

# SIMULAREA CU AJUTORUL INSTRUMENTAȚIEI VIRTUALE A UNOR PROCESE INDUSTRIALE ROBOTIZATE

Șef lucr. dr. ing. Barbu Cristian BRAUN<sup>1</sup>, Șef lucr. dr. ing. Corneliu Nicolae DRUGĂ<sup>1</sup>

<sup>1</sup> Universitatea „Transilvania” din Brașov, România<sup>1</sup>

**REZUMAT.** În lucrare este descrisă o metodă prin care diferite cicluri industriale de producție robotizate pot fi simulate cu ajutorul unei interfețe de instrumentație virtuală. Concret sunt prezentate câteva exemple de cicluri de asamblare a unor sisteme hidro-pneumatice cu rol funcțional activ. Exemplele prezentate se bazează pe algoritmi de programare a ciclurilor de asamblare aferente, pornind de la funcționarea și modul de programare al unui robot de tip Melfa Basic IV, dispus pe unul dintre modulele de lucru ale unei linii automatizate de producție.

**Cuvinte cheie:** Ciclu, robotizat, simulare, algoritmi, virtual, eveniment

**ABSTRACT.** The paper describes a method by which different robotic industrial production cycles can be simulated using a virtual instrumentation interface. Concretely, some examples of assembly cycles of hydro-pneumatic systems with an active functional role are presented. The presented examples are based on algorithms for programming related assembly cycles, starting from the operation and programming mode of a Melfa Basic IV type robot, placed on one of the work modules of an automated production line.

**Keywords:** Cycle, robotic, simulation, algorithm, virtual, event

## 1. IMPORTANȚA ROBOȚILOR INDUSTRIALI

În ultimii ani dezvoltarea și utilizarea roboților a cunoscut o dinamică în continuă creștere, aceștia fiind utilizați nu doar în domeniul industrial ci și în domeniul medical sau chiar casnic [1], [2]. Chiar dacă în prezent utilizarea roboților în domeniul medical și cel casnic este răspândită și se estimează o utilizare pe scară tot mai largă, domeniul industrial rămâne cel mai vizat în privința folosirii roboților [3], [4]. Acest fapt se explică prin faptul că operațiile industriale, în marea lor majoritate sunt sintetizate în cicluri automatizate, fiind deosebit de importantă productivitatea [4], [5]. Cu mici excepții, liniile tehnologice de producție implică prelucrarea de componente, asamblare și testare în serii mari și foarte mari de repere. Aceasta înseamnă că programarea unui astfel de ciclu, odată făcută și testată, ea poate apoi fi implementată și utilizată la scară largă, pe perioade mai scurte sau mai lungi pentru asigurarea acestor cicluri de producție. Cu toate acestea, în continuare intervenția operatorului uman este indispensabilă din două puncte de vedere. În primul rând este nevoie de cel puțin un programator care să conceapă algoritmul, să scrie programul în codul mașină, să-l compileze și apoi să îl testeze și să aducă remedierile necesare. Pe de altă parte, din anumite motive, un ciclu de producție

programat, în timpul desfășurării acestuia poate întâmpina anumite sincope. Ele pot fi de natură subiectivă (programatorul a omis ceva sau nu a testat programul în integralitate, în toate condițiile) sau de natură obiectivă (poate interveni o pană de curent sau un eveniment imprevizibil ce poate perturba desfășurarea ciclului). Ca urmare, și într-un caz și în altul este necesară supravegherea ciclurilor de producție, o intervenție făcându-se doar în situații de urgență. Totodată, o reprogramare a unor cicluri de producție este de asemenea necesară, știut fiind faptul că cererile de producție pe piață se schimbă într-un mod foarte dinamic [5].

## 2. SIMULAREA PROGRAMĂRII CICLURILOR, CA ROL EDUCAȚIONAL

Ca urmare a faptului că programarea unor cicluri de producție pe liniile tehnologice este esențială din punctul de vedere al siguranței și al eficienței, rigurozitatea acestei etape este absolut indispensabilă. În consecință, etapele de programare a unui ciclu de producție pe scară medie și largă trebuie să fie parcurse cu maximă atenție și rigurozitate. Una dintre cele mai importante etape constă în testarea programării ciclului respectiv, mai concret să se verifice dacă în ciclul respectiv sunt parcurși în mod corect toți pașii. Mai mult decât atât, trebuie să se

verifice că nu apare nici un risc de coliziune, implicit de avariere a unor utilaje, a unor scule aşchietoare, a unor reperi etc. în timpul rulării ciclului. În faza testării, riscurile să se constate anumite disfuncţionalităţi ce pot pune în pericol întreg procesul poate fi încă destul de mare. Aceasta se explică prin faptul că operatorul şi programatorul pot scăpa din vedere anumite aspecte (cum ar fi, de exemplu să acţioneze oprirea de urgenţă în timp util). De aceea simulatoarele de programare şi-au găsit în ultima perioadă un rol din ce în ce mai important [6]. Pornind de la aceste considerente, în ultimii ani s-au dezvoltat diferite medii software dedicate exclusiv simulărilor în condiţii cât mai apropiate de cele reale. În acest mod, în etapa testării unui program de ciclu automatizat de producţie se poate observa, prin simulare modul în care se succed evenimentele, putând fi observate toate sincopile şi riscurile ce pot apărea. În felul acesta, în timpul testării nu există nici un fel de pericol de a se produce vreo coliziune sau vreo daună la nivelul liniei de producţie şi astfel de probleme pot fi rezolvate mult mai eficient şi fără riscuri [6], [7].

Un alt aspect deosebit al simulării programării ciclurilor de producţie este de natură educaţională. Concret, prin simulări, o persoană proaspăt angajată cu funcţia de programator, poate să se iniţieze într-un timp mult mai scurt şi cu costuri mult mai mici. Aceasta se explică prin faptul că persoana în cauză va putea înţelege mult mai rapid şi mai uşor logica algoritmilor şi a evenimentelor logice pe care le implică un ciclu de programare. Mai mult, ea va înţelege şi va putea mult mai eficient să remedieze diferite sinope ce pot surveni în primele etape ale testării unui ciclu programat iniţial.

Se ştie faptul că, din motive financiar-economice, o companie de producţie, în zilele noastre rar îşi permite să angajeze persoane fără experienţă în domeniu. Aceasta înseamnă că o persoană proaspăt angajată pe postul de programator trebuie să aibă un minim bagaj de cunoştinţe şi de competenţe în programarea roboţilor industriali. Înseamnă, deci că doar abilitatea de a se forma rapid, datorită însuşirii cunoştinţelor de operare într-un mediu software dedicat simulărilor nu ar fi suficientă. Prin urmare, persoana care intenţionează să se angajeze ca programator ar trebui să îşi formeze o serie de astfel de competenţe înainte de angajare. Din acest motiv, tot mai mult se pune accentul pe formarea din faza educaţională, mai concret, studenţii cu profil tehnic să poată dobândi cunoştinţe şi chiar competenţe practice în acest sens. Mediile software dedicate pentru simularea ciclurilor programabile ale roboţilor industriali pot fi deosebit de utile şi binevenite, facilitând nu doar accesul mult mai uşor la cunoştinţe, ci şi actul educaţional, în sine.

Însă problema cea mai mare se referă, pe de-o parte la costul de achiziţie şi de mentenanţă al unor astfel de licenţe software şi, pe de altă parte necesitatea unor resurse puternice ale sistemelor de calcul. Este cunoscut faptul că, în special în ţările mai puţin dezvoltate, prea puţine instituţii de învăţământ îşi pot permite să achiziţioneze sisteme de calcul foarte performante şi licenţe software cu costuri ridicate. Mai mult, nici măcar companiile de producţie nu întotdeauna îşi permit asemenea investiţii, mai ales dacă este vorba doar de actul educaţional. În consecinţă, din păcate, în foarte multe situaţii, soluţia achiziţionării unor licenţe software dedicate simulărilor (ce necesită resurse puternice) nu poate fi adoptată.

Pentru a putea fi depăşită totuşi această problemă, soluţia realizării unor interfeţe software mai simple, dar care să aibă acelaşi scop, acela de a permite simulări în condiţii cât mai apropiate de condiţiile reale, pare a fi o soluţie viabilă. Acest lucru ar putea fi posibil prin crearea unor interfeţe software în medii software mult mai accesibile atât ca preţ de cost, cât şi ca resurse informaţionale.

### 3. DEZVOLTAREA UNEI INTERFEŢE SOFTWARE UTILĂ SIMULĂRII CICLURILOR DE PRODUCŢIE ALE ROBOŢILOR INDUSTRIALI

În ideea celor descrise în capitolul II, în lucrarea de faţă este prezentat un exemplu prin care se poate arăta faptul că într-un mediu software specific instrumentaţiei virtuale (LabVIEW) se poate crea o astfel de interfaţă. În capitolul III este prezentat modul în care s-a conceput, s-a programat o astfel de interfaţă software utilă simulării unor cicluri simple de programare. Trebuie menţionat faptul că, în prezent, aceasta este folosită intens atât la activităţile de curs, cât şi la cele de laborator pentru iniţierea studenţilor de la specializarea Mecatronică, din cadrul Universităţii Transilvania din Braşov.

În vederea înţelegerii, în etapa educaţională a procedurii specifice programării unor cicluri simple de producţie, interfaţa software s-a axat pe simularea unor cicluri de asamblare a unor componente pneumatice pe linia de asamblare FESTO Didactic. Interfaţa a fost concepută în aşa fel încât să se poată înţelege simplu şi rapid succesiunea evenimentelor şi algoritmi de programare specifici diferitor cicluri de asamblare. Ordinea ciclurilor ce pot fi simulate în cadrul interfeţei a fost stabilită în aşa fel încât să se meargă de la simplu la mai complex. Concret, s-a stabilit o listă de cicluri simulabile după cum urmează: 1 - mutarea unei carcase de culoare roşie din staţia de alimentare în staţia de asamblare;

## SIMULAREA CU AJUTORUL INSTRUMENTAȚIEI VIRTUALE A UNOR PROCESE INDUSTRIALE

2 - mutarea unei carcase de culoare neagră din stația de alimentare în stația de asamblare; 3 - mutarea unei carcase de culoare argintie din stația de alimentare în stația de asamblare; 4 - asamblare carcasă roșie + piston negru; 5 - asamblare carcasă neagră + piston argintiu; 6 - asamblare carcasă argintie + piston negru; 7 - asamblare carcasă roșie + piston negru + arc; 8 - asamblare carcasă neagră + piston argintiu + arc și 9 - asamblare carcasă argintie + piston negru + arc. Semnificația culorilor se referă la dimensiunile carcaselor, respectiv ale pistoanelor. Carcasa de culoare roșie are dimensiunea mai mare a diametrului interior (în care se introduce pistonul) și înălțimea mai mare. Carcasa de culoare neagră are dimensiunea mai mică a diametrului interior și înălțimea mai mare iar carcasa argintie are dimensiunea mai mare a diametrului interior și înălțimea mai mică. De asemenea, pistonul de culoare neagră are diametrul mai mare, iar pistonul argintiu are diametrul mai mic.

Pentru a se înțelege mai ușor procesul de simulare a evenimentelor de programare a ciclurilor enumerate mai sus, în figura 3.1 este prezentată stația de asamblare cu ajutorul robotului Mitsubishi Melfa Basic IV RV-2AJ:

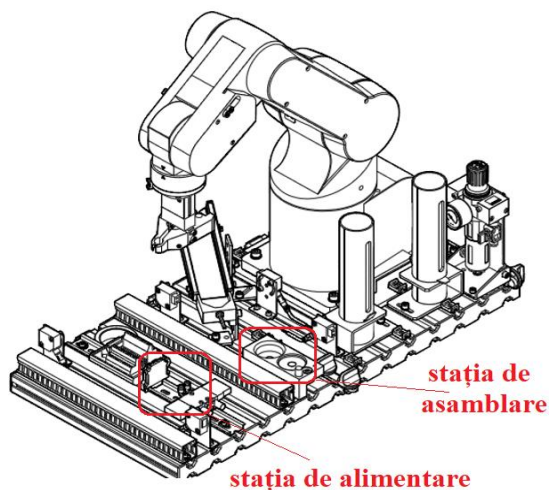


Fig. 3.1. Prezentarea stației de asamblare a robotului Mitsubishi Melfa Basic IV RV-2AJ din cadrul liniei FESTO Didactic.

În cele ce urmează vor fi exemplificate și descrise două dintre ciclurile programabile enumerate mai sus. Este vorba de ciclul de asamblare carcasă argintie cu piston negru și ciclul de asamblare carcasă neagră – piston argintiu – arc.

### A. Ciclul de asamblare carcasă – piston

În figura 3.2 este ilustrate etapele de asamblare în stația de lucru:

Definirea listei de alocare pentru pozițiile necesare parcurgerii ciclului s-a considerat că este foarte importantă, fiind prezentată mai jos: POS P1 – poziția de alimentare carcasă nouă; POS P2 –

poziția intermediară nr. 1; POS P3 – poziție depozitare carcasă în stația de asamblare; POS P4 – poziția intermediară nr. 2; POS P5 – poziție preluare piston; POS P7 – poziție depozitare piston în interiorul carcasei; POS P99 – poziție de pornire/finalizare ciclu. În afară de acestea ar mai fi utilă o listă de alocare pentru pozițiile auxiliare: POS P11 – poziția pentru verificarea prezenței carcasei în stația de alimentare (P1); POS P12 – poziția pentru verificarea prezenței pistonului în stația (P5).

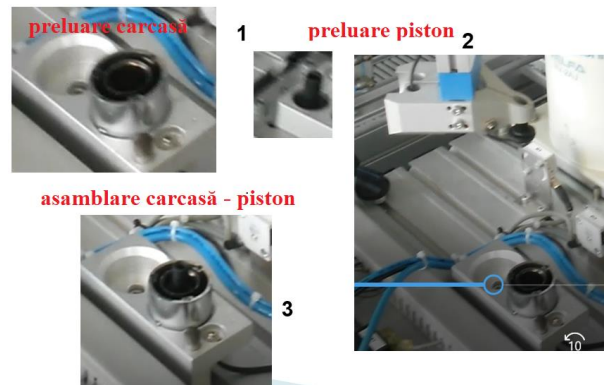


Fig. 3.2. Etapele de asamblare carcasă argintie – piston negru.

În ceea ce privește programarea ciclului, în continuare este prezentată rutina, cu liniile de comentariu aferente:

### - definirea variabilelor de tip poziție:

- 10 DEF POS P11 – definirea variabilei poziției P11 (pt. verificare prezență carcasă);
- 20 DEF POS VECX25 – definirea cotei relative pentru poziția auxiliară P11 față de poziția predefinită P1;
- 30 VECX25 = (25,0,0,0,0) – atribuirea coordonatelor carteziene pentru cota relativă;
- 40 P11 = P1 + VECX25 – atribuirea coordonatelor specifice poziției auxiliare P11;
- 50 DEF POS VECZ100 – definirea unei poziții auxiliare pentru coborârea și ridicarea prehsorului în scopul evitării coliziunilor;
- 60 VECZ100 = (0,0,100,0,0) – atribuirea coordonatelor specifice poziției de ridicare / coborâre).

### - definirea variabilelor de tip Boolean:

- 70 DEF IO START = BIT,1 – definirea variabilei Booleene START prin atribuirea bitului nr. 1 din registrul de intrări ieșiri, pentru condiționarea pornirii ciclului;
- 80 DEF IO PCARCASĂ = BIT,2 - definirea variabilei Booleene PREZENȚĂ CARCASĂ, prin atribuirea bitului nr. 2 din registrul de intrări ieșiri;
- 90 DEF IO PPISTON = BIT,3 - definirea variabilei Booleene PREZENȚĂ PISTON, prin atribuirea bitului nr. 3 din registrul de intrări ieșiri.

## CREATIVITATE, INVENTICĂ, ROBOTICĂ

### *- definirea variabilelor de tip întreg (Integer):*

- 100 DEF INTE MICĂ – definirea variabilei de tip nr. întreg pentru viteză redusă de deplasare;  
110 MICĂ% = 25 – atribuirea valorii procentuale de 25% din viteza maximă pentru viteza de deplasare redusă;  
120 DEF INTE MEDIE - definirea variabilei de tip nr. întreg pentru viteză medie de deplasare;  
130 MEDIE% = 55 - atribuirea valorii procentuale de 55% din viteza maximă pentru viteza de deplasare medie;  
140 DEF INTE MARE - definirea variabilei de tip nr. întreg pentru viteză mare de deplasare;  
150 MARE% = 80 - atribuirea valorii procentuale de 80% din viteza maximă pentru viteza de deplasare mare.

### *Pornire ciclul:*

- 160 IF START = 1 THEN GOTO 170, ELSE GOTO 160 – condiționarea pornirii ciclului de apăsarea butonului START;

### *Preluare carcasă cu viteză mare*

- 170 SPD MARE% - stabilirea vitezei mari (80%) pentru preluarea carcasei;  
180 MOV P11 + VECZ100 – deplasarea liberă deasupra poziției P11 pentru verificare;  
190 MVS P11 – coborâre prin interpolare liniară în poziția de verificare P11;  
200 DLY 2.0 – temporizare 2 secunde necesară verificării;  
210 IF PCARCASĂ = 1 THEN GOTO 220, ELSE GOTO 160 – impunerea condiției ca doar dacă este sesizată o carcasă în magazia de alimentare ciclul să continue;  
220 MVS P11 + VECZ100 – revenire prin interpolare liniară deasupra poziției de verificare;  
230 MOV P1 + VECZ100 – deplasare liberă deasupra poziției de preluare (P1);  
240 HOPEN 1 – deschiderea prealabilă a prehensurului;  
250 MVS P1 – coborâre prin interpolare liniară în poziție de preluare (P1);  
260 DLY 0.5 – temporizare de 0,5 secunde pentru preluarea în siguranță a carcasei;  
270 HCLOSE 1 – închiderea prehensurului și preluarea carcasei;  
280 DLY 0.5 – temporizare de 0,5 secunde pentru ridicare în siguranță;  
290 MVS P1 + VECZ100 – ridicare cu carcasa prin interpolare liniară deasupra poziției de preluare;  
300 MOV P2 – deplasare liberă în prima poziție intermediară P2;  
310 MOV P3 + VECZ100 - deplasare liberă deasupra poziției de depozitare (P3);

- 320 MVS P3 – coborâre prin interpolare liniară în stația de depozitare (P3);  
330 DLY 0.5 – temporizare de 0,5 secunde pentru eliberarea carcasei în condiții de siguranță;  
340 HOPEN 1 – eliberarea carcasei în stația de asamblare (P3);  
350 DLY 0.5 – temporizare de 0,5 secunde pentru ridicarea în condiții de siguranță;  
360 MVS P3 + VECZ100 – ridicarea prin interpolare liniară deasupra poziției de depozitare;

### *Preluare piston cu viteză mică*

- 370 SPD MICĂ% - apelarea variabilei viteză mică pentru subrutina de preluare a pistonului;  
380 MOV P4 deplasarea liberă (cu viteza mică, de 25% din viteza maximă) în cea de a 2-a poziție intermediară (între (P3) și (P5));  
390 MOV P12 + VECZ100 – deplasare liberă deasupra poziției de verificare (P12);  
400 MVS P12 – coborâre prin interpolare liniară în poziția de verificare;  
410 DLY 2.0 – temporizare 2 secunde necesară verificării prezenței pistonului în stația de alimentare (P5);  
420 IF PPISTON = 1 THEN GOTO 430 ELSE GOTO 370 – impunerea condiției ca ciclul să poată fi continuat doar în prezența pistonului în stația (P5);  
430 MVS P12 + VECZ100 – ridicare prin interpolare liniară deasupra poziției de verificare piston;  
440 MOV P5 + VECZ100 – deplasare liberă deasupra poziției de preluare piston;  
450 HOPEN 1 – deschidere prealabilă a prehensurului pentru prinderea pistonului;  
460 MVS P5 – coborâre în poziția de preluare piston (P5);  
470 DLY 0,5 – temporizare pentru preluarea în siguranță;  
480 HCLOSE 1 – închiderea prehensurului și apucarea pistonului;  
490 DLY 0.5 – temporizare pentru ridicarea în siguranță;  
500 MVS P5 + VECZ100 – revenire prin interpolare liniară deasupra poziției de preluare piston;  
510 MOV P7 + VECZ100 – deplasare liberă deasupra poziției de depozitare a pistonului în interiorul carcasei (P7);  
520 MVS P7 – coborâre în poziția de depozitare, (P7);  
530 DLY 0.5 - temporizare pentru depozitare în siguranță;  
540 HOPEN 1 – deschidere prehensur și depozitare piston în interiorul carcasei;  
550 DLY 0.5 - temporizare pentru ridicarea în siguranță;

## SIMULAREA CU AJUTORUL INSTRUMENTAȚIEI VIRTUALE A UNOR PROCESE INDUSTRIALE

560 MVS P7 + VECZ100 – revenire prin interpolare liniară deasupra poziției de depozitare;

### **Preluare ansamblu carcasă + piston cu viteză medie**

570 SPD MEDIE% - apelarea vitezei medii (fiind de 55% din viteza maximă) pentru secvența de preluare și depozitare a ansamblului carcasă + piston;

580 MOV P3 + VECZ100 – deplasarea liberă (cu viteză medie) deasupra poziției de preluare ansamblu (P3);

590 HOPEN 1 – deschidere prealabilă a prehensurului pentru preluarea ansamblului;

600 MVS P3 – coborâre prin interpolare liniară în poziția de preluare (P3);

610 DLY 0.5 - temporizare pentru preluarea în siguranță;

620 HCLOSE 1 – închiderea prehensurului pentru preluarea ansamblului;

630 DLY 0.5 - temporizare pentru ridicarea în siguranță;

640 MVS P3 + VECZ100 – ridicare cu ansamblul deasupra poziției de preluare (P3);

650 MOV P1 + VECZ100 – deplasare liberă deasupra poziției de depozitare pe conveyor;

660 MVS P1 – coborâre prin interpolare liniară în poziția de depozitare (P1);

670 DLY 0.5 – temporizare pentru depozitare în siguranță;

680 HOPEN 1 – eliberare ansamblu în poziția (P1) pe conveyor;

690 DLY 0.5 - temporizare pentru ridicare în siguranță;

700 MVS P1 + VECZ100 – revenire prin interpolare liniară deasupra poziției de depozitare (P1);

800 MOV P99 – deplasare liberă în poziția de așteptare (P99);

810 HLT – impunerea condiției ca rularea ciclului de asamblare să se facă o singură dată;

820 END – finalul ciclului.

### **B. Ciclul de asamblare carcasă – piston – arc**

În figura 3.3 sunt ilustrate etapele necesare parcurgerii ciclului de asamblare. Lista de alocare a pozițiilor în acest caz este identică cu lista de alocare specifică ciclului prezentat la punctul A, la care se mai adaugă pozițiile specifice arcului: POS P8 – poziția pentru preluarea arcului de compresiune; POS P9 – poziția pentru depozitarea (cu prehensurul mic) a arcului, peste piston, în interiorul carcasei, în stația de asamblare. Programarea ciclului este prezentată mai jos.

### **Variabilele de tip poziție:**

POS P11 – definirea variabilei poziției P11 (pt. verificare prezență carcasă);

20 DEF POS VECX25 – definirea cotei relative pentru poziția auxiliară P11 față de poziția predefinită P1;

30 VECX25 = (25,0,0,0,0) – atribuirea coordonatelor carteziene pentru cota relativă;

40 P11 = P1 + VECX25 – atribuirea coordonatelor specifice poziției auxiliare P11.



Fig. 3.3 Etapele de asamblare carcasă - piston - arc.

50 DEF POS VECZ100 – definirea unei poziții auxiliare pentru coborârea și ridicarea prehensurului în scopul evitării coliziunilor;

60 VECZ100 = (0,0,100,0,0) – atribuirea coordonatelor specifice poziției de ridicare / coborâre).

### **Definirea variabilelor de tip Boolean:**

70 DEF IO START = BIT,1 – definirea variabilei Booleene START prin atribuirea bitului nr. 1 din registrul de intrări ieșiri, pentru condiționarea pornirii ciclului;

80 DEF IO PCARCASĂ = BIT,2 - definirea variabilei Booleene PREZENȚĂ CARCASĂ, prin atribuirea bitului nr. 2 din registrul de intrări ieșiri;

90 DEF IO PPISTON = BIT,3 - definirea variabilei Booleene PREZENȚĂ PISTON, prin atribuirea bitului nr. 3 din registrul de intrări ieșiri.

### **Definirea variabilelor de tip întreg (Integer):**

100 DEF INTE MICĂ – definirea variabilei de tip nr. întreg pentru viteză redusă de deplasare;

110 MICĂ% = 20 – atribuirea valorii procentuale de 20% din viteza maximă pentru viteza de deplasare redusă;

120 DEF INTE MEDIE - definirea variabilei de tip nr. întreg pentru viteză medie de deplasare;

130 MEDIE% = 60 - atribuirea valorii procentuale de 60% din viteza maximă pentru viteza de deplasare medie;

## CREATIVITATE, INVENTICĂ, ROBOTICĂ

- 140 DEF INTE MARE - definirea variabilei de tip nr. întreg pentru viteză mare de deplasare;  
150 MARE% = 90 - atribuirea valorii procentuale de 90% din viteza maximă pentru viteza de deplasare mare.

### *Pornire ciclu:*

- 160 IF START = 1 THEN GOTO 170, ELSE GOTO 160 – condiționarea pornirii ciclului de apăsarea butonului START;

### *Preluare carcasă cu viteză mare*

- 170 SPD MARE% - stabilirea vitezei mari (90%) pentru preluarea carcasei;  
180 MOV P11 + VECZ100 – deplasarea liberă deasupra poziției P11 pentru verificare;  
190 MVS P11 – coborâre prin interpolare liniară în poziția de verificare P11;  
200 DLY 2.0 – temporizare 2 secunde necesară verificării;  
210 IF PCARCASĂ = 1 THEN GOTO 220, ELSE GOTO 160 – impunerea condiției ca doar dacă este sesizată o carcasă în magazia de alimentare ciclul să continue;  
220 MVS P11 + VECZ100 – revenire prin interpolare liniară deasupra poziției de verificare;  
230 MOV P1 + VECZ100 – deplasare liberă deasupra poziției de preluare (P1);  
240 HOPEN 1 – deschiderea prealabilă a prehensurului;  
250 MVS P1 – coborâre prin interpolare liniară în poziție de preluare (P1);  
260 DLY 0.5 – temporizare de 0,5 secunde pentru preluarea în siguranță a carcasei;  
270 HCLOSE 1 – închiderea prehensurului și preluarea carcasei;  
280 DLY 0.5 – temporizare de 0,5 secunde pentru ridicare în siguranță;  
290 MVS P1 + VECZ100 – ridicare cu carcasa prin interpolare liniară deasupra poziției de preluare;  
300 MOV P2 – deplasare liberă în prima poziție intermediară P2;  
310 MOV P3 + VECZ100 - deplasare liberă deasupra poziției de depozitare (P3);  
320 MVS P3 – coborâre prin interpolare liniară în stația de depozitare (P3);  
330 DLY 0.5 – temporizare de 0,5 secunde pentru eliberarea carcasei în condiții de siguranță;  
340 HOPEN 1 – eliberarea carcasei în stația de asamblare (P3);  
350 DLY 0.5 – temporizare de 0,5 secunde pentru ridicarea în condiții de siguranță;  
360 MVS P3 + VECZ100 – ridicarea prin interpolare liniară deasupra poziției de depozitare;

### *Preluare piston cu viteză mică*

- 370 SPD MICĂ% - apelarea variabilei viteză mică pentru subrutina de preluare a pistonului;  
380 MOV P4 deplasarea liberă (cu viteza mică, de 20% din viteza maximă) în cea de a 2-a poziție intermediară (între (P3) și (P5));  
390 MOV P12 + VECZ100 – deplasare liberă deasupra poziției de verificare (P12);  
400 MVS P12 – coborâre prin interpolare liniară în poziția de verificare;  
410 DLY 2.0 – temporizare 2 secunde necesară verificării prezenței pistonului în stația de alimentare (P5);  
420 IF PPISTON = 1 THEN GOTO 430 ELSE GOTO 370 – impunerea condiției ca ciclul să poată fi continuat doar în prezența pistonului în stația (P5);  
430 MVS P12 + VECZ100 – ridicare prin interpolare liniară deasupra poziției de verificare piston;  
440 MOV P5 + VECZ100 – deplasare liberă deasupra poziției de preluare piston;  
450 HOPEN 1 – deschidere prealabilă a prehensurului pentru prinderea pistonului;  
460 MVS P5 – coborâre în poziția de preluare piston (P5);  
470 DLY 0,5 – temporizare pentru preluarea în siguranță;  
480 HCLOSE 1 – închiderea prehensurului și apucarea pistonului;  
490 DLY 0.5 – temporizare pentru ridicarea în siguranță;  
500 MVS P5 + VECZ100 – revenire prin interpolare liniară deasupra poziției de preluare piston;  
510 MOV P7 + VECZ100 – deplasare liberă deasupra poziției de depozitare a pistonului în interiorul carcasei (P7);  
520 MVS P7 – coborâre în poziția de depozitare, (P7);  
530 DLY 0.5 - temporizare pentru depozitare în siguranță;  
540 HOPEN 1 – deschidere prehensur și depozitare piston în interiorul carcasei;  
550 DLY 0.5 - temporizare pentru ridicarea în siguranță;  
560 MVS P7 + VECZ100 – revenire prin interpolare liniară deasupra poziției de depozitare;

### *Preluare arc cu viteză medie*

- 570 SPD MEDIE% - apelarea vitezei medii (fiind de 60% din viteza maximă) pentru secvența de preluare și depozitare a ansamblului carcasă + piston;  
580 MOV P8 + VECZ100 – deplasarea liberă (cu viteză medie) deasupra poziției de preluare arc (P8);  
590 HOPEN 1 – deschidere prealabilă a prehensurului pentru preluarea arcului;

## SIMULAREA CU AJUTORUL INSTRUMENTAȚIEI VIRTUALE A UNOR PROCESE INDUSTRIALE

- 600 MVS P8 – coborâre prin interpolare liniară în poziția de preluare arc (P8);
- 610 DLY 0.5 - temporizare pentru preluarea în siguranță;
- 620 HCLOSE 1 – închiderea prehensurului pentru preluarea arcului;
- 630 DLY 0.5 - temporizare pentru ridicarea în siguranță;
- 640 MVS P8 + VECZ100 – ridicarea cu arcul deasupra poziției de preluare arc (P8);
- 650 MOV P9 + VECZ100 – deplasarea liberă deasupra poziției P9 pentru depozitarea arcului în interiorul carcasei;
- 660 MVS P9 – coborârea în poziția de depozitare arc (P9);
- 670 DLY 0.5 – temporizare de 0,5 secunde pentru depozitarea în siguranță;
- 680 HOPEN 1 – eliberarea arcului în interiorul carcasei, peste piston;
- 690 DLY 0.5 - temporizare de 0,5 secunde pentru ridicarea în siguranță;
- 700 MVS P9 + VECZ100 – ridicare prin interpolare liniară după eliberarea arcului;

### ***Preluare ansamblu cu viteză medie***

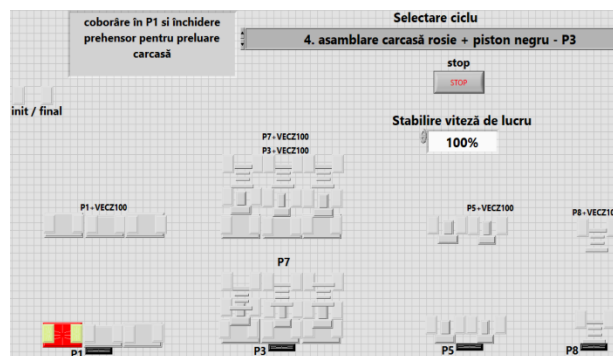
- 710 MOV P3 + VECZ100 – re poziționarea robotului a.î. să ajungă deasupra poziției de preluare a ansamblului;
- 720 HOPEN 1 – deschidere prealabilă a prehensurului;
- 730 DLY 0.5 – temporizarea de siguranță la preluare;
- 740 HCLOSE 1 – preluarea ansamblului din stația (P3);
- 750 DLY 0.5 - temporizarea de siguranță după preluare;
- 760 MOV P3 + VECZ100 – revenire deasupra poziției de preluare;
- 770 MOV P1 + VECZ100 – deplasare liberă deasupra poziției de depozitare ansamblu pe conveyor (P1);
- 780 MVS P1 – coborâre prin interpolare liniară în poziția de depozitare ansamblu;
- 790 DLY 0.5 – temporizarea de siguranță la depozitare;
- 800 HOPEN 1 – eliberarea ansamblului în poziția (P1);
- 810 DLY 0.5 - temporizarea de siguranță după depozitare;
- 820 MVS P1 + VECZ100 – revenire fără ansamblu deasupra poziției de depozitare;

### ***Retragerea cu viteză mare în poziția de așteptare***

- 830 SPD MARE% - apelarea vitezei mari (de 90% din viteza maximă) pentru retragere;
- 840 MOV P99 - retragere cu viteză mare în poziția de așteptare pentru pornirea unui nou ciclu (P99);
- 850 HLT – impunerea condiției ca ciclul de asamblare să se desfășoare o singură dată;
- 860 END – finalul ciclului.

Transpunerea ciclurilor exemplificate în interfața software pentru simulare s-a făcut prin programarea

în mediul de instrumentație virtuală LabVIEW, aspectul interfeței fiind prezentat în figura 3.4. Interfața prezentată este constituită în fereastra-panou, iar programarea acesteia s-a realizat într-o altă fereastră pentru apelarea și interconectarea accesoriilor, funcțiilor și structurilor de programare, în fereastra-diagramă.



**Fig. 3.4** Prezentarea interfeței software pentru simularea ciclurilor de asamblare.

Entitățile ce definesc elementele de asamblare, precum și pozițiile specific parcurgerii unui ciclu s-au definit ca butoane virtuale de tip LED, cu 2 stări logice (stins și aprins). Astfel, aprinderea lor trebuie să marcheze evenimentul curent din timpul parcurgerii ciclului (de exemplu dacă a fost preluată carcasa din postul de alimentare atunci entitatea de tip LED specific acesteia se va colora) (fig. 3.4) etc.

Din punctual de vedere al programării, pentru fiecare ciclu de asamblare, în cadrul unei structuri multicazuale s-a definit un număr de funcții de tip Boolean egal cu numărul de evenimente logice. Fiecare astfel de funcție True-False a fost relaționată de o entitate sau de un grup de entități de tip LED, astfel încât, în funcție de fiecare etapă a ciclului în cauză să se aprindă (să se coloreze) entitățile respective marcând acel eveniment logic. Aceasta a presupus relaționarea fiecărei astfel de funcții la un operator logic (SAU logic), având ca intrări constant de timp cu diferite valori. Acestea au constituit interval de timp care să definească atât momentul, cât și durata desfășurării fiecărui eveniment logic, în parte (a aprinderii entităților marcând evenimentul). De exemplu s-a stabilit că pentru a se marca evenimentul preluării unei carcase de culoare roșie este nevoie de un interval de timp cuprins între constantele 7 și 18. Aceste constante de timp, la rândul lor au fost relaționate de un selector cu 3 cazuri, specific regimului de viteză, mai exact un increment al constantei de timp să corespundă unei durate anume. Spre exemplu, această durată a fost stabilită la 625 ms pentru viteza de 25% din cea maximă posibilă, respective la 0,5 s pentru viteza de 50% din

cea maximă. Pentru viteze mari (75% și 100% din viteza maximă) au fost stabilite duratele de 375 ms, respectiv de 0,25 s, după cum se poate observa în exemplul din figura 3.5. În acest exemplu sunt prezentate toate cele 4 setări specific regimurilor de viteză. În figura 3.6 este prezentată o secvență în care se arată modul în care s-a folosit o funcție Booleană pentru relaționarea constantelor de timp, specific definirii intervalului pentru marcarea evenimentului preluare carcasă roșie.

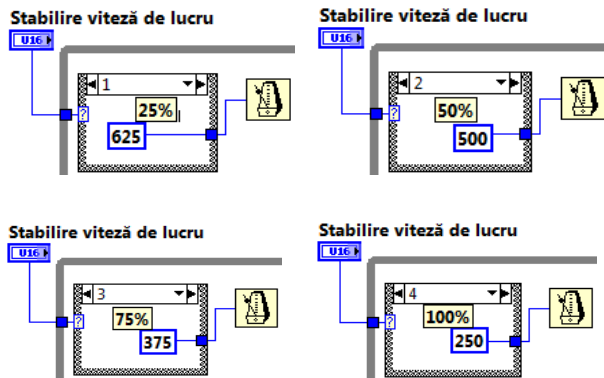


Fig. 3.5 Programarea constantelor de timp pentru durata unui increment ce stă la baza stabilirii intervalului de timp pentru marcarea unui eveniment logic.

Un alt aspect legat de programarea succedării evenimentelor logice se referă la necesitatea folosirii unei funcții de tip ciclu în buclă închisă While –Loop, al cărui index să poată fi relaționat de valorile ce definesc intervalele de timp (figura 3.6)

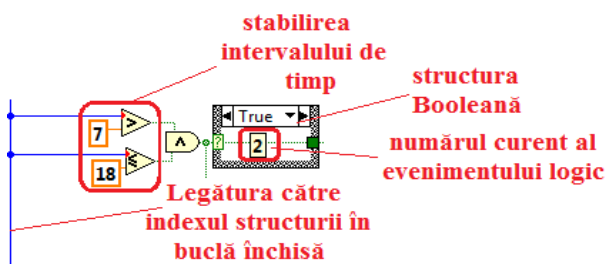


Fig. 3.6. Exemplificarea unei secvențe de program pentru evenimentul nr. 2 specific simulării preluării unei carcase roșii din stația de alimentare.

În același mod s-a procedat pentru fiecare eveniment logic, evident modificându-se valorile numerice ale constantelor de timp, astfel încât acestea să stabilească momentele și intervalele de timp specific fiecărui eveniment. Un exemplu privind secvențele de programare specific aprinderii LED-urilor ce marchează toate etapele specific evenimentului preluare carcasă roșie este prezentat în figura 3.7.

Pentru aceasta s-a folosit o structură secvențială conținând un număr egal de secvențe cu numărul fazelor ce constituie evenimentele logice.

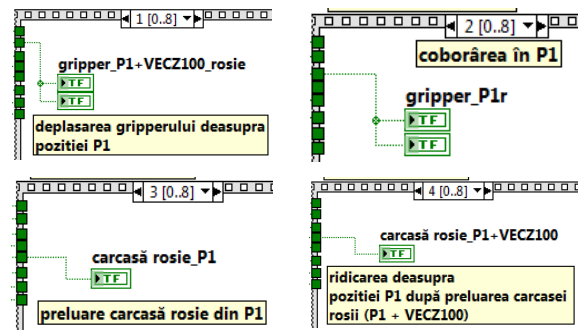


Fig. 3.7 Secvențele de program pentru aprinderea succesivă a entităților ce marchează evenimentul preluare carcasă roșie din stația de alimentare .

Un astfel de algoritm prin care s-au definit toate intervalele necesare de timp, relaționate prin structuri Booleene și secvențiale pentru marcarea succesivă a evenimentelor logice a fost programat pentru fiecare caz în parte. Mai exact, pentru fiecare tip de ciclu simulate, în parte s-a procedat la programarea unui astfel de algoritm. Cu toate că, pentru fiecare caz, în parte, algoritmul de programare a fost asemănător, alte entități de tip LED virtual au fost definite, pentru ca interfața de simulare să poată permite vizualizarea tuturor tipurilor de cicluri. Mai mult, fiecare ciclu simulate trebuie să marcheze succesiv evenimentele logice specific diferitor mărimi și culori de carcasă, respectiv diferitor mărimi și culori de pistoane.

Utilizarea interfeței pentru simularea unui anumit ciclu presupune parcurgerea următorilor pași: selectarea din listă a ciclului ce se dorește a fi simulate. Spre exemplu dacă se dorește simularea unui ciclu prin care să se asambleze un piston negru într-o carcasă roșie, atunci se va alege din listă opțiunea nr. 4 (figura 3.8).

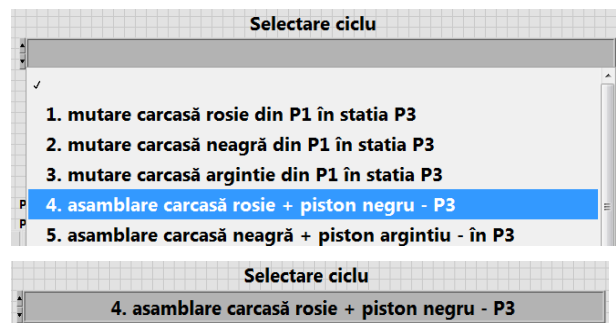


Fig. 3.8. Exemplificarea modului prin care se alege din listă ciclul pentru asamblarea unei carcase roșii cu un piston negru.

Dacă se dorește simularea unui ciclu prin care să se asambleze un piston argintiu cu un piston negru și cu un arc, atunci din listă se va alege opțiunea nr. 9 (figura 3.9).

Următoarea etapă este aceea prin care se selectează regimul de viteză. În figura 3.10, spre exemplu se arată cum se procedează pentru alegerea vitezei minime (25% din viteza maximă).



## SIMULAREA CU AJUTORUL INSTRUMENTAȚIEI VIRTUALE A UNOR PROCESE INDUSTRIALE

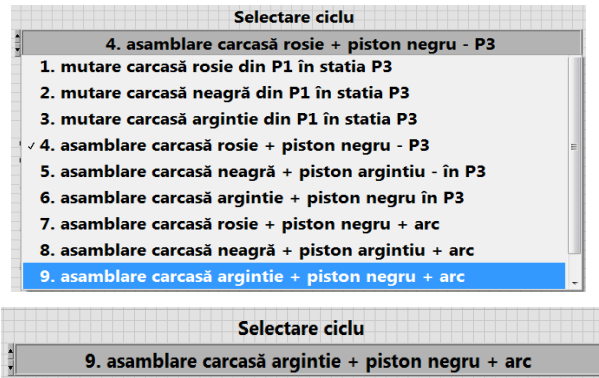


Fig. 3.9. Exemplificarea modului prin care se alege din listă ciclul pentru asamblarea unei carcase argintii cu un piston negru și cu un arc

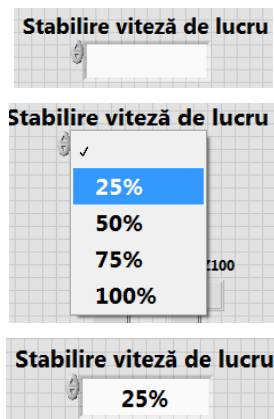


Fig. 3.10. Stabilirea vitezei de 25% din viteza maximă.

Selectarea celorlalte regimuri de viteză se face în mod similar.

În figurile 3.11, 3.12 și 3.13 este exemplificată succedarea a 3 evenimente, după cum urmează: preluare carcasă roșie, depozitare carcasă roșie și preluare piston negru.

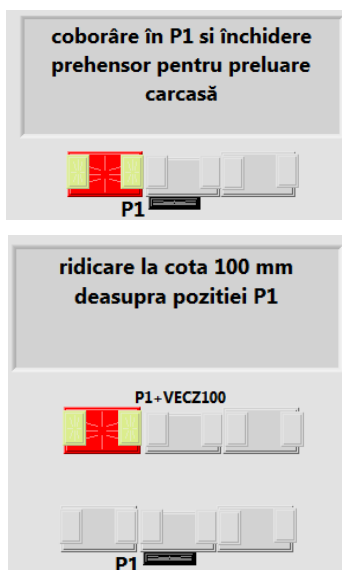


Fig. 3.11. Exemplificarea marcării evenimentului preluare carcasă roșie din postul de alimentare P1.

În mod similar pot fi exemplificate rând pe rând toate celelalte evenimente, atât pentru ciclul în cauză (asamblare carcasă roșie cu piston negru), cât și pentru celelalte cicluri de asamblare.

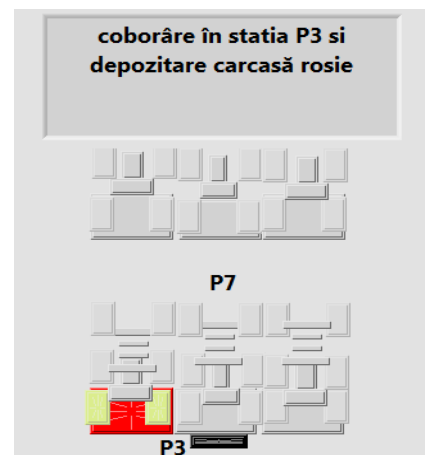
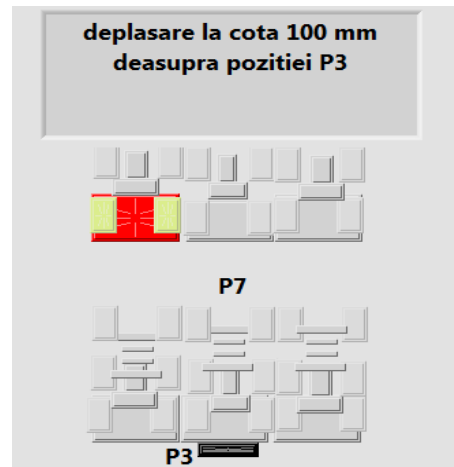


Fig. 3.12. Exemplificarea marcării evenimentului depozitare carcasă roșie din postul de asamblare P3.

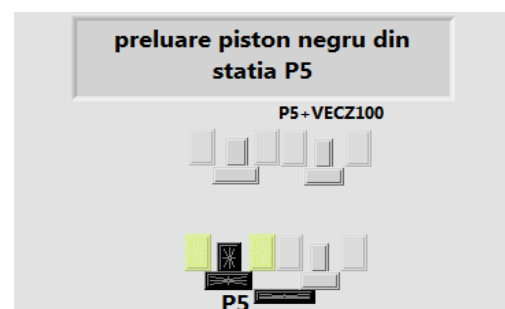


Fig. 3.13. Exemplificarea marcării evenimentului preluare piston negru din postul P5.

Finalizarea simulării ciclului se realizează în mod automat, însă dacă se dorește întreruperea mai iute a simulării se apasă butonul virtual STOP (figura 3.14).

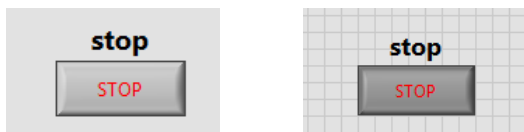


Fig. 3.14 Finalizarea forțată a unui ciclu de simulare

## 4. CONCLUZII

Odată programată interfața pentru simularea celor 9 cicluri de asamblare, aceasta a fost testată și apoi îmbunătățită, în așa fel încât animația grafică să fie una clară și ușor de înțeles. Mai mult, s-a insistat pe sincronizarea afișării mesajelor-text ce descrie fiecare eveniment logic în parte cu marcarea aceluși eveniment.

Până în prezent interfața a fost prezentată studenților din anul 3, specializarea Mecatronică, aceștia fiind foarte receptivi și implicați în procesul educațional de înțelegere a modului de programare a ciclurilor de asamblare. În consecință, s-a putut observa o eficientizare sensibilă a procesului de

acumulare de cunoștințe și competențe practice în privința programării unor cicluri de producție simple și uzuale pe stația robotizată a liniei FESTO Didactic.

## BIBLIOGRAFIE

- [1] Almad, T, Samy, E., *Medical Robotics*. Prototyping of Robotic Systems: Applications of Design and Implementation, First Edition, Publisher: IGI Global, 2012, USA.
- [2] Spyros G., Eleftheria, M., *Classification of domestic robots*, Conference: ARSA, 1<sup>st</sup> Virtual International Conference Advanced Research in Scientific Areas, decembrie 2012.
- [3] Zhong, J., Ling, C., Cangelosi, A, Lotfi, A, Liu, X., *On the gap between domestic robotic application and computational Intelligence*, Electronics Review, China, 2021.
- [4] Hägele, M., Klas, N., J. Norberto Pires, N., *Industrial Robotics*, Research Gate, decembrie 2007.
- [5] Karabegović, I., Catovic, F., Hodžić, D., *Industrial Robots Applications in the Process Industry*, Trends in the Development of Machinery and Associated Technology, TMT Conference, Istanbul, Turcia, ianuarie, 2008.
- [6] Shivraj, N., *Simulation in Robotics*, National Conference on Recent Advances in Manufacturing Engineering & Technology (RAMET 2011)At: VNRVJIET HyderabadVolume: 1, ianuarie 2011.
- [7] Lopez, N., Antonio, R., Josechu, G., *Simulation tools for active learning in robot control and programming*, EAEEIE Annual Conference, 2009, IEEE Explore, iulie, 2009.

---

## Despre autori

Sef lucr. dr. ing. **Barbu BRAUN**

Universitatea „Transilvania” din Brașov, Brașov, România

Braun Barbu este absolvent al Facultății de Mecanică, specializarea Mecatronică, promoția 2000. A urmat cursurile de Studii Aprofundate (specializarea Structuri de Mecanică Fină pentru Managementul Sistemelor Tehnice) în cadrul aceleiași facultăți. Din 2001, este angajat al Universității Transilvania din Brașov, la Facultatea de Inginerie Mecanică, iar din 2009 este Doctor în domeniul Inginerie Mecanică. În perioada 2010 – 2013 a urmat un stagiu post-doctoral, în cadrul programului POS-DRU, având ca tematică biomecanica și modelarea susținătorilor plantari. A publicat în calitate de autor sau coautor: 7 cărți și peste 145 de articole în reviste de specialitate și în volumele unor conferințe internaționale și naționale.

Sef lucr. dr. ing. **Corneliu DRUGĂ**

Universitatea „Transilvania” din Brașov, Brașov, România

Drugă Corneliu este absolvent al Facultății de Mecanică, specializarea Mecanică Fină (promoția 1998). A urmat cursurile de Studii Aprofundate (specializarea Structuri de Mecanică Fină pentru Managementul Sistemelor Tehnice) în cadrul aceleiași facultăți. Din 2001, este angajat al Universității Transilvania din Brașov, la Facultatea de Inginerie Mecanică, iar din 2011 este Doctor în domeniul Inginerie Mecanică. Din 2013, este șef de lucrări la Departamentul de Design de Produs, Mecatronică și Mediu din cadrul Universității Transilvania din Brașov. A publicat în calitate de autor sau coautor: 9 cărți și peste 130 de articole în reviste de specialitate și în volumele unor conferințe internaționale și naționale.