

Denumire Etapa 1:

Modelarea cinematica a trei prototipuri de sisteme robotice autonome:

Prototipul 1 (Autonomous Wheelchair 2DW/2FW): Scaunul cu roțile pentru persoane varstnice și cu dizabilitati, conceput ca robot mobil cu doua roți motoare și doua roți libere(2DW/2FW);

Rezumat Etapa 1:

În cadrul acestei etape s-a determinat modelul cinematic a sistemului autonom cu două roți motoare și două libere (2DW/2FW). Acest model este necesar pentru realizarea celei de a doua etape a proiectului: "Conducerea și navigatia bazata pe sistem video-biometric și sistem laser a scaunului cu roțile". De asemenea s-a determinat experimental modelul matematic al sistemului format din servoamplificator, motor de curent continuu și reductor de turație necesar dezvoltării sistemului de comandă a celor două roți motoare. În această etapă au fost calculați parametrii reguletoarelor de tip PI necesare conducerii scaunului cu roțile. S-a realizat și implementat fizic și o soluție software pentru determinării în timp real a poziției sistemului autonom cu două roți motoare și două libere utilizând două encodere. Tot în cadrul acestei etape s-a realizat testarea Prototipului 1 (2DW/2FW Cirrus Power Wheelchair) cu sistemul de navigatie bazat pe videobiometria ochiului. În cadrul acestei activități s-a implementat în Matlab un algoritm pentru identificarea mișcării ochiului utilizatorului pe cele trei direcții de deplasare: înainte, stânga și dreapta.

Act. 1.1 (A.2 - Cercetare Industrială)

Modelarea cinematica a sistemului autonom corespunzator Prototipului 1, modelare ca robot mobil cu doua roți motoare și doua libere (2DW/2FW). Grad de realizare 100%

Modelul cinematic

În Figura 1 este reprezentat un model geometric al unui scaun cu roțile care definește principalele variabile necesare pentru a obține modelul cinematic.

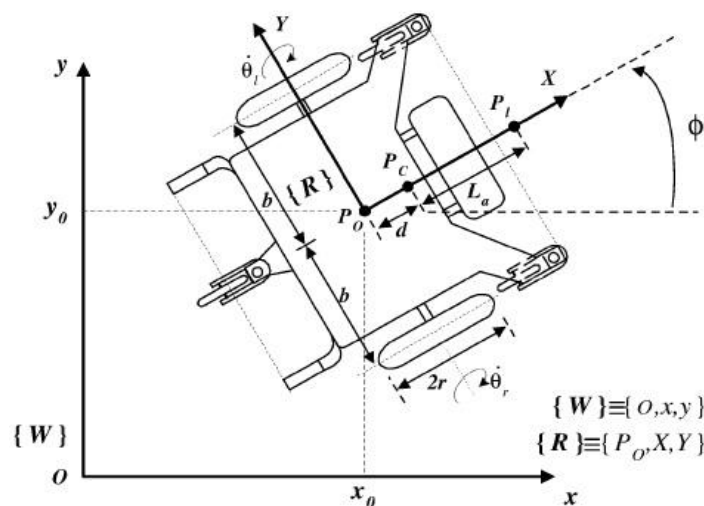


Figura 1. Scaun mobil cu două roți motoare și două roți libere pentru persoanele cu handicap locomotor

Scaunul cu roțile are două roți motoare diametral opuse, având raza r , și două roți libere (necesare realizării stabilității scaunului) de tip castor. Ambele roți motoare au atașate actuatori și

encodere pentru detecția poziției și/sau a vitezei, în timp ce roțile castor nu conțin niciun fel de senzor sau actuator, acestea nefiind luate în considerare la crearea modelului cinematic.

Originea sistemului de coordonate ale robotului cu coordonatele (x_c, y_c) este definită de P_c , și se presupune că este pe axa X la o distanță d de P_0 . P_l este un punct virtual de referință atașat platformei cu coordonatele (x_l, y_l) , și este plasat pe axa X la o distanță $L\alpha$ de P_c .

$2b$ este lungimea axei dintre cele două roțile ale robotului mobil. ϕ este unghiul format de axa x care aparține lui $\{W\}$ și axa X care aparține lui $\{R\}$. θ_r și θ_l sunt vitezele unghiulare ale roților din dreapta, respectiv stânga în jurul axei Y .

Echilibrul robotului este menținut de către roțile castor al caror efect va fi ignorat. Astfel, $q = [x_c, y_c, \phi, \theta_r, \theta_l]^T$ denotă configurația sistemului, i.e., cele 5 coordonate generalizate ($n=5$).

Pentru modelul cinematic, se presupune că pentru fiecare contact, există o mișcare pură în rostogolire. Presupunând că viteza lui P_0 este în direcția axei de simetrie (axa X) și roțile nu prezintă alunecări, următorul set de constrângeri ($m=3$), este obținut:

$$\dot{y}_c \cos \phi - \dot{x}_c \sin \phi - \dot{\phi} d = 0$$

$$\dot{x}_c \cos \phi - \dot{y}_c \sin \phi + b\dot{\phi} - r\dot{\theta}_r = 0$$

$$\dot{x}_c \cos \phi - \dot{y}_c \sin \phi - b\dot{\phi} - r\dot{\theta}_l = 0$$

Modelul cinematic este dat de către relația:

$$\dot{q} = S(q)v(t) \text{ cu } v = [v_1 \ v_2]^T = [\dot{\theta}_r \ \dot{\theta}_l]^T.$$

unde:

$$A(q) = \begin{bmatrix} -\sin \phi & \cos \phi & -d & 0 & 0 \\ -\cos \phi & -\sin \phi & -b & r & 0 \\ -\cos \phi & -\sin \phi & b & 0 & r \end{bmatrix}$$

$$S(q) = \begin{bmatrix} c(b \cos \phi - d \sin \phi) & c(b \cos \phi + d \sin \phi) \\ c(b \sin \phi + d \cos \phi) & c(b \sin \phi - d \cos \phi) \\ c & -c \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

iar $c = r/2b$

Modelarea reguletoarelor de tip PI pentru controlul celor două motoare.

Pentru controlul unui motor de curent continuu este necesar să dispunem de modelul matematic al motorului. Modelul matematic trebuie să aibă posibilitatea determinării unei relații între viteza unghiulară a rotorului și tensiunea aplicată acestuia.

Pentru determinarea modelului matematic se va pleca de la schema electrică a motorului de curent continuu:

$$e_b(t) = K_b \frac{d\theta(t)}{dt} = K_b \omega(t)$$

$$e_a = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e_b(t)$$

$$T_m(t) = J \frac{d^2\theta(t)}{dt^2} + B \frac{d\theta}{dt} = K_T i_a(t)$$

unde: R_a - rezistența armăturii; L_a - inductanța armăturii; i_a - curentul prin armătură; E_a - tensiunea aplicată la intrare; E_b - forța electromotoare de opunere; T_m - cuplu motor; Ω - viteza unghiulară.

Funcția de transfer a motorului de curent continuu, ținând cont de tensiunea de intrare poate fi scrisă astfel:

$$H(s) = \frac{\Omega(s)}{E_a(s)} = \frac{K_t}{(L_a s + R_a)(J s + B) + K_b K_t}$$

Deoarece în practică inductanța armăturii este foarte mică, atunci funcția de transfer se poate simplifica și poate fi scrisă sub forma:

$$H(s) = \frac{\Omega(s)}{E_a(s)} = \frac{K_m}{\tau s + 1}$$

unde:

$$K_m = \frac{K_t}{R_a B + K_t K_b}, \quad \tau = \frac{R_a J}{R_a B + K_t K_b}$$

În cazul scaunului cu roțile pentru controlul motoarelor de curent continuu avem nevoie de modelul matematic al sistemului format din servoamplificator, motor de curent continuu și reductor de turație. Pentru comanda motoarelor de cc cu perii s-a utilizat servoamplificatorul 50A8 (a se vedea figura 2a). Acesta poate primi informații de la placa de achiziție iar pe baza semnalului primit, servoamplificatorul va trimite o comandă de tip PWM către motorul de curent continuu, semnal cu factor de umplere modificabil dinamic. Pentru achiziția de date precum și pentru trimiterea comenzilor către cele două servomotoare s-a utilizat o placă de achiziție de tip NI-6024E (vezi figura 2b). Aceasta are 16 canale de intrări analogice, două canale de ieșiri analogice și 8 linii digitale de intrare-ieșire. Slotul prin care se realizează comunicația dintre placa de bază și placa de achiziție este de tip PCI. Acest dispozitiv face legătura dintre PC și servoamplificatoare în vederea comandării motoarelor de curent continuu, precum și între cele două encodere (vezi figura 2c) și PC în vederea preluării de semnale necesare implementării sistemului de odometrie.



Figura 2. a) servoamplificatorul 50A8, b) placă de achiziție de tip NI-6024E, c) Encoder incrementl Hengstler RI-58

Pentru identificarea funcțiilor de transfer a celor două sisteme de comanda s-a utilizat instrumentul din MATLAB „*System Identification*”

Instrumentul System Identification determină modelul matematic a unor sisteme plecând de la datele de intrare și cele de ieșire a sistemului. Acest instrument ne ajută la determinarea modelelor matematice a unor sisteme, a căror determinare este posibilă utilizând principiile de proiectare și specificațiile sistemului. Datele de intrare-ieșire pot fi din domeniul timp dar și din domeniul frecvențelor pentru determinarea funcțiilor de transfer în timp continuu sau discret.

Caracteristici: determinarea funcțiilor de transfer, modelului procesului și reprezentarea în spațiu stărilor a unor sisteme pe baza datelor din domeniu frecvențelor sau din domeniul timp; blocuri pentru

utilizarea sistemelor identificate în Simulink; identificarea modelelor nelineare ARX și a modelelor Hammerstein-Weiner cu neliniariități ale intrărilor și ieșirilor cum ar fi zone moarte și zone de saturație

Pentru obținerea funcției de transfer care caracterizează fiecare ansamblu servoamplificator, motor, reductor trebuie să achiziționăm date pe care le vom procesa cu ajutorul instrumentului *Sistem Identification*. Datele sunt formate din semnalele aplicate sistemului la intrare și valoarea vitezei unghiulare.

Datele achiziționate de la cei doi senzori ce măsoară viteza de rotație a celor două roți motoare sunt prezentate în tabelele următoare precum și în figurile 3 și 4:

input[V]	output[rad/s]	input[V]	output[rad/s]
1	2,445087719	1	2,246951754
1	2,545087719	1	2,206951754
1	2,580877194	1	2,216951754
1	2,645087719	1	2,226951754
1	2,545087719	1	2,226951754
1	2,545087719	1	2,286951754
1	2,485087719	1	2,256951754
1	2,505087719	1	2,186951754
4	10,10035088	4	10,84662281
4	10,18035088	4	10,81662281
4	10,18035088	4	10,76662281
4	10,20035088	4	10,78662281
4	10,24035088	4	10,84662281
4	10,20035088	4	10,80662281
4	10,16035088	4	10,81662281
4	10,17035088	4	10,81662281

Tabel 1: Datele achiziționate utilizând o placă de achizie de date de tip (NI) pentru motorul stâng (a) și drept (b)

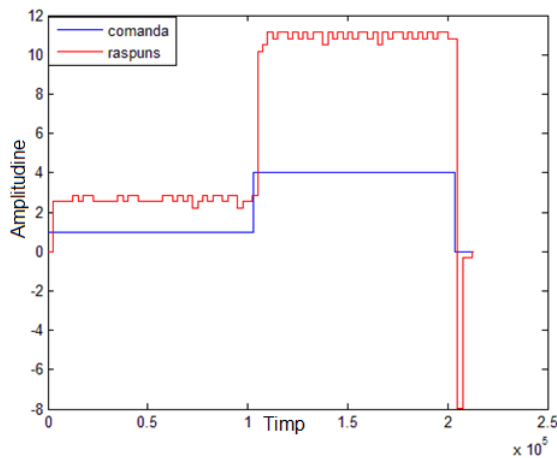


Figura 3. Turația motorului stâng în funcție de variația intrării (tensiune aplicată servoamplificatorului)

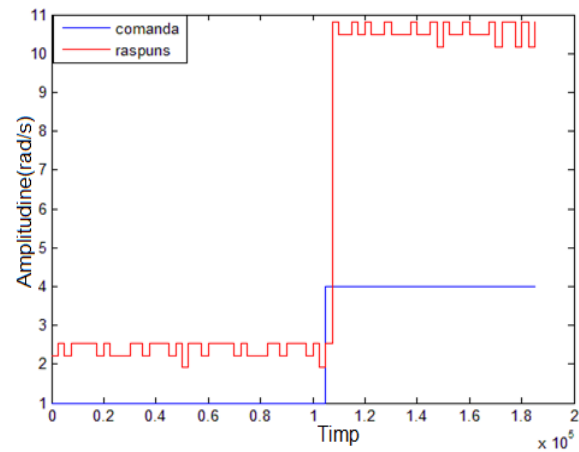


Figura 4. Turația motorului drept în funcție de variația intrării (tensiune aplicată servoamplificatorului)

În urma identificării au rezultat următoarele funcții de transfer:

$$H_L(s) = \frac{2.6}{0.28*s+1}$$

$$H_R(s) = \frac{2.7}{0.3*s+1}$$

În cazul sistemelor de ordinul I, utilizarea unui regulator de tip PI permite alocarea exactă a polilor sistemului rezultat, prin alegerea corespunzătoare a componentelor K_p și K_i .

Pentru acordarea regulatorului de tip PI [3],[4],[7] se va folosi metoda alocării polilor. Metoda alocării este o metodă de proiectare analitică și constă în transpunerea performanțelor impuse într-o anumită repartitie a polilor și zerourilor funcției de transfer a sistemului în buclă închisă.

În urma calculelor au rezultat următoarele valori:

$$K_{pL} = 0.6, K_{iL} = 2;$$

$$K_{pR} = 0.5, K_{iR} = 2.1;$$

Rezultatele experimentale (reale) utilizând reglatoarele PI integrate în mediul RealTime Windows Target din Matlab sunt prezentate în figurile următoare:

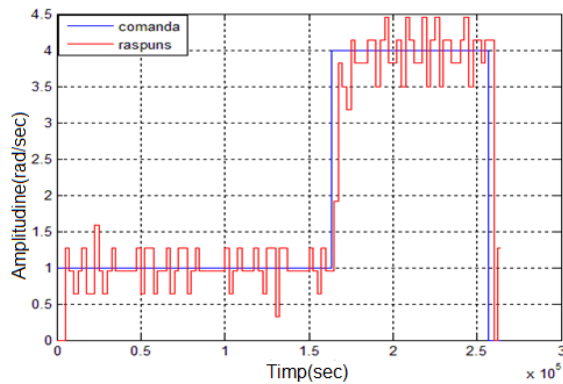


Figura 5. Turația motorului stâng în funcție de referință

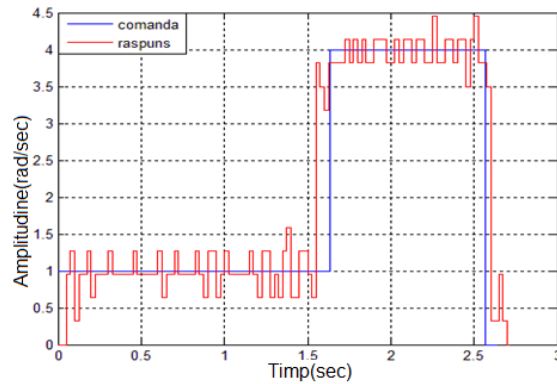


Figura 6. Turația motorului drept în funcție de referință

Schema Simulink ce poate fi folosită mai departe pentru proiectarea sistemului de conducere și navigație bazată pe sistem video-biometric (etapa viitoare a proiectului) a scaunului cu roțile este prezentată în figura 7.

Soluție software pentru implementarea encoderelor

Pentru implementarea soluției software se utilizează placa de achiziții PCI-6024E și biblioteca dedicată Matlab/Simulink Real-Time Windows Target ce permite lucru în timp real.

Deoarece rezoluția encoderelor este de 10.000 de pulsuri pe rotație se lucrează la o perioadă de eșantionare de 0.0001 s, cu toate că perioada de eșantionare este foarte mică se folosește un generator extern de semnal care generează un impuls odată la 0,1 sec pentru a determina momentul în care să se determine parametrii ce trebuie calculați.

Viteza unghiulară a fiecărei roți poate fi determinată utilizând următoarele ecuații:

$$\omega = \frac{(\theta_{pas} \cdot k - \theta_{pas} \cdot k_{ant})}{t}$$

$$\theta_{pas} = \frac{360^\circ}{nr.pulsuri\ pe\ rotație} \cdot \frac{\pi}{180^\circ}$$

unde: - θ_{pas} unghiul, în rad;
 - k nr. de pulsuri la pasul curent;
 - k_{ant} nr. de pulsuri de la pasul anterior;
 - t perioada de eșantionare.

Nr. de pulsuri pe rotație reprezintă numărul total de pulsuri realizat la o rotație completă a encoderului.

În figura 8 sunt arătate grafic datele extrase experimental utilizând formulele de mai sus.

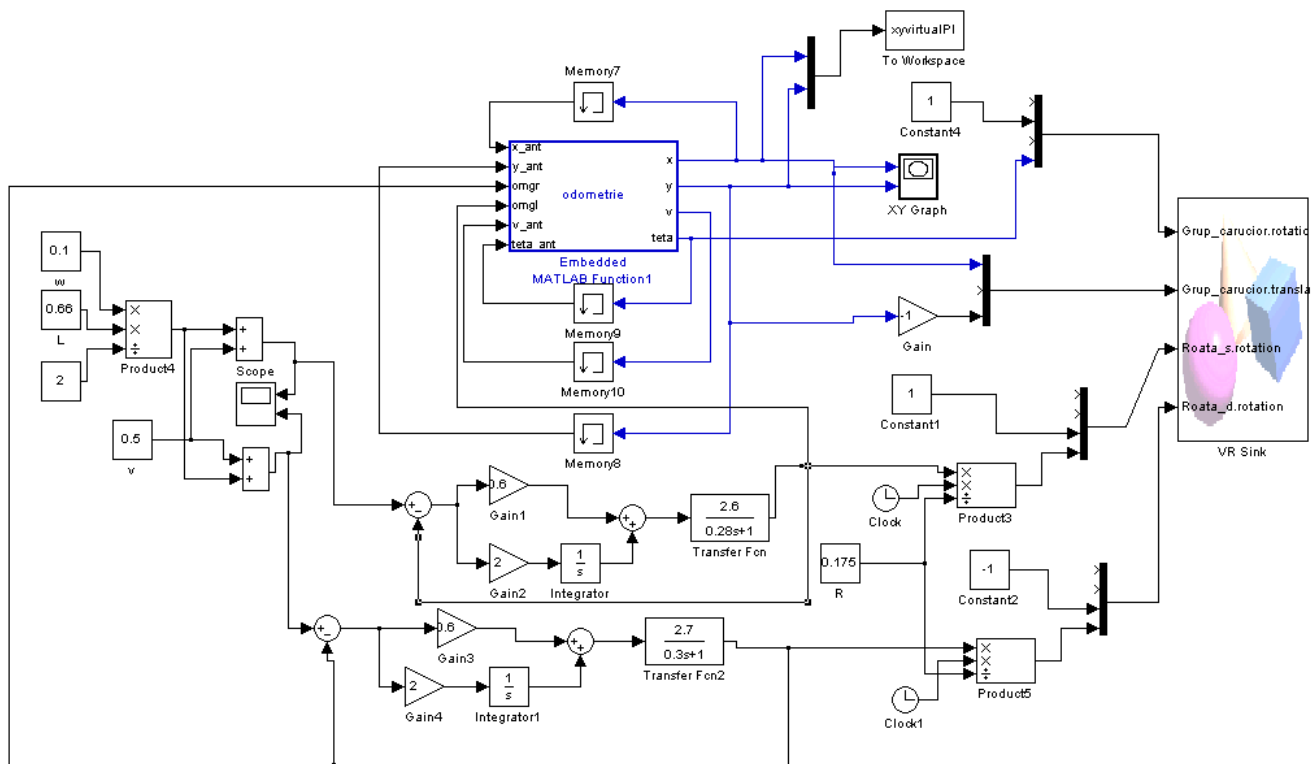


Figura 7. Modelul cinematic al scaunului cu roțile utilizat în mediul Simulink necesar dezvoltării metodelor de conducere și navigare.

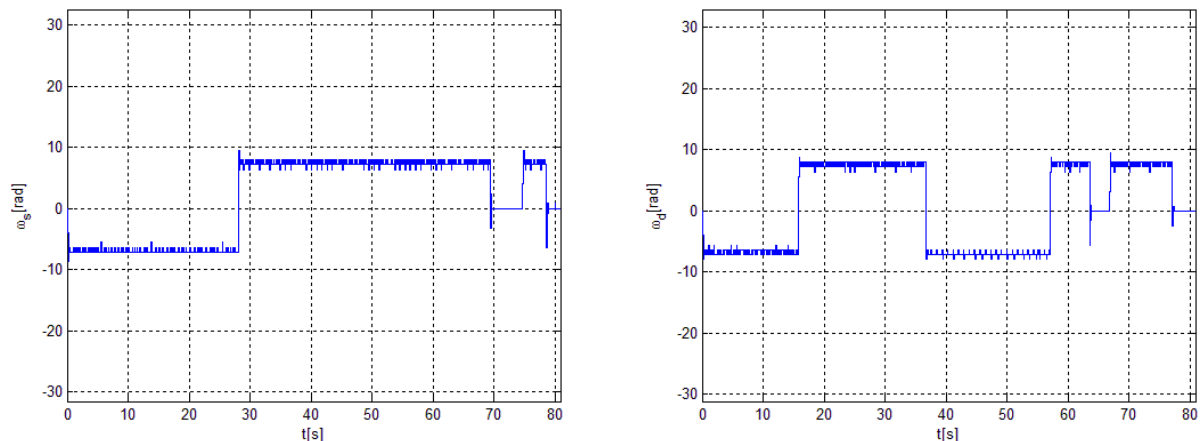


Figura 8. Exemplu de achiziție de date (viteza unghiulară la cele două roți motoare).

Act 1.4 (A.2 - Cercetare Industrială)

Testarea Prototipului 1 (2DW/2FW Cirrus Power Wheelchair) cu sistemul de navigație bazat pe videobiometria ochiului. Grad de realizare 100%

Sistemul de navigație constă din: sistemul de prelucrare a imaginilor, interfață grafică cu utilizatorul și controlul scaunului electric mobil. Sistemul de prelucrare a imaginilor are rolul de a defini comenzile pentru direcție. Întregul ansamblu de informație din cadrul procesării de imagine include detecția ochiului, identificarea direcției și validarea acesteia prin medierea valorilor a 10 cadre. Poziția ochiului generează o comandă, care poate fi de forma șirului de caractere ÎNAINTE, STANGA,

DREAPTA, pentru direcția înainte, stânga, respectiv dreapta sau atunci când se dorește oprirea elementelor de execuție se va genera comanda STOP prin nedetecția irisului, la închiderea ochiului. Aceste comenzi sunt trimise către interfață grafică cu utilizatorul realizată în mediul de programare LabVIEW.

Etapele parcurse pentru identificarea irisului și a centrului acestuia sunt ilustrate în Figura 9.

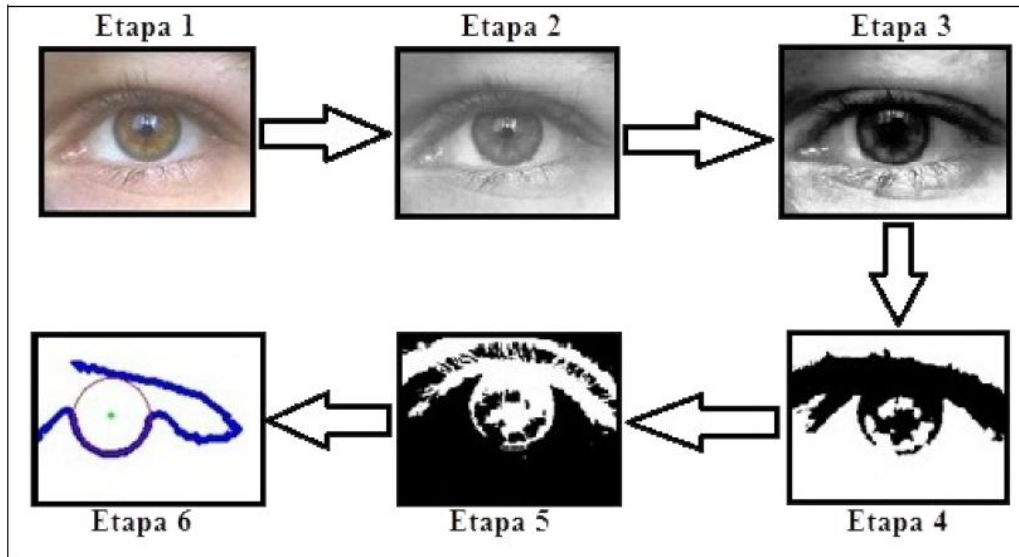


Figura 9. Etapele parcurse pentru identificarea irisului și a centrului acestuia

Pentru identificarea direcției, utilizând centrul cercului înscris în jurul irisului, s-a utilizat un algoritm de măsurare a centrului față de poziția pe axa OX (0-320 pixeli) a imaginii. Atunci când centrul cercului se află sub limita de 100, direcția va fi stânga; atunci când este peste 200, direcția va fi dreapta, iar pentru centru se va lua lungimea valorilor intermediare. Comenzile vor fi trimise către interfață grafică cu utilizatorul sub forma șirurilor de caractere: ÎNAINTE – pentru direcția înainte, STANGA – pentru direcția stânga, DREAPTA – pentru direcția dreapta și STOP – comanda de oprire a elementelor de execuție ale scaunului. Comanda STOP, este generată prin nedetectarea irisului, la închiderea ochiului utilizatorului, atunci când acesta dorește să se oprească.

S-au definit 2 benzi de siguranță pe axa OY (0÷240 pixeli), una superioară cuprinsă în intervalul (180÷240) și una inferioară cuprinsă în intervalul (0÷40). Aceste benzi au rolul de a filtra cercurile false care pot apărea în urma mișcărilor involuntare ale ochiului. Figura 10 descrie implementarea soluției pentru identificarea direcției prin mișcările ochiului utilizatorului.

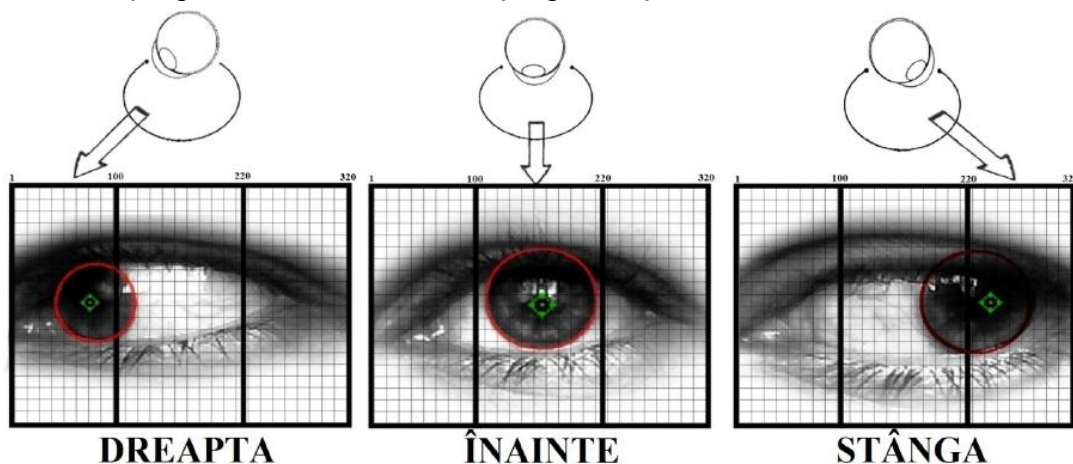


Figura 10. Ilustrare principiu pentru identificarea direcției

Interfața grafică cu utilizatorul, creată cu LabVIEW, este ilustrată în Figura 11. S-a evidențiat cu roșu săgețile indicatoare pentru direcție inactivă, și cu verde direcția aleasă de către utilizator.

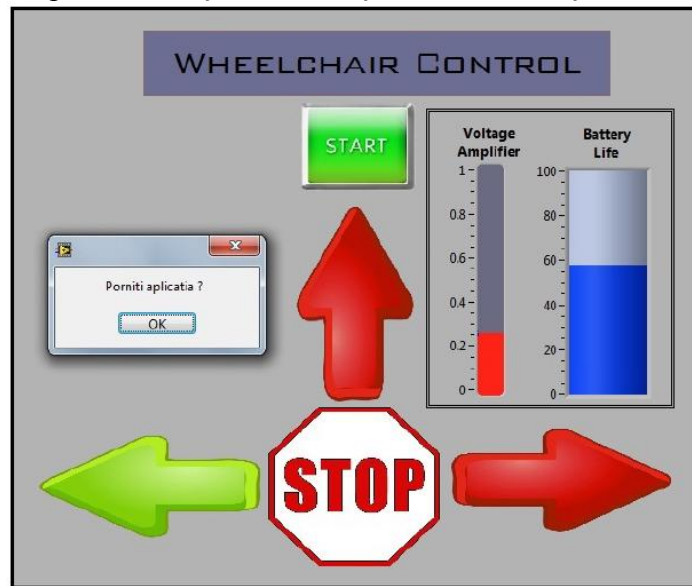


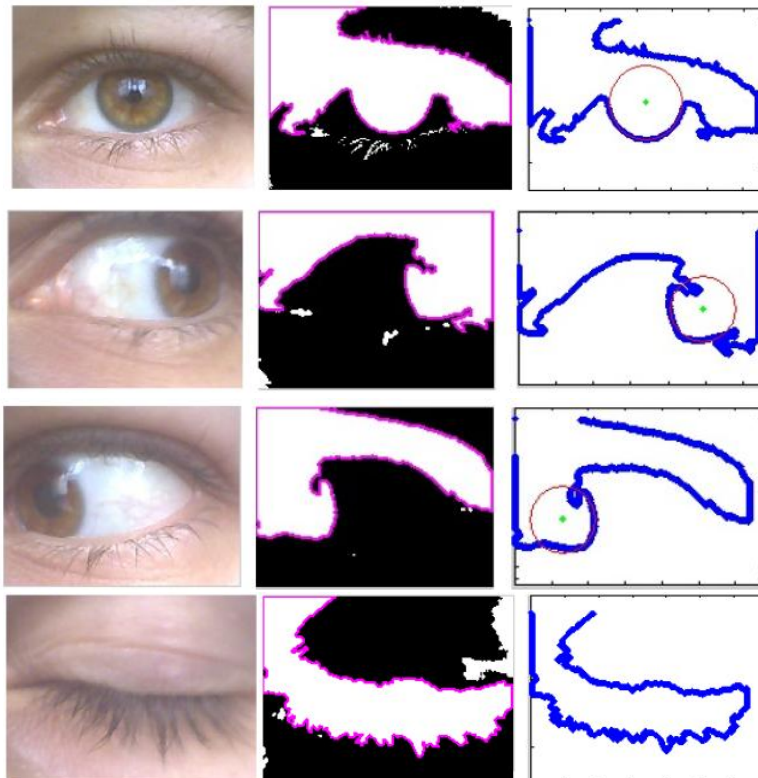
Figura 11. Interfața grafică cu utilizatorul

Diagrama bloc LabVIEW asociată interfeței grafice cu utilizatorul este ilustrată în Figura 12.

Rezultate experimentale

La evaluarea rezultatelor experimentale și al acurateții acestora s-a ținut cont de o serie de factori cum ar fi puterea de procesare al computerului instalat pe scaunul electric mobil, elementele perturbatoare în identificarea irisului etc.

Prezentarea rezultatelor experimentale pentru direcția ÎNAINTE, STÂNGA, DREAPTA, STOP:



Prototipul 2 (Autonomous Robot 2DW/1FW, 2DW/2FW), robot mobil cu două roți motoare și una sau două roți libere, (2DW/1FW, 2DW/2FW), echipat cu manipulator robotic cu 6 grade de libertate (6-DOF)

Rezumat Etapa 1:

Prototipul 2 este conceput ca sistem robotic autonom cu două roți motoare și una sau două roți libere (2DW/1FW sau 2DW/2FW), având un sistem de conducere de la distanță bazată pe tehnici neliniare avansate (sliding-mode și backstepping) și o structură de navigație și evitare obstacole, bazată pe ultrasunete și eventual laser. Platforma mobilă se va echipa cu un manipulator cu 6 grade de libertate (6-DOF) al cărui sistem de conducere și poziționare de la distanță sunt bazate pe tehnici neliniare sliding-mode și eventual servoing vizuale. Prototipul va fi capabil să transporte și să manipuleze sarcini de greutate mică, medii și mari, fiind destinat a opera în regim colaborativ pentru deservirea liniilor flexibile de asamblare și de prelucrare, pentru a le face reversibile, adică să permită și dezasamblare, respectiv reprecizie.

Act. 1.2 (A.2 - Cercetare Industrială)

Modelarea cinematică a sistemului autonom corespunzător Prototipului 1, modelare ca robot mobil cu două roți motoare și două libere (2DW/2FW). Grad de realizare 100%

Modelul cinematic

Se consideră un WMR cu 2 roți motoare și două directoare (2DW/2SW) prezentat în Figura 13, caracterizat de variabila generalizată $q = (q_1, q_2, \dots, q_n)$. Se presupune că roțile vehiculului se rotesc fără a aluneca, astfel robotul este supus constrângerilor nonholonomice descrise de ecuația:

$$A(q) \cdot \dot{q} = 0 \quad (2.1),$$

unde $A(q)$ este matricea asociată constrângerilor.

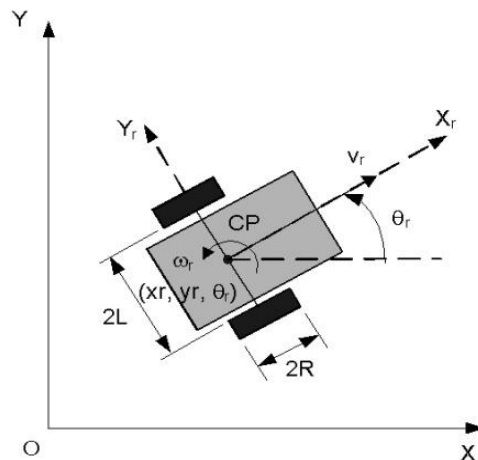


Figura 13 Modelul cinematic al roboților mobili

Pentru robotul mobil din Figura 13, Oxy este sistemul de coordonate, CPX_r, Y_r este sistemul de coordonate atașat robotului, distanța dintre CP și centrul de greutate este d , CP se află la mijlocul distanței dintre cele 2 roți motoare, în acest caz avem constrângerile:

$$\dot{y}_r \cdot \cos \theta_r - \dot{x}_r \cdot \sin \theta_r - d \cdot \dot{\theta} = 0 \quad (2.2)$$

$$x_r \cdot \cos \theta_r + y_r \cdot \sin \theta_r + L \cdot \dot{\theta} = R \cdot \dot{\phi}_d \quad (2.3)$$

$$x_r \cdot \cos \theta_r + y_r \cdot \sin \theta_r - L \cdot \dot{\theta} = R \cdot \dot{\phi}_s \quad (2.4)$$

Matricea $A(q)$ devine:

$$A(q) = \begin{bmatrix} \sin \theta_r & -\cos \theta_r & d & 0 & 0 \\ \cos \theta_r & \sin \theta_r & b & -r & 0 \\ \cos \theta_r & \sin \theta_r & -b & 0 & -r \end{bmatrix} \quad (2.5)$$

Configurația robotului poate fi reprezentată utilizând cinci variabile generalizate $q = [x_r \ y_r \ \theta_r \ \phi_d \ \phi_s]$, unde (x_r, y_r) sunt coordonatele lui CP, θ_r este orientarea, iar ϕ_d, ϕ_s sunt unghiurile roților motoare. Fie matricea $S(q)$ astfel încât:

$$S^T(q) \cdot A^T(q) = 0 \quad (2.6)$$

$$S(q) = \begin{bmatrix} \frac{r}{2 \cdot L} (L \cdot \cos \theta_r - d \cdot \sin \theta_r) & \frac{r}{2 \cdot L} (L \cdot \cos \theta_r + d \cdot \sin \theta_r) & & & \\ \frac{r}{2 \cdot L} (L \cdot \sin \theta_r + d \cdot \cos \theta_r) & \frac{r}{2 \cdot L} (L \cdot \sin \theta_r - d \cdot \cos \theta_r) & & & \\ & \frac{r}{2 \cdot L} & & -\frac{r}{2 \cdot L} & \\ & 1 & & 0 & \\ & 0 & & 1 & \end{bmatrix} \quad (2.7)$$

Modelul cinematic al robotului devine:

$$\dot{q} = S(q) \cdot v \quad (2.8)$$

unde $v = [v_d \ v_s]$ reprezintă vitezele unghiulare ale roților motoare.

Din (2.8) se obține:

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \\ \dot{\phi}_d \\ \dot{\phi}_s \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos \theta_r & \frac{r}{2} \cos \theta_r \\ \frac{r}{2} \sin \theta_r & \frac{r}{2} \sin \theta_r \\ \frac{r}{2 \cdot L} & -\frac{r}{2 \cdot L} \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_d \\ v_s \end{bmatrix} \quad (2.9)$$

Se cunoaște relația dintre viteza liniară și viteza unghiulă și vitezele unghiulare ale roților motoare:

$$\begin{bmatrix} v_d \\ v_s \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{L}{r} \\ \frac{1}{r} & \frac{1}{r} \end{bmatrix} \cdot \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \quad (2.10)$$

Utilizând relația (2.10) în (2.9) modelul cinematic se poate simplifica astfel:

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos \theta_r & 0 \\ \sin \theta_r & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \quad (2.11)$$

în care:

x_r - reprezintă poziția robotului pe axa Ox

y_r - reprezintă poziția robotului pe axa Oy

θ_r - reprezintă orientarea robotului

v_r - reprezintă viteza liniară a robotului

ω_r - reprezintă viteza unghiulară a robotului.

O altă formă pentru sistemul (2.11) este:

$$\begin{cases} \dot{x}_r = v_r \cdot \cos \theta_r \\ \dot{y}_r = v_r \cdot \sin \theta_r \\ \dot{\theta}_r = \omega_r \end{cases} \quad (2.12)$$

Modelele cinematice descriu mișcarea robotului sau a vehiculului și nu iau în calcul forțele care acționează asupra lor și sunt folosite la calculul comenzii pentru conducerea WMR.

Pentru determinarea modelului cinematic al WMR am considerat variabilele generalizate ale sistemului, rotirea roților robotului fără alunecare și aplicat constrângerile nonholomice specifice acestui caz. Modelul rezultat astfel are cinci variabile: 2 variabile reprezintă *centrul geometric* al WMR, o variabilă reprezentând *unghiul de direcție* al robotului și două variabile reprezentând *unghiurile fiecărei roți*. Aceste variabile depind de vitezele de rotire ale roților. Modelul este simplificat pentru că s-a dorit doar calculul coordonatelor carteziene ale centrului geometric și unghiul de direcție, înlocuind vitezele celor două roți cu viteza liniară și viteza unghiulară a robotului.

Studiu de caz 1

Cercetarea se realizează utilizând robotul mobil Pioneer P3-DX de la Adept Mobile Robots. Robotul mobil Pioneer 3-DX PowerBot este un robot condus diferențial (2DW/2SW) și echipat cu sonare. Robotul Pioneer va fi echipat cu un braț articulat cu 7 grade de libertate și un gripper.

Platforma robotică vine echipată cu motoare și controlere, toate controlate de un microcontroller încorporat care acționează ca un server și software client pentru roboții mobili. Dezvoltarea de software include Advanced Robotics Interface for Applications (ARIA) și ArNetworking, dezvoltate sub licență publică GNU, și copmpletă cu librării complet documentate pentru C++, Java și Python și cod sursă.

Roboții sunt prevăzuți cu o conexiune serială RS232 pentru comunicația cu exteriorul, iar conectarea la calculator se realizează utilizând o conexiune wireless, utilizând un access point wireless și un universal device server pentru conversia de la protocolul RS232 la protocolul Ethernet.

Simularea a fost realizată cu ajutorul software-ului MobileSim de la Mobile Robots - soft pentru simularea platformelor MobileRobots/ActivMedia și mediile lor și experimentare cu ARIA. Înlocuiește SRIsim distribuit anterior cu ARIA. MobileSim are la bază simulatorul Stage, creat de Richard Vaughan, Andrew Howard, și alții ca parte din proiectul Player/Stage, cu niște modificări făcute de MobileRobots. MobileSim poate simula comportamentul tuturor roboților produși de MobileRobots. Pentru simulare a fost creată o hartă în Mapper3basic, încărcată ulterior, în MobileSim. Programele sunt scrise în C++ și compilate în Visual Studio. Comunicarea cu simulatorul se realizează cu ajutorul funcțiilor ARIA.

Se propun doua studii de caz care difera prin traiectoria pe care se deplaseaza robotul, si prin algoritmul de deplasare al acestuia.

Având în vedere că robotul trebuie să se deplaseze pe o linie de fabricație, acesta se va mișca într-o încăpere, după o traiectorie prestabilită, spre posturile de asamblare și/sau dezasamblare.

Studiul de caz propus, în continuare, propune într-o harta, 7 posturi de lucru, spre care robotul se deplasează în 4 situații care se aleg la începutul acțiunii: transport de la postul 1 la postul 7 (caz 1), transport de la postul 7 spre postul 1 (caz 2), transport de la postul 1 la postul 7 efectuând funcții de asamblare/dezasamblare (caz 3) și transport de la postul 7 la postul 1 efectuând funcții de asamblare/dezasamblare (caz 4). Harta, traseul de mișcare și numărul posturilor de lucru, pot fi modificate, prin introducerea unor noi date în fișierul *harta.txt*.

În această situație controlul este în buclă deschisă. În capitolele ulterioare se va gestiona un control în buclă închisă discontinuă, care conține comutații la frecvențe mari. Avantajul major îl reprezintă robustețea la variația parametrilor și perturbații, precum și faptul că sistemul se comportă ca un sistem de ordin redus în comparație cu procesul și faptul că alunecarea pe suprafețele de comutație nu este afectată de incertitudini de model sau perturbații.

Harta propusă în studiile de caz este prezentată în figura 14. Se observă casa robotului în punctul de coordonate (0,0) și traiectoria de mișcare pe care se identifică 10 puncte.

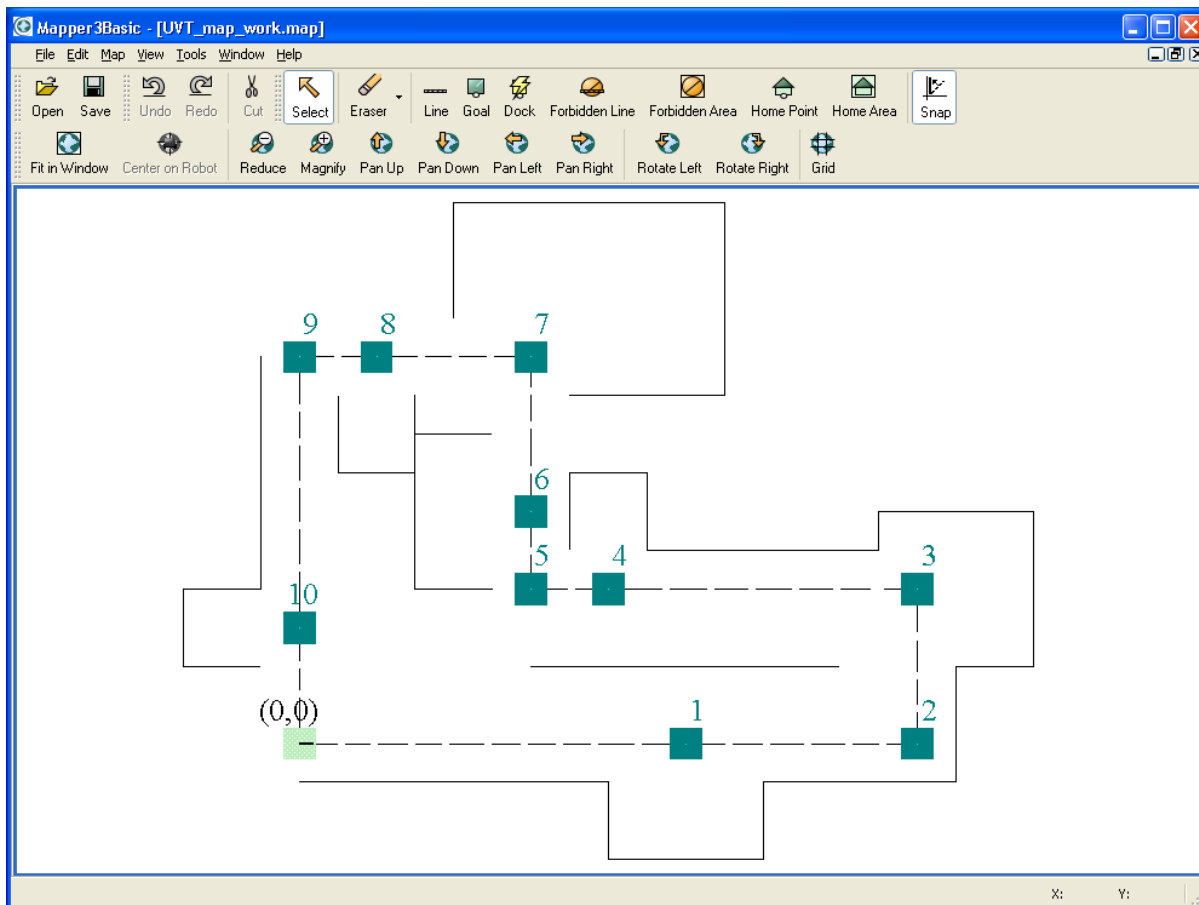


Figura 14. Harta aplicației

Harta este prezentată și în figura 15 în stânga, utilizând de această dată WordPad pentru fișierul *UVT_map.map*. În figura 15, dreapta se poate observa conținutul fișierului *harta.txt* ce conține parametrii de mișcare – *coordonate (x,y) punct* spre care se îndreaptă robotul, *unghiul de mișcare* executată când robotul este pe loc pentru a se poziționa spre următorul punct de pe traseu, *unghiul de rotație* pe loc la intrarea în postul de lucru, *coordonate x și y pentru punctul de asamblare/dezasamblare* respectiv *flag-ul de semnalizare* a existenței (1) sau inexistenței (0) unui post de lucru în poziția cu coordonatele indicate.

Anterior se citește din fișier numărul de puncte ale traseului și numărul de posturi de lucru, necesare pentru a inițializa matricea de mișcare din utilizată în programul C. Pentru cazul PID primii trei parametrii din fișier reprezintă k_p, k_i, k_d folosiți în conducerea cu PID.

```

UVT_map.map - Notepad
File Edit Format View Help
2D-Map
LineMinPos: -10000 -10000
LineMaxPos: 20000 20000
NumLines: 29
LinesAreSorted: true
LINES
3000 1000 7000 1000
0 -500 4000 -500
4000 -500 4000 -1500
4000 -1500 6000 -1500
6000 -1500 6000 -500
6000 -500 8500 -500
8500 -500 8500 1000
8500 1000 9500 1000
9500 1000 9500 3000
9500 3000 7500 3000
7500 3000 7500 2500
7500 2500 4500 2500
4500 2500 4500 3500
4500 3500 3500 3500
3500 3500 3500 2500
2500 2000 1500 2000
1500 2000 1500 4000
1500 4000 2500 4000
3500 4500 5500 4500
5500 4500 5500 7000
5500 7000 2000 7000
2000 7000 2000 5500
1500 4500 1500 3500
1500 3500 500 3500
500 3500 500 4500
-500 5000 -500 2000
-500 2000 -1500 2000
-1500 2000 -1500 1000
-1500 1000 -500 1000
DATA
Ln 1, Col 1

harta.txt - Notepad
File Edit Format View Help
0.1 0.2 0.3
12 7 750
0 0 0 0 0 0 0
5000 0 0 270 0 -1000 1
8000 0 0 0 0 0 0
8000 2000 90 0 1000 0 1
4000 2000 180 90 0 1000 1
3000 2000 180 0 0 0 0
3000 3000 90 180 -1000 0 1
3000 5000 90 45 1000 1000 1
1000 5000 180 270 0 -1000 1
0 5000 180 0 0 0 0
0 1500 270 180 -1000 0 1
0 0 270 0 0 0 0
Ln 1, Col 1

```

Figura 15. Harta aplicației în NotPad (stânga) și Matricea de mișcare (dreapta)

Punctele spre care se îndreaptă robotul sunt numerotate cu 0 până la 11 putând să fie vizualizate în figura 14 iar coordonatele acestor puncte sunt identificabile în figura 15 stânga.

Simularea presupune lansarea în execuție a hărții, cu ajutorul programului MobileSim și poziționarea robotului pe coordonatele (0,0). În acest sens am creat un fișier de comenzi *UVT_Mapp.bat*.

```

UVT_Mapp.bat - Notepad
File Edit Format View Help
@echo off
cls
cd C:\Program Files\MobileRobots\MobileSim
echo Se lanseaza in executie Mobilsim.exe
echo Acesta ruleaza acum !
Mobilesim -m UVT_map.map -r p2de:robot1 -start 0,0,0
Ln 1, Col 1

```

Figura 16. Fișierul de comenzi pentru lansarea aplicației MobileSim și poziționarea robotului

După lansarea în execuție a programului *UVT_Robot.DebugVC10.exe* se permite din meniu prin tastarea comenzilor de la 1 la 8 realizarea unei acțiuni specifice. În figurile 17 și 18 sunt prezentate cazurile 1 și 3 din cele 4 expuse anterior.

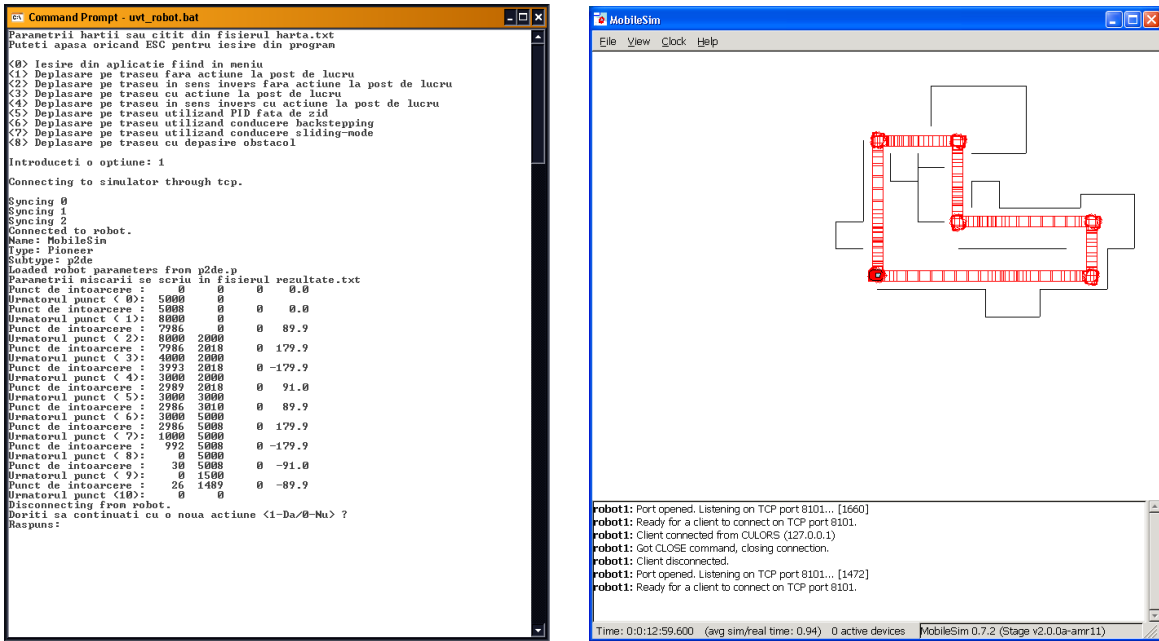


Figura 17. Rezultate aplicație în cazul 1 (stânga) și traseul mișcării (dreapta)

Mișcarea robotului utilizează funcțiile ARIA, *robot.setHeading()* pentru setarea unghiului, urmată de *gotoPoseAction.setGoal()* pentru deplasare spre o poziție definită și în final de funcția *Post_Lucru()*, funcție definită de utilizator prin care se pătrundere pentru execuția unei acțiuni pe linia de fabricație la un post de lucru, dacă acest lucru este setat.

În cazurile 1 și 2, robotul îndeplinește doar funcția de transport.

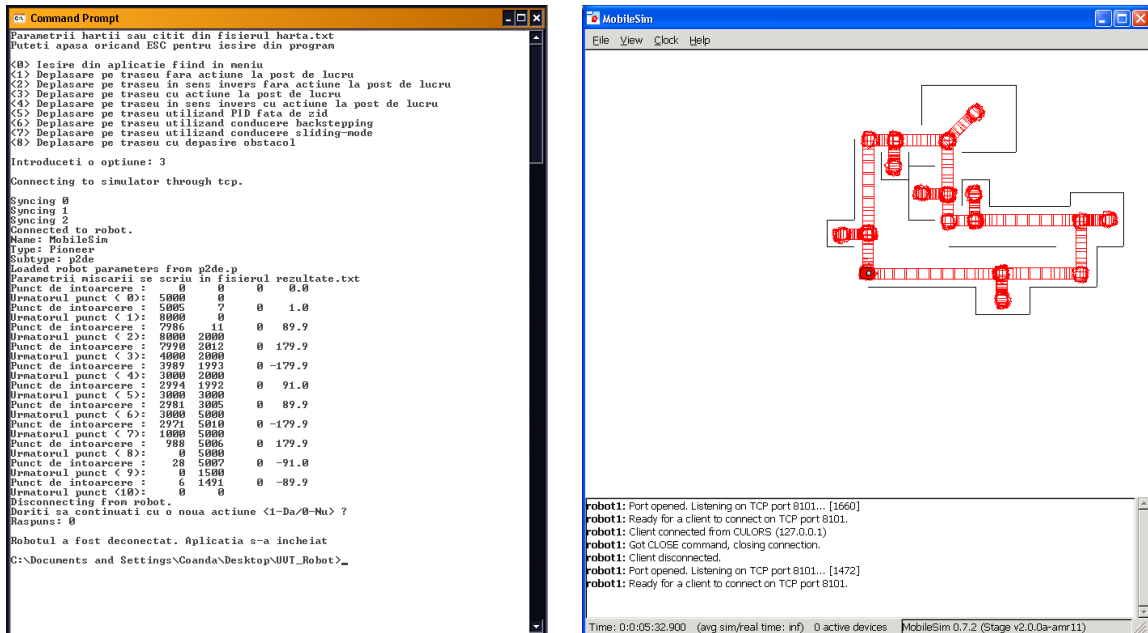


Figura 18. Rezultate aplicație în cazul 3 (stânga) și traseul mișcării (dreapta)

Linie	x	y	viteza
3	0	30	0.0
9	0	60	0.0
17	0	90	0.0
29	0	121	0.0
46	0	151	0.0
64	0	181	0.0
84	0	212	0.0
109	0	244	0.0
137	0	274	0.0
168	0	304	0.0
202	0	334	0.0
237	0	364	0.0
277	0	394	0.0
321	0	424	0.0
365	0	454	0.0
413	0	484	0.0
466	0	514	0.0
521	0	544	0.0
579	0	574	0.0
641	0	604	0.0
704	0	634	0.0
770	0	664	0.0
840	0	695	0.0
914	0	725	0.0
989	0	750	0.0
1066	0	750	0.0
1141	0	750	0.0
1216	0	750	0.0
1292	0	750	0.0
1367	0	750	0.0
1443	0	750	0.0
1518	0	750	0.0
1595	0	750	0.0
1670	0	750	0.0
1745	0	750	0.0

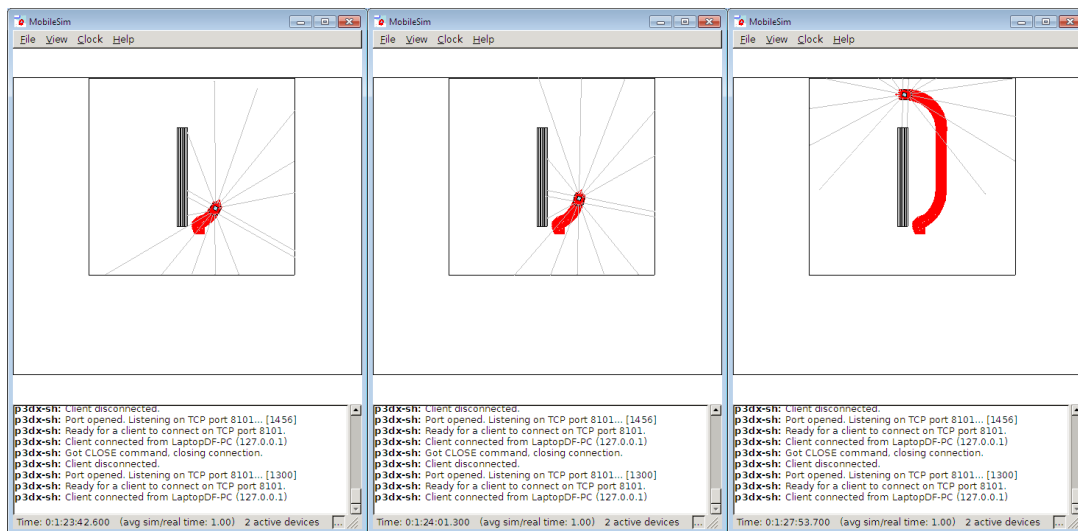
Figura 19. Fișier de rezultate caz 1

În cazul 3 prezentat în figura 18, precum și în cazul 4 robotul îndeplinește pseudo-funcția de asamblare/ dezasamblare, acolo unde gripper-ul respectiv camera video vor activa.

În fișierul *rezultate.txt* se înregistrează coordonatele x și y ale robotului la un moment dat în raport cu punctul de start, viteza de deplasare (viteza maximă posibilă fiind definită în fișierul de configurare – *harta.txt*, în exemplul nostru la valoarea maximă de 750) și unghiul sub care acesta se deplasează raportat la poziția de start a acestuia. Ele permit evaluarea erorilor în deplasare și în mod evident concură la necesitatea unui control în buclă închisă prezentat în cele ce urmează. Se observă cum viteza de deplasare crește spre maxim sau scade funcție de distanța liberă de deplasare și de valoarea maximă setată.

Studiu de caz 2

În cel de-al studiu de caz se propune deplasarea robotului pe o traiectorie rectilinie cu mentinerea constantă a distanței față de „perete”. Algoritmii impuși are la bază conducerea în buclă închisă cu regulator tripozitional. Regulatorul comandă viteza și unghiul de înaintare a WMR atunci când eroarea atinge valorile limită prestabilite.



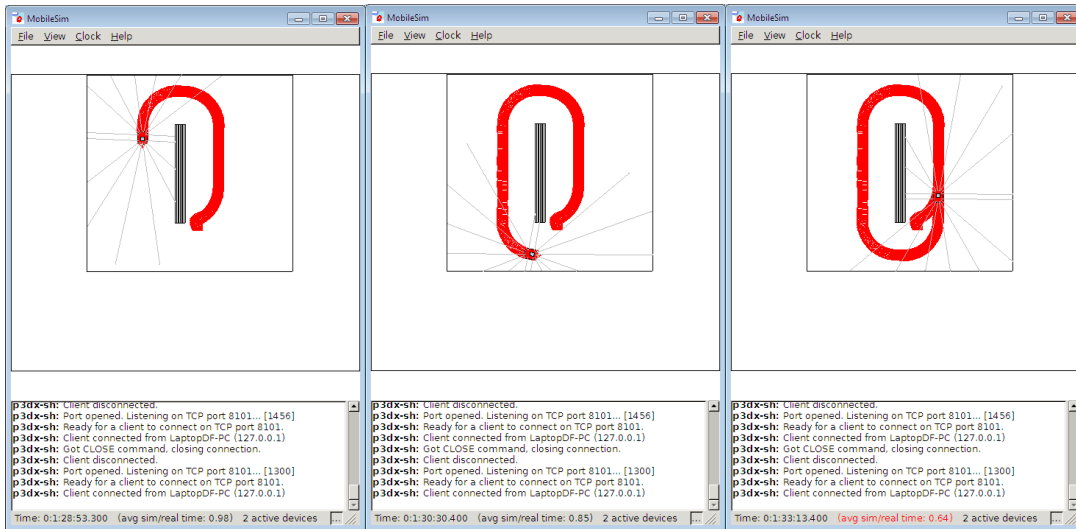
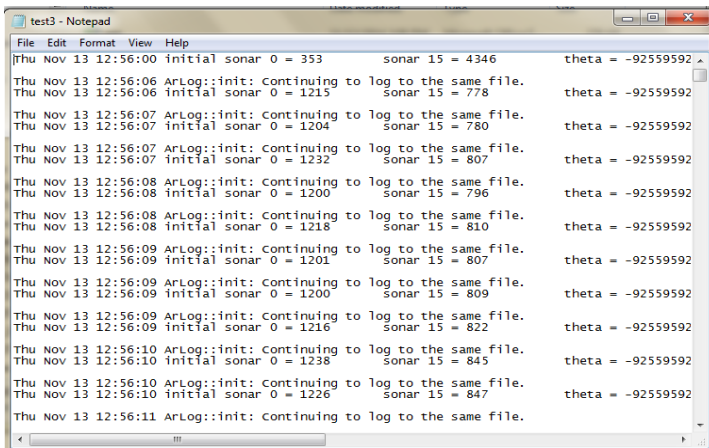


Figura 20. Deplasare robot



Evaluarea pozitei curente a robotului se face prin comparatia valorilor returnate de sonarul 0 si 15 (cele 16 sonare sunt dispuse radial, sonarul 0 si 15 sunt aflati in vecinatate. Eroarea de pozitie se calculeaza iterativ, la interval de timp de o secunda.

Figura 21. Fişierul de rezultate caz 2

Aceste valori au fost trimise intr-un fisier extern (Figura 21), ceea ce a permis reprezentarea grafica a variatiei sale. In zonele in care robotul se repositioneaza pentru a urmari zidul (zonele cu intoarceri la 180 grade si reasezarea pe traiectorie) eroarea are variatii semnificative pana ajunge sa se stabilizeze din nou la zero (Figura 22. si Figura 23.).

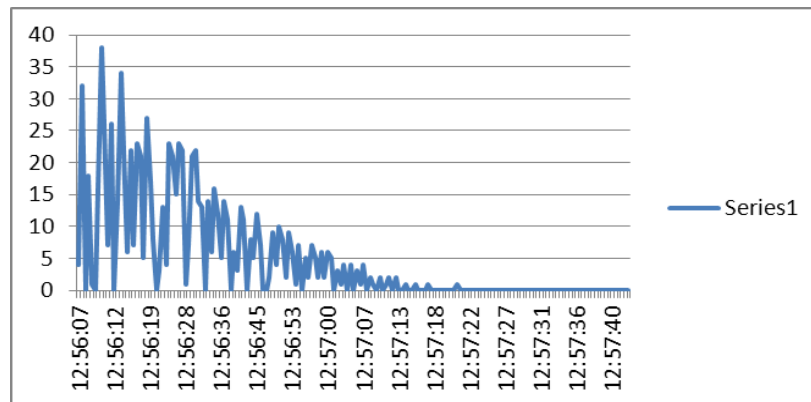


Figura 22. Eroarea de pozitie in timp real, in zona primei intoarceri cu 180⁰ a WMR

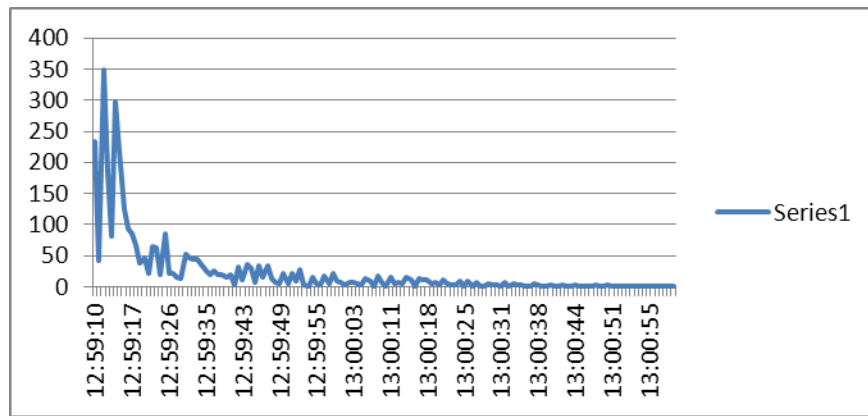


Figura 23. Eroarea de pozitie in timp real, in zona celei de-a doua intoarceri cu 180° a WMR

Concluzii

S-a studiat cinematica sistemului robotic autonom al prototipului 2, modelat ca robot mobil cu două roți motoare și una sau două roți libere (2DW/1FW sau 2DW/2FW) în două abordări distincte. Prima abordarea a realizat conducerea robotului în locații predefinite, fără a include algoritmi de optimizare a deplasării și poziționării în punctele de destinație. În cel de-al doilea studiu de caz s-a implementat deplasarea robotului cu menținerea constantă a distanței față de un obstacol liniar. Viteza de deplasare a robotului, precum și unghiul de înaintare s-au determinat prin conducerea în buclă închisă cu regulator tripozitional.

Prototipul 3 (Autonomous Omnidirectional Vehicle 4DW/SW): Vehicul autonom tip robot mobil cu patru roți motoare și directoare omnidirectionale (4DW/SW) echipat cu manipulator robotic cu 6 grade de libertate (6-DOF), care deservește procese din industria metalurgică și sticla.

Rezumat Etapa 1:

În cadrul acestei etape s-a determinat modelul cinematic al sistemului robotic autonom corespunzător Prototipului 3 modelat ca vehicul autonom omnidirecțional cu 4 roți motoare și directoare (4DW/SW). Acest model este necesar pentru realizarea celei de a doua etape a proiectului: ”Conducerea de la distanță și navigația vehiculului autonom 4DW/SW echipat cu manipulator 6-DOF, în interior și exterior, pe diferite tipuri de carosabil”.

De asemenea s-au efectuat simulări pentru testarea modelului cinematic cât și soluției de control propuse.

Act. 1.2 (A.2 - Cercetare Industrială)

Modelarea cinematică a sistemului robotic autonom al Prototipului 3, modelare ca vehicul autonom omnidirecțional cu 4 roți motoare și directoare (4DW/SW)

Modelului cinematic (realizat 100 %)

În Figura 1 este reprezentat un model geometric al unui vehicul autonom care definește principalele variabile (incluzând alunecarea laterală) necesare pentru a obține modelul cinematic.

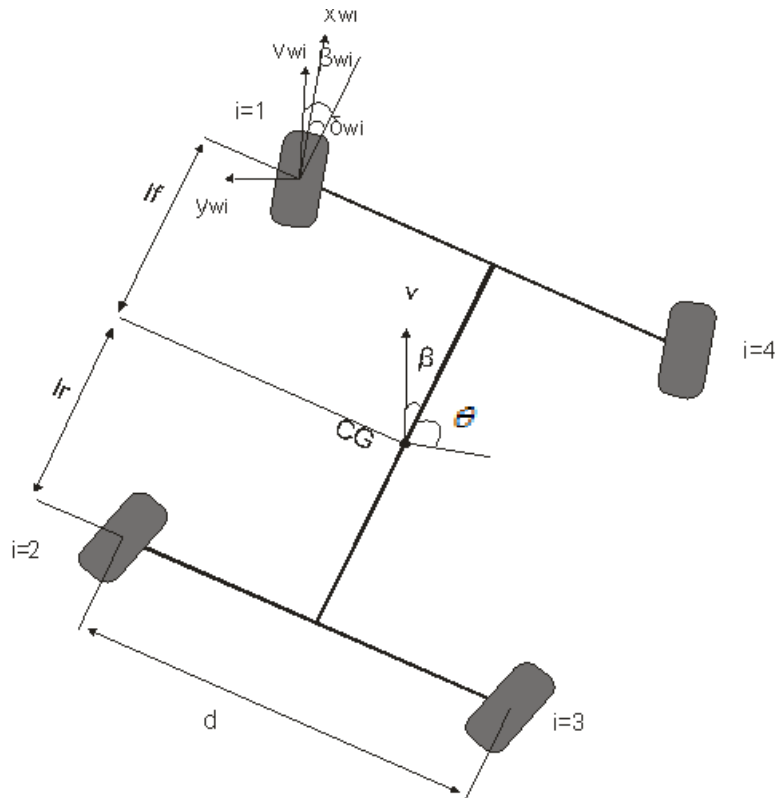


Figura 1. Modelul cinematic al vehiculului autonom SEEKUR

Modelul cinematic al vehiculului autonom descrie mișcarea vehiculului fără a lua în considerare masa sau forțele care acționează asupra vehiculului. Modelul cinematic determinat va fi folosit la proiectarea conducerii backstepping și sliding-mode.

Fiecare roată are un anumit unghi director δ_i și un unghi al alunecării β_i . Unghiul de alunecare definește cât de mare este alunecarea laterală și se calculează în funcție de viteza longitudinală și liniară a roții:

$$\beta_i = \tan^{-1} \left(\frac{v_{y_{wi}}}{v_{x_{wi}}} \right)$$

unde indicele i denotă numărul roții așa cum e arătat în Fig.1, (X_{CG}, Y_{CG}, θ) definește poziția și orientarea centrului de greutate al vehiculului, (x_{wi}, y_{wi}) definește poziția roții i , v și v_{wi} sunt vitezele vehiculului și a fiecărei roți. β reprezintă unghiul de alunecare a vehiculului, l_f , l_r sunt distanțele de la centrul de greutate la roțile din față și din spate. Luând în considerare alunecările, constrângerile nonholonomice se pot exprima astfel:

$$\dot{x}_{CG} \cdot \sin(\beta + \theta) + \dot{y}_{CG} \cdot \cos(\beta + \theta) = 0 ;$$

$$\dot{x}_{wi} \cdot \sin(\beta_{wi} + \theta_{wi}) + \dot{y}_{wi} \cdot \cos(\beta_{wi} + \theta_{wi}) = 0 .$$

Constrângerile geometrice dintre fiecare roată și centrul de greutate pot fi scrise astfel:

$$\left\{ \begin{array}{l} x_{w1} = X_{CG} + l_f \cdot \cos \theta - \frac{d}{2} \cdot \sin \theta \\ x_{w2} = X_{CG} - l_f \cdot \cos \theta - \frac{d}{2} \cdot \sin \theta \\ x_{w3} = X_{CG} - l_r \cdot \cos \theta + \frac{d}{2} \cdot \sin \theta \\ x_{w4} = X_{CG} + l_r \cdot \cos \theta + \frac{d}{2} \cdot \sin \theta \end{array} \right. ; \quad \left\{ \begin{array}{l} y_{w1} = y_{CG} + l_f \cdot \sin \theta - \frac{d}{2} \cdot \cos \theta \\ y_{w2} = y_{CG} - l_f \cdot \sin \theta - \frac{d}{2} \cdot \cos \theta \\ y_{w3} = y_{CG} + l_r \cdot \sin \theta + \frac{d}{2} \cdot \cos \theta \\ y_{w4} = y_{CG} + l_r \cdot \sin \theta + \frac{d}{2} \cdot \cos \theta \end{array} \right. .$$

Considerând roțile 1 și 2 și constrângerile obținem în forma matriceală:

$$A_{12} \cdot \dot{q}_0 = 0 ,$$

$$A_{12} = \begin{bmatrix} \sin \phi_{w1} & -\cos \phi_{w1} & -l_f \cdot \cos(\phi_{w1} - \theta) - \frac{d}{2} \cdot \sin(\phi_{w1} - \theta) \\ \sin \phi_{w2} & -\cos \phi_{w2} & -l_f \cdot \cos(\phi_{w2} - \theta) - \frac{d}{2} \cdot \sin(\phi_{w2} - \theta) \\ \sin \phi_0 & -\cos \phi_0 & 0 \end{bmatrix} ,$$

$$\dot{q}_0 = \begin{bmatrix} \dot{x}_{CG} \\ \dot{y}_{CG} \\ \dot{\theta} \end{bmatrix} ,$$

$$\phi_{wi} = \beta_{wi} + \delta_{wi} + \theta ,$$

$$\phi_0 = \beta + \delta .$$

Considerând o mișcare planară, corp rigid și fără alunecarea roții, modelul poate fi aproximat de modelul bicicletei (Figura 2).

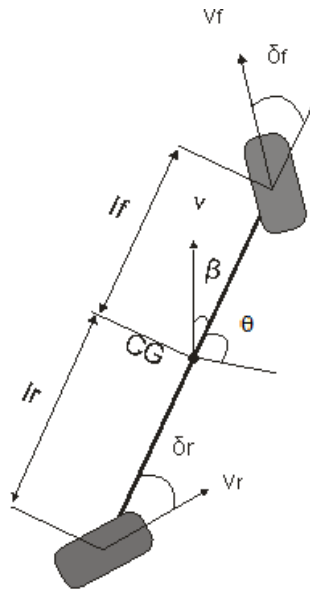


Fig. 2 Modelul cinematic al bicicletei

$$\begin{bmatrix} \sin(\delta_f + \theta) & -\cos(\delta_f + \theta) & -l_f \cdot \cos \delta_f \\ \sin(\delta_r + \theta) & -\cos(\delta_r + \theta) & -l_f \cdot \cos \delta_r \\ \sin(\beta + \theta) & -\cos(\beta + \theta) & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{x}_{CG} \\ \dot{y}_{CG} \\ \dot{\theta} \end{bmatrix} = 0.$$

Se poate obține vectorul \dot{q}_0 care satisface relația:

$$\begin{bmatrix} \dot{x}_{CG} \\ \dot{y}_{CG} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\beta + \theta) \\ \sin(\beta + \theta) \\ \frac{\cos \beta (\tan \delta_f - \tan \delta_r)}{l_f + l_r} \end{bmatrix} \cdot v$$

unde v este viteza liniară a vehiculului și în care:

$$\beta = \arctan \frac{l_f \cdot \tan \delta_r + l_r \cdot \tan \delta_f}{l_f + l_r},$$

Presupunem că unghiul roților vehiculului este limitat de formula:

$$-\frac{\pi}{2} \leq \delta_f, \delta_r \leq \frac{\pi}{2}.$$

Presupunem de asemenea ca vehiculul se deplasează utilizând manevra cu alunecare laterală zero (Zero-side-slip maneuver), propusă de Danwei și Feng. În acest caz unghiul de alunecare este 0 pe întreaga durată a deplasării, iar orientarea vehiculului se potrivește cu unghiul tangent la traiectoria dorită,

$$\theta(t) = \theta_d(t), t : 0 \rightarrow t_f.$$

Unghiurile roților, în acest caz, sunt $\delta_f = -\delta_r$, situație prezentată în Fig. 2. Modelul cinematic al vehiculului în cazul manevrei cu alunecare laterală zero devine

$$\begin{bmatrix} \dot{x}_{CG} \\ \dot{y}_{CG} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\beta + \theta) \\ \sin(\beta + \theta) \\ \frac{(\tan \delta_f - \tan \delta_r)}{2 \cdot l_f} \end{bmatrix} \cdot v \quad \text{sau} \quad \begin{cases} \dot{x}_{CG} = v \cdot \cos \theta \\ \dot{y}_{CG} = v \cdot \sin \theta \\ \dot{\theta} = v \cdot \tan \delta_f / L \end{cases},$$

Unde:

v reprezintă viteza liniară;

δ_f reprezintă unghiul roții din față;

(x_{CG}, y_{CG}) reprezintă coordonatele carteziene ale centrului vehiculului;

θ reprezintă unghiul dintre direcția de mers și axa Ox;

$L = 2 \cdot l_f$ reprezintă distanța dintre roți.

Conducerea sliding-mode în timp discret a vehiculului autonom cu 4 roți motoare și 4 directoare (4DW/SW) (realizat 100 %)

Intrucat algoritmul de conducere este implementat pe sisteme digitale se va folosi conducerea sliding-mode pentru timp discret.

Conducerea sliding-mode în timp discret a roboților mobili și vehiculelor autonome este realizată pornind de la ecuațiile cinematice ale centrului de greutate. Arhitectura conducerii sliding-mode în timp discret a vehiculului autonom cu 4 DW/SW este prezentată în Fig. 3. Modul Odometry din Fig. 3 primește datele de la encoderele vehiculului autonom și calculează poziția, orientarea, viteza liniară și viteza unghiulară a robotului. Acest modul este implementat în softul ARIA de la Mobile Robots și datele calculate pot fi obținute apelând în program funcțiile din ARIA.

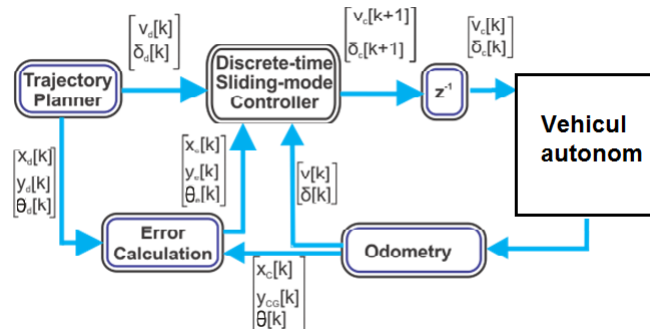


Fig. 3 Arhitectura conducerii sliding-mode în timp discret a vehiculului autonom

Conducerea presupune rezolvarea problemei urmăririi traiectoriei. Rezolvarea acestei probleme necesită proiectarea unui controller capabil de urmărirea traiectoriei dorite. În acest

scop se consideră un robot virtual având traiectoria dorită $q_d(t) = [x_d(t) \ y_d(t) \ \theta_d(t)]^T$

și model cinematic determinat mai sus.

Sistemul poate fi rescris în timp discret astfel:

$$\begin{cases} x_d[k+1] = x_d[k] + v_d[k] \cdot \cos \theta_d[k] \cdot T_s \\ y_d[k+1] = y_d[k] + v_d[k] \cdot \sin \theta_d[k] \cdot T_s \\ \theta_d[k+1] = \theta_d[k] + \frac{v_d[k]}{L} \cdot \tan \delta_d[k] \cdot T_s \end{cases} .$$

În timp discret dinamica erorilor se exprimă astfel:

$$\begin{cases} x_e[k+1] = x_e[k] + (v_r[k] \cdot \cos \theta_e[k] - v_d[k] + y_e[k] \cdot \frac{v_d[k]}{L} \cdot \tan \delta_d[k]) \cdot T_s \\ y_e[k+1] = y_e[k] + (v_r[k] \cdot \sin \theta_e[k] - x_e[k] \cdot \frac{v_d[k]}{L} \cdot \tan \delta_d[k] \cdot T) \cdot T_s \\ \theta_e[k+1] = \theta_e[k] + (\frac{v_r[k]}{L} \tan \delta_r[k] - \frac{v_d[k]}{L} \tan \delta_d[k]) \cdot T_s \end{cases} .$$

Controlerul sliding-mode în timp discret este un controler cu structură variabilă, care efectuează măsurători și calculează comenzile la intervale regulate de timp și păstrează semnalul de comandă constant între intervale. O importantă proprietate a conducerii sliding-mode în timp discret este controlul discontinuu.

Iar comenzile pentru viteza liniară și viteza unghiulară sunt următoarele:

$$\begin{aligned} v[k+1] &= \frac{1}{\cos \theta_e[k] \cdot T_s} \cdot [-(1 - q_1 \cdot T_s) \cdot s_1[k] + \varepsilon_1 \cdot T_s \cdot \operatorname{sgn}(s_1[k] - x_e[k+1] \cdot (1 + k_1) - \\ &\quad - (v_d[k+1] - v_r[k] \cdot \theta_e[k+1] \cdot \sin \theta_e[k] - w_d[k+1] \cdot y_e[k] - w_d[k] \cdot y_e[k+1]) \cdot T_s)] \\ \delta[k] &= \arctan\left(\frac{L}{v_r[k] \cdot \cos \theta_e[k] + k_0 \cdot \operatorname{sgn} y_e[k+1]} \cdot [-(1 - q_2 \cdot T_s) \cdot s_2[k] + \varepsilon_2 \cdot T_s \cdot \operatorname{sgn}(s_2[k]) - \theta_e[k] - \right. \\ &\quad \left. - y_e[k+1] \cdot (k_2 + 1) - (v_r[k+1] \cdot \sin \theta_e[k] + w_d[k+1] \cdot x_e[k] - w_d[k] \cdot x_e[k+1]) \cdot T_s]\right) + \frac{L}{v_r} w_d[k] \end{aligned}$$

Controlerul propus pentru conducerea roboților mobili și vehiculelor autonome a fost testat în simulări realizate cu ajutorul softului MobileSim de la Mobile Robots. MobileSim este un soft pentru simularea platformelor MobileRobots/ActivMedia și mediile lor și experimentare cu ARIA.

MobileSim poate simula comportamentul tuturor roboților produși de MobileRobots.

Comunicarea cu simulatorul se realizează cu ajutorul funcțiilor ARIA.

Softul Aria realizează conectarea automată la simulatorul MobileSim în cazul în care nu este detectat nici un robot conectat la portul COM1. Simulatorul are implementate modelele cinematice ale roboților, funcții pentru simularea sonarelor și laserelor, care sunt folosite pentru a simula comportamentul unui robot real. Programul scris în C++ apelează funcțiile ARIA în cazul în care se dorește trimiterea unor comenzi către simulator sau citirea datelor simulate.

Testul își propune urmărirea unei traiectorii în forma din Fig. 4 utilizând vehiculul autonom SEEKUR (Fig. 6) folosind conducerea sliding-mode în timp discret. Parametrii constanți folosiți în acest experiment sunt:

$$q_1 = 0.9, \quad q_2 = 0.9, \quad \varepsilon_1 = 0.01, \quad \varepsilon_2 = 0.75, \quad k_0 = 30, \quad k_1 = 0.75, \quad k_2 = 15.$$

Acești parametri au fost obținuți prin identificare în urma unor simulări succesive utilizând diferite valori ale parametrilor. În Fig. 5 este prezentată arhitectura conducerii folosite. În acest caz modulul Trajectory Planner folosește planificatorul de traiectorii pentru generarea vitezei unghiulare și unghiului director. Modulul Odometry este implementat de softul ARIA și viteza liniară, unghiul director, poziția și orientarea vehiculului sunt obținute utilizând funcțiile ARIA corespunzătoare. În Fig. 4 este prezentată traiectoria simulată utilizând conducerea sliding-mode în timp discret cu o linie continuă roșie și traiectoria dorită cu o linie întreruptă albastră a vehiculului autonom SEEKUR.

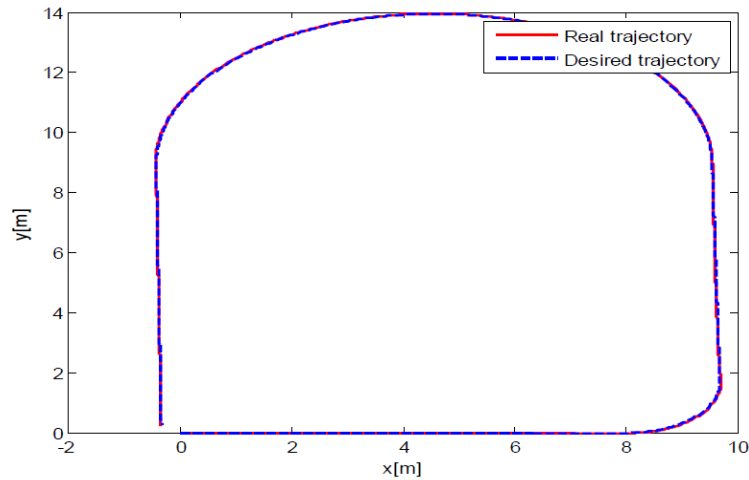


Fig. 4 Traiectoria obținută prin simulare în MobileSim și traiectoria impusă la conducerea sliding-mode în timp discret a vehiculului autonom 4DW/SW SEEKUR.

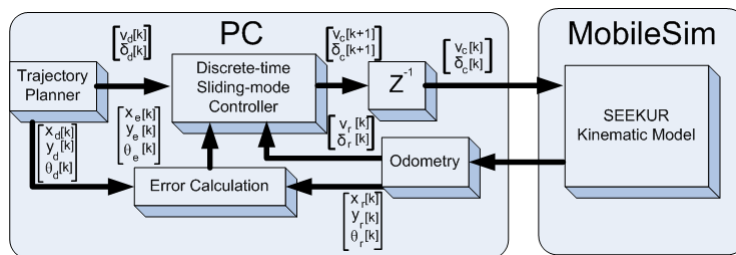


Fig. 5 Arhitectura de conducere sliding-mode în timp discret a Vehiculului autonom 4DW/SW SEEKUR în MobileSIM.



Fig. 6 Vehicul autonom Seekur

Analizand graficele de erori (Fig. 7, 8, 9, 10, 11) se observă performanțe ridicate la urmărirea traiectoriei propuse, cu erori de urmărire mici, iar suprafețele de alunecare oscilează în apropierea valorii zero.

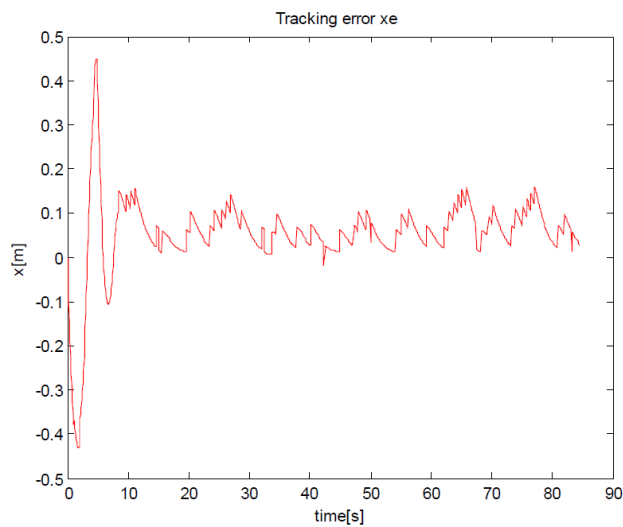


Fig. 7 Eroarea de urmărire pe axa X obținută prin simulare în MobileSim la conducerea sliding-mode în timp discret a vehiculului autonom 4DW/SW SEEKUR.

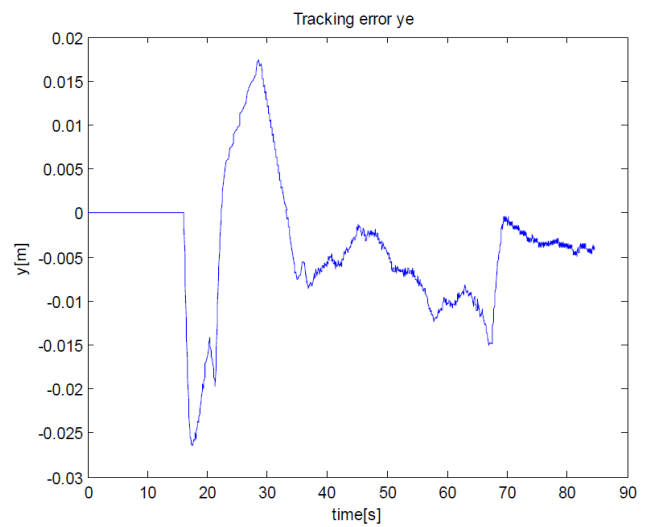


Fig. 8 Eroarea de urmărire pe axa Y obținută prin simulare în MobileSim la conducerea sliding-mode în timp discret a vehiculului autonom 4DW/SW SEEKUR.

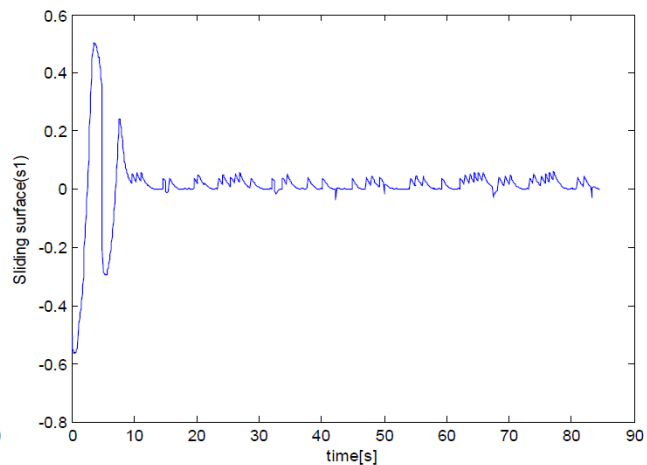
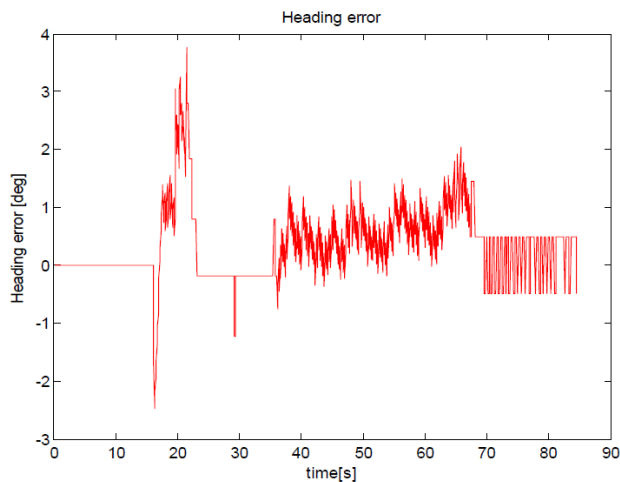


Fig. 9 Eroarea de urmărire a direcției obținută prin simulare în MobileSim la conducerea sliding-mode în timp discret a vehiculului autonom 4DW/SW SEEKUR.

Fig. 10 Suprafața de comutație s1 obținută prin simulare în MobileSim la conducerea sliding-mode în timp discret a vehiculului autonom 4DW/SW SEEKUR.

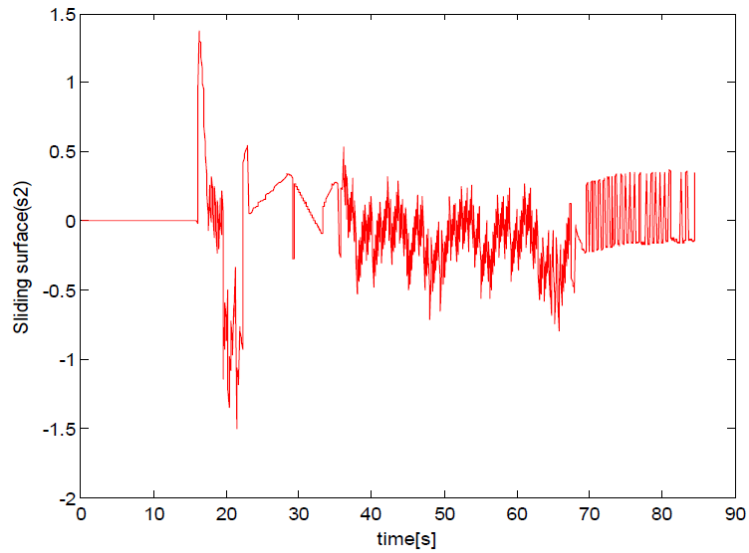


Fig. 11 Suprafața de comutație s2 obținută prin simulare în MobileSim la conducerea sliding-mode în timp discret a vehiculului autonom 4DW/SW SEEKUR.