Etapa 4-Raport Stiinţifico-Tehnic

Rezumat Etapa 4 - Prototipul 1 (2DW/2FW Cirrus Power Wheelchair): Conducerea in timp-real so testarea scaunului cu rotile, la sectia de neurologie Sp. clinic de urgenta “Sf. Apostiol Andrei”Galati. Prototipul 2 (2DW/1FW Pioneer 3DX echipat cu Pioneer 6-DOF Arm si 2DW/2SW PatrolBot echipat cu Cyton 6-DOF Arm): Integrarea sistemelor robotice echipate cu manipulatoare in linia de posturi autonome de asamblare generalala intreprinderea "DACIA RENAULT", Pitesti; Prototipul 3(4DW/SW Autonoumous Ominidirectional Vehicle SEEKUR echipat cu 6-DOF SEEKUR outdoor manipulator): Utilizarea vehiculului autonom echipat cu manipulator la operatii de transport si manipulare la"STICLA"Avrig.

Activitate 4.1.-Testarea Prototipului 1 (2DW/2FW Cirrus Power Wheelchair) in spitalul de urgenta din Galati pe persoane cu dizabilitati neuro-loco motorii severe. Testarea sistemului de navigatie, bazata pe video- biometria fetei si a ochiului.

Conducerea în timp real utilizând biometria ochiului: Pentru testatea în timp real s-a ţinut cont de o serie de factori cum ar fi performanțele camerei video folosită, puterea de procesare a calculatorului instalat pe scaunul electric mobil, elementele perturbatoare în identificarea irisului, etc. Datorită numărului redus de etape parcurse pentru extragerea irisului din imaginile video achiziționate in timp real, soluția propusă este mult mai rapidă în comparaţie cu alte soluții propuse in literatură. Sistemul permite portabilitatea acestuia pe orice platformă robotică şi controlul acestuia la distanţă. Mai jos sunt prezentate rezulatele în timp real pentru cele trei direcții de navigare: direcția ÎNAINTE și STOP Fig.1, direcția STÂNGA Fig.2 și direcția DREAPTA Fig.3.

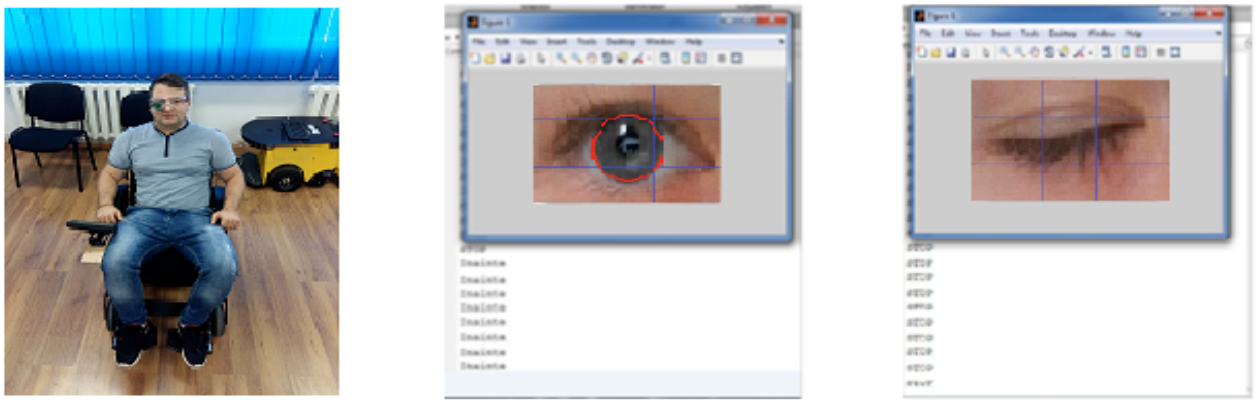

Fig.1. Rezultatele în timp real obținute pentru direcția ÎNAINTE și STOP

Fig.2. Rezultatele în timp real obținute pentru direcția STÂNGA

Fig.3. Rezultatele în timp real obținute pentru direcția DREAPTA

Pentru validarea sistemului odometric implementat se foloseşte un joystick virtual cu care se va face conducerea scaunului cu rotile şi se va observa evoluţia parametrilor odometrici.

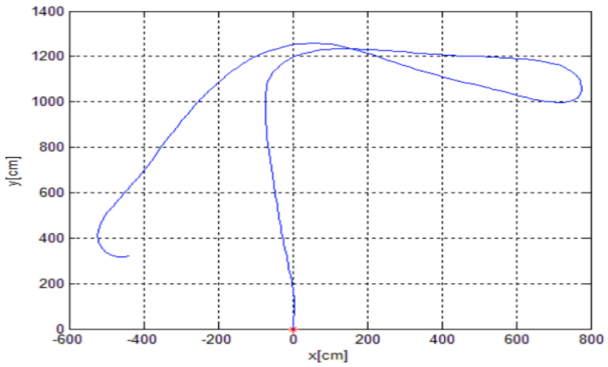

Fig.4. Deplasare cu ajutorul unui joystick pe o traiectorie

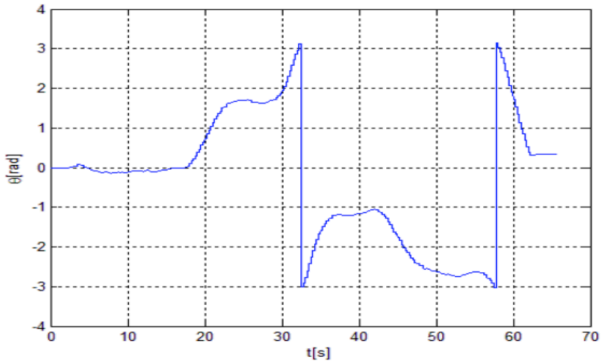

Fig.5. Evoluţia unghiului θ de-a lungul traiectoriei

Implementarea în timp real şi testarea în regim de laborator a structurii de conducere sliding-mode (Fig. 6) şi super-twisting sliding-mode control (fig. 7), pentru urmărirea unei traiectorii impuse a Prototipului 1.

S-a efectuat un studiu comparativ între două cele mai cunoscute metode (de tip sliding-mode) folosite pentru a controla un robot mobil cu două roți motoare pentru a urma o traiectorie dorită. Experimentele în timp real sunt efectuate pe un robot mobil real cu parametri incerți și perturbații externe pentru ambele metode de control.

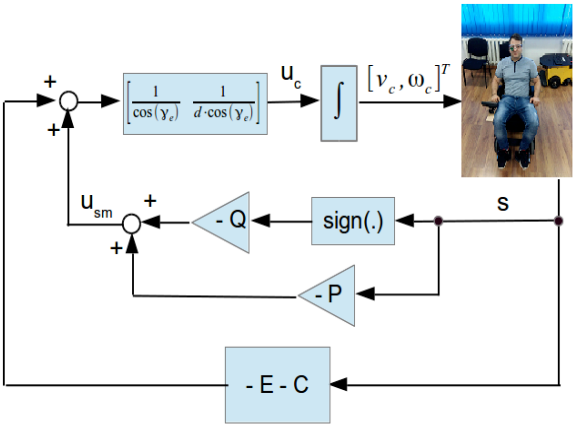

Fig.6. Schema bloc pentru conducerea sliding-mode

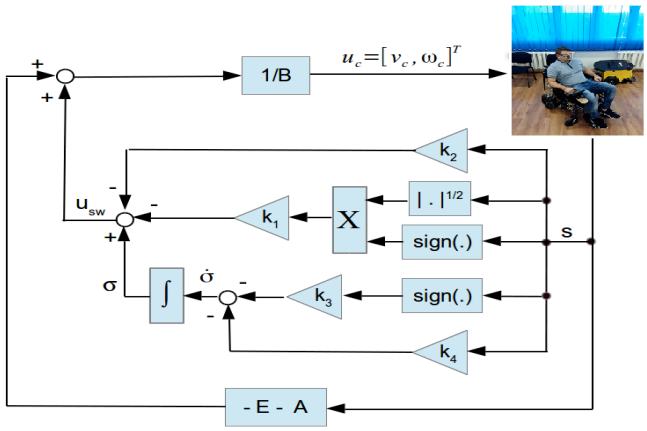

Fig.7. Schema bloc pentru "Super Twist Sliding-Mode Control"

Rezultatele obținute arată performanțe îmbunătățite în cazul utilizării *super-twisting sliding-mode control* în ceea ce privește scăderea oscilațiilor și creșterea robusteții la perturbații în comparație cu *sliding-mode control*. Mai mult decât atât, schema de control în cazul abordării STSMC este mai simplă și mai ușor de implementat în timp real pentru un robot mobil cu roți. Prima traiectorie testată este una circulară (cazul I), iar cea de-a doua este liniară (cazul II).În Tab.1. sunt date condițiile inițiale pentru cele două cazuri. În Tab.2. sunt enumerați parametrii (celor două controllere) utilizați în testarea în timp real.

Tab.1. Condiţiile iniţiale

Caz	v_d [m/s]	ω_d [rad/s]	x_0 [m]	y_0 [m]	θ_0 [deg]	Lh_d [m]	ψ_d [deg]
I-cerc	0.2	-0.2	0.5	0.0	0.0	0.5	-3π/4
II-linie	0.2	0.0	0.5	0.2	π/4	0.5	3π/4

Tab.2. Parametri controller-elor

	q_1	q_2	p_1	p_2	k_{Lh}	k_v	k_0
SMC	0.05	0.01	0.1	0.05	1.5	0.5	0.01
	k_1	k_2	k_3	k_4	k_{Lh}	k_v	k_0
STSMC	0.05	0.1	0.004	0.005	1.5	0.5	0.01

Algoritmii de control de tip *sliding-mode şi super-twisting sliding-mode control* sunt realizaţi în limbajul C++ şi rulează în timp real cu un pas de eşantionare Ts = 100ms pe un calculator.

Rezultatele experimentelor în timp real sunt prezentate în figurile următoare:

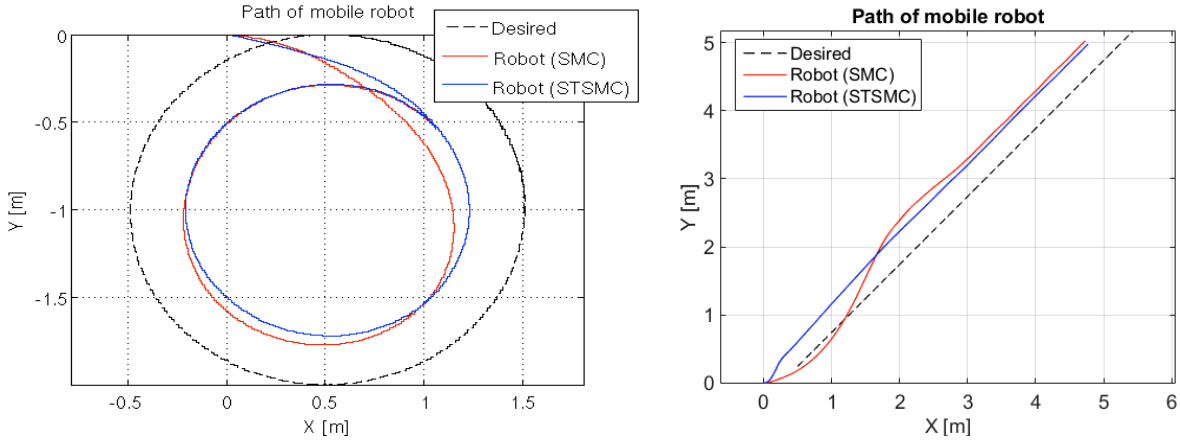

Fig.8. Rezultatul în timp real pentru cazul I şi II.

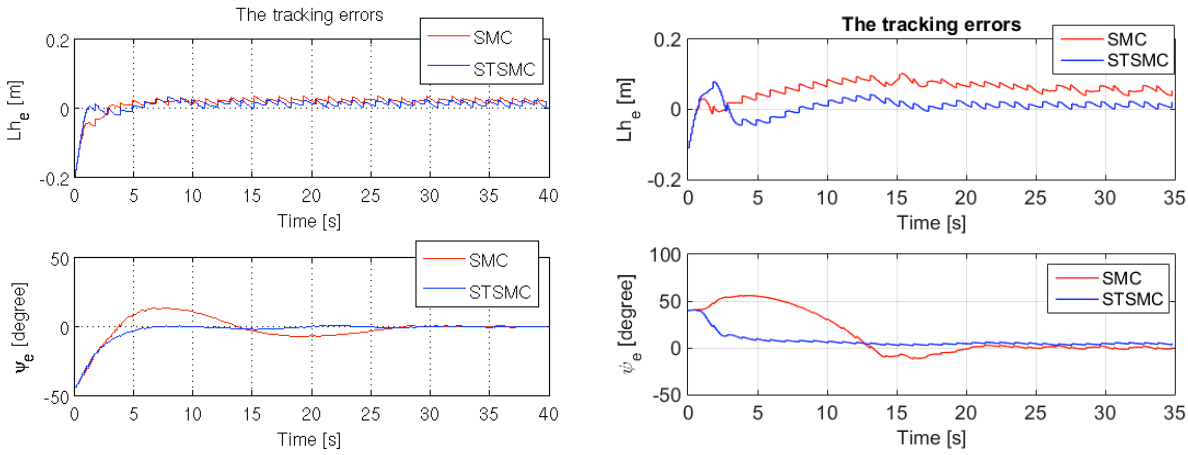

Fig.9. Erorile de urmărire **- Caz** I şi Caz II.

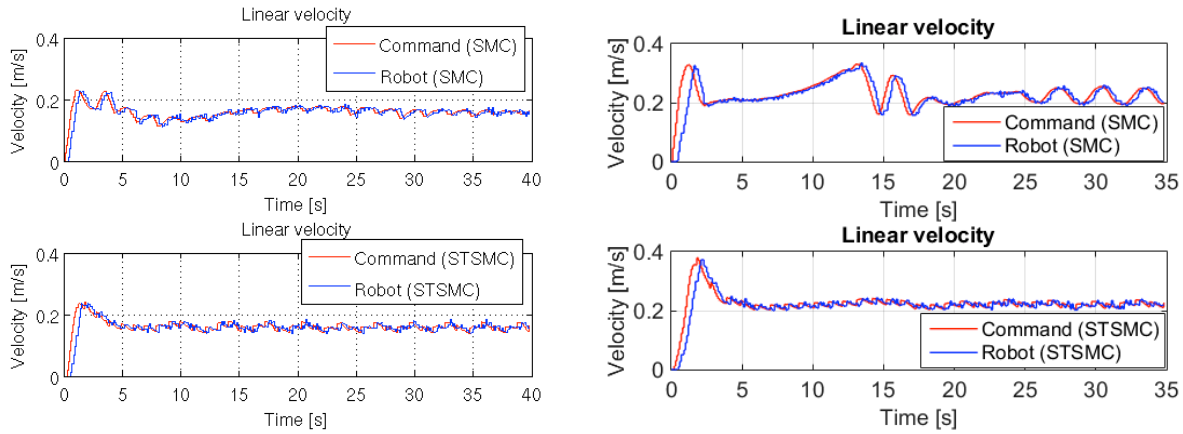

Fig.10. Rezultatele în timp real **pentru** viteza liniară **- Caz I şi Caz II.**

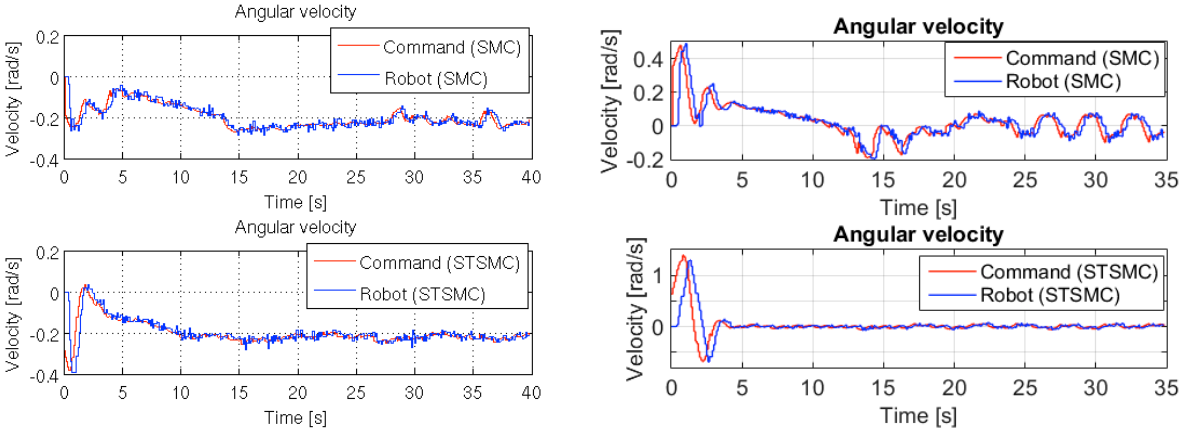

Fig.11. Rezultatul în timp real **pentru** viteza unghiulară **- Caz I** şi **Caz II.**

O sinteză a rezultatelor experimentelor în timp real pentru STSMC cât şi pentru SMC este prezentată în Tab.3 în care se pot observa avantajele oferite de STSMC.

Tab.3. Rezultatele experimentelor în timp real pentre STSMC cât şi pentru SMC

Controller	Caz I - cerc			Caz II - linie		
	Lh_e	ψ_e	θ_e	Lh_e	ψ_e	θ_e
SMC	0.036	0.154	0.218	0.063	0.464	0.635
STSMC	0.025	0.138	0.141	0.024	0.179	0.193

O clasă de perturbaţii inportantă pentru roboţi mobili este variaţia sarcinii utile. Pentru a evidenţia robusteţea conducerii STSMC s-au efectuat alte experimente cu şi fără sarcină utilă suplimentară. În acest caz, vitezele liniare şi unghiulare ale traiectoriei dorite au fost: v_d = 0,5 m / s şi ω_d = - 0,5 rad/s.

În Fig.12. şi Fig.13. sunt reprezentate viteza de comandă şi vitezele reale ale robotului mobil pentru cele două cazuri (fără şi cu sarcină utilă). Aceste cifre arată doar primele secunde pentru viteze lineare şi unghiulare, deoarece pentru restul experimentului diferenţele dintre ele sunt nesemnificative. În Fig.10 se observă diferenţele dintre vitezele liniare ale robotului şi comenzile datorate întârzierii (*~300ms*) şi inerţiei robotului.

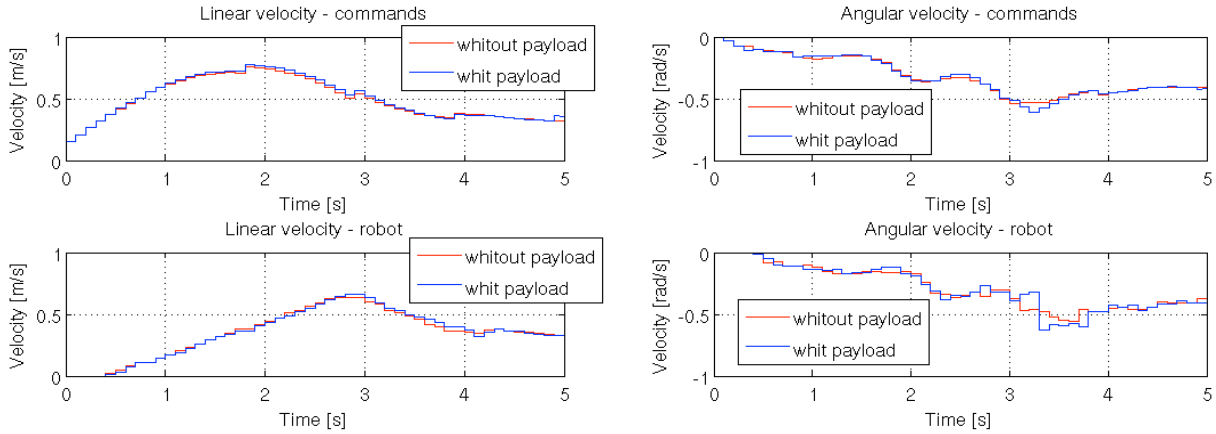

Fig.12. Viteza liniară de comandă şi viteza liniară reală **Fig.13. V**iteza unghiulară de comandă şi viteza unghiulară reală

Activitatea 4.2-Integrarea si testarea sistemelor robotice autonome din gama Prototipului 2 (2DW/1FW Pioneer 3DX, echipat cu Pioneer 6-DOF Arm si 2DW/2SW PatrolBot echipat cu Cyton 5-DOF Arm) in liniile asamblare preliminară şi generală aferente modelelor Dacia Logan şi Sandero la intreprinderea "DACIA-RENAULT", Pitesti;

Integrarea si testarea sistemelor robotice autonome din gama Prototipului 2 a presupus:

1) *modificarea platformei de lucru* – Raspberry PI 3 care funcţiona doar ca releu pentru datele expediate de PC către robot a fost înlocuită de LattePanda; considerentele care au condus la acest demers vizând *utilizarea sistemului de operare Windows*, *o putere de calcul ridicată* care a permis şi conectarea unei camere video ce poate transmite date în timp real astfel încât o posibilă recunoaştere de forme să fie implementată în viitor, *rularea codului executabil direct de pe placă* (independenţă faţă de PC/Laptop), *comanda de la distanţă a braţului robotic prin intermediul unei conexiuni de tip TeamViewer*;
2) *programarea activităţii braţului robotic Mover6* astfel încât acesta să fie comandat de la distanţă sau să execute programe independent de calculatorul de control acesta fiind utilizat doar pentru recepţia datelor din sistem;

Arhitectura hardware a sistemului propus s-a modificat astfel în etapa a IV-a a proiectului prin înlocuirea calculatorului Raspberry PI cu LattePanda şi eliminarea modulului WiFi de emulare a comunicaţiei seriale sau de conectare peer-to-peer (Fig. 14).

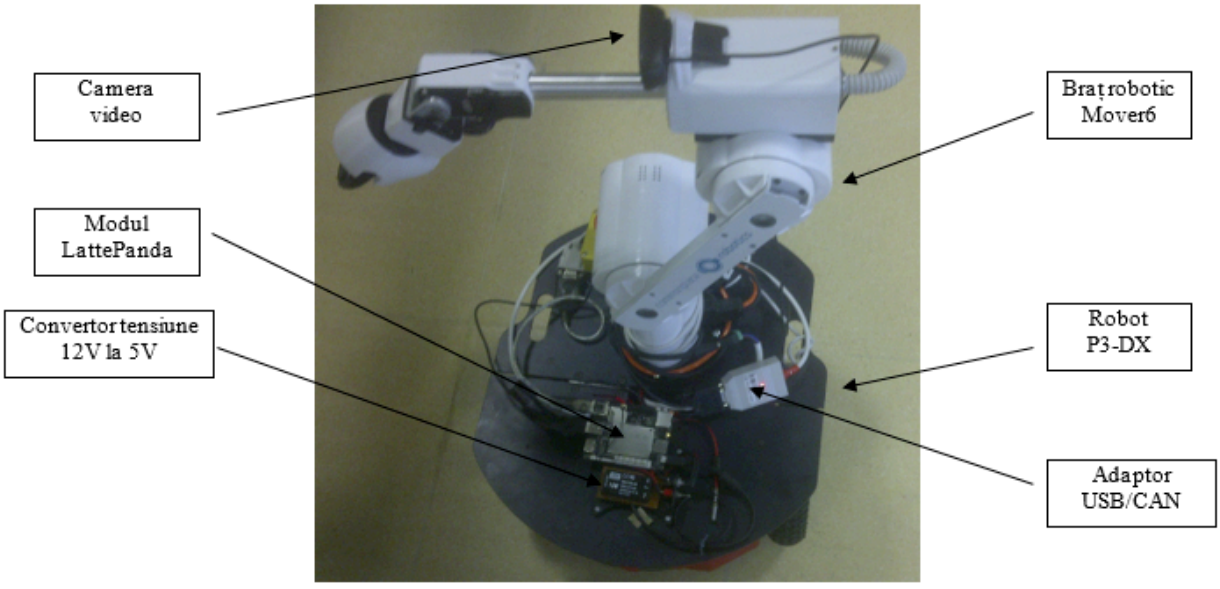

Fig.14. Schemă bloc a Sistemului robotic autonom 2DW/2FW echipat cu manipulator 6-DOF şi calculator LattePanda

Fig. 15 oferă un punct de vedere asupra arhitecturii hardware implementate în etapa a IV-a.

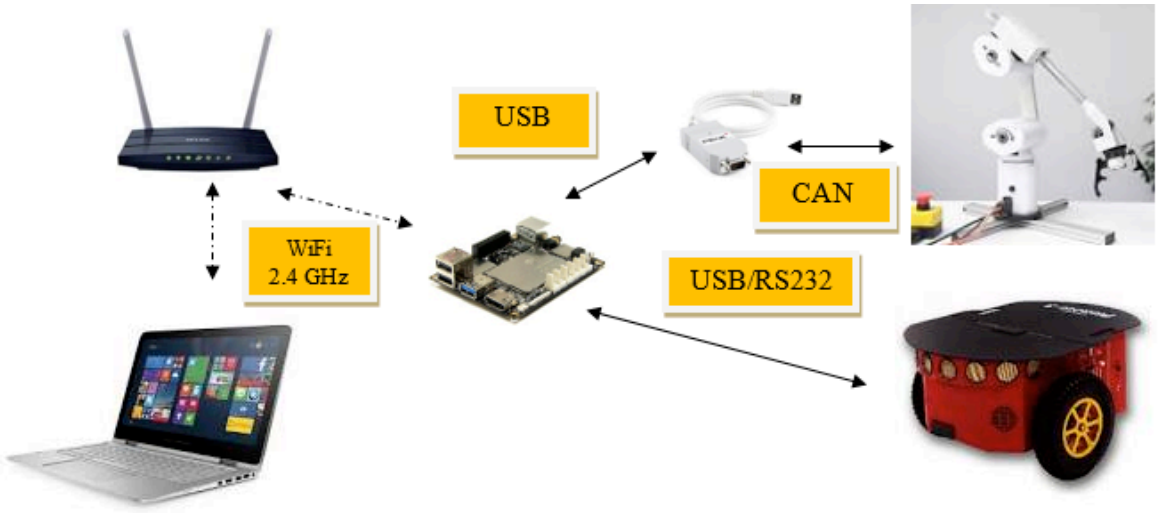

Fig. 15 Arhitectură hardware prototip 2 în etapa 4

Programele rulează direct pe calculatorul oferit de LattePanda comandându-se astfel robotul Pioneer 3DX și brațul robotic Mover6, modulul WiFi al plăcii transferând date de tip rezultate execuție program, prin intermediul unui router WiFi către un PC Desktop/Laptop pe care poate rula o interfață Web. Totodată, prin intermediul unei conexiuni prin TeamViewer, sub-sistemele - robot și braț robotic pot să fie comandate pentru deplasare / manipulare la puncte de lucru prestabilite.

Integrarea si testarea prototipului în mediu industrial

- *robotul 3DX a fost testat prin comandă de la distanță pentru deplasarea într-o zonă de producție* – la 3 puncte de lucru (programul a rulat de pe LattePanda comanda fiind oferită prin TeamViewer);folosindu-se funcțiile ARIA și răspunsuril de la encodere; traseul descris de robot este oferit în figura 17;
- *robotul 3DX a fost testat prin execuția unui program cu date de traseu prestabilite pentru deplasarea într-o zonă de producție* – la 3 puncte de lucru (programul a rulat de pe LattePanda comanda fiind oferită un fișier de configurare local sistemului) folosind sliding-mode rezultatele fiind prezentate în figurile 18 și 19 (erorile față de traseul optim pentru axele X, Y);
- *prototipul a fost testat prin deplasare pe traseul pentru cele 3 puncte de lucru, în calea acestuia fiind poziționat un obstacol* – în figura 20 este prezentat traseul urmat de robot iar în figura 21 erorile apărute față de traseul clasic, prin revenirea după depășirea obstacolului;
- *brațul robotic Mover6 a fost testat prin manipularea automată a 3 obiecte în cele 3 posturi de lucru* – manipularea automată a fost realizată prin aplicația ce rulează pe LattePanda în fereastra de timp oferită de robotul Pioneer 3DX când acesta ajunge în postul de lucru; în figura 21 se pot observa erorile de poziționare;

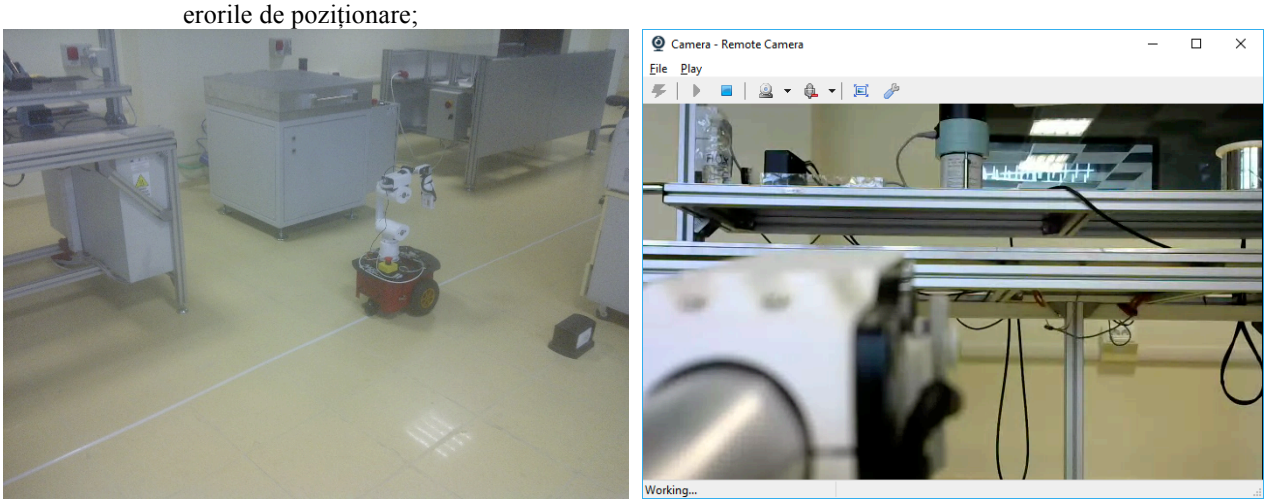

Fig. 16. Prototipul 2 în mediul industrial, captură video (cameră robot)

Prototipul a parcurs un traseu impus (figura 4) suprafața acoperită fiind de 2900 x 7200 mm^2 ajungând în final în punctul de start, iar brațul robotic Mover6 a trebuit să execute 3 programe prin care a manipulat 2 obiecte, sub 400g fiecare, (în posturile de lucru 1 și 2 există câte un obiect care trebuie deplasat în postul imediat următor). Bancul de piese este la limita maximă a deschiderii articulațiilor prototipului, acest aspect putând să fie vizualizat în figura 22.

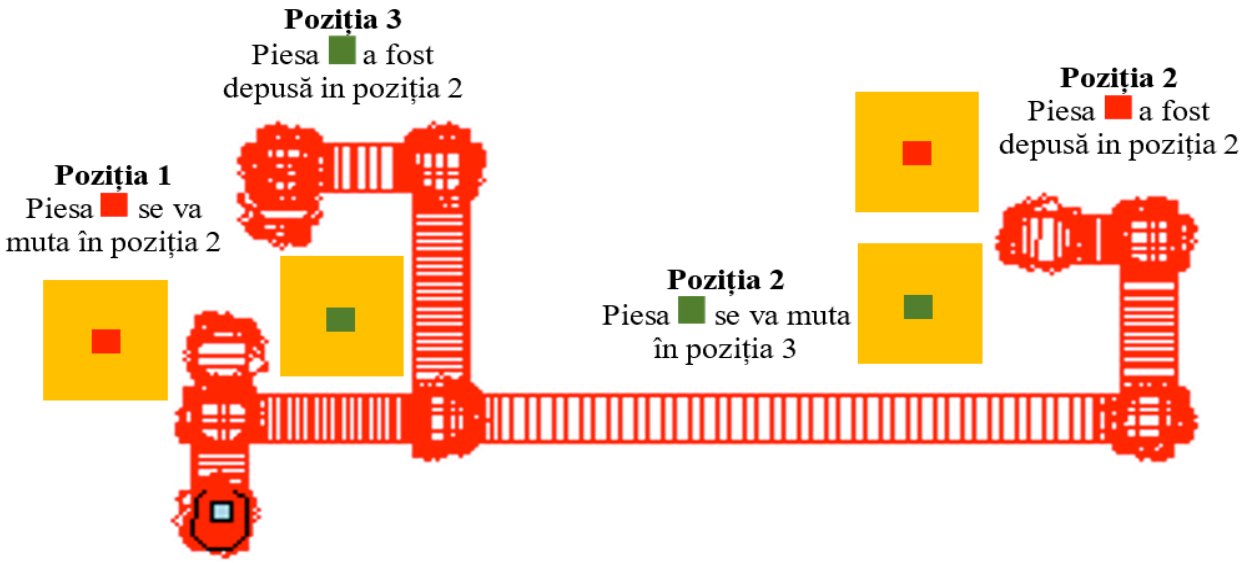

Fig.17. Traseu robot în hala de producție

Robotul 3DX a fost testat prin comandă de la distanță pentru deplasarea într-o zonă de producție – la 3 puncte de lucru (programul a rulat de pe LattePanda comanda fiind oferită prin TeamViewer / Remote Utilities Viewer). În figurile care urmează se pot observa, eroarea de urmările pe axa X, eroarea de urmărire pe axa Y și erorile unghiulare pentru cele 2 situații control ARIA prin encodere respectiv control sliding-mode. Pentru control sliding-mode s-au folosit Q1=0.05, Q2=0.5, P1=0.5, P2=0.75 (în legea de control), k0=30, k1=0.75, k2=25 (pentru suprafețele de comutație), v_d = 0.5m/s, 200 de iterații, ($w=\omega_d=\phi_d'=0$).

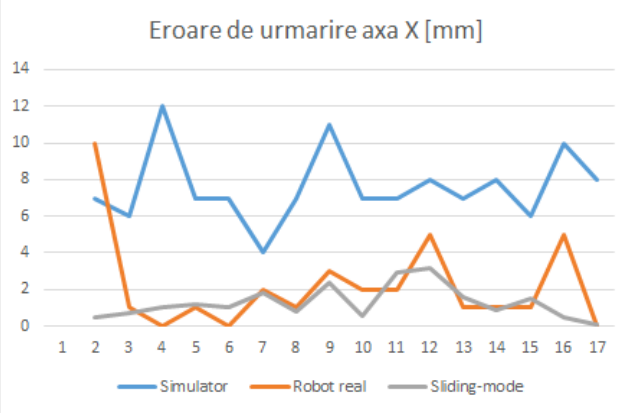

Fig.18. Eroarea de urmărire pe axa X

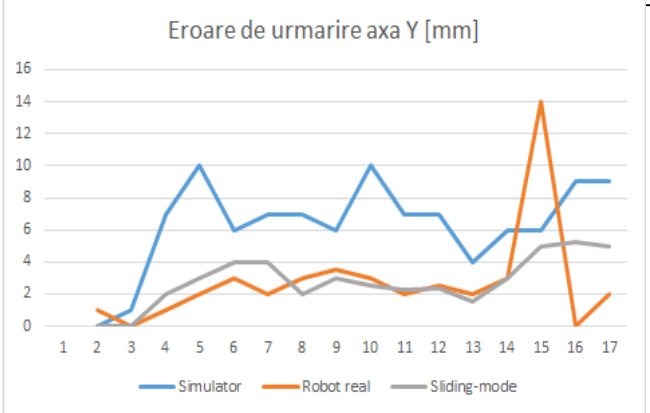

Fig.19. Eroarea de urmărire pe axa Y

Se poate observa că eroarea maximă, în toate cazurile, este obținută pentru simulator iar eroarea cea mai mică prin utilizarea sliding-mode. Deplasarea automată pe traseu, fără control în buclă de reacție (doar utilizând encoderele robotului) se dovedește destul de nefericită pentru viteze ridicate ale robotului, fiind datorată în special unghiului de întoarcere și calității suprafeței de alunecare. Sliding-mode oferă erori mici dar setarea parametrilor de control este dificilă.

Implementarea în timp real și testarea sistemului de navigație a Prototipului 2

Prototipul a fost testat prin deplasare pe traseul pentru cele 3 puncte de lucru, în calea acestuia fiind poziționat un obstacol – în figura 7 este prezentat traseul urmat de robot și erorile apărute față de traseul clasic, prin revenirea după depășirea ostacolului. S-a folosit comanda automată cu control fără algoritm, doar encodere și controlul prin sliding-mode. Obstacolul a fost prezent în fața robotului atât pe traseul dinspre postul de lucru 1 spre 2 cât și pe traseul invers. În ambele situații datorită prezenței unei singure zone libere acesta a luat-o spre stânga respectiv spre dreapta (la întoarcere). Erorile au valori pozitive și negative astfel încât în grafice a fost considerat modulul acestora (25 puncte de măsurare).

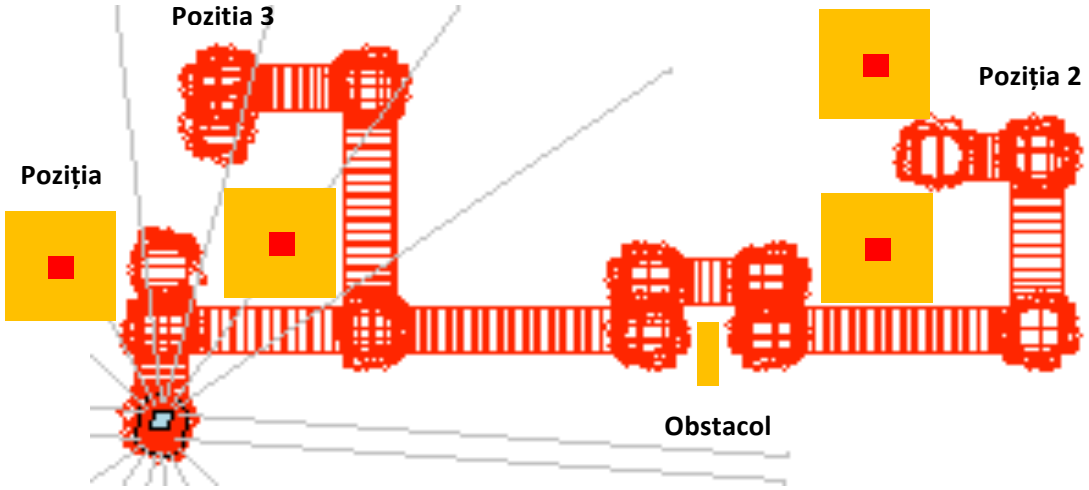

Fig. 20. Deplasare automată pe traseu predefinit, utilizând sistemul de navigație (sonar) – traiectorie descrisă de robot

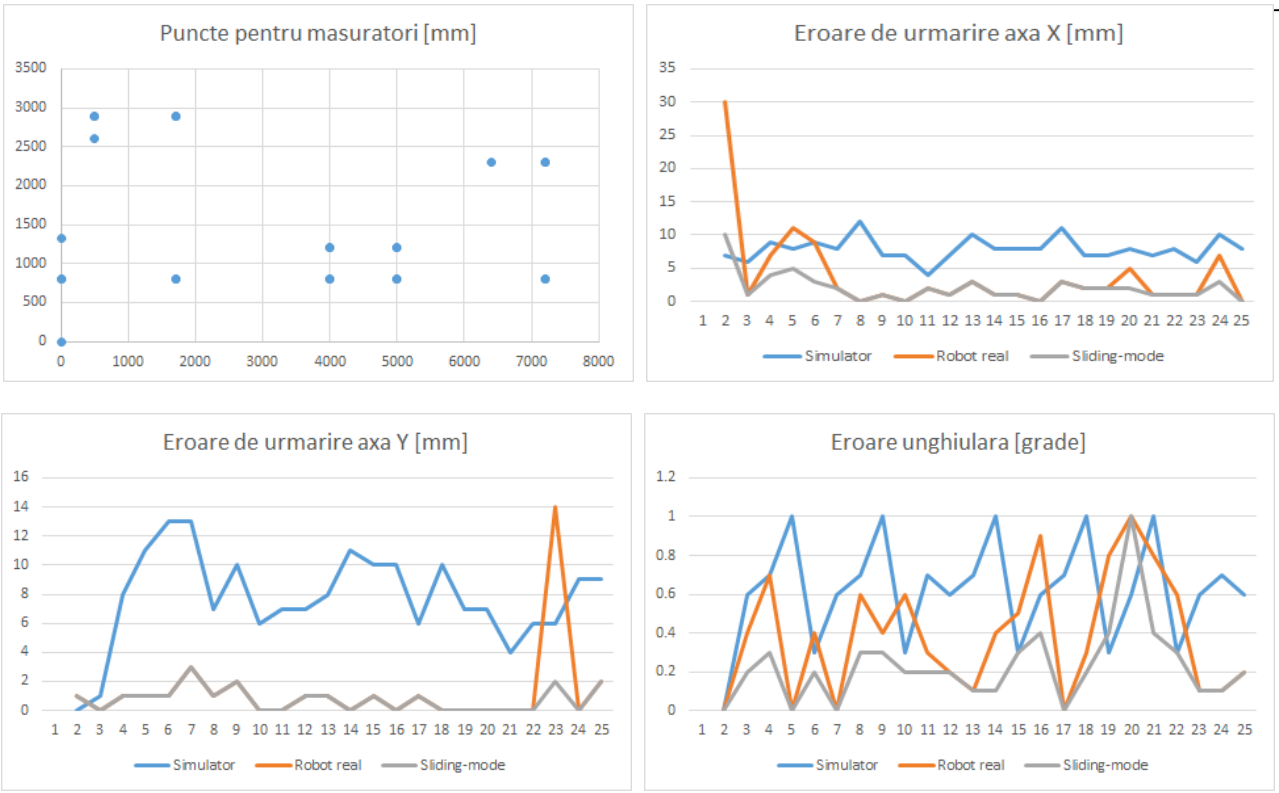

Fig. 21. Erori de urmările pe axele X, Y și eroare unghiulară (simulator și robot real)

Implementarea în timp real și testarea acțiunii de poziționare a manipulatorului care echipează Prototipul 2

Brațul robotic Mover6 a fost testat prin manipularea automată a 3 obiecte în cele 3 posturi de lucru – manipularea manuală a fost efectuată de la distanță (TeamViewer); în tabelul 1 se pot observa erorile de poziționare. Eroarea pentru fiecare motor a fost calculată prin mediere pentru 5 rezultate obținute rulând programul propus pentru fiecare post de lucru. Rezultatele sunt cele de mai jos. Erorile sunt mai mari decat în laborator (raport 3), poziționarea robotului P3DX în punctul ideal în raport cu care brațul robotic trebuia să aibă erori minime, fiind afectată de condițiile din hala de producție, suprafață lucioasă, pete de mizerie, etc. În figura 10 se poate observa prototipul în cele 3 posturi de lucru și deplasându-se pe traseu.

Motoare / Erori	Punct 1	Punct 2	Punct 3
A1	0.23 %	0.55 %	0.58 %
A2	0.82 %	1.35 %	1.25 %
A3	0.61 %	1.27 %	1.15 %
A4	0.50 %	0.60 %	0.67 %
A5	0.55 %	0.76 %	0.78 %
A6	0.30 %	0.40 %	0.40 %

Tabel 1. Acțiune de poziționare a brațului robotic cu comandă de la distanță – erori de poziționare

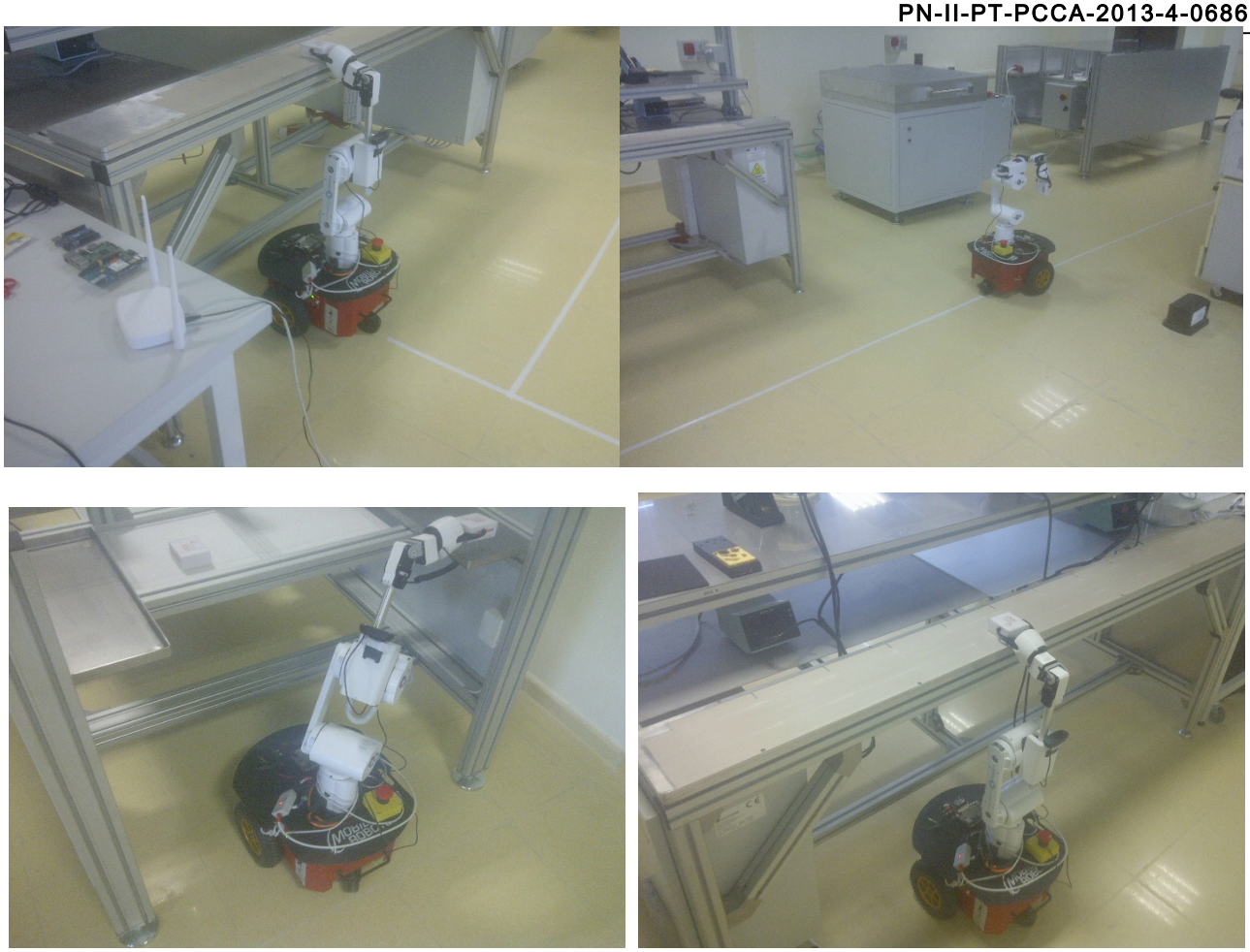

Fig. 22. Manipulare în hala ICSTM – post 1 (sus-stanga), traseu spre post 2 (sus-stanga), post 2 (jos-stanga), post 3 (jos-dreapta)

Integrarea si testarea sistemelor robotice autonome din gama Prototipului 2 (2DW/1FW Pioneer 3DX, echipat cu Pioneer 6-DOF Arm) in liniile de asamblare preliminară şi generală aferente modelelor Dacia Logan şi Sandero la intreprinderea "DACIA-RENAULT", Pitesti s-a realizat în Institutul de Cercetare Ştiinţifică şi Tehnologică Multidisciplinară din UVT (ICSTM). Am adoptat această soluţie deoarece capabilităţile prototipului nu sunt, pentru moment, în parametrii corespunzători unor teste într-o linie de fabricaţie “Dacia-Renault”, dar au fost relativ decente pentru o linie de fabricaţie din ICSTM sau o companie cu activitate în zona de automatizări linii de fabricaţie de pe plan local.

Activitatea 4.3-Integrarea si testarea Prototipului 3 (4DW/SW Autonomous Ominidirectional Vehicle echipat cu 6-DOF manipulator) in procese de fabricatie la sticla (”STICLA” Avrig).

1. Integrare în aplicaţie

În urma finalizării operaţiunilor şi dezvoltării aferente etapei 3, proiectul a continuat cu paşii descrişi în următoarele capitole în vederea funcţionării robotului autonom.

2. Implementarea scannerelor de zonă pe robot

În vederea utilizării informaţiei scannerelor, ieşirea analogică a fiecăruia a fost conectată la un modul de intrare analogică de pe automatul programabil. Informaţia analogică, corespunzătoare distanţei până la cel mai apropiat obstacol, are valoare doar în măsura în care se poate determina şi unghiul în care respectiva măsurătoare a fost realizată.

2.1. Transformarea din coordonate timp, distanţă în x, y

Pentru determinarea unghiului α s-au avut în vedere următoarele considerente:

1. Ecuația $x(t)$ are o formă ce poate fi considerată sinusoidă, conform celor determinate experimental mai sus;
2. Variabilele ce trebuiesc definite pentru coordonare sunt perioada *T* de execuție a unei curse complete, între un capăt și altul al cursei, și unghiul între care se execută cursa;
3. Motorul ce rotește senzorul descrie o mișcare continuă, dar datorită variațiilor inerente într-un astfel de sistem, perioada va varia cu timpul și nu poate fi considerată constantă;

Determinarea perioadei s-a folosit prin citirea contactului celui de-al 3-lea pin al motorului, ce indică atingerea capătului de cursă, în automatul programabil. Intrarea în zona capătului de cursă este semnalizată prin închiderea unui contact între tensiunea de alimentare și această bornă. Durata cât contactul este făcut poate varia, dar poziția a fost considerată relativ fixă.

Astfel, s-a putut determina perioadă din contorizarea timpului între 2 momente de timp în care contactul a fost închis. Prin medierea acestei valori de timp pe parcursul ultimelor 3 perioade, se consideră valoarea *T* determinată, de altfel cu rezultate bune.

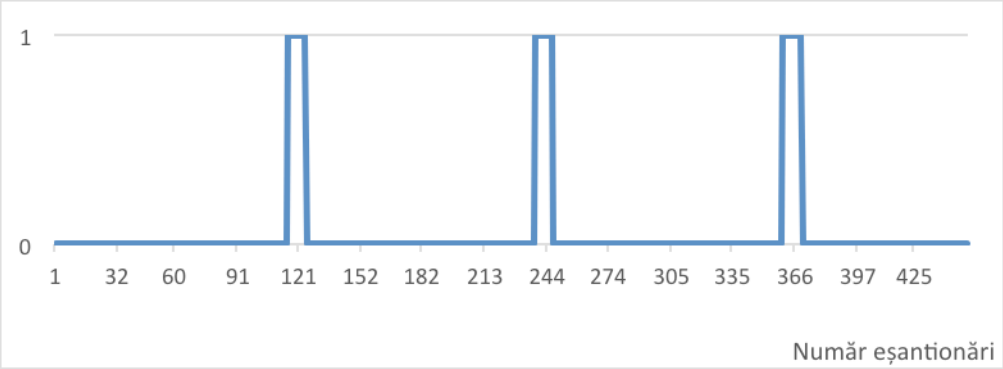

Fig. 23 Măsurarea duratei variabile a cursei servomotorului

Unghiul între cele 2 capete de cursă s-a măsurat direct pe echipament.

Folosind informațiile de mai sus, s-a obținut o formă standard de transformare din coordonatele (d,t) în (x,y):

$$x(d,t) \quad d \cdot \cos\left(\frac{t}{T} \cdot 2\pi \quad \theta\right)$$

$$y(d,t) \quad d \cdot \sin\left(\frac{t}{T} \cdot 2\pi \quad \theta\right)$$

Relațiile vor trebui implementate pe un automat programabil, unde calculele trigonometrice sunt realizate, cel mai probabil, prin dezvoltare în serie Taylor. Timpul de rezolvare este mult prea mare pentru a permite o execuție rapidă, deci ecuațiile vor trebui simplificate și parametrii trigonometrici interpolați liniar.

Ecuațiile devin:

$$x(d,t) \quad d \cdot \cos\left(\frac{t}{T} \cdot 2\pi \quad \theta\right) \quad d\left(\cos\frac{2\pi t}{T} \cdot \cos\theta \;\; \sin\frac{2\pi t}{T} \cdot \sin\theta\right)$$

$$y(d,t) \quad d \cdot \sin\left(\frac{t}{T} \cdot 2\pi \quad \theta\right) \quad d\left(\sin\frac{2\pi t}{T} \cdot \cos\theta \quad \cos\frac{2\pi t}{T} \cdot \sin\theta\right)$$

Factorii *sin* θ și *cos* θ au fost calculați în afara mediului de calcul al automatului programabil, la măsurarea unghiului maxim al deschiderii motorului.

2.2. Optimizarea calculului trigonometric

Factorii sin și $\cos\frac{2\pi t}{T}$ au fost calculați pentru toate valorile posibile ale unghiului și mutați într-un tabel de interpolare cu 10 de variabile intermediare. Împărțirea este realizată la fiecare iterație, dar cu un efort de calcul numeric relativ redus.

În graficul următor se poate observa eroarea între funcția calculată și cea interpolată:

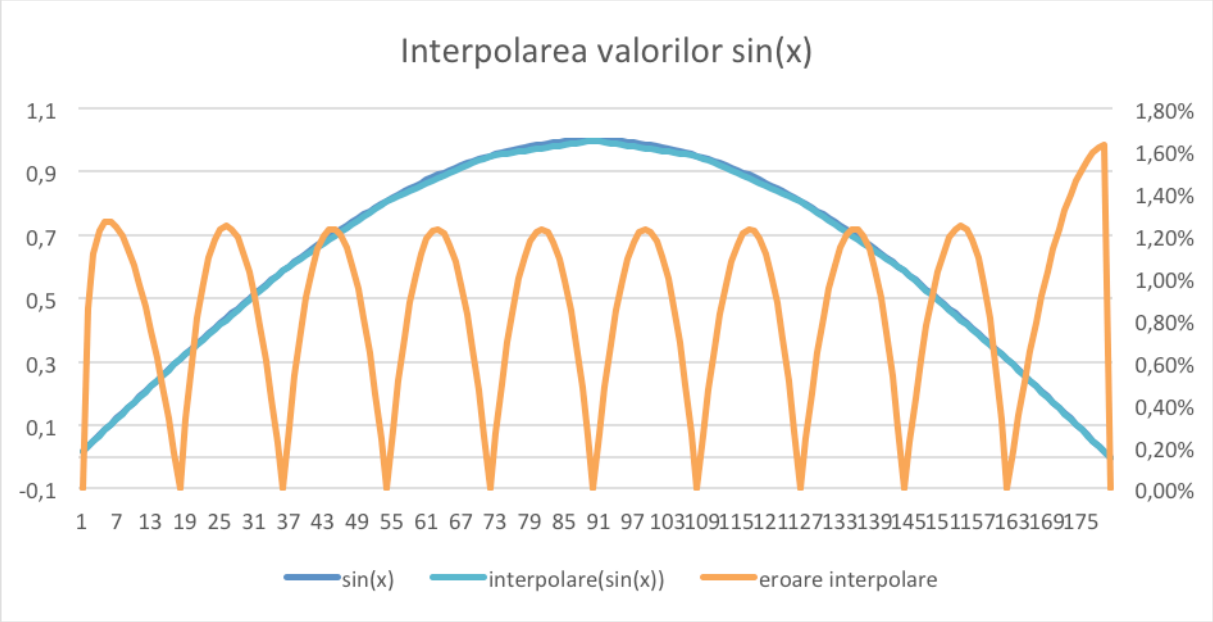

Fig.24 Estimarea funcțiilor trigonometrice prin interpolare

Programul din automatul programabil permite o astfel de abatere, în condițiile în care este corectată în regim dinamic.

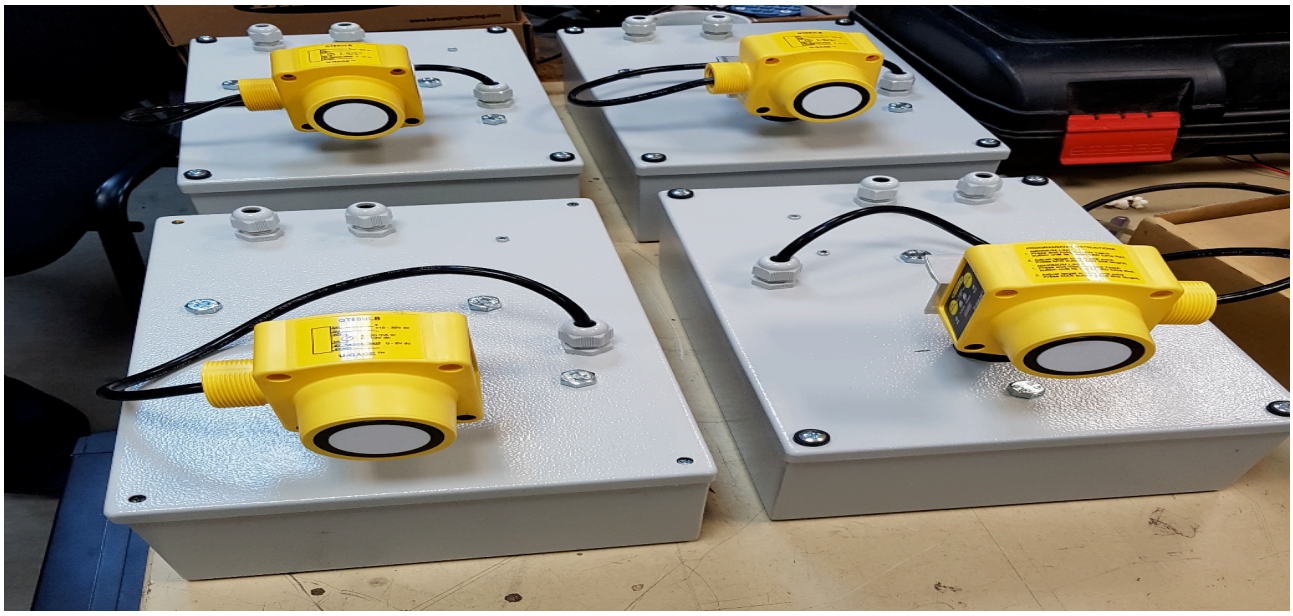

Fig. 25 Ansamblul celor 4 senzori ultrasonici montați pe cutiile servomotoarelor

Rularea procesului duce la obținerea unei hărți în timp real a împrejurimilor senzorului, cu o precizie suficient de bună încât să poată fi folosită.

Fig.26. Zona de dectatat

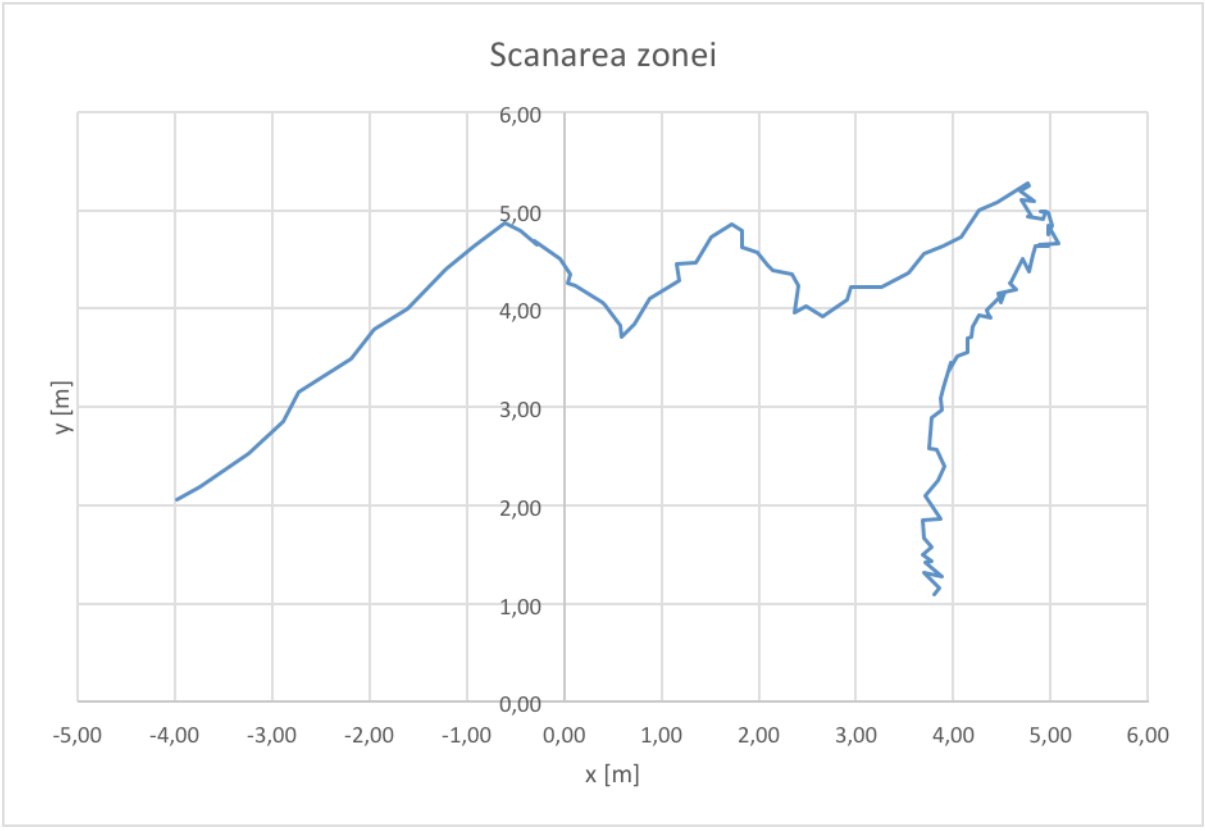

Fig. 27. Rezultatul scanării

2.3. Simplificarea hărții

Pentru a obține un număr suficient de fin de puncte pentru a putea evita coliziunea cu obiectele cu suprafață mică, pe de-o parte, apoi pentru a putea recunoaște traseul, pe de altă parte, viteza unghiulară nu poate fi prea mare. O serie de teste a dus la un număr de aprox. 200 de puncte pentru 150° deschidere între capetele de cursă ale servomotorului, rezultând un punct la aprox. 1,2°.

Majoritatea obiectelor din împrejurimile senzorului au suprafață plană, spre exemplu pereții unei hale. Astfel, pentru determinarea optimă a unei traiectorii este avantajos să simplificăm harta zonei trimisă către algoritmul de navigare, pentru a îmbunătăți viteza de procesare.

Îmbunătățirea se realizează prin unirea cu un segment continuu a punctelor cu o deviație mică:

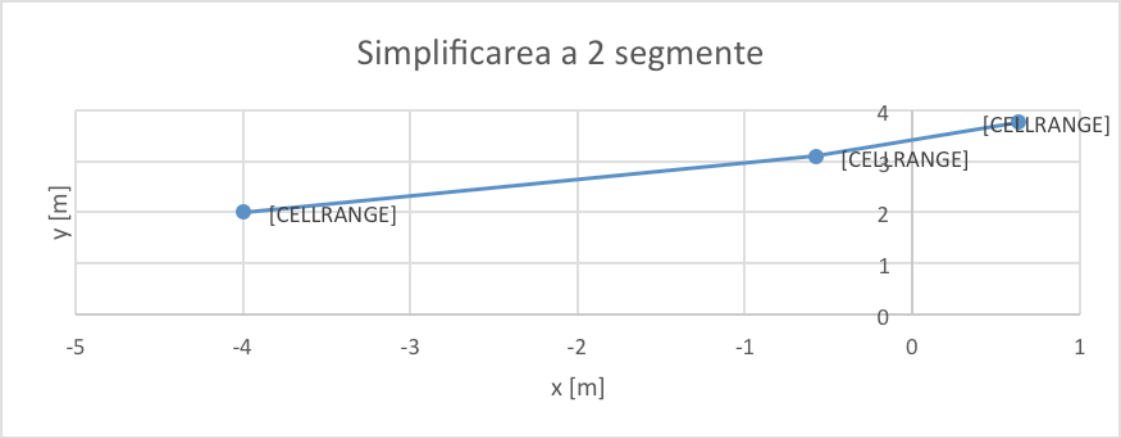

Fig. 28 Coordonatele a 3 puncte, din care p2 poate fi anulat

Algoritmul de anulare a unui punct, după notațiile din graficul de mai sus, este:

1. Se calculează

$$\alpha_1 \quad \frac{y_3 \quad y_2}{x_3 \quad x_2}$$

$$\alpha_2 \quad \frac{y_3 \quad y_1}{x_3 \quad x_1}$$

unde α_1 și α_2 sunt valorile pantelor pentru dreptele dintre 2 segmente.

2. Dacă

$$|\alpha_1 \quad \alpha_2| < \alpha_{max}$$

unde α_{max} este un factor de abatere maximă între pantele a 2 segmente, atunci punctul nou introdus ar putea înlocui punctul median.

Acest algoritm se dovedește, în schimb, nefuncțional când îi sunt prezentate forme curbe. Din experiențe, panta relativ mică dar constantă între 2 puncte face ca toate punctele în interiorul capetelor să fie simplificate. Este necesară o funcție de corecție, implementată astfel:

3. Se memorează diferența între pantele a 3 puncte succesive:

$$\alpha_{d1} \quad \alpha_1 \quad \alpha_2$$

4. La următorul pas, se compară cu rezultatul obținut pentru următorul punct:

$$\alpha_{d2} \quad \alpha_2 \quad \alpha_3$$

$$\alpha_d \quad \alpha_{d1} \quad \alpha_{d2}$$

Dacă α_d este mai mic decât un α_{min} rezultă că suprafața prezintă o curbură continuă și punctul intermediar nu ar trebui anulat.

Din analiza de mai sus vor rezulta 4 hărți simplificate, pentru fiecare din cele 4 laterale ale robotului.

Algoritmul trebuie să rezolve următoarele sarcini de bază:

1. Determinarea distanței minime între corpul robotului și obiectele din jur, pentru a preveni un accident
2. Prevenirea deplasărilor ce pot micșora această distanță

În urma achiziției datelor realizată la pasul anterior, harta împrejurimilor este stocată sub forma unor perechi de coordonate ce definesc puncte dintr-un poligon. Pentru prevenirea coliziunii ne interesează segmentele acestui poligon aflate la o distanță mai mică decât un factor de siguranță *ds* de corpul robotului.

2.4. Evitarea coliziunii

Algoritmul trebuie să determine distanța minimă între extremitatea robotului și fiecare dintre segmente:

$$d = \frac{|(y_2 - y_1)x_0 - (x_2 - x_1)y_0 + x_2y_1 - y_2x_1|}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}}$$

Unde *d* este distanța minimă, y_1y_2, x_1x_2 coordonatele capetelor segmentului și x_0y_0 coordonatele capătului robotului.

Din experiențe, în schimb, lungimile segmentelor s-au dovedit a fi suficient de mici încât distanța între lateralele robotului și punct să fie o aproximare suficient de bună. În același timp, calculul ecuației anterioare este foarte solicitant din punct de vedere numeric, încât creșterea de precizie nu este justificată.

Rezultă că algoritmul trebuie să recunoască prezența unui obiect prin includerea lui într-o zonă de protecție, pentru simplificare considerată tot rectangulară, în jurul robotului.

Astfel, dacă

$$|y_1| < d_s$$

și

$$|x_1| < l_{senzor} + d_s$$

atunci mișcările ce ar putea apropia laterala robotului de obstacol sunt inhibate.

Pentru un obstacol în zona de *x* pozitiv, mișcările de rotație în sensul ce ar putea apropia robotul de obstacol sunt inhibate. Similar pentru *x* negativ.

Pentru un obstacol în care s-a atins jumătate din distanța minimă, toate mișcările robotului sunt inhibate și se așteaptă intervenția operatorului uman.

3. Deplasarea automată

Robotul trebuie să execute mișcări de deplasare automată, între 2 puncte din spațiu.

Pentru recunoașterea destinației, el trebuie să fie capabil să recunoască forma mediului înconjurător. Memorarea punctului final și a traseului inițial va fi realizată printr-o cursă condusă de operator, în care automatul programabil va memora mediul și mișcările realizate astfel încât să poată să le reproducă parcursul următor, având în vedere alunecările variabile.

Memorarea mediului destinație se realizează prin memorarea celor 4 hărți ale celor 4 senzori.

Pentru stabilirea parcursului între 2 destinații, sistemul va reține succesiunea mișcărilor parcurse, duratele lor și câteva coordonate ale mediului.

Pentru simplificarea parcurgerii în condițiile în care robotul poate determina forma mediului înconjurător se va utiliza ca reper, pentru toate mișcările, cel puțin 1 suprafață pe durata întregii mișcări.

Ambele cerințe necesită o metodă de recunoaștere a similitudinilor dintre 2 hărți, una memorată, alta măsurată în timp real, și de determinare a transformărilor necesare pentru a le suprapune, dacă este posibil.

3.1. Recunoaşterea formelor din mediu

Se va urmări recunoaşterea unei forme definită ca bază, un tip de hartă ce va trebui recunoscut ca poziție de destinație, atât pentru tur cât și pentru retur. Forma aceasta permite recunoaşterea și poziționarea automată a robotului cu precizie suficient de bună, și un efort minim de construcție.

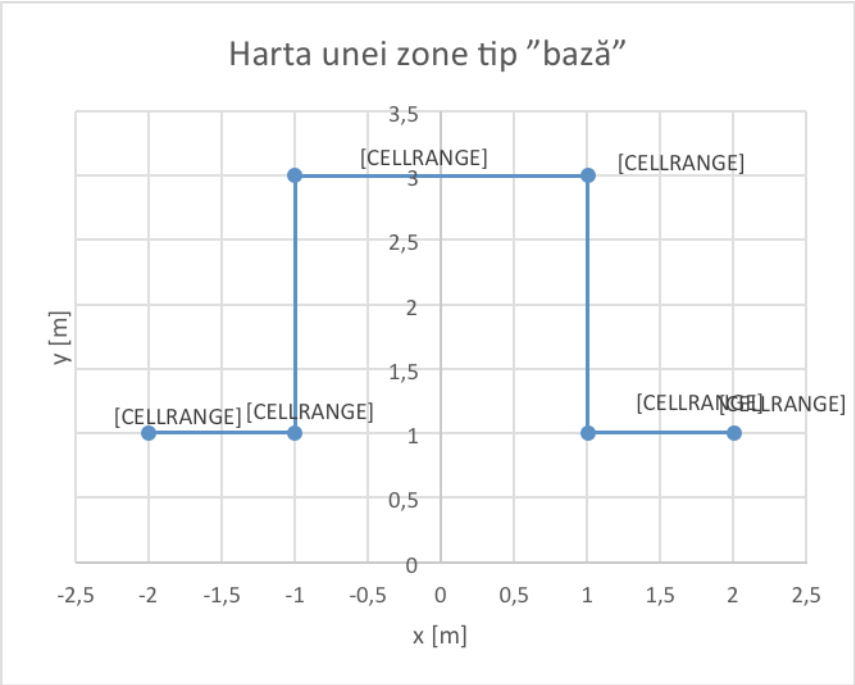

Fig. 29 Harta unei zone tip ”bază”

Se caută puncte din grafic ce să fie în unghiul a 2 segmente: panta punctelor anterioare și panta punctelor ulterioare să fie la un unghi suficient de apropiat de 90°. Sunt necesare 4 astfel unghiuri succesive într-o parcurgere a listei de puncte pentru a identifica o ”bază”.

3.2. Algoritmul de recunoaştere

Condiția pentru un unghi de 90° este definită prin calculul unghiului făcut între 3 puncte succesive.

1. Calculul începe de la un punct din serie, pentru punctul 1, notat cu indexul *p1*.
Se caută punctul *p2* pentru care distanța între *p1* și *p2* este mai mare decât cea a brațului *db1*.

2. Se parcurg toate punctele între *p1* și *p2* pentru a constata dacă pot fi considerate în linie.
Se va folosi formula descrisă mai sus:

$$d = \frac{|(y_2 - y_1)x_0 - (x_2 - x_1)y_0 + x_2y_1 - y_2x_1|}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}}$$

unde, în acest caz, *d* este distanța dintre punct și linia celor 2 puncte. Algoritmul este solicitant de executat pentru fiecare calculație, dar se poate optimiza foarte simplu memorând toți parametrii ce nu depind de x_0y_0. Astfel, calculul se simplifică la doar 2 înmulțiri, ce pot fi executate în timp suficient de scurt.

Dacă se identifică un punct ce iese în afara segmentului ce poate fi considerat drept, indexul lui *p1* ia valoarea acestui punct și se reia calculația de la pasul 1.

3. Dacă toate punctele pot fi incluse în segment, se calculează panta α_1 între *p1* și *p2*, conform algoritmului arătat mai sus.

$$\alpha_1 = \frac{y_2 - y_1}{x_2 - x_1}$$

4. Se caută *p3* pentru care distanța între *p2* și *p3* este mai mare decât *db2*.
5. La identificarea unui astfel de punct, se reia procedura de verificare a liniarității segmentului între *p3* și *p2*.

Dacă se identifică un punct ce iese în afara segmentului ce poate fi considerat drept, se incrementează simultan valorile de start pentru toți cei 3 indecși, *p1*, *p2* și *p3*, și se reiau verificările de la primul pas.

6. Dacă se găsește un set de *p1*, *p2* și *p3* pentru care *p1-p2* este un segment drept, *p2-p3* alt segment drept, se calculează panta α_2 între *p3* și *p2* conform:

$$\alpha_2 \quad \frac{y_3 \quad y_2}{x_3 \quad x_2}$$

7. Identificarea perpendicularității între cele 2 segmente se face ținând cont că 2 segmente perpendiculare vor avea pantele respectând relația:

$$\alpha_1 \quad \frac{1}{\alpha_2}$$

Având în vedere că valorile sunt măsurate din câmp, cu precizie relativ mică, egalitatea va trebui ponderată cu un factor de toleranță.

8. Algoritmul trebuie să determine dacă vârful *p2* descrie un unghi concav sau convex, raportat la planul senzorului ultrasonic. Din analiza posibilităților de amplasare, condițiile necesare și suficiente sunt ca

$$y_{p3} > y_{p1}$$

$$y_{p2} < y_{p3}$$

Dacă toate condițiile sunt îndeplinite rezultă că punctele *p1*, *p2* și *p3* definesc un segment ce poate fi considerat ca partea *db1-db2* din graficul anterior.

Pentru segmentul *db2-db3* este necesară reluarea algoritmului. Punctul *p2* devine punctul *p1* iar punctul *p3* devine punctul *p2*, din algoritmul de mai sus.
Liniaritatea lui *db2* a fost verificată. În schimb, este posibil ca între punctul *p3* ce definește segmentul *db2* și punctul de start al lui *db3* să existe neconcordanțe, datorate măsurii imperfecte. Algoritmul compensează prin introducerea unui pas suplimentar:

2.1. Se găsește punctul la distanța maximă a abaterii tolerate $d_{abatere}$ de lungime a segmentului *db2*. Se verifică dacă din acest punct până în *db2* punctele pot fi considerate în continuare coliniare, în urma calculului

$$d \quad \frac{|(y_2 \quad y_1)x_0 \quad (x_2 \quad x_1)y_0 + x_2y_1 \quad y_2x_1|}{\sqrt{(y_2 \quad y_1)^2 + (x_2 \quad x_1)^2}}$$

Dacă nu, se reia tot algoritmul de la pasul 1, cu indexul lui *p3* ca fiind *p1*: s-a identificat doar un unghi de 90°, nu toate cele 4 unghiuri necesare.
Dacă da, algoritmul continuă în mod similar cu pașii 3-8, cu diferența verificării, de această dată, a convexității unghiului dintre *db2* și *db3*, echivalentul pasului 8:

$$y_{p2} > y_{p1}$$

$$y_{p1} < y_{p3}$$

Algoritmul continuă în același mod și pentru segmentele *db4* și *db5*.

În acest mod, este posibilă utilizarea sistemului pentru determinarea corespondenței între 2 hărți.

3.3. Modificări în implementare

Implementând acest algoritm, prima constatare a fost că întreaga ”bază” este greu de vizualizat, având în vedere că senzorul nu poate măsura decât distanțele aflate în fața lui. Colțurile obturează vederea senzorului și distorsionează imaginea mediului suficient de mult pentru a nu permite recunoașterea ei completă.

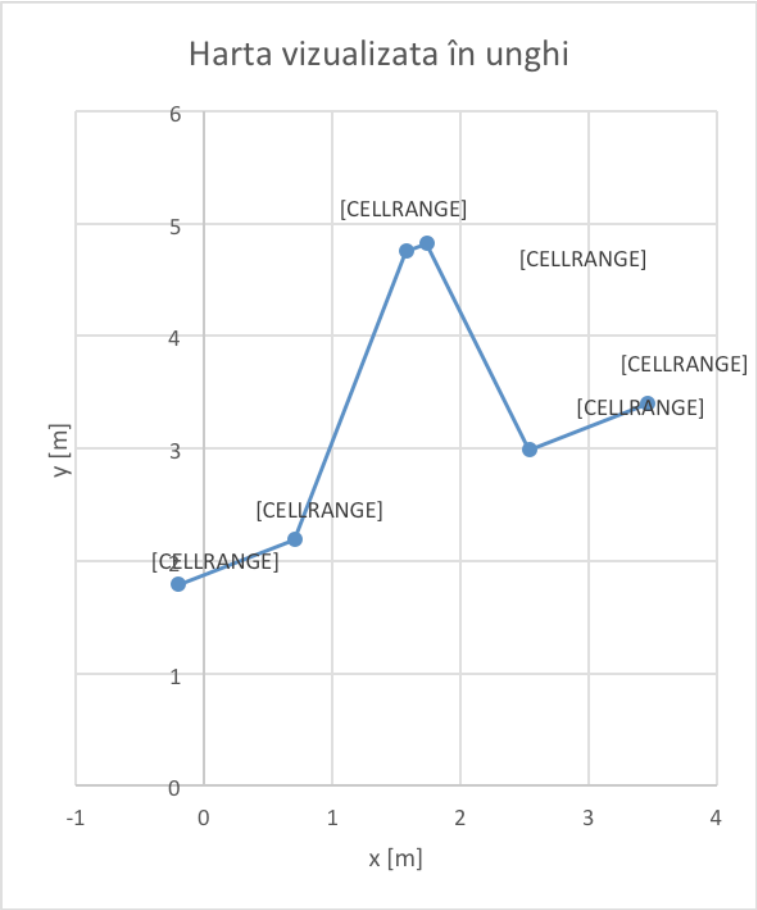

Fig. 30 Harta unei baze parțial obturată

Din cele 4 unghiuri de detectat, doar unul singur este vizibil, și și acela la limita de a fi parțial obturat. Pentru a vizualiza complet este nevoie de următorul pas, modificarea poziției pentru a permite vizualizarea completă a spațiului.

Din detectarea pantei dreptei *p2-p3* și a coordonatei x a punctului *p2* deducem că sunt necesare 2 tipuri de mișcări de executat de către axele robotului:

1. De translație, pentru a aduce punctul *p2* în dreptul zonei inițiale
2. De rotație, pentru a aduce panta *p2-p3* la unghiul din zona memorată

Mișcările de executat sunt trimise către sistemul de poziționare, la viteză mică. Măsurarea continuă a zonei din fața senzorului duce la oprirea mișcării când aceasta a fost corespunzător executată.

3.4. Concluzii

Sistemul de navigare funcționează suficient de bine pentru a permite navigarea și poziționarea echipamentului între 2 repere, iar alunecările între cele 2 puncte sunt compensate de capacitatea de adaptare.

Sistemul poate fi folosit pentru transportul și amplasarea pe poziție a brațului robotic.

Integrarea si testarea Prototipului 3 (4DW/SW Autonomous Ominidirectional Vehicle echipat cu 6-DOF manipulator) in procese de fabricatie la sticla (”STICLA” Avrig).

La baza procesului de fabricație într-o fabrică de sticla este cuptorul pentru topit sticlă. Un cuptor de sticlă, în pricipiu, cotine zona de încărcare 1, de topire 2, zona de liniștire 3 și de menținere 4. Sticlarii preiau sticlă din zona de menținere 4 prin guri de acces.

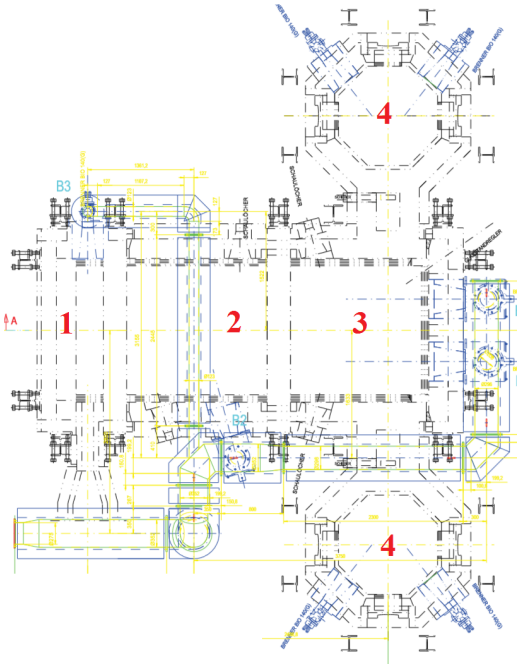

Fig. 31 Zonele unui cuptor de sticlă

Fig.32 Gura pentru extragerea sticlei

Pentru creşterea competitivităţii fabricile de sticlă au nevoie de echipamente robotizate în vederea producţiei de serie medie şi mare de produse uzuale din sticlă.

Fig. 33 Exemple de produse din sticla pretabile pentru productia de serie cu ajutorul braţului robotic

Astfel vehiculul robotizat va deservi cuptorul de sticlă executând operaţiunea de extragere a unei cantităţi funcţie de dimensiunea bilei ceramice cu care este echipată lancea şi plasarea sticlei topite într-o presă care modelează sticlă.

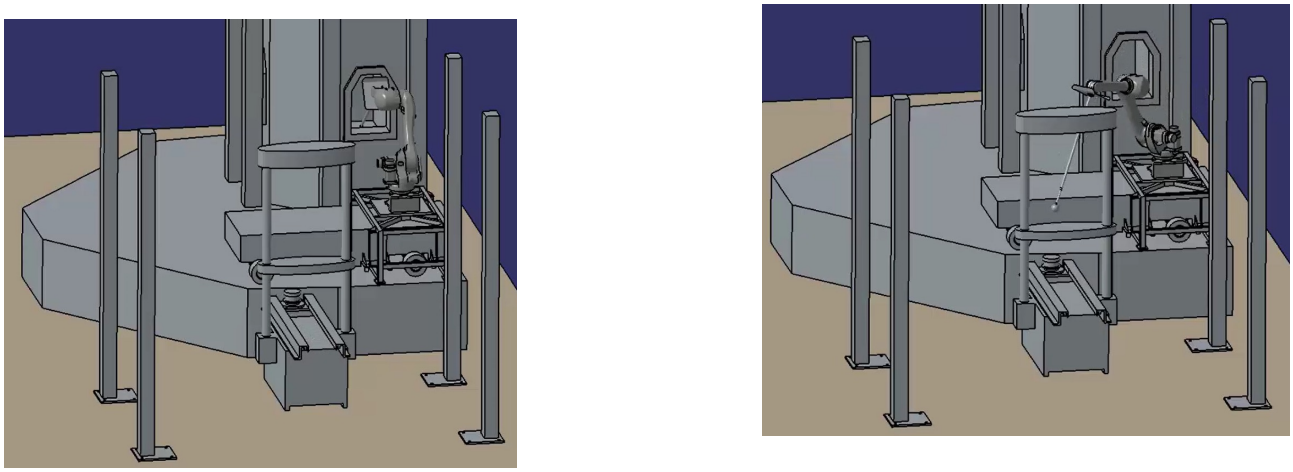

Fig. 34 Simulari ale ansamblului cuptor - robot – presa

Fig. 35 Ansamblul vehicul – brat robotic

Manipulatorul robotic este programat să reproducă mișcările manuale pentru extragerea sticlei dar cu o rată de 10 piese pe minut și o repetabilitate de +/-0,08 mm.

Vehiculul pe care este montat brațul robotic face posibilă mutarea în alt punct de lucru într-un timp de ordinul minutelor.

Activitatea 4.4 Participare la manifestari tehnico-stiintifice din domeniii specifice proiectului (mese rotunde, workshopuri, simpozioane nationale / internationale, targuri nationale / internationale)

1. George Ciubucciu, Razvan Solea, Adrian Filipescu, Adriana Filipescu, "Visual Servoing and Obstacle Avoidance Method based Control Autonomous Robotic Systems Servicing a Mechatronics Manufacturing Line", Proceedings of the 2017 IEEE 9th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) 21-23, Sep, 2017 Bucharest Vol. 2, pp. 874- 879, 20EEE Catalog number: CFP17803-USB, ISBN 978-1-5386-0697-1/17/$31.00 ©2017 IEEE.

2. Filipescu Adrian, Minca Eugenia, Cernega Daniela, Filipescu Adriana, Solea Razvan, SHPN Models Based Simulation and Control of Mobile Robotic Systems Integrated into A/DML Proceedings of the 21th IEEE, International Conference on System Theory, Control and Computing, ICSTCC2017 19-21, Oct. Sinaia, 2017, ISBN: 978-1-5386-3841-5, accepted paper.

3. Filipescu Adrian, Solea Razvan, Petrea George, Cernega Daniela, Filipescu Adriana, Ciubucciu George, SHPN Modelling, Visual Servoing and Control of WMR with RM Integrated into P/RML, Proceedings of the 21th IEEE, International Conference on System Theory, Control and Computing, ICSTCC2017 19-21, Oct. Sinaia, 2017, ISBN: 978-1-5386-3841-5,accepted paper.

4. H.G. Coandă, E. Mincă, I. Caciulă, F. Ion, *New solutions for robotic systems using LattePanda*, Journal of Electrical Engineering, Electronics, Control and Computer Science (JEEECCS), indexat BDI, ISSN 2207-3528, Volum 3, nr.2, Issue 8, 2017 (*în evaluare*).

Director Proiect:PN_II_PT_PCCA_2013_4_0686
Prof. Dr. Ing. Adrian FILIPESCU