(date6,"\n");
    fclose(date6);
    date6=fopen("D:/Pioneerviteza_ciclu_complet.txt","a+");
    fprintf(date6,"%4.4f \n",robot.getVel()/1000);
    fclose(date6);
    arm.moveTo(1,50,20);
    arm.moveTo(2,55,20);
    arm.moveTo(3,0,20);
    arm.moveTo(5,45,20);
    ArUtil::sleep (5000);//4000
    arm.moveTo(6, 100, 20);
    ArUtil::sleep (1000);//2000
}

void eliberaredoi()
{
    date6=fopen("D:/Pioneerviteza_ciclu_complet.txt","a+");
    fprintf(date6,"\n");
    fclose(date6);
    date6=fopen("D:/Pioneerviteza_ciclu_complet.txt","a+");
    fprintf(date6,"%4.4f \n",robot.getVel()/1000);
    fclose(date6);
//    arm.moveTo(1,85,20);
    arm.moveTo(2,55,20);
    arm.moveTo(3,0,20);
    arm.moveTo(5,45,20);
    ArUtil::sleep (3000);//4000
    arm.moveTo(6, 100, 20);
    ArUtil::sleep (1000);//2000
}

void eliberaretrei()
{
    date6=fopen("D:/Pioneerviteza_ciclu_complet.txt","a+");
    fprintf(date6,"\n");
    fclose(date6);
    date6=fopen("D:/Pioneerviteza_ciclu_complet.txt","a+");
    fprintf(date6,"%4.4f \n",robot.getVel()/1000);
    fclose(date6);
//    arm.moveTo(1,80,20);
    arm.moveTo(2,60,20);
    arm.moveTo(3,0,20);
    arm.moveTo(5,45,20);
    ArUtil::sleep (3000);//4000
    arm.moveTo(6, 100, 20);

```

```

ArUtil::sleep (1000); //2000

}
int main(int argc, char **argv )
{
FILE *date1;
FILE *date2;
int start;
Aria::init();
ArArgumentParser parser(&argc, argv);
parser.loadDefaultArguments();
ArSimpleConnector simpleConnector(&parser);
ArAnalogGyro gyro(&robot);
robot.addRangeDevice(&sonar);
// Make a key handler, so that escape will shut down the program
// cleanly
ArKeyHandler keyHandler;
Aria::setKeyHandler(&keyHandler);
robot.attachKeyHandler(&keyHandler);
printf("You may press escape to exit\n");
// Collision avoidance actions at higher priority
ArActionLimiterForwards limiterAction("speed limiter near", 300, 600,
250);
ArActionLimiterForwards limiterFarAction("speed limiter far", 300, 11000,
400);
//ArActionLimiterForwards limiterFarAction("speed limiter far", 600,
2200, 800);
ArActionLimiterTableSensor tableLimiterAction;
robot.addAction(&tableLimiterAction, 100);
robot.addAction(&limiterAction, 95);
robot.addAction(&limiterFarAction, 90);
// Goto action at lower priority
ArActionGoto gotoPoseAction("goto");
robot.addAction(&gotoPoseAction, 50);
ArSimpleConnector con(&argc, argv);
if(!Aria::parseArgs())
{
Aria::logOptions();
Aria::shutdown();
return 1;
}
ArLog::log(ArLog::Normal, "armExample: Connecting to the robot.");
if(!con.connectRobot(&robot))
{
ArLog::log(ArLog::Terse, "armExample: Could not connect to the robot.
Exiting.");
Aria::shutdown();
return 1;
}
robot.runAsync(true);
arm.setRobot(&robot);
if (arm.init() != ArP2Arm::SUCCESS)
{
ArLog::log(ArLog::Terse, "armExample: Error initializing the P2 Arm!");
return 1;
}
// Print out some of the settings
P2ArmJoint *joint;
printf("Current joint info:\nJoint Vel Home Center\n");
for (int i=1; i<=ArP2Arm::NumJoints; i++)
{
joint = arm.getJoint(i);

```

```

    printf(" %2i: %5i %5i %5i\n", i, joint->myVel, joint->myHome,
joint->myCenter);
}
printf("\n");
// Put the arm to work
printf("Powering on (takes a couple seconds to stabilize)\n");
arm.powerOn();
// Request one status packet and print out the arm's status
printf("Current arm status:\n");
arm.requestStatus(ArP2Arm::StatusSingle);
ArUtil::sleep(200); // Give time to get the packet
printf("Arm Status: ");
if (arm.getStatus() & ArP2Arm::ArmGood)
    printf("Good=1 ");
else
    printf("Good=0 ");
if (arm.getStatus() & ArP2Arm::ArmInited)
    printf("Inited=1 ");
else
    printf("Inited=0 ");
if (arm.getStatus() & ArP2Arm::ArmPower)
    printf("Power=1 ");
else
    printf("Power=0 ");
if (arm.getStatus() & ArP2Arm::ArmHoming)
    printf("Homing=1 ");
else
    printf("Homing=0 ");
printf("\n\n");
    arm.moveTo(1,65,20);
    arm.moveTo(2,95,20);
    arm.moveTo(3,0,20);
    arm.moveTo(5,0,20);
    ArUtil::sleep (3000);//3000
timp=0.0;
Ts=0.1;
Q1 = 0.05;           //0.05           0.5
Q2 = 0.5;             //0.5           0.75
P1 = 0.5;             //0.5           1.75
P2 = 0.75;            //1.75           0.75
alpha = 0.5;
gama0 = 30;           //20
gama_x = 0.75;         //1.25
gama_y = 25;           //15
ref_xr = 0.0;
ref_yr = 0.0;
ref_Th = 0.0;
    dateplot=fopen("d:/Pioneer/test_d3.txt","w+");
ArUtil::sleep (200);
    Delay();

// Prima ridicare
{ int p,m;
{date1=fopen("D:/Pioneera1.txt","w+");
date2=fopen("D:/Pioneerb1.txt","w+");
apucareunu();
p=1;m=1;
fprintf(date1,"%d \n",p);
fprintf(date2,"%d \n",m);
fclose(date1);
fclose(date2);
}
ArUtil::sleep (250);//1000
{ date1=fopen("D:/Pioneera2.txt","w+");

```

```

        date2=fopen("D:/Pioneerbl.txt","a+");
        ridicare();
        p=0;m=0;
        fprintf(date1,"%d \n",p);
        fprintf(date2,"%d \n",m);
        fclose(date1);
        fclose(date2);
    }
    ArUtil::sleep (250); //1000
    { // Deplasare inainte 70 cm
        date1=fopen("D:/Pioneera3.txt","w+");
        date2=fopen("D:/Pioneerbl.txt","a+");
        // sliding(100,0.1);
        sliding(80,0.1,0);
        p=0;m=0;
        fprintf(date1,"%d \n",p);
        fprintf(date2,"%d \n",m);
        fclose(date1);
        fclose(date2);
    }

    robot.setVel2(0,0);
    ArUtil::sleep (100);

        //Intoarcere
    {
        date1=fopen("D:/Pioneera6.txt","w+");
        date2=fopen("D:/Pioneerbl.txt","a+");
        //sliding(65,-0.1);

        sliding(155, 0, -0.1);
        p=0;
        fprintf(date1,"%d \n",p);
        fprintf(date2,"%d \n",m);
        fclose(date1);
        fclose(date2);
    }

    robot.setVel2(0,0);
    ArUtil::sleep (100);
    {
        date1=fopen("D:/Pioneera9.txt","w+");
        date2=fopen("D:/Pioneerbl.txt","a+");
        //sliding(68,0.1);
        sliding(190,0.1, 0);
        p=0;m=0;
        fprintf(date1,"%d \n",p);
        fprintf(date2,"%d \n",m);
        fclose(date1);
        fclose(date2);
    }

    robot.setVel2(0,0);
    ArUtil::sleep (100);
    ArUtil::sleep (5000); //5000
    arm.moveTo(6, 0, 20);
    // Intoarcere
    {
        date1=fopen("D:/Pioneera12.txt","w+");
        date2=fopen("D:/Pioneerbl.txt","a+");
        // sliding(33,0.1);
        //////////////
        ArUtil::sleep (5000); //5000

        sliding(155, 0, -0.1);
        p=0;m=0;
        fprintf(date1,"%d \n",p);
        fprintf(date2,"%d \n",m);
        fclose(date1);
        fclose(date2);
    }
    arm.moveTo(6, 0, 20);

```

```

    }
    robot.setVel2(0,0);
    ArUtil::sleep(5000); //5000
    arm.moveTo(6, 0, 20);
{
    date1=fopen("D:/Pioneer15.txt","w+");
    date2=fopen("D:/Pioneerbl.txt","a+");
    // sliding(36,0.1);
    sliding(40,0.1,0);
    p=0;m=0;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}

robot.setVel2(0,0);
ArUtil::sleep(2050); //2000
{
    date1=fopen("D:/Pioneer16.txt","w+");
    date2=fopen("D:/Pioneerbl.txt","a+");
    eliberare();
    p=1;m=1;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}
{
    date1=fopen("D:/Pioneer17.txt","w+");
    date2=fopen("D:/Pioneerbl.txt","a+");
    ridicare();
    p=0;m=0;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}
ArUtil::sleep(250); //2000
{
    date1=fopen("D:/Pioneer18.txt","w+");
    date2=fopen("D:/Pioneerbl.txt","a+");
    sliding(40,-0.1, 0);
    p=0;m=0;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}

robot.setVel2(0,0);
{
    date1=fopen("D:/Pioneer21.txt","w+");
    date2=fopen("D:/Pioneerbl.txt","a+");
    sliding(157,0,0.1);
    p=0;m=0;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}

robot.setVel2(0,0);
{
    date1=fopen("D:/Pioneer24.txt","w+");
    date2=fopen("D:/Pioneerbl.txt","a+");
    sliding(176,-0.1, 0);
    p=0;m=0;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}

```

```

    }
    robot.setVel2(0,0);
    ArUtil::sleep(250); //2000
    {
        date1=fopen("D:/Pioneera27.txt","w+");
        date2=fopen("D:/Pioneerb1.txt","a+");
        sliding(158,0,0.1);
        p=0;m=0;
        fprintf(date1,"%d \n",p);
        fprintf(date2,"%d \n",m);
        fclose(date1);
        fclose(date2);
    }
    robot.setVel2(0,0);

{
    date1=fopen("D:/Pioneera30.txt","w+");
    date2=fopen("D:/Pioneerb1.txt","a+");
    sliding(73,-0.1,0);
    p=0;m=0;
    fprintf(date1,"%d \n",p);
    fprintf(date2,"%d \n",m);
    fclose(date1);
    fclose(date2);
}

    robot.setVel2(0,0);
    ArUtil::sleep (100);
    ArUtil::sleep (750); //3000
}

    fclose(dateplot);
    Aria::shutdown();

    return 0;
}

```

ANEXA C: Programul de conducere în timp real al liniei flexibile HERA & Horstmann din cadrul procesului de asamblare și dezasamblare totală a unui produs, dezasamblare deservită de robotul mobil Pioneer P3-DX echipat cu manipulator Pioneer 5-DOF Arm implementat în automatul programabil SIEMENS Simatic S7-300

FC1 - <offline>

"Montage_Schrittkette"

Version: 0.1

Block version: 2

Time stamp Code: Interface:

06/26/2012 04:33:04 PM

01/22/2008 05:00:29 PM

Lengths (block/logic/data): 01982 01648 00002

Block: FC1 Montaj

Network: 1 Initializare

AN	"M_SKM_Prod_Warn_PS"	M200.7
		M152.6
AN	"M_LT_voll"	M40.1
A	"M_Start_Montage"	M200.0
AN	"M_SKM_Prod_Fehler_PS"	M152.0
FP	"M_SK_Montage_Flanke_pos1"	M205.0
=	L 1.0	
A	L 1.0	
BLD	102	
S	"M_Schritt_00"	M210.0
	A L 1.0	
	BLD 102	
R	"M_SK_Montage_ENDE"	M200.3

	A	L	1.0		
		BLD	102		
	S	"M_SK_Montage_T_u_Z_reset"	M205.1		
		A	L	1.0	
		BLD	102		
	R	"M_Start_Montage"	M200.0		
		A	L	1.0	
		BLD	102		
	S	"M_FB_Lagerturm_enable"	M205.2		
		A	L	1.0	
L		0			
T		"Piesa ce va fi demontata"	MW41		
NOP		0			
A		L	1.0		
BLD		102			
R		T	60		
A		L	1.0		
BLD		102			
A		L	1.0		
BLD		102			
R		"M_Schritt			
A		L	1.0		
BLD		102			
R		"M_Schritt			
A		L	1.0		
BLD		102			
	R	"M_Schritt_02"	M210.2		
A		L	1.0		
BLD		102			
	R	"M_Schritt_02"	M210.2		
A		L	1.0		
BLD		102			
	R	"M_Schritt_03"	M210.3		
A		L	1.0		
BLD		102			
	R	"M_Schritt_04"	M210.4		
A		L	1.0		
BLD		102			
	R	"M_Schritt_05"	M210.5		
A		L	1.0		
BLD		102			
	R	"M_Schritt_06"	M210.6		
A		L	1.0		
BLD		102			
	R	"M_Schritt_07"	M210.7		
A		L	1.0		
BLD		102			
	R	"M_Schritt_08"	M211.0		
A		L	1.0		
BLD		102			
	R	"M_Schritt_09"	M211.1		
A		L	1.0		
BLD		102			
	R	"M_Schritt_10"	M211.2		
A		L	1.0		
BLD		102			
	R	"M_Schritt_11"	M211.3		
A		L	1.0		
BLD		102			
	R	"M_Schritt_12"	M211.4		
A		L	1.0		
BLD		102			
	R	"M_Schritt_13"	M211.5		
A		L	1.0		
BLD		102			
	R	"M_Schritt_14"	M211.6		
A		L	1.0		
BLD		102			
	R	"M_Schritt_15"	M211.7		
A		L	1.0		
BLD		102			
	R	"M_Schritt_16"	M212.0		
A		L	1.0		
BLD		102			
	R	"M_Schritt_17"	M212.1		

A	L			1.0	
BLD	102				
		R	"M_Schritt_18"	1.0	M212.2
A	L				
BLD	102				
		R	"M_Schritt_19"	1.0	M212.3
A	L				
BLD	102				
		R	"M_Schritt_20"	1.0	M212.4
A	L				
BLD	102				
		R	"M_Schritt_21"	1.0	M212.5
A	L				
BLD	102				
		R	"M_Schritt_22"	1.0	M212.6
A	L				
BLD	102				
		R	"M_Schritt_23"	1.0	M212.7
A	L				
BLD	102				
		R	"M_Schritt_24"	1.0	M213.0
A	L				
BLD	102				
		R	"M_Schritt_25"	1.0	M213.1
A	L				
BLD	102				
		R	"M_Schritt_26"	1.0	M213.2
A	L				
BLD	102				
		R	"M_Schritt_27"	1.0	M213.3
A	L				
BLD	102				
		R	"M_Schritt_28"	1.0	M213.4
		A	L	1.0	
			BLD	102	
		R	"M_Schritt_29"	1.0	M213.5
			A	L	
			BLD	102	
		R	"M_Schritt_30"	1.0	M213.6
			A	L	
			BLD	102	
		R	"M_Schritt_31"	1.0	M213.7
			A	L	
			BLD	102	
		R	"M_Schritt_32"	1.0	M214.0
			A	L	
			BLD	102	
		R	"M_Schritt_33"	1.0	M214.1
			A	L	
			BLD	102	
		R	"M_Schritt_34"	1.0	M214.2
			A	L	
			BLD	102	
		R	"M_Schritt_35"	1.0	M214.3
			A	L	
			BLD	102	
		R	"M_Schritt_36"	1.0	M214.4
			A	L	
			BLD	102	
		R	"M_Schritt_37"	1.0	M214.5
			A	L	
			BLD	102	
		R	"M_Schritt_38"	1.0	M214.6
			A	L	
			BLD	102	
		R	"M_Schritt_39"	1.0	M214.7
			A	L	
			BLD	102	
		R	"M_Schritt_40"	1.0	M215.0
			A	L	
			BLD	102	
		R	"M_Schritt_41"	1.0	M215.1
			A	L	
			BLD	102	
		R	"M_Schritt_42"	1.0	M215.2
			A	L	
			BLD	102	
		R	"M_Schritt_43"	1.0	M215.3
			A	L	

		BLD	102	
R	"M_Schritt_44"			M215.4
	A	L	1.0	
		BLD	102	
R	"M_Schritt_45"			M215.5
	A	L	1.0	
		BLD	102	
R	"M_Schritt_46"			M215.6
	A	L	1.0	
		BLD	102	
R	"M_Schritt_47"			M215.7
	A	L	1.0	
		BLD	102	
R	"M_Schritt_48"			M216.0
	A	L	1.0	
		BLD	102	
R	"M_Schritt_49"			M216.1
	A	L	1.0	
		BLD	102	
R	"M_Schritt_50"			M216.2
	A	L	1.0	
		BLD	102	
R	"M_Schritt_51"			M216.3
	A	L	1.0	
		BLD	102	
R	"M_Schritt_52"			M216.4
	A	L	1.0	
		BLD	102	
R	"M_Schritt_53"			M216.5
	A	L	1.0	
		BLD	102	
R	"M_Schritt_54"			M216.6
	A	L	1.0	
		BLD	102	
R	"M_Schritt_55"			M216.7
	A	L	1.0	
		BLD	102	

A	L	1.0
BLD	102	

R	"M_Schritt_56"			M217.0
R	"M_Schritt_57"			M217.1
	A	L	1.0	
		BLD	102	
R	"M_Schritt_58"			M217.2
	A	L	1.0	
		BLD	102	
R	"M_Schritt_59"			M217.3
	A	L	1.0	
		BLD	102	
R	"M_Schritt_60"			M217.4
	A	L	1.0	
		BLD	102	
R	"M_Schritt_61"			M217.5
	A	L	1.0	
		BLD	102	
R	"M_Schritt_62"			M217.6
	A	L	1.0	
		BLD	102	
R	"M_Schritt_63"			M217.7
	A	L	1.0	
		BLD	102	
R	"M_Schritt_64"			M218.0
	A	L	1.0	
		BLD	102	
R	"M_Schritt_65"			M218.1
	A	L	1.0	
		BLD	102	
R	"M_Schritt_66"			M218.2
	A	L	1.0	
		BLD	102	
R	"M_Schritt_67"			M218.3
	A	L	1.0	
		BLD	102	
R	"M_Schritt_68"			M218.4
	A	L	1.0	
		BLD	102	
R	"M_Schritt_69"			M218.5

```

      A      L      1.0
      BLD 102
R      "M_Schritt_70"      M218.6
      A      L      1.0
      BLD 102
R      "M_Schritt_71"      M218.7
      A      L      1.0
      BLD 102
R      "M_Schritt_72"      M219.0
      A      L      1.0
      BLD 102
R      "M_Schritt_73"      M219.1
      A      L      1.0
      BLD 102
R      "M_Schritt_74"      M219.2
      A      L      1.0
      BLD 102
R      "M_Schritt_75"      M219.3
      A      L      1.0
      BLD 102
R      "M_Schritt_76"      M219.4
      A      L      1.0
      BLD 102
R      "M_Schritt_77"      M219.5
      A      L      1.0
      BLD 102
R      "M_Schritt_78"      M219.6
      A      L      1.0
      JNB _002
      L      0
      T      MB 161
      _002: NOP 0
      A      L      1.0
      JNB _003
      L      0
      T      MB 162
      _003: NOP 0

```

Network: 2 Initializare

Daca turnul este plin nu se porneste - se genereaza eroare

```

      M200.7
A      "M LT voll"      M40.1
A      "M Start Montage"      M200.0
AN      "M_SKM_Prod_Fehler_PS"      M152.0

FP      "M_SK_Montage_Flanke_pos1"      M205.0
R      "M_Start_Montage"      M200.0
S      "M_SKM_Fehler_LT_voll"      M152.5
S      "M_SKM_Prod_Fehler_PS"      M152.0

```

Network: 3 pasul 00

```

#####
#####
#####      #####
#####      Banda 1 magazia de paleti      #####
#####      #####
#####
#####
#####
#####

```

```

AN      "M SK nicht bearbeiten"      M200.7
A      "M_Schritt_00"      M210.0
AN      "M_Schritt_01"      M210.1
S      "A_FB1_Lift_out"      Q12.5
S      "M_FB1_Rechtslauf"      M20.1
S      "M_SK_Montage_T_u_Z_reset"      M205.1
S      "A_AsiLamp_orange"      Q23.5
S      "M_Schritt_01"      M210.1
R      "M_Schritt_00"      M210.0

```

Urca liftul si se aprinde lampa

Network: 4 Schritt 01

Se deschide zavorul de la o paleta

AN	"M SK nicht bearbeiten"	M200.7
A	"E_FB1_Sensor_Lift_out"	I12.2
A	"M_Schritt_01"	M210.1
AN	"M_Schritt_00"	M210.0
S	"A_FB1_Spenden"	Q12.6
R	"A_FB1_Lift_out"	Q12.5
R	"M_SK_Montage_T_u_Z_reset"	M205.1
S	"M_Schritt_02"	M210.2
R	"M_Schritt_01"	M210.1

Network: 5 Schritt 02

Se inchide zavorul

AN	"M SK nicht bearbeiten"	M200.7
A	"E_FB1_Sensor_Palette_out"	I12.0
A	"M_Schritt_02"	M210.2
AN	"M_Schritt_03"	M210.3
R	"A_FB1_Spenden"	Q12.6
S	"M_Schritt_03"	M210.3
R	"M_Schritt_02"	M210.2

Network: 6 Schritt 03

Se pune opritor la banda 2

AN	"M SK nicht bearbeiten"	M200.7
A	"E_FB1_Lift_in"	I12.3
A	"M_Schritt_03"	M210.3
AN	"M_Schritt_04"	M210.4
S	"M_FB_1_Rechtslauf"	M20.1
S	"A_FB2_Stopper_out"	Q14.5
S	"M_Schritt_04"	M210.4
R	"M_Schritt_03"	M210.3

Network: 7 Schritt 04

```
#####  
#####  
#####  
#####   Banda 2 Partea inferioara   #####  
#####  
#####  
#####  
#####
```

Cand se ajunge la capatul din dreapta porneste banda 2

AN	"M SK nicht bearbeiten"	M200.7
AN	"E_FB1_Bandendlage_links"	I11.5
A	"M_Schritt_04"	M210.4
AN	"M_Schritt_05"	M210.5
S	"M_FB_2_Rechtslauf"	M20.4
S	"M_Schritt_05"	M210.5
R	"M_Schritt_04"	M210.4

Network: 8 Schritt 05

AN	"M SK nicht bearbeiten"	M200.7
A	"E_FB2_Sensor_Teil_da"	I14.1
A	"M_Schritt_05"	M210.5
AN	"M_Schritt_06"	M210.6
S	"M_FB_1_Stopp"	M20.2
S	"M_Schritt_06"	M210.6
R	"M_Schritt_05"	M210.5

Network: 9 Schritt 06

Se deschide zavorul de la partea de jos (a cutiei)

AN	"M SK nicht bearbeiten"	M200.7
A	"M_Schritt_06"	M210.6
AN	"M_Schritt_07"	M210.7
S	"A_FB2_Spenden"	Q14.4

R	"M_Schritt_06"	M210.6
S	"M_Schritt_07"	M210.7

Network: 10 Schritt 07

Cand a iesit piesa de baza (de jos) din magazie se inchide zavorul

AN	"M SK nicht bearbeiten"	M200.7
AN	"E_FB2_Sensor_Teil_out"	I14.0
A	"M_Schritt_07"	M210.7
AN	"M_Schritt_08"	M211.0
R	"A_FB2_Spenden"	Q14.4
S	"M_Schritt_08"	M211.0
R	"M_Schritt_07"	M210.7

Network: 11 Schritt 08

Se retrage opritorul (deblocam calea)
cand a coborat piesa

AN	"M SK nicht bearbeiten"	M200.7
A	"E_FB2_Sensor_Teil_out"	I14.0
A	"M_Schritt_08"	M211.0
AN	"M_Schritt_09"	M211.1
R	"A_FB2_Stopper_out"	Q14.5
S	"M_Schritt_09"	M211.1
R	"M_Schritt_08"	M211.0

Network: 12 Schritt 09

Cand se ajunge la capatul benzii 2
Se porneste banda 3 si
se pune si un opritor pe calea 3

AN	"M SK nicht bearbeiten"	M200.7
AN	"E_FB2_Bandendlage_links"	I13.5
A	"M_Schritt_09"	M211.1
AN	"M_Schritt_10"	M211.2
S	"M_FB_3_Rechtslauf"	M20.7
S	"A_FB3_Stopper_out"	Q16.5
S	"M_Schritt_10"	M211.2
R	"M_Schritt_09"	M211.1

Network: 13 Schritt 10

```
#####
#####
#####
#####   Banda 3 Statia de capace   #####
#####
#####
#####
#####
```

AN	"M SK nicht bearbeiten"	M200.7
A	"E_FB3_Sensor_Teil_da"	I16.1
A	"M_Schritt_10"	M211.2
AN	"M_Schritt_11"	M211.3
S	"M_FB_2_Stopp"	M20.5
S	"M_Schritt_11"	M211.3
R	"M_Schritt_10"	M211.2

Network: 14 Schritt 11

Se deschide zavorul de la capace

AN	"M SK nicht bearbeiten"	M200.7
A	"M_Schritt_11"	M211.3
AN	"M_Schritt_12"	M211.4
S	"A_FB3_Spenden"	Q16.4
S	"M_Schritt_12"	M211.4
R	"M_Schritt_11"	M211.3

Network: 15 Schritt 12

Se asteapta sa iasa piesa din magazie

AN	"M SK nicht bearbeiten"	M200.7
A	"E_FB3_Sensor_Teil_out"	I16.0
A	"M_Schritt_12"	M211.4
AN	"M_Schritt_13"	M211.5
S	"M_Schritt_13"	M211.5
R	"M_Schritt_12"	M211.4

Network: 16 Schritt 13

Cand piesa coboara se inchide zavorul

AN	"M_SK_nicht_bearbeiten"	M200.7
A	"E_FB3_Sensor_Teil_da"	I16.1
A	"M_Schritt_13"	M211.5
AN	"M_Schritt_14"	M211.6
R	"A_FB3_Spenden"	Q16.4
R	"A_FB3_Stopper_out"	Q16.5
S	"M_Schritt_14"	M211.6
R	"M_Schritt_13"	M211.5

Network: 17 Schritt 14

AN	"M SK nicht bearbeiten"	M200.7
AN	"E_FB3_Bandendlage_links"	I15.5
A	"M_Schritt_14"	M211.6
AN	"M_Schritt_15"	M211.7
S	"M_FB_4_Rechtslauf"	M21.2
S	"M_Schritt_15"	M211.7
R	"M_Schritt_14"	M211.6

Network: 18 Schritt 15

```
#####
#####
#####
#####   Banda 4 Statia de montare bolturi   #####
#####
#####
#####
```

AN	"M SK nicht bearbeiten"	M200.7
AN	"E_FB4_Sensor_Pos1"	I4.0
A	"M_Schritt_15"	M211.7
AN	"M_Schritt_16"	M212.0
S	"M_FB_3_Stopp"	M21.0
S	"M_FB_4_Stopp"	M21.3
S	"A_FB4_Bolzen_spenden"	Q4.5
S	"M_Schritt_16"	M212.0
R	"M_Schritt_15"	M211.7
R	"M_FB_4_Rechtslauf"	M21.2

Network: 19 Schritt 16

Se actioneaza(se retrage) pistonul de introducere bolt in piesa

AN	"M_SK_nicht_bearbeiten"	M200.7
A	"E_FB4_Zylinder_out"	I4.3
A	"M_Schritt_16"	M212.0
AN	"M_Schritt_17"	M212.1
R	"A_FB4_Bolzen_spenden"	Q4.5
S	"M_Schritt_17"	M212.1
R	"M_Schritt_16"	M212.0

Network: 20 Schritt 17

Porneste banda

AN	"M SK nicht bearbeiten"	M200.7
A	"E_FB4_Zylinder_in"	I4.2
A	"M_Schritt_17"	M212.1

AN	"M_Schritt_18"	M212.2
S	"M_FB_4_Rechtslauf"	M21.2
S	"M_Schritt_18"	M212.2
R	"M_Schritt_17"	M212.1

Network: 21		Schritt 18
-------------	--	------------

Se monteaza boltul la pozitia 2		
---------------------------------	--	--

AN	"M_SK_nicht_bearbeiten"	M200.7
AN	"E_FB4_Sensor_Pos2"	I4.1
A	"M_Schritt_18"	M212.2
AN	"M_Schritt_19"	M212.3
S	"M_FB_4_Stopp"	M21.3
R	"M_FB_4_Rechtslauf"	M21.2
S	"A_FB4_Bolzen_spenden"	Q4.5
S	"M_Schritt_19"	M212.3
R	"M_Schritt_18"	M212.2

Network: 22		Schritt 19
-------------	--	------------

se retrage pistonul(cilindrul)		
--------------------------------	--	--

AN	"M_SK_nicht_bearbeiten"	M200.7
A	"E_FB4_Zylinder_out"	I4.3
A	"M_Schritt_19"	M212.3
AN	"M_Schritt_20"	M212.4
R	"A_FB4_Bolzen_spenden"	Q4.5
		M160.0
S	"M_Schritt_20"	M212.4
R	"M_Schritt_19"	M212.3

Network: 23		Schritt 20
-------------	--	------------

Dupa ce s-a retras cilindrul porneste banda(aparent bagat din mers)		
--	--	--

AN	"M_SK_nicht_bearbeiten"	M200.7
A	"E_FB4_Zylinder_in"	I4.2
A	"M_Schritt_20"	M212.4
AN	"M_Schritt_21"	M212.5
S	"M_FB_4_Rechtslauf"	M21.2
R	"Reset_Counter_Clock"	M160.0
S	"M_Schritt_21"	M212.5
R	"M_Schritt_20"	M212.4

Network: 24		Schritt 21
-------------	--	------------

#####		
#####		
#####		#####
##### Banda 4 Statia de testare		#####
#####		#####
#####		
#####		

Se numara impulsuri de la senzorul pos01		
--	--	--

AN	"M_SK_nicht_bearbeiten"	M200.7
A(
AN	"E_FB4_Senosr_Pos01"	I3.2
FP	M 170.1	
)		
A	"M_Schritt_21"	M212.5
AN	"M_Schritt_22"	M212.6
BLD	101	
A	"Reset_Counter_Clock"	M160.0
L S	C#0	
NOP	C 30	
L	0	
T	C 30	
NOP	MW 140	
	0	

NOP

0

Network: 25 Schritt 21

Statia de testare: Se merge pana la senzorul Sens_Pos_01 de patru ori (PSS)
activeaza si o perioada de asteptare de o secunda

```

AN      "M_SK_nicht_bearbeiten"      M200.7
A(
L      MW      140
L      4
)
A      "M_Schritt_21"      M212.5
AN      "M_Schritt_22"      M212.6
S      "M_FB_4_Stopp"      M21.3
R      "M_FB_4_Linkslauf"      M21.1      -- Motor
      etat Q3.0
R      "M_FB_4_Rechtslauf"      M21.2
L      S5T#1S
SS      T      60
S      "M_Schritt_22"      M212.6
R      "M_Schritt_21"      M212.5

```

Network: 26 Schritt 22

```

      A      "M_Schritt_22"      M212.6
      AN      "M_Schritt_23"      M212.7
      =      L      1.0
      A      L      1.0
      A      "E_FB4_Sensor_OT_da"      I3.7
S      "TS_Upper_part_there"      M161.0      -- 1=there; 0=not there
      A      L      1.0
      A      "E_FB4_Sensor_Bolzen_da"      I5.6
S      "TS_Bolt1_There"      M161.1      -- 1=there; 0=not there
      A      L      1.0
      AN      "E_FB4_Sensor_Bolzen_ind"      I5.7
S      "TS_Bolt1_Material"      M161.2      -- 1=plastic; 0=metal
      A      L      1.0
      BLD      102
      S      "M_FB_4_Rechtslauf"      M21.2
      A      L      1.0
      BLD      102
      R      "M_FB_4_Linkslauf"      M21.1
      A      L      1.0
      BLD      102
      R      "M_FB_4_Stopp"      M21.3
      A      L      1.0
      BLD      102
      S      "M_Schritt_23"      M212.7
      A      L      1.0
      BLD      102
      R      "M_Schritt_22" M212.6

```

```

AN      "M SK nicht bearbeiten"      M200.7
A(
AN      "E_FB4_Senosr_Pos01"      I3.2
FP      M      170.0
)
A      "M_Schritt_23"      M212.7
AN      "M_Schritt_24"      M213.0
CU      C      30
BLD      101
A      "Reset_Counter_Clock"      M160.0
L      C#0
S      C      30
NOP      0
L      C      30
T      MW      140
NOP      0
NOP      0

```

```

S      "TS_Bolt2_There"      M161.3      -- 1=there; 0=not there
      A      L      1.0
      AN      "E_FB4_Sensor_Bolzen_ind"      I5.7
S      "TS_Bolt2_Material"      M161.4      -- 1=plastic; 0=metal
      A      L      1.0
      AN      "E_TS_PalletCode_Bit1"      I5.5

```

S	"TS_PalletCode_Bit1"	M180.0	-- 1=Screw there; 0=Screw not there
		A L 1.0	
	AN "E_TS_PalletCode_Bit2"	I5.4	
S	"TS_PalletCode_Bit2"	M180.1	-- 1=Screw there; 0=Screw not there
		A L 1.0	
	AN "E_TS_PalletCode_Bit3"	I5.3	
S	"TS_PalletCode_Bit3"	M180.2	-- 1=Screw there; 0=Screw not there
		A L 1.0	
	AN "E_TS_PalletCode_Bit4"	I5.2	
S	"TS_PalletCode_Bit4"	M180.3	-- 1=Screw there; 0=Screw not there
		A L 1.0	
	AN "E_TS_PalletCode_Bit5"	I5.1	
S	"TS_PalletCode_Bit5"	M180.4	-- 1=Screw there; 0=Screw not there
		A L 1.0	
	AN "E_TS_PalletCode_Bit6"	I5.0	
S	"TS_PalletCode_Bit6"	M180.5	-- 1=Screw there; 0=Screw not there
		A L 1.0	
		BLD 102	
	S "M_FB_4_Rechtslauf"	M21.2	
		A L 1.0	
		BLD 102	
	R "M_FB_4_Linkslauf"	M21.1	
		A L 1.0	
		BLD 102	
	R "M_FB_4_Stopp"	M21.3	
		A L 1.0	
		BLD 102	
	S "M_Schritt_25"	M213.1	
		A L 1.0	
		BLD 102 se porneste banda 5	
AN	"M SK nicht bearbeiten"	M200.7	
AN	"E_FB4_Endlage_links"	I3.5	
A	"M_Schritt_25"	M213.1	
AN	"M_Schritt_26"	M213.2	
S	"M_FB_5_Rechtslauf"	M21.5	
S	"M_Schritt_26"	M213.2	
R	"M_Schritt_25"	M213.1	
Network: 31 Schritt 26			
AN	"M SK nicht bearbeiten"	M200.7	
AN	"E_FB5_Endlage_links"	I17.5	
A	"M_Schritt_26"	M213.2	
AN	"M_Schritt_27"	M213.3	
S	"M_FB_4_Stopp"	M21.3	
R	"M_FB_4_Rechtslauf"	M21.2	
S	"A_TK1_Bandlauf_links"	Q19.0	
S	"M_Schritt_27"	M213.3	
R	"M_Schritt_26"	M213.2	
Network: 32 Schritt 27			
AN	"M SK nicht bearbeiten"	M200.7	
AN	"E_TK1_Bandendlage_rechts"	I19.1	-- senzor
A	"M_Schritt_27"	M213.3	
AN	"M_Schritt_28"	M213.4	
S	"M_FB_5_Stopp"	M21.6	
R	"M_FB_5_Rechtslauf"	M21.5	
R	"A_TK1_Bandlauf_links"	Q19.0	-- A_TK1
		Q20.3	
S	"M_Schritt_28"	M213.4	
R	"M_Schritt_27"	M213.3	
Network: 33 Schritt 28			
AN	"M SK nicht bearbeiten"	M200.7	
A	"LT_Gabel_unten"	I20.6	
A	"M_Schritt_28"	M213.4	
AN	"M_Schritt_27"	M213.3	
S	"A_TK1_Bandlauf_links"	Q19.0	
R	"A_LT_Spindel_down"	Q20.3	
S	"M_Schritt_29"	M213.5	
R	"M_Schritt_28"	M213.4	

Network: 34 Schritt 29

```
#####
#####
#####
#####          Turnul de stocare          #####
#####
#####
#####
#####
```

Exista un palet pe furca?
 Porneste timerul 70 (timp de o secunda)
 opresc banda 6
 Activeaza turnul de stocare(seteaza un bit)

```

AN      "M_SK_nicht_bearbeiten"          M200.7
A        "LT_WS_auf_Gabel"                I20.7
A        "M_Schritt_29"                   M213.5
AN      "M_Schritt_30"                     M213.6
=        L          1.0
A        L          1.0
L        S5T#1S
SS       T          70
A        "M_SK_Montage_T_u_Z_reset"        M205.1
R        T          70
NOP      0
NOP      0
NOP      0

                                A      L      1.0

R        "A_TK1_Bandlauf_links"            Q19.0          -- A_TK1 Banda stânga
                                A      L      1.0
                                BLD     102
S        "M_FB_Lagerturm_enable"          M205.2
                                A      L      1.0
                                BLD     102
S        "M_Schritt_30"                   M213.6
                                A      L      1.0
                                BLD     102

R        "M_Schritt_29" M213.5Daca dupa temporizarea 70 (de o secunda) turnul de stocare e
                                pregatit:
                                se porneste timerul 17 (de doua secunde)
                                se seteaza bitul ce da comanda de depozitare

```

```

A        T          70
AN      "M_SK_nicht_bearbeiten"          M200.7
A        "M_LT_READY"
A        "M_Schritt_30"                   M213.6
AN      "M_Schritt_31"                     M213.7
=        L          1.0
A        L          1.0
L        S5T#2S
SS       T          17
A        "M_SK_Montage_T_u_Z_reset"        M205.1
R        T          17

NOP      0
NOP      0
NOP      0
A        L          1.0
BLD     102
S        "M_LT_Einlagern"                 M205.3
A        L          1.0
BLD     102
S        "M_Schritt_31"                   M213.7
A        L          1.0
BLD     102
R        "M_Schritt_30"                   M213.6

```

Network: 36 Schritt 31

```

AN      "M_SK_nicht_bearbeiten"          M200.7
.        T          17
.        "M_Schritt_3"                    M213.7
.        "M_Schritt_3"                    M214.0
.        "M_LT_Einlag"                    M205.3

```

```

: "A_AsiLamp_o      Q23.5          -- lampa portocalie
  "M_Schritt_3      M214.0
: "M_Schritt_3      M213.7
  1"

```

Network: 37 Schritt 32

```

#####
#####
#####

```

```

##### (fara erori) #####
#####
#####

```

Finalizare operatie de depozitare?
se seteaza bitul de finalizare montaj
se mai seteaza si bitul de sfarsit montaj

```

AN  "M_SK nicht bearbeiten"      M200.7
A    "M_LT_READY"                M40.0
A    "M_Schritt_32"              M214.0
S    "M_Montage_fertig"          M200.1
S    "M_FB_Lagerturm_enable"     M205.2
S    "M_SK_Montage_ENDE"         M200.3
R    "M_Schritt_32"              M214.0
S    Q      0.4

```

FC2 - <offline>

"Schrittkette Demontage"

Block version: 2

Time stamp Code: Interface:

06/26/2012 02:23:58 PM

01/28/2008 09:59:58 AM

Lengths (block/logic/data): 01208 00894 00002

Block: FC2 Demontaj

Network: 1

```

AN  "M_DM_Schritt_00"      M225.0
AN  "M_SK nicht bearbeiten" M200.7
A    "E_SPS_Schalter_Dig_IN_4" I2.4
=    L      1.0
A    L      1.0
BLD  102
R    "Sterge atribuire LT"   M208.0
A    L      1.0
JNB  _001
L    2
T    "Piesa ce va fi demontata"
A    L      1.0
BLD  102
S    "M_Start_Demontage"    M200.4
A    L      1.0
BLD  102
R    Q      0.4

```

Network: 2

```

AN  "M_SK nicht bearbeiten" M200.7
AN  "M_SKM_Prod_Warn_PS"    M152.6
      rnung vorliegt
A    "M_Start_Demontage"    M200.4
AN  "M_SKM_Prod_Fehler_PS" M152.0
      oare de productie

A    L      1.0
JNB  _002
L    0
T    MD 225
A    L      1.0
JNB  _003
L    0
T    MD 229
A    L      1.0
BLD  102
S    "M_DM_Schritt_00"      M225.0

```

```

A      L      1.0
BLD    102
R      "M_Start_Demontage"      M200.4
Network: 3      Schritt 00

```

```

#####
#####
#####
#####          Lagerturm          #####
#####
#####
#####
#####
#####
#####

```

```

AN      "M_SK_nicht_bearbeiten"      M200.7
A      "M_DM_Schritt_00"      M225.0
AN      "M_DM_Schritt_01"      M225.1
=      L      1.0
A      L      1.0
A(
L      "Piesa ce va fi demontata"      MW41
L      1
==I
)
JNB      _004
L      "M_WScore_Lagerpl_1"      MB128
T      "MI01_WS_Ident_Code"      MB101
A      L      1.0
A(
L      "Piesa ce va fi demontata"      MW41
L      2
==I
)
JNB      _005
L      "M_WScore_Lagerpl_2"      MB129
T      "MI01_WS_Ident_Code"      MB101
A      L      1.0
A(
L      "Piesa ce va fi demontata"      MW41
L      3
==I
)
JNB      _006
L      "M_WScore_Lagerpl_3"      MB130
T      "MI01_WS_Ident_Code"      MB101
A      L      1.0
A(
L      "Piesa ce va fi demontata"      MW41
L      4
==I
)
JNB      _007
L      "M_WScore_Lagerpl_4"      MB131
T      "MI01_WS_Ident_Code"      MB101
A      L      1.0
A(
L      "Piesa ce va fi demontata"      MW41
L      5
==I
)
JNB      _008
L      "M_WScore_Lagerpl_5"      MB132
T      "MI01_WS_Ident_Code"      MB101
A      L      1.0
A(
L      "Piesa ce va fi demontata"      MW41
L      6
==I
)
JNB      _009
L      "M_WScore_Lagerpl_6"      MB133
T      "MI01_WS_Ident_Code"      MB101
A      L      1.0
A(
L      "Piesa ce va fi demontata"      MW41
L      7
==I
)
JNB      _00a
L      "M_WScore_Lagerpl_7"      MB134
T      "MI01_WS_Ident_Code"      MB101

```

```

A      L      1.0
A(
L      "Piesa ce va fi demontata"  MW41
L      8
==I
)
JNB    _00b
L      "M_WSCode_Lagerpl_8"          MB135
T      "MI01_WS_Ident_Code"          MB101
A      L      1.0
BLD    102
A      L      1.0
BLD    102
R      "M_SK_Demontage_TuZ_reset"    M200.5
A      L      1.0
BLD    102
R      "M_SK_Demontage_ENDE"         M200.6
A      L      1.0
BLD    102
R      T      35
A      L      1.0
BLD    102
S      "M_DM_Schritt_01"             M225.1
A      L      1.0
BLD    102
R      "M_DM_Schritt_00"             M225.0

Start schimbarea
--> Auslagern starten

```

```

AN      "M_SK_nicht_bearbeiten"      M200.7
A      "E_LT_Drehpos_middle"         I20.1
A      "LT_Gabel_unten"              I20.6
A      "M_DM_Schritt_01"             M225.1
AN      "M_DM_Schritt_02"            M225.2
= A      L      1.0
BLD    L      1.0
S      102
A      "M_LT_Auslagern"              M205.4
L      L      1.0
SS      S5T#1S
A      T      35
R      "M_SK_Demontage_TuZ_reset"    M200.5
NOP     T      35
        0
NOP     0
NOP     0
A      L      1.0
BLD    102
S      "M_DM_Schritt_02"             M225.2
A      L      1.0
BLD    102
R      "M_DM_Schritt_01"             M225.1

```

```

Network: 5      Schritt 02

```

```

A      T      35
AN      "M_SK_nicht_bearbeiten"      M200.7
A      "M_DM_Schritt_02"             M225.2
AN      "M_DM_Schritt_03"            M225.3
R      "M_LT_Auslagern"              M205.4
S      "M_DM_Schritt_03"             M225.3
R      "M_DM_Schritt_02"             M225.2

```

```

Network: 6      Schritt 03
Se porneste Banda
Camera de turn retrage aprobarea
Bandlauf starten

```

```

AN      "M SK nicht bearbeiten"      M200.7
A      "M_LT_READY"                  M40.0
A      "LT_WS_auf_Gabel"              I20.7
A      "LT_Gabel_unten"              I20.6
A      "M_DM_Schritt_03"             M225.3
AN      "M_DM_Schritt_04"            M225.4
R      "M_FB_Lagerturm_enable"       M205.2
        Q19.1

```

S	"M_DM_Schritt_04"	M225.4
R	"M_DM_Schritt_03"	M225.3

Network: 7 Schritt 04

Bandend locat,ie (de transfer nodul 1) se ajunge la
Schimba banda Förderband_5

AN	"M SK nicht bearbeiten"	M200.7
AN	"E TK1 Bandendlage rechts"	
A	"M_DM_Schritt_04"	M225.4
AN	"M_DM_Schritt_05"	M225.5
S	"M_FB_5_Linkslauf"	M21.4
) setat Q17.0
S	"M_DM_Schritt_05"	M225.5
R	"M_DM_Schritt_04"	M225.4

Network: 8 Schritt 05

Wenn Bolzen_1 da und Bolzen_1 rot
--> Band anhalten und Zylinder ausfahren

AN	"M SK nicht bearbeiten"	M200.7	
		I18.5	
A	"M_DM_Schritt_05"	M225.5	-
AN	"M_DM_Schritt_06"	M225.6	
S	"M_FB_5_Stopp"	M21.6	
R	"M_FB_5_Linkslauf"	M21.4	-
) setat Q17.0	
R	"M_FB_5_Rechtslauf"	M21.5	
S	"A_FB5_Bolzen_alu_Zyl_out"	Q18.6	-
		- retrage piston)	
S	"M_DM_Schritt_06"	M225.6	
R	"M_DM_Schritt_05"	M225.5	

Bolzen_1 Zylinder ausgefahren
--> Zylinder einfahren

AN	"M SK nicht bearbeiten"	M200.7	
A	"E_FB5_Bolzen_rot_Zyl_out"	I18.7	-
A	"M_DM_Schritt_06"	M225.6	
AN	"M_DM_Schritt_07"	M225.7	
R	"A_FB5_Bolzen_alu_Zyl_out"	Q18.6	-
		- retrage piston)	
		Q19.1	-
S	"M_DM_Schritt_07"	M225.7	
R	"M_DM_Schritt_06"	M225.6	

Network: 10 Schritt 07

Bolzen_1 Zylinder ausgefahren

AN	"M SK nicht bearbeiten"	M200.7
A	"E_SPS_Schalter_Dig_IN_2"	I2.2
A	"E_FB5_Bolzen_rot_Zyl_in"	
A	"M_DM_Schritt_07"	M225.7
AN	"M_DM_Schritt_08"	M226.0
S	"M_FB_5_Linkslauf"	M21.4
		setat
S	"M_DM_Schritt_11"	M226.3
R	"M_DM_Schritt_07"	M225.7

Network: 11 Schritt 11 - scoate bolt 2 la piston 2

Wenn Bolzen_1 da und Bolzen_1 alu
--> Band anhalten und Zylinder aus

AN	"M SK nicht bearbeiten"	M200.7	
		I18.1	-
A	"M_DM_Schritt_11"	M226.3	

AN	"M_DM_Schritt_12"	M226.4	
S	"M_FB_5_Stopp"	M21.6	
R	"M_FB_5_Linkslauf"	M21.4	-
) setat Q17.0	
R	"M_FB_5_Rechtslauf"	M21.5	
S	"A_FB5_Bolzen_rot_Zyl_out"	Q18.2	-
		-	
S	"M_DM_Schritt_12"	M226.4	
R	"M_DM_Schritt_11"	M226.3	

Bolzen_1 Zylinder ausgefahren
--> Zylinder einfahren

AN	"M_SK nicht bearbeiten"	M200.7	
A	"E_Zyl_out"	I18.3	
A	"M_DM_Schritt_12"	M226.4	
AN	"M_DM_Schritt_13"	M226.5	
R	"A_FB5_Bolzen_rot_Zyl_out"	Q18.2	-
		-	
S	"M_DM_Schritt_13"	M226.5	
S	M 80.0		
R	"M_DM_Schritt_12"	M226.4	

Network: 13 Schritt 13

Bolzen_1 Zylinder ausgefahren

AN	"M_SK nicht bearbeiten"	M200.7	
A	"E_Zyl_in"	I18.2	
A	"M_DM_Schritt_13"	M226.5	
A	"E_SPS_Schalter_Dig_IN_3"	I2.3	
AN	"M_DM_Schritt_14"	M226.6	
S	"M_FB_5_Linkslauf"	M21.4	-
S	"M_FB_4_Linkslauf"	M21.1	-
			-
S	"M_DM_Schritt_14"	M226.6	
R	"M_DM_Schritt_13"	M226.5	

Network: 14 Schritt 14

AN	"M_SK nicht bearbeiten"	M200.7	
AN	"E_TK2_Dreh_Endl_rechts"	I3.4	
A	"M_DM_Schritt_14"	M226.6	
AN	"M_DM_Schritt_15"	M226.7	
S	"M_FB_5_Stopp"	M21.6	
R	"M_FB_5_Linkslauf"	M21.4	
		setat Q17.0	
R	"M_FB_3_Stopp"	M21.0	
S	"M_FB_3_Linkslauf"	M20.6	
		setat prin Q15.0	
R	M 80.0		
S	"M_DM_Schritt_15"	M226.7	
R	M 250.0		
R	Q 0.0		
R	"M_DM_Schritt_14"	M226.6	

setat Q3.0

I0.0 este 1

AN	"M_SK nicht bearbeiten"	M200.7	
A(
AN	"E_FB3_Bandendlage_links"	I15.5	
FP	M 90.0		
)			
A	"M_DM_Schritt_15"	M226.7	
AN	"M_DM_Schritt_16"	M227.0	
CU	C 31		
BLD	101		
A	M 80.0		

L	C#0	
S	C	31
NOP	0	
L	C	31
T	MW	60
NOP	0	
NOP	0	

Network: 16 pas 15

Network: 18

AN	"E_FB1_Bandendlage_links"	I11.5
CU	C	12
BLD	101	
L	C#6	
S	C	12

I4.0

R	C	12
L	C	12
T	MW	3

AN	"E_Bandendlage_links_FB5"	I17.4
NOP	0	
NOP	0	

Network: 19 Motor Conveior Statie nr. 1

A	C	12
A(
L	MW	3
L	12	
==I		
)		
R	"A_FB1_Linkslauf"	Q11.0
		prin M20.0
R	"A_FB2_Linkslauf"	Q13.0
S	"M_FB_3_Stopp"	M21.0

Network: 20 Schritt 18

AN	"M_SK nicht bearbeiten"	M200.7
A	"M_DM_Schritt_18"	M227.2
AN	"M_DM_Schritt_19"	M227.3
S	"M_FB_1_Stopp"	M20.2
S	"M_DM_Schritt_19"	M227.3
R	"M_DM_Schritt_18"	M227.2

Network: 21 Schritt 19

Demontage beendet

AN	"M_SK nicht bearbeiten"	M200.7
A	"M_DM_Schritt_19"	M227.3
R	"M_LT_Auslagern"	M205.4
R	"M_SK_Demontage_TuZ_reset"	M200.5
S	"M_SK_Demontage_ENDE"	
S	"M_FB_Lagerturm_enable"	M205.2
R	M	250.0
R	Q	0.0
R	"M_DM_Schritt_19"	M227.3

Network: 22

```

AN      "M_SK_nicht_bearbeiten"      M200.7
A      "E_FB3_Bandendlage_links"      I15.5
CU      C      10
BLD     101
AN      "E_FB4_Sensor_Pos1"           I4.0
L      C#6
S      C      10
AN      "E_Bandendlage_links_FB5"      I17.4
R      C      10
L      C      10
T      MW      5

NOP     0

```

Network: 23 Motor Conveior Statie nr. 3 - mers inapoi

```

A      C      10
A(
L      MW      5
L      12
==I
)
R      "A_FB3_Linkslauf"   Q15.0                    -- Motor

```

Network: 24 Porneste banda 3 din DAQ

```

A      "E SPS Schalter Dig IN 0"   I2.0
S      "A_FB3_Linkslauf"            Q15.0

```

```

Network: 25
S      M      250.0
prin M20.3

```

Network: 29 Motor Conveior Statie nr. 2 - mers inapoi

```

A      "E SPS Schalter Dig IN 1"   I2.1
S      "A_FB2_Linkslauf"            Q13.0
                                           - setat prin M20.3
R      "A_FB3_Linkslauf"            Q15.0
                                           - setat prin M20.6
R      "A_FB4_Linkslauf"            Q3.0
                                           - setat prin M21.1

```
