



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI
ȘI PROTECȚIEI SOCIALE
AMPOSDRU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
CERCETĂRII
TINERETULUI
ȘI SPORTULUI

OIPOSDRU



UNIVERSITATEA "DUNĂREA DE JOS" DIN GALAȚI

ȘCOALA DOCTORALĂ A FACULTĂȚII DE AUTOMATICĂ,
CALCULATOARE, INGINERIE ELECTRICĂ ȘI ELECTRONICĂ

REZUMATUL TEZEI DE DOCTORAT

CONTRIBUTII LA CONDUCEREA, NAVIGAȚIA ȘI EVITAREA DE OBSTACOLE A ROBOȚILOR MOBILI ȘI VEHICULELOR AUTONOME

Conducător științific
Prof. Univ.Dr. Ing. Adrian Filipescu

Doctorand
Ing. Bogdan Dumitrașcu

Galați
2012



UNIUNEA EUROPEANĂ



Investește în oameni !

FONDUL SOCIAL EUROPEAN

Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007 – 2013

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

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

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

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

Rezumatul Tezei de Doctorat

CONTRIBUȚII LA CONDUCEREA, NAVIGAȚIA ȘI EVITAREA DE OBSTACOLE A ROBOȚILOR MOBILI ȘI VEHICULELOR AUTONOME

Domeniul: *Ingineria Sistemelor*

Doctorand: Ing. Bogdan DUMITRASCU

Componența Comisiei de doctorat:

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

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

REFERENT OFICIAL: Prof.univ.dr.ing. Corneliu LAZĂR
Universitatea Tehnică “Gheorghe Asachi” din Iași

REFERENT OFICIAL: Prof.univ.dr.ing. Octavian-Cezar PĂSTRĂVANU
Universitatea Tehnică “Gheorghe Asachi” din Iași

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

Cuprins

Capitolul 1. Introducere	5
1.1 Stadiul actual	5
1.2 Structura lucrării.....	7
1.3 Lista publicațiilor	8
Capitolul 2. Tipologii de senzori utilizați în conducerea roboților mobili și vehiculelor autonome.....	10
2.1. Unități de măsură inerțială	10
2.1.1. Erorile înregistrate în funcționarea IMU	10
2.3.2. Reprezentarea atitudinii	13
2.2.4. Implementarea practică a filtrului Kalman	14
2.3.5. Rezultate obținute prin implementarea algoritmului de filtrare Kalman în estimarea atitudinii	17
Capitolul 3. Conducerea trajectory-tracking a roboților mobili și vehiculelor autonome	19
3.1 Determinarea modelului cinematic al roboților mobili și vehiculelor autonome.....	19
3.1.1 Determinarea modelului cinematic al roboților mobili	19
3.1.2. Determinarea modelului cinematic al vehiculelor autonome	20
3.2. Conducerea sliding-mode cu timp continuu.....	21
3.2.1 Conducerea sliding-mode cu timp continuu a roboților mobili cu 2 roți motoare și 2 libere(WMR 2DW/2FW)	21
3.2.2 Conducerea sliding- mode a vehiculului autonom cu 4 roți motoare și 4 directoare (4DW/SW).....	23
3.2 Conducerea sliding-mode cu timp discret.....	25
3.3.1 Conducerea sliding-mode cu timp discret a WMR cu 2 DW/2FW	25
3.3.2 Conducerea sliding-mode cu timp discret a vehiculelor autonome.....	27
3.4. Conducerea backstepping.....	30
3.4.1 Conducerea backstepping aWMR cu 2DW/2FW	30

3.4.2 Conducerea backstepping a vehiculului autonom cu 4 DW/SW.....	31
3.5 Concluzii	32
Capitolul 4. Algoritmi de conducere a vehiculelor autonome și a roboților mobili, destinați evitării obstacolelor	34
4.1 Algoritmi de evitare a obstacolelor implementați în conducerea roboților mobili	34
4.2. Algoritmi de evitare a obstacolelor implementați în conducerea vehiculelor autonome	39
4.3. Concluzii	41
Capitolul 5. Implementarea conducerii în timp real.Rezultate experimentale.....	42
5.1. Simularea conducerii roboților mobili și vehiculelor autonome.....	42
5.2 Rezultatele experimentale ale conducerii în timp real ale roboților mobili și vehiculelor autonome	49
Capitolul 6. Concluzii	51
6.1. Contribuții privind conducerea și evitarea obstacolelor utilizând roboți mobili și vehicule autonome.....	51
6.1.1 Estimarea atitudinii.....	52
6.1.2 Conducerea trajectory-tracking a roboților mobili și vehiculelor autonome.....	52
6.1.3 Evitarea obstacolelor	53
6.1.4 Testarea algoritmilor propuși.....	54
6.2. Direcții de cercetare deschise.....	54
7. Bibliografie.....	55

Capitolul 1.

Introducere

Roboții mobili și vehiculele autonome au devenit subiecte de cercetare foarte importante în ultimii ani. Progresul tehnologic a permis crearea de roboți din ce în ce mai complexi și la prețuri din ce în ce mai mici. Din acest motiv, roboții mobili sunt folosiți foarte des în industrie și chiar de către utilizatorii casnici. Vehiculele autonome pot înlocui șoferii umani, astfel fiind eliminat comportamentul imprevizibil în trafic al șoferilor. În prezent cauza majoră a accidentelor rutiere este eroarea șoferilor. Vehiculele autonome prezintă avantajul că pot naviga în condiții maxime de siguranță.

Roboții mobili și vehiculele autonome au nevoie de date despre poziția și orientarea lor, viteza de deplasare și distanța până la obstacole. Pentru a obține aceste date este nevoie de senzori, cum ar fi: sonarele, senzorii inerțiali, GPS-urile, encoderele.

1.1 Stadiul actual

Un număr mare de companii au construit diverse tipuri de roboți mobili și vehicule autonome, și un număr mare de cercetători au cercetat acest domeniu. Din acest motiv analiza tuturor articolelor științifice sau roboților mobili și vehiculelor autonome nu este posibilă. În această secțiune sunt prezentate câteva dintre realizările notabile din acest domeniu.

Mai multe companii au dezvoltat diverși roboți mobili autonomi capabili să execute diverse sarcini în gospodărie. Câteva exemple de astfel de roboți sunt prezentate mai jos.

Compania iRobot comercializează gama Roomba de roboți-aspirator autonomi capabili să aspire autonom podelele, compania comercializează și gama Scooba de roboți autonomi proiectați pentru spălarea podelelor, care funcționează similar roboților Roomba.

Husqvarna comercializează gama Automower de roboți autonomi de tuns iarba.

La institutul japonez pentru cercetarea roboților, ATR, se află în curs de dezvoltare robotul Robovie-II un robot proiectat să realizeze cumpărăturile de la magazin.

Unul dintre primele vehicule autonome a fost VaMp construit de echipa lui Ernst Dickmanns de la Universitatea Bundeswehr și Mercedes-Benz în 1990. Vehiculul poate parcurge distanțe lungi în trafic intens fără intervenția umană, și să recunoască obstacole în mișcare și să le evite automat. În 2010 a fost lansat testul Vislab Intercontinental Autonomous Challenge la care au participat 4 vehicule autonome, VIAC care au străbatut

distanța de 15000 km de la Parma, Italia, până la Shanghai, China, fără intervenția operatorului uman.

Roboții mobili și vehiculele autonome sunt sisteme nonholonomice, descrise de sisteme de ecuații neliniare. Din acest motiv conducerea folosind regulatoare PID nu dă rezultate satisfăcătoare. Pentru conducerea proceselor neliniare s-au dezvoltat mai multe metode, cum ar fi: metodele backstepping, sliding-mode, fuzzy și rețelele neuronale.

Metodele de conducere a roboților mobili și vehiculelor autonome au ca scop proiectarea unor regulatoare capabile să calculeze viteza liniară și viteza unghiulară a vehiculului pentru ca acesta să urmărească o traiectorie dorită.

Unii cercetători [5], [29] și [30], au propus conducerea backstepping pentru a rezolva problema conducerii sistemelor nonholonomice. Backstepping este o procedură recursivă care descompune o problemă de proiectare a întregului sistem în probleme de proiectare pentru sisteme de ordin inferior. Proiectarea backstepping are la bază integratorul backstepping, acesta fiind aplicat recursiv pentru rezolvarea problemei sistemelor complexe, procedură descrisă de Chen [5].

Altă metodă propusă pentru conducerea sistemelor nonholonomice este conducerea sliding-mode. Conducerea sliding-mode a fost analizată de mai mulți cercetători, cum ar fi Utkin [49], [50], Gao, [22], [23], și a fost aplicată de mai mulți cercetători în [7], [8], [9], [10], [26], [28] și [37], pentru conducerea roboților mobili și vehiculelor autonome, cu rezultate foarte bune.

Navigația constă în localizarea robotului și stabilirea unei traiectorii pentru ajungerea la destinația dorită. W. Danwei și Qi Feng, [11], au propus o metodă de planificare a traiectoriilor vehiculelor cu 4 roți directoare.

Kanarat [33] propune o metodă de planificare a traiectoriei care determină un traseu optimal pentru un robot mobil într-un mediu, bazat pe incertitudinea maximă permisă, care reprezintă un număr repartizat fiecărei configurații libere din mediu.

În timpul deplasării robotului pot apare obstacole neprevăzute și este nevoie de o componentă capabilă să evite obstacolele. Lumelsky et al. , [39], a propus algoritmul „bug” pentru evitarea obstacolelor, care presupune înconjurarea obstacolului pentru determinarea punctului aflat la distanță minimă față de țintă și continuarea traseului din acest punct.

Alt algoritm de evitare a obstacolelor este algoritmul câmpului de potențial în [2] și [3]. Acest algoritm consideră că robotul este atras de forțe virtuale către țintă și respins de forțe virtuale generate de obstacole, iar traiectoria este rezultatul compunerii acestor forțe. Există situații în care soluția nu este găsită de acest algoritm. Situațiile în care nu este găsită

soluția pot fi evitate folosind câmpurile de potențial adaptive din[1], care folosesc mai multe puncte auxiliare de atracție pentru a permite robotului să evite obstacolele mari sau grupate foarte aproape unul de altul. Configurația câmpului de potențial optim este calculată de un algoritm genetic.

Algoritmul Vector Field Histogram din [65] elimină problema erorilor de măsură ale senzorilor prin utilizarea unei histograme polare care conține mai multe măsurări succesive recente.

1.2 Structura lucrării

Structura tezei este următoarea:

În capitolul 1 a fost evidențiat **Stadiul actual** al implementării structurilor robotice în diferite tipologii de vehicule autonome.

În capitolul 2, intitulat **Tipologii de senzori utilizați în conducerea roboților mobili și vehiculelor autonome**, a fost prezentată o descriere scurtă a celor mai importanți senzori folosiți de WMR și vehiculele autonome și au fost prezentate în detaliu caracteristicile senzorilor IMU. Scopul acestui capitol este prezentarea rezultatelor cercetării obținute prin utilizarea senzorilor IMU realizată cu sprijinul Institutului de Sisteme și Robotică din cadrul Universității din Coimbra. Au fost prezentate erorile care caracterizează mărimile măsurate de senzorii IMU și a fost analizat senzorul IMU Xsens Mti aplicând metoda Allan. În secțiunea următoare modul de reprezentare a atitudinii vehiculelor utilizând cuaternioni a fost analizat, apoi a fost prezentat FK, urmat de implementarea folosită pentru estimarea erorilor atitudinii furnizate de IMU. Filtrul corectează erorile furnizate de giroscopul 3D utilizând măsurătorile de la accelerometre și senzorii magnetici. În secțiunea următoare sunt prezentate atitudinile estimate.

În capitolul 3, intitulat **Conducerea trajectory-tracking a roboților mobili și vehiculelor autonome**, a fost prezentată determinarea modelului cinematic al WMR cu 2DW/2FW și modelului cinematic al vehiculului autonom electric SEEKUR(4DW/SW). Pentru simplificarea calculelor modelul vehiculului autonom SEEKUR a fost simplificat și s-a obținut un model similar modelului bicicletei. Aceste modele au fost folosite pentru proiectarea a 3 regulatoare pentru conducerea trajectory-tracking a roboților mobili 2DW/2FW și 3 regulatoare pentru conducerea trajectory-tracking a vehiculelor autonome 4DW/SW. Regulatoarele au fost calculate aplicând metodologia sliding-mode cu timp continuu, metodologia sliding-mode cu timp discret și metodologia backstepping.

În capitolul 4, intitulat **Algoritmi de conducere a vehiculelor autonome și a roboților mobili, destinați evitării obstacolelor**, a fost prezentată o metodă de evitare de obstacole pentru roboții mobili care utilizează sonarele și o metodă de evitare a obstacolelor pentru vehiculele autonome care folosește detecția laser. Robotul urmărește o traiectorie impusă și folosește una din metodele de conducere menționate în capitolul 3. La fiecare pas se verifică dacă a fost detectat un obstacol, iar în cazul detectării unui obstacol se caută o traiectorie pentru evitarea obstacolului folosind Quintic equations și se calculează vitezele și accelerațiile care descriu traiectoria de evitare a obstacolelor. Noua traiectorie este urmărită de procedura de conducere, iar când obstacolul a fost evitat robotul revine la traiectoria impusă.

În capitolul următor, Capitolul 5, intitulat **Implementarea conducerii în timp real. Rezultate experimentale**, a fost prezentată implementarea metodelor propuse și rezultate experimentale pentru validarea metodelor propuse.

Capitolul final prezintă **Concluzii**.

1.3 Lista publicațiilor

Această teză are la baza următoarele publicații:

1. [12] **Dumitrascu B.** and Filipescu A., Discrete-time sliding-mode controller for wheeled mobile robots, Proceedings of the 18th International Conference on Control Systems and Computer Science, vol. 1, Bucuresti, pag. 397-403, mai, 2011.
2. [13] **Dumitrascu B.** and Filipescu A., Sliding mode control of lateral motion for four driving-steering wheels autonomous *vehicle*, *Annals of the University of Craiova*, vol. 7 (34), pag 20-25, 2010.
3. [14] **Dumitrascu B.** and Filipescu A., Sliding mode controller for steering of four-wheels driving and steering *vehicle*, *Proceedings of the 14th International Conference on System Theory and Control*, pag. 202-206, Sinaia, 2010.
4. [15] **Dumitrascu B.**, Filipescu A., Backstepping control of wheeled mobile *robots*, *Proceedings of the 15th International Conference on System Theory and Control*, Sinaia, pag. 206-211, 2011.
5. [16] **Dumitrascu B.**, Filipescu A., Radaschin A., Filipescu A. Jr., Minca E., Discrete-time sliding mode control of wheeled mobile *robots*, *Proceedings of 2011 8th Asian Control Conference (ASCC)*, Kaohiung, Taiwan, pag. 771-776, mai, 2011.

6. [17] **Dumitrascu B.**, Filipescu A., Vasilache C., Minca E., Filipescu A. Jr., Discrete-time sliding-mode control of four driving/steering wheels mobile platform, Proceedings of the 19th Mediterranean Conference on Control and Automation, Corfu, Grecia, iunie, 2011, pag. 1076-1081.
7. [18] **Dumitrascu B.**, Filipescu A., Minzu V., Voda A., Minca E., Discrete-Time Sliding-Mode Control of Four Driving-Steering Wheels Autonomous Vehicle, *Proceedings of the 30th Chinese Control Conference*, pag. 3620-3625, 2011.
8. [20] Filipescu A., Minzu V., **Dumitrascu B.** and Filipescu A., Trajectory-tracking and discrete-time sliding-mode control of wheeled mobile robots, *The 2011 IEEE International Conference on Information and Automation*, Shenzhen, pag. 27-32, iunie, 2011.
9. [44] Solea R., Filipescu A., Filipescu S, and **Dumitrascu B.**, Sliding-mode controller for four- wheel-steering vehicle: trajectory-tracking problem, *Proceedings of the 8th World Congress on Intelligent Control and Automation*, Jinan, China, pag. 1185-1190, 2010.

Capitolul 2.

Tipologii de senzori utilizați în conducerea roboților mobili și vehiculelor autonome

Vehiculele autonome și roboții mobili sunt dotați cu mai multe tipuri de senzori pentru a fi posibilă determinarea poziției, orientării și distanței până la obstacole, aceste informații fiind vitale pentru funcționarea autonomă.

Cele mai importante tipuri de senzori sunt: encoderele, senzorii inerțiali (IMU), sonarele și laserele.

2.1. Unități de măsură inerțială

Unitățile de măsură inerțială(IMU) sunt compuse din accelerometre 3D și giroscopae 3D și uneori senzori magnetici 3D. Acești senzori pot detecta mișcarea chiar și în cazurile în care encoderele nu pot furniza corect datele.

IMU-urile sunt folosite pentru a calcula viteza, poziția și atitudinea vehiculelor, navelor și avioanelor. Pentru a se obține viteza semnalul de la accelerometru este integrat, iar pentru a afla poziția accelerația este dublu integrată. Prin integrarea datelor de la giroscop se obține unghiul de ruliu, unghiul de tangaj și orientarea. Semnalele primite de la IMU conțin erori care se acumulează în urma integrării datelor. Din acest motiv IMU reprezintă un senzor bun pentru perioade de timp scurte, dar acumulează erori din ce în ce mai mari dacă timpul de funcționare este mai lung. Aceste erori trebuie identificate și corectate pentru a obține date corecte. Corectarea datelor de la IMU se realizează utilizând filtrul Kalman.

Cercetarea senzorilor IMU Xsens Mti a fost realizată cu sprijinul Institutul de sisteme și robotică din cadrul Universității din Coimbra, unde am efectuat stagiul extern din cadrul proiectului POSDRU.

2.1.1. Erorile înregistrate în funcționarea IMU

Un IMU oferă date precise despre atitudine și poziție pentru vehicule doar în cazul perioadelor scurte de timp, în cazul intervalelor de timp lungi integrarea micilor erori ale IMU conduc la erori mari. Soluția pentru o funcționare corectă este identificarea și eliminarea erorilor care afectează senzorul. Erorile înregistrate în funcționarea IMU și modelarea stohastică a IMU au fost prezentate în [40]. O metodă pentru identificarea erorilor este metoda Allan. Această metodă a fost folosită de cercetători ca [19] și [24].

Metoda Allan este o tehnică de analiză în domeniul timp, care a fost inițial folosită pentru analiza stabilității oscilatoarelor, dar apoi a fost folosită pentru analiza senzorilor inerțiali. Această metodă presupune că incertitudinea este cauzată de diferite caracteristici ale componentelor zgomotului. Covarianța fiecărei componente este evaluată folosind datele din analiza Allan.

1. Se obține un set lung de date și se împarte în grupuri de lungimea τ . Setul de date trebuie să poată fi împărțit în minim 9 grupuri pentru a păstra semnificația;

2. Se formează o listă cu valoarea medie din fiecare grup ($a(\tau)_1, a(\tau)_2, \dots, a(\tau)_n$);

3. Se utilizează formula

$$AVAR(\tau) = \frac{1}{2(n-1)} \sum (a(\tau)_{i+1} - a(\tau)_i)^2; \quad (2.1)$$

4. Caracteristicile zgomotului se determină din graficul deviației Allan ca o funcție de τ pe o scală logaritmică. Fiecare componentă a zgomotului este reprezentată de o pantă cu un gradient diferit.

Semnalul de la giroscopae conține viteza de rotație perturbată de zgomot și deviație. Cele mai semnificative erori ale giroscopelor sunt: bias constant(eroarea sistematică constantă), angle random walk(mersul aleatoriu al unghiului), flicker noise, efectul temperaturii și eroarea de calibrare. Biasul constant produce o eroare care crește liniar cu timpul($\theta_e = \varepsilon \cdot t$).

Angle random walk este cauzat de integrarea zgomotului alb prezent în semnalul giroscopului. Pe graficul Allan este reprezentat de o pantă cu gradient -1/2. Valoarea numerică se citește din grafic la timpul $\tau = 1$.

Stabilitatea biasului unui giroscop se modifică în timp datorită zgomotului de comutație în componentele electronice. Zgomotul de comutație are frecvența scăzută și apare ca variații ale biasului în semnal. Instabilitatea este reprezentată de zona plată din zona minimului și valoarea numerică este valoarea minimă din grafic.

Rate random walk este un proces cu origini incerte și e reprezentat cu o pantă de 1/2, iar magnitudinea poate fi citită la $\tau = 3$.

Coeficientul de pantă este o eroare deterministică care apare în timpul intervalelor de timp mari și e definit[27]

$$\Omega = R \cdot t, \quad (2.2)$$

unde R este coeficientul rampei. Rampa este reprezentată de o pantă cu gradientul 1 și amplitudinea poate fi calculată la $\tau = \sqrt{2}$.

Zgomotul de cuantizare este cauzat de natura digitală a circuitului și e reprezentat cu panta -1 și magnitudinea poate fi citită la $\tau = \sqrt{3}$.

Graficul Allan pentru cele 3 giroscopae ale Xsens Mti sunt prezentate în Fig. 2.1.

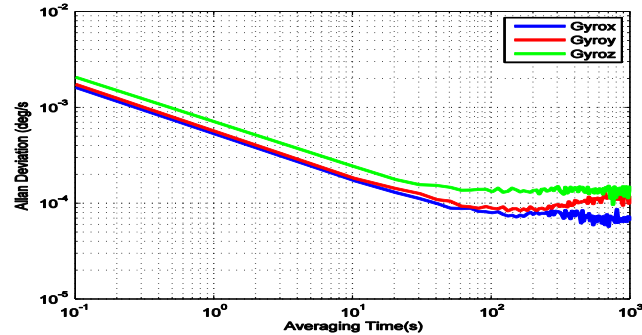


Fig. 2.1 Deviația Allan a giroscopaelor

Biasul accelerometrului este valoarea medie a accelerației (m/s^2) când accelerometrul nu este supus accelerației. Valoarea medie a unui set lung de date dacă nu e afectată de gravitație este biasul. Poziția este rezultatul dublei integrări a valorilor de la accelerometru și se obține o eroare care evoluează astfel

$$p(t) = \varepsilon \cdot \frac{t^2}{2}. \quad (2.3)$$

Velocity random walk se măsoară în $m/s/\sqrt{h}$ și e cauzat de integrarea zgomotului alb. Zgomotul alb generează random walk de ordinul doi în poziție, cu medie zero și deviație standard

$$\sigma_p(t) = \sigma \cdot t^{3/2} \cdot \sqrt{\frac{\delta t}{3}}. \quad (2.4)$$

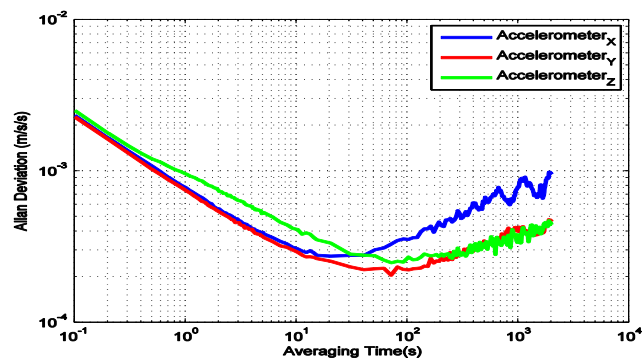


Fig. 2.2 Deviația Allan a accelerometrelor

Stabilitatea biasului accelerometrului determină variația biasului în timp.

Deviația Allan pentru accelerometre este prezentată în Fig. 2.2. Zgomotul alb apare ca o pantă cu valoarea -0.5. Valoarea pentru random walk este citită la $\tau = 1$. Instabilitatea

biasului apare ca regiunea plată din zona minimumului și valoarea este valoarea minima a curbei.

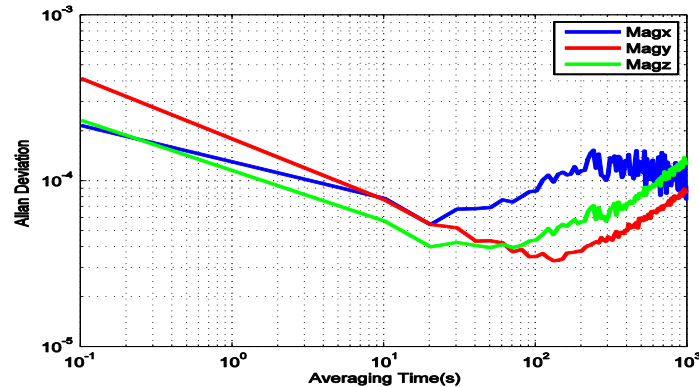


Fig. 2.3 Deviația Allan pentru senzorii magnetici

Deviația Allan pentru senzorii magnetici este prezentată în Fig. 2.3.

2.3.2. Reprezentarea atitudinii

Atitudinea este formată din unghiul de ruluu, unghiul de tangaj și orientarea. În cazul roboților mobili și vehiculelor autonome cel mai important unghi este unghiul care reprezintă orientarea, dar pentru o estimare mai bună a orientării trebuie folosite cele 3 unghiuri.

Atitudinea poate fi exprimată folosind unghiurile Euler, dar unghiurile Euler conțin o singularitate. Această problemă a dus la stabilirea altor reprezentări: direct cosine formulation, Euler-axis formulation, Euler-Rodrigues quaternion formulation. Aceste reprezentări sunt prezentate în [41].

Algebra cuaternionilor elimină singularitatea și are nevoie de mai puțin timp de calcul, fiind o foarte bună metodă de reprezentare a atitudinii. Proprietățile cuaternionilor sunt descrise de [25].

Un cuaternion e reprezentat de un vector cu 4 elemente

$$q = [q_0 \quad q_1 \quad q_2 \quad q_3]^T = [q_0 \quad \vec{q}]^T, \quad (2.5)$$

unde q_0 este partea scalară a cuaternionului și q este partea vectorială, q_0, q_1, q_2, q_3 sunt numere reale.

Cuaternionul descrie o atitudine cu 3 grade de libertate. Relația dintre componente este

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = \cos^2 \frac{\theta}{2} + (r_x^2 + r_y^2 + r_z^2) \cdot \sin^2 \frac{\theta}{2} = 1. \quad (2.6)$$

2.2.4. Implementarea practică a filtrului Kalman

Atitudinea vehiculului este estimată folosind accelerometre 3D, giroscopae 3D și compasuri magnetice 3D conținute în pachetul Xsens Mti. Datele de la senzori sunt filtrate folosind filtrul Kalman. FK este folosit pentru estimarea atitudinii în [6], [31] și [32]. În [35] este prezentată detectarea inclinației senzorului IMU utilizând accelerometrele din senzorul IMU, și se constată posibilitatea corectării orientării cu ajutorul accelerațiilor înregistrate de accelerometre.

Pentru determinarea atitudinii se definește o stare nominală și o stare eroare. Schema bloc a algoritmului de estimare este prezentată în Fig. 2.4.

Folosim Filtrul Kalman indirect pentru a minimiza starea eroare. Primul pas este precizarea stării nominale a cuaternionului integrând valorile primite de la giroscopae.

Semnalele giroscopaelor sunt perturbate de zgomot și valoarea prezisă trebuie corectată utilizând

$$q = \tilde{q}_e \otimes \hat{q}, \quad (2.7)$$

unde \tilde{q}_e este eroarea estimată dintre valoarea reală și cea prezisă.

Formula pentru integrare este [31]

$$\dot{\hat{q}} = \frac{1}{2} \cdot \Omega(w) \cdot \hat{q}, \quad (2.8)$$

unde

$$\Omega(w) = \begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & w_z & -w_y \\ w_y & -w_z & 0 & w_x \\ w_z & w_y & -w_x & 0 \end{bmatrix} \quad (2.9)$$

Pentru că \tilde{q}_e depinde doar de zgomotul și biasul semnalului și are valori foarte mici, care permit aproximarea

$$\tilde{q}_e = \begin{bmatrix} 1 \\ q_e \end{bmatrix}. \quad (2.10)$$

Filtrul Kalman estimează eroarea și corectează cuaternionul q folosind ecuația (2.9).

Eroare este aproximată ca [31]

$$\hat{q}_e = -y_g \times q_e - \frac{1}{2} \cdot (b_g + v_g), \quad (2.11)$$

unde y_g este rata unghiulară măsurată, b_g este biasul giroscopelor, v_g este zgomotul.

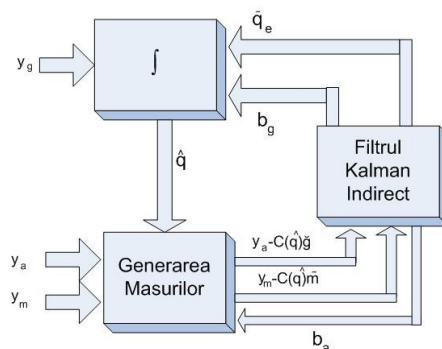


Fig. 2.4 Schema bloc a algoritmului de filtrare

Ieșirile accelerometrelor, giroscopelor și senzorilor magnetic au forma

$$\begin{cases} y_a = C(q)\tilde{g} + b_a + v_a + a_b \\ y_g = \omega + b_g + v_g \\ y_m = C(q)\tilde{m} + v_m \end{cases} \quad (2.12)$$

$$\tilde{g} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \quad \tilde{m} = \begin{bmatrix} \cos i \cdot \cos d \\ \cos i \cdot \sin d \\ -\sin i \end{bmatrix},$$

unde g este accelerația gravitațională și i este înclinația vectorului câmpului magnetic terestru, iar d declinația, i și d sunt valori specifice zonei în care se efectuează testul.

Definim starea ca

$$x = \begin{bmatrix} q_e \\ b_g \\ b_a \end{bmatrix}. \quad (2.13)$$

Ecuția procesului este:

$$x_{k+1} = A_k x_k + \begin{bmatrix} -0.5v_g \\ v_{bg} \\ v_{ba} \end{bmatrix}, \quad (2.14)$$

$$A_k = I + A \cdot T + \frac{1}{2} A^2 \cdot T^2, \quad (2.15)$$

$$A = \begin{bmatrix} -[y_g \times] & -0.5I & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (2.16)$$

Ecuțiile măsurilor sunt:

$$y_a - C(\hat{q})\tilde{g} = 2[C(\hat{q})\tilde{g} \times]q_e + a_b + v_a + b_a; \quad (2.17)$$

$$y_m - C(\hat{q})\tilde{m} = 2[C(\hat{q})\tilde{m} \times]q_e + v_m. \quad (2.18)$$

Predicția stării se face utilizând ecuația (2.14):

$$\hat{x}_{k+1}^- = A_k \hat{x}_k; \quad (2.19)$$

$$P_{k+1}^- = A_k P_k A_k^T + Q. \quad (2.20)$$

Corecția se face în 2 etape pentru a minimaliza erorile magnetice mari care pot fi generate când senzorul este în apropierea unor obiecte metalice. Prima etapă asigură corecția utilizând informațiile de la accelerometre și în etapa a doua se aplică corecția magnetică.

- Etapa 1

Măsurătoarea $y_a - C(\hat{q})\tilde{g}$ este folosită pentru corecția estimării.

$$K_{a,k} = P_k^- \cdot H_{a,k}^T (H_{a,k} P_k^- H_{a,k}^T + R_a + \hat{Q}_{a,k})^{-1}; \quad (2.21)$$

$$\hat{x}_{a,k} = \hat{x}_k^- + K_{a,k} (z_{a,k} - H_{a,k} \hat{x}_k^-); \quad (2.22)$$

$$P_{a,k} = (I - K_{a,k} H_{a,k}) P_k^- (I - K_{a,k} H_{a,k})^T + K_{a,k} (R_a + \hat{Q}_{a,k}) K_{a,k}^T; \quad (2.23)$$

$$P_k^+ = \frac{1}{2} (P_{a,k} + P_{a,k}^T); \quad (2.24)$$

$$H_{a,k} = [2[C(\hat{q})\tilde{g} \times] \quad 0 \quad I]; \quad (2.25)$$

$\hat{Q}_{a,k}$ este definită conform:

$$\hat{Q}_{a,k} = U_k - (H_{a,k} P_k^- H_{a,k}^T + R_a), \quad (2.26)$$

U_k este simetrică și există vectori eigen $u_{i,k} \in R^{3 \times 1}$ și valori corespunzătoare eigen $\lambda_{i,k} \in R$ astfel încât

$$U_k = \sum_{i=1}^3 \lambda_{i,k} \cdot u_{i,k} \cdot u_{i,k}^T \quad (2.27)$$

În cazul în care nu există accelerație externă ($\max(\lambda_{i,k} - u_{i,k}) < \zeta$), $\hat{Q}_{a,k}$ este zero. Dacă există accelerații externe atunci

$$\hat{Q}_{a,k} = \sum_{i=1}^3 \max(\lambda_{i,k} - u_{i,k}, 0) u_{i,k} \cdot u_{i,k}^T. \quad (2.28)$$

După calculul estimării corectăm atitudinea

$$q_e = \hat{x}_{a,k}(1:3); \quad (2.29)$$

$$\hat{q} = \hat{q} \otimes \bar{q}_e; \quad (2.30)$$

$$\hat{q} = \hat{q} / \|\hat{q}\|; \quad (2.31)$$

$$\hat{x}_{a,k}(1:3) = 0. \quad (2.32)$$

- Etapa 2. Corecția magnetică:

$$P_{m,k}^- = P_k^+; \quad (2.33)$$

$$K_{m,k} = P_{m,k}^- H_{m,k}^T (H_{m,k} P_{m,k}^- H_{m,k}^T + R_m)^{-1}; \quad (2.34)$$

$$\hat{x}_k = \hat{x}_{a,k} + K_{m,k} (z_{a,k} - H_{m,k} \hat{x}_{a,k}); \quad (2.35)$$

$$H_{m,k} = [2[C(\hat{q})\tilde{m} \times] \quad 0 \quad 0]; \quad (2.36)$$

$$z_{m,k} = y_{m,k} - C(\hat{q}_k)\tilde{m}; \quad (2.37)$$

$$q_e = \hat{x}_k(1:3); \quad (2.38)$$

Se corectează atitudinea folosind (2.30),(2.31).

2.3.5. Rezultate obținute prin implementarea algoritmului de filtrare Kalman în estimarea atitudinii

Datele au fost colectate în urma rulării vehiculului Yamaha în parcarea laboratorului. Vehiculul Yamaha este dotat cu un calculator care folosește sistemul de operare ROS pentru a colecta datele de la toți senzorii echipați și pentru a funcționa în regim autonom. Datele colectate de la vehicul au fost prelucrate în MATLAB utilizând algoritmul prezentat în secțiunea anterioară. Rezultatele unuia din testele efectuate pentru validarea algoritmului de estimare a atitudinii sunt prezentate în figurile următoare. În grafice atitudinea exprimată în quaternioni este convertită în atitudinea exprimată în grade pentru o mai bună înțelegere a rezultatelor. Conversia de la cuaternioni la unghiuri Euler furnizează rezultate aflate în intervalul în intervalul $[-180^\circ, 180^\circ]$.

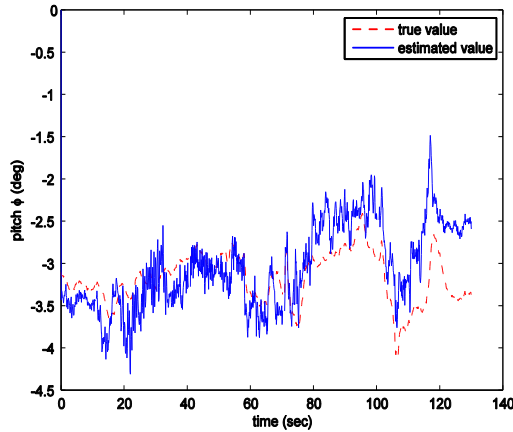


Fig. 2.5 Variația în timp a unghiului de tangaj

Fig. 2.5 prezintă variația în timp a unghiului de tangaj estimat de senzorul Xsens Mti(roșu) și variația în timp a unghiului de tangaj estimat utilizând filtrul propus(albastru). Fig. 2.6 prezintă variația în timp a unghiului de ruluu estimat de senzorul Xsens (roșu) și variația în timp a unghiului de tangaj estimat utilizând filtrul propus(albastru) în partea stângă și unghiul direcției în partea dreaptă.

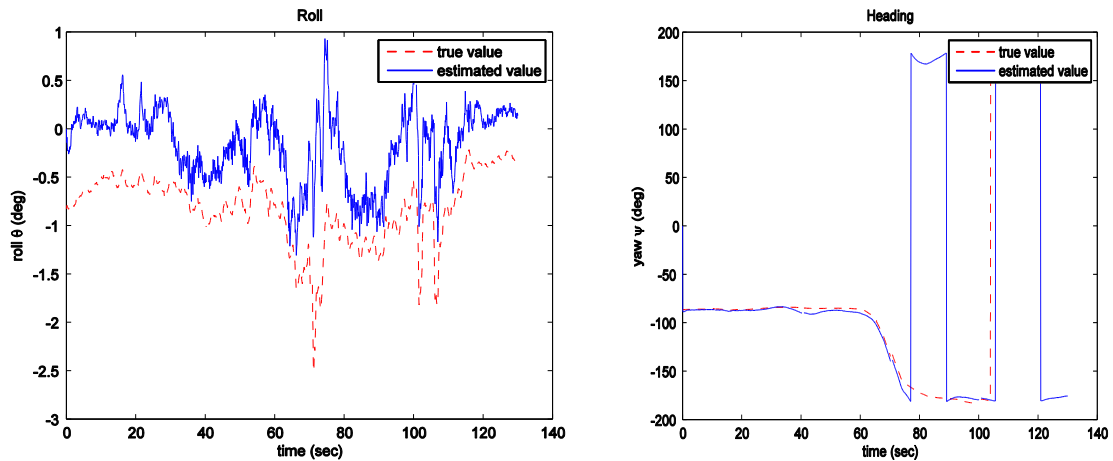


Fig. 2.6 Variația în timp a unghiului de ruluu(stânga) și Variația în timp a unghiului de direcție(dreapta)

Rezultatele filtrului utilizat au furnizat grafice similare celor furnizate de Xsens, se observă că Xsens folosește valori diferite pentru condițiile inițiale, aceste valori afectând diferențele dintre grafice. Pentru că unghiul de direcție are valori mai mici de -180° se observă o trecere bruscă spre 180° , cauzată de modul de reprezentare a cuaternionilor. Din cele 3 grafice se observă că filtrul propus realizează o bună estimare a atitudinii.

Capitolul 3.

Conducerea trajectory-tracking a roboților mobili și vehiculelor autonome

3.1 Determinarea modelului cinematic al roboților mobili și vehiculelor autonome

Există două modele folosite pentru conducerea WMR și vehiculelor: modelul cinematic și modelul dinamic. Modelul cinematic descrie mișcarea fără a considera forțele implicate. Modelul dinamic descrie mișcarea considerând și forțele care acționează asupra roboților.

În continuare vor fi prezentate modelele cinematice ale WMR și vehiculului SEEKUR, care vor fi folosite ulterior la elaborarea conducerii.

3.1.1 Determinarea modelului cinematic al roboților mobili

Se consideră un WMR cu 2 roți motoare și două directoare(2DW/2FW) prezentat în Fig. 3.1, 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

$$A(q) \cdot \dot{q} = 0, \quad (3.1),$$

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

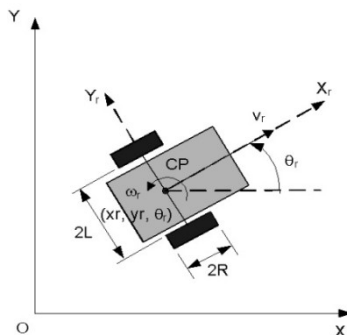


Fig. 3.1 Modelul cinematic al WMR cu 2DW/2FW

Considerăm robotul mobil din Fig. 3.1. 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. Modelul cinematic rezultat este

$$\begin{cases} \dot{x}_r = v_r \cdot \cos \theta_r \\ \dot{y}_r = v_r \cdot \sin \theta_r \\ \dot{\theta}_r = \omega_r \end{cases}, \quad (3.2)$$

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.

3.1.2. Determinarea modelului cinematic al vehiculelor autonome

Modelul cinematic al vehiculului autonom electric SEEKUR Robot Base descrie mișcarea vehiculului fără a lua în considerare masa sau forțele care acționează asupra vehiculului. SEEKUR este un AV care poate fi modelat considerând un WMR cu 4DW/SW. Modelul cinematic determinat va fi folosit la proiectarea conducerii backstepping și sliding-mode. Modelul cinematic al AV 4DW/SW poate fi aproximat de modelul bicicletei (Fig. 3.2).

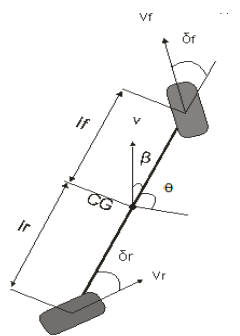


Fig. 3.2 Modelul cinematic al bicicletei

Presupunem că unghiul roților vehiculului este limitat de formula

$$-\frac{\pi}{2} \leq \delta_f, \delta_r \leq \frac{\pi}{2}. \quad (3.3)$$

Presupunem de asemenea ca vehiculul se deplasează utilizând manevra cu alunecare laterală zero (Zero-side-slip maneuver), propusă de Danwei și Feng [11]. În acest caz unghiul de alunecare este 0 pe întreaga durată a deplasării, iar orientarea vehiculului θ se potrivește cu unghiul tangent la traiectoria dorită.

$$\theta(t) = \theta_d(t), t: 0 \rightarrow t_f \quad (3.4)$$

Unghiurile roților, în acest caz, sunt $\delta_f = -\delta_r$, situație prezentată în Fig. 3.2.

Modelul cinematic al vehiculului în cazul manevrei cu alunecare laterală zero devine

$$\begin{cases} \dot{x}_{CG} = v \cdot \cos \theta \\ \dot{y}_{CG} = v \cdot \sin \theta \\ \dot{\theta} = v \cdot \tan \delta_f / L \end{cases}, \quad (3.5)$$

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.

3.2. Conducerea sliding-mode cu timp continuu

Chiar dacă conducerea sliding-mode a fost inventată în anii 60, folosirea ei în exteriorul Rusiei a început abia după anii 70. A evoluat într-o metodă de elaborare a comenzilor care poate fi aplicată unui număr foarte mare de sisteme și în special pentru sistemele neliniare.

O lege de conducere este definită pentru a conduce traiectoria de stare a sistemului într-o stare prescrisă a suprafeței în spațiul stărilor și să mențină traiectoria de stare a sistemului pe această traiectorie. Această suprafață se numește suprafața de comutație.

Conducerea sliding-mode a fost analizată de cercetători ca Utkin [49], [50], Gao, [22]. Mai mulți cercetători în [42], [43], [44], [45], [46] și [47], au ales conducerea sliding-mode pentru conducerea roboților mobili și vehiculelor autonome, caracterizate de sisteme nonholonomice. În [36] a fost tratată problema reducerii efectului de chattering în conducerea sliding-mode.

3.2.1 Conducerea sliding-mode cu timp continuu a roboților mobili cu 2 roți motoare și 2 libere (WMR 2DW/2FW)

Arhitectura conducerii sliding-mode cu timp continuu a WMR cu 2 DW/2FW este prezentată în Fig. 3.3. Această arhitectură de conducere permite robotului să urmărească o traiectorie dorită cu un profil de viteză impus. Trajectory-tracking consideră că robotul urmărește un WMR virtual care se deplasează pe traiectoria dorită cu profilul de viteză impus. Considerând că traiectoria dorită $q_d(t) = [x_d \ y_d \ \theta_d]^T$ este generată de un WMR virtual (Fig. 3.4).

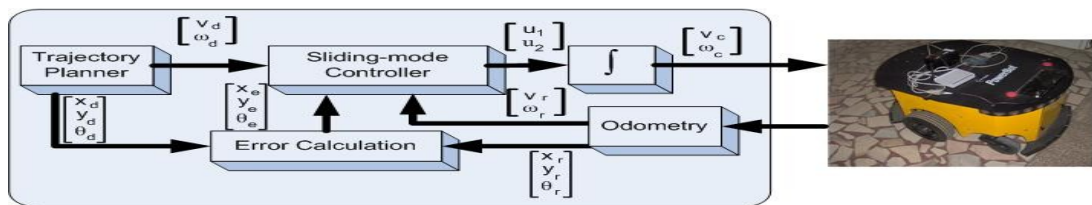


Fig. 3.3 Arhitectura conducerii sliding-mode a WMR 2DW/2FW

Modelul cinematic al robotului virtual este:

$$\begin{cases} \dot{x}_d = v_d \cdot \cos \theta_d \\ \dot{y}_d = v_d \cdot \sin \theta_d \\ \dot{\theta}_d = \omega_d \end{cases} \quad (3.6)$$

unde (x_d, y_d) reprezintă coordonatele Carteziene ale centrului geometric, v_d este viteza liniară, θ_d reprezintă orientarea și ω_d reprezintă viteza unghiulară.

Când robotul este condus pentru a urmări o anumită traiectorie se obțin erori de urmărire(Fig. 3.4). Erorile de urmărire[34] sunt:

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos \theta_d & \sin \theta_d & 0 \\ -\sin \theta_d & \cos \theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x_d \\ y_r - y_d \\ \theta_r - \theta_d \end{bmatrix} \quad (3.7)$$

Dinamica erorilor de urmărire este

$$\begin{cases} \dot{x}_e = -v_d + v_r \cdot \cos \theta_e + \omega_d \cdot y_e \\ \dot{y}_e = v_r \cdot \sin \theta_e - \omega_d \cdot x_e \\ \dot{\theta}_e = \omega_r - \omega_d \end{cases} \quad (3.8)$$

Se presupune că $|\theta_e| < \pi/2$, astfel orientarea robotului nu este perpendiculară pe traiectoria dorită.

Considerând erorile de urmărire (3.7) și derivata lor (3.8) se definesc suprafețele de comutație

$$\begin{cases} s_1 = \dot{x}_e + k_1 \cdot x_e \\ s_2 = \dot{y}_e + k_2 \cdot y_e + k_0 \cdot \text{sgn}(y_e) \cdot \theta_e \end{cases} \quad (3.9)$$

unde $k_1, k_2, k_3 \geq 0$ sunt parametri constanți pozitivi, x_e, y_e, θ_e sunt erorile de urmărire.

Dacă s_1 converge la zero, atunci x_e converge la zero. Dacă s_2 converge la 0, atunci $\dot{y}_e = -k_2 \cdot y_e + k_0 \cdot \text{sgn}(y_e) \cdot \theta_e$. Dacă $y_e > 0$ atunci $\dot{y}_e < 0$ doar dacă $k_0 < k_2 \cdot |y_e|/|\theta_e|$. Se observă că dacă y_e și \dot{y}_e converg la zero atunci θ_e converge la zero.

Derivatele suprafețelor devin

$$\begin{cases} \dot{s}_1 = \ddot{x}_e + k_1 \cdot \dot{x}_e \\ \dot{s}_2 = \ddot{y}_e + k_2 \cdot \dot{y}_e + k_0 \cdot \text{sgn}(y_e) \cdot \dot{\theta}_e \end{cases} \quad (3.10)$$

Legea de conducere este folosită în forma

$$\dot{s} = -Q \cdot \text{sgn}(s) - P \cdot s, \quad (3.11)$$

unde
$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}, \quad P = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix}, \quad Q_1, Q_2, P_1, P_2 \geq 0,$$

$$\text{sgn}(s) = [\text{sgn}(s_1) \quad \text{sgn}(s_2)]^T, \quad s = [s_1 \quad s_2]^T$$

Din (3.7), (3.8), (3.9), (3.10) și (3.11) se obține legea de comandă sliding mode:

$$\dot{v}_c = \frac{-Q_1 \cdot \text{sign}(s_1) - P_1 \cdot s_1 - k_1 \cdot \dot{x}_e - \dot{\omega}_d \cdot y_e - \omega_d \cdot \dot{y}_e + v_r \cdot \dot{\theta}_e \cdot \sin \theta_e + \dot{v}_d}{\cos \theta_e}; \quad (3.12)$$

$$\omega_c = \frac{-Q_2 \cdot \text{sign}(s_2) - P_2 \cdot s_2 - k_2 \cdot \dot{y}_e - \dot{v}_r \cdot \sin \phi_e + \dot{\omega}_d \cdot x_e + \omega_d \cdot \dot{x}_e}{v_r \cdot \cos \phi_e + k_0 \cdot \text{sgn}(y_e)} + \omega_d \quad (3.13)$$

Fie $V = \frac{1}{2} \cdot s^T \cdot s$ o funcție Lyapunov. Derivata acestei funcții este:

$$\dot{V} = s_1 \cdot \dot{s}_1 + s_2 \cdot \dot{s}_2 = s_1 \cdot (-Q_1 \cdot \text{sgn}(s_1) - P_1 \cdot s_1) + s_2 \cdot (-Q_2 \cdot \text{sgn}(s_2) - P_2 \cdot s_2) \quad (3.14)$$

Derivata funcției Lyapunov este negativă semi-definită dacă alegem parametrii $Q_i, P_i \geq 0$.

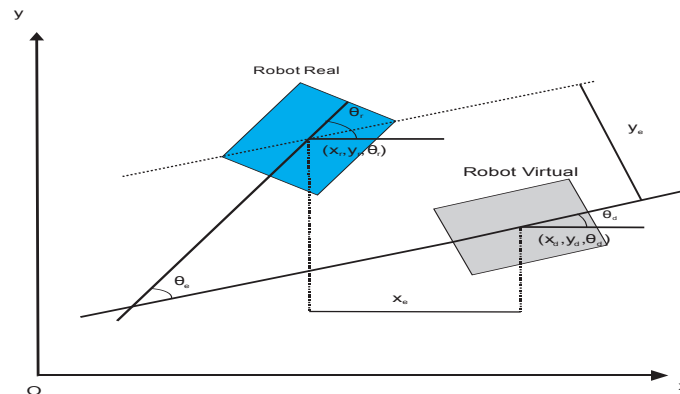


Fig. 3.4 Erorile de urmărire în cazul roboților mobili

3.2.2 Conducerea sliding-mode a vehiculului autonom cu 4 roți motoare și 4 directoare (4DW/SW)

Calculul comenzii se realizează pornind de la modelul cinematic simplificat al vehiculului din ecuația (3.5) și considerând un vehicul virtual care se deplasează pe traiectoria dorită. Problema urmăririi traiectoriei presupune proiectarea unui regulator care să calculeze viteza liniară și viteza unghiulară care să permită vehiculului real să urmărească traiectoria vehiculului virtual cu erori de poziționare cât mai mici.

Arihitectura conducerii sliding-mode cu timp continuu a AV 4DW/SW este prezentată în Fig. 3.5.

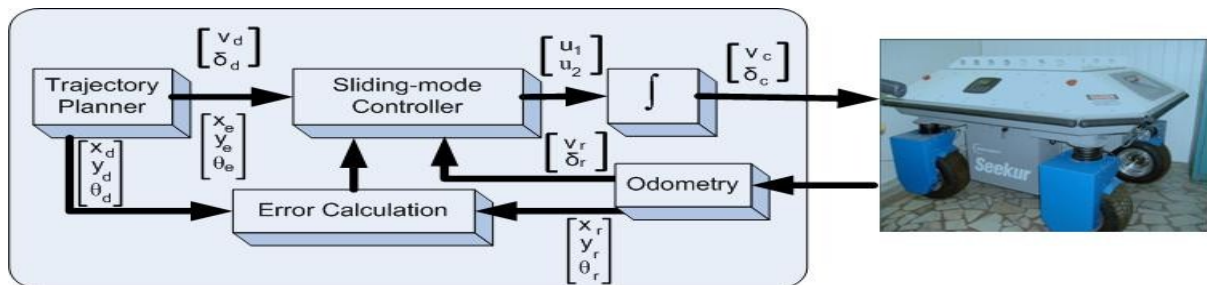


Fig. 3.5 Arihitectura conducerii sliding-mode cu timp continuu a AV 4DW/SW

Modelul cinematic al vehiculului virtual este

$$\begin{cases} \dot{x}_d = v_d \cdot \cos \theta_d \\ \dot{y}_d = v_d \cdot \sin \theta_d \\ \dot{\theta}_d = \frac{v_d \cdot \tan \delta_d}{L} \end{cases}, \quad (3.15)$$

unde (x_d, y_d) reprezintă coordonatele Carteziene ale centrului geometric, v_d este viteza liniară, θ_d reprezintă orientarea și ω_d reprezintă viteza unghiulară.

Erorile de urmărire[34] sunt

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos \theta_d & \sin \theta_d & 0 \\ -\sin \theta_d & \cos \theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x_d \\ y_r - y_d \\ \theta_r - \theta_d \end{bmatrix}. \quad (3.16)$$

Dinamica erorilor de urmărire este

$$\begin{cases} \dot{x}_e = -v_d + v_r \cdot \cos \theta_e + \frac{v_d}{L} \cdot \tan \delta_d \cdot y_e \\ \dot{y}_e = v_r \cdot \sin \theta_e - \frac{v_d \cdot \tan \delta_d}{L} \cdot x_e \\ \dot{\theta}_e = \frac{v_r \cdot \tan \delta_r}{L} - \frac{v_d \cdot \tan \delta_d}{L} \end{cases}. \quad (3.17)$$

Erorile de urmărire ale vehiculului autonom electric SEEKUR, considerând modelul cinematic simplificat, reprezentat de modelul cinematic al bicicletei sunt descrise în Fig. 3.6.

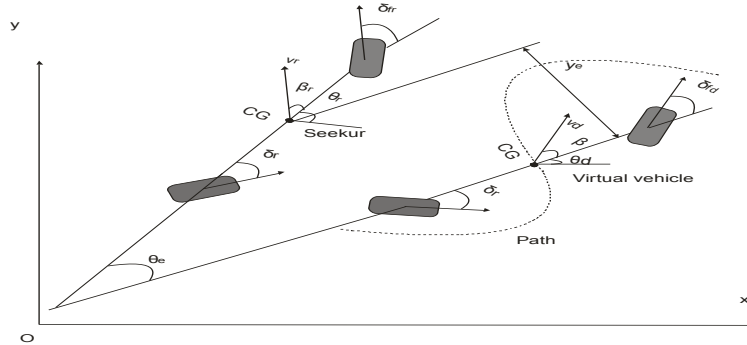


Fig. 3.6 Erorile de urmărire în cazul vehiculului SEEKUR

Considerând erorile de urmărire (3.16) și derivata lor (3.17) se definesc suprafețele de comutație:

$$s_1 = \dot{x}_e + k_1 \cdot x_e; \quad (3.18)$$

$$s_2 = \dot{y}_e + k_2 \cdot y_e + k_0 \cdot \text{sgn}(y_e) \cdot \theta_e. \quad (3.19)$$

Legea de conducere este folosită în forma:

$$\dot{s} = -Q \cdot \text{sgn}(s) - P \cdot s. \quad (3.20)$$

Din (3.16), (3.17), (3.18), (3.19) și (3.20) se obține legea de comandă sliding mode:

$$\dot{v}_c = \frac{-Q_1 \cdot \text{sign}(s_1) - P_1 \cdot s_1 - k_1 \cdot \dot{x}_e - \dot{\omega}_d \cdot y_e - \omega_d \cdot \dot{y}_e + v_r \cdot \dot{\theta}_e \cdot \sin \theta_e + \dot{v}_d}{\cos \phi_e}; \quad (3.21)$$

$$\delta_c = \arctan\left(\frac{L \cdot (-Q_2 \cdot \text{sign}(s_2) - P_2 \cdot s_2 - k_2 \cdot \dot{y}_e - \dot{v}_r \cdot \sin \theta_e + \dot{\omega}_d \cdot x_e + \omega_d \cdot \dot{x}_e)}{v_r \cdot \cos \theta_e + k_0 \cdot \text{sgn}(y_e)} + \frac{L}{v_r} \omega_d\right). \quad (3.22)$$

Fie $V = \frac{1}{2} \cdot s^T \cdot s$ o funcție Lyapunov. Derivata acestei funcții este

$$\dot{V} = s_1 \cdot \dot{s}_1 + s_2 \cdot \dot{s}_2 = s_1 \cdot (-Q_1 \cdot \text{sgn}(s_1) - P_1 \cdot s_1) + s_2 \cdot (-Q_2 \cdot \text{sgn}(s_2) - P_2 \cdot s_2). \quad (3.23)$$

Derivata funcției Lyapunov este negativă semi-definită dacă alegem parametrii $Q_i, P_i \geq 0$.

3.2 Conducerea sliding-mode cu timp discret

Conducerea sliding-mode a fost aplicată cu succes la rezolvarea problemei conducerii sistemelor nonholonomice. Conducerea sliding-mode cu timp discret a fost aplicată mai rar în cazul sistemelor nonholonomice decât conducerea sliding-mode pentru timp continuu și pentru că algoritmul de conducere este implementat pe sisteme digitale se va folosi conducerea sliding-mode pentru timp discret. Conducerea sliding-mode cu timp discret are structura din Fig. 3.7. Conducerea sliding-mode cu timp discret a fost propusă de [12], [16], [17] și [18].

3.3.1 Conducerea sliding-mode cu timp discret a WMR cu 2 DW/2FW

Regulatorul primește de la modulul de planificare a traiectoriei viteza liniară și viteza unghiulară dorită, și erorile de poziționare de la modulul de calcul al erorilor și calculează comanda pentru viteza liniară și viteza unghiulară.

Conducerea sliding-mode cu timp discret a roboților mobili cu 2DW/2FW este realizată pornind de la ecuațiile cinematice ale centrului de greutate.

Considerând perioada de eșantionare T_s și metoda zero order hold sistemul neliniar (3.2) poate fi scris în timp discret astfel

$$\begin{cases} x_r[k+1] = x_r[k] + v_r[k] \cdot \cos \theta_r[k] \cdot T_s \\ y_r[k+1] = y_r[k] + v_r[k] \cdot \sin \theta_r[k] \cdot T_s \\ \theta_r[k+1] = \theta_r[k] + \omega_r[k] \cdot T_s \end{cases} \quad (3.24)$$

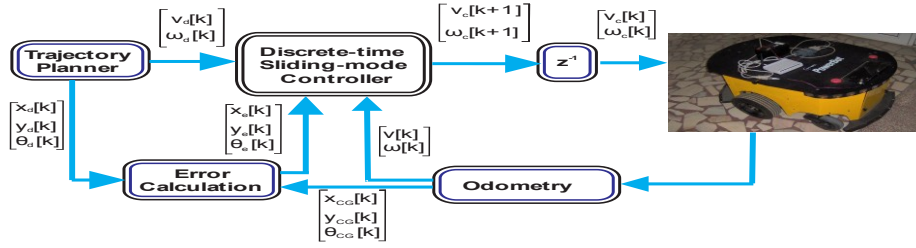


Fig. 3.7 Arhitectura de conducere a WMR cu 2DW/2FW

Conducerea își propune rezolvarea problemei urmării traiectoriei. Rezolvarea acestei probleme necesită proiectarea unui regulator 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 următorul model cinematic în timp discret

$$\begin{cases} x_d[k+1] = x_d[k] + v_d[k] \cdot \cos \theta_d[k] \cdot T_s \\ y_d[k+1] = y_d[k] + v_d[k] \cdot \sin \theta_d[k] \cdot T_s \\ \theta_d[k+1] = \theta_d[k] + \omega_d[k] \cdot T_s \end{cases} \quad (3.25)$$

în care: x_d - reprezintă poziția dorită pe axa Ox, y_d - reprezintă poziția dorită pe axa Oy, θ_d - reprezintă orientarea dorită, v_d - reprezintă viteza liniară dorită, ω_d - reprezintă viteza unghiulară dorită.

Erorile de urmărire din Fig. 3.4 sunt

$$\begin{cases} x_e[k] = x_r[k] - x_d[k] \cdot \cos \theta_e[k] + y_r[k] - y_d[k] \cdot \sin \theta_e[k] \\ y_e[k] = -x_r[k] - x_d[k] \cdot \sin \theta_e[k] + y_r[k] - y_d[k] \cdot \cos \theta_e[k] \\ \theta_e[k] = \theta_r[k] - \theta_d[k] \end{cases} \quad (3.26)$$

În timp timp discret dinamica erorilor se exprimă

$$\begin{cases} x_e[k+1] = x_e[k] + (v_r[k] \cdot \cos \theta_e[k] - v_d[k] + y_e[k] \cdot \omega_d[k]) \cdot T_s \\ y_e[k+1] = y_e[k] + (v_r[k] \cdot \sin \theta_e[k] - x_e[k] \cdot \omega_d[k]) \cdot T_s \\ \theta_e[k+1] = \theta_e[k] + (\omega_r[k] - \omega_d[k]) \cdot T_s \end{cases} \quad (3.27)$$

Regulatorul sliding-mode cu timp discret este un regulator 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 cu timp discret este controlul discontinuu.

Gao et al. [23] a propus următoarea condiție pentru existență sliding-mode:

$$s[k] \cdot (s[k+1] - s[k]) < 0 \quad (3.28)$$

Legea de conducere propusă de Gao et al. [23] este:

$$s[k+1] = (1 - q \cdot T_s) \cdot s[k] - \varepsilon \cdot T_s \cdot \text{sgn}(s[k]); \quad (3.29)$$

$$0 < 1 - q \cdot T < 1; \quad (3.30)$$

$$0 < \varepsilon \cdot T_s < 1; \quad (3.31)$$

în care: $T_s > 0$ este perioada de eşantionare, $\varepsilon > 0$ este viteza de aducere, $q > 0$ este convergența exponențială,

Se definesc suprafețele de comutație

$$\begin{cases} s_1[k] = x_e[k+1] + k_1 \cdot x_e[k] \\ s_2[k] = y_e[k+1] + k_2 \cdot y_e[k] + k_0 \cdot \text{sign}(y_e[k]) \cdot \theta_e[k] \end{cases} \quad (3.32)$$

în care: k_0, k_1, k_2 sunt constante pozitive x_e, y_e și θ_e sunt erorile de urmărire.

Din (3.29) și (3.32) obținem

$$s_1[k+1] = x_e[k+2] + k_1 \cdot x_e[k+1] = (1 - q \cdot T_s) \cdot s_1[k] - \varepsilon \cdot T_s \cdot \text{sgn}(s_1[k]) \quad (3.33)$$

$$\begin{aligned} s_2[k+1] &= y_e[k+2] + k_2 \cdot y_e[k+1] + k_0 \cdot \text{sgn}(y_e[k]) \cdot \theta_e[k] = \\ &= (1 - q \cdot T_s) \cdot s_2[k] - \varepsilon \cdot T_s \cdot \text{sgn}(s_2[k]) \end{aligned} \quad (3.34)$$

Din (3.33) și (3.34) se obțin comenzile pentru viteza liniară și viteza unghiulară astfel:

$$\begin{aligned} v[k+1] &= \frac{1}{\cos \theta_e[k] \cdot T_s} \cdot [-(1 - q_1 \cdot T_s) \cdot s_1[k] + \varepsilon_1 \cdot T_s \cdot \text{sgn}(s_1[k]) - x_e[k+1] \cdot (1 + k_1) - \\ &- (v_d[k+1] - v_r[k] \cdot \theta_e[k+1] \cdot \sin \theta_e[k] - w_d[k+1] \cdot y_e[k] - w_d[k] \cdot y_e[k+1]) \cdot T_s] \quad ; \end{aligned} \quad (3.35)$$

$$\begin{aligned} \omega[k] &= \frac{1}{v_r[k] \cdot \cos \theta_e + k_0 \cdot \text{sgn } y_e[k+1]} \cdot [-(1 - q_2 \cdot T_s) \cdot s_2[k] + \varepsilon_2 \cdot T_s \cdot \text{sgn}(s_2[k]) - \theta_e[k] - \\ &- y_e[k+1] \cdot (k_2 + 1) - (v_r[k+1] \cdot \sin \theta_e[k] + w_d[k+1] \cdot x_e[k] - w_d[k] \cdot x_e[k+1]) \cdot T_s] + w_d[k] \end{aligned} \quad (3.36)$$

3.3.2 Conducerea sliding-mode cu timp discret a vehiculelor autonome

Conducerea sliding-mode cu timp discret vehiculelor autonome este realizată pornind de la ecuațiile cinematice ale centrului de greutate. Arhitectura conducerii sliding-mode cu timp discret a vehiculului autonom cu 4 DW/SW este prezentată în Fig. 3.8.

Considerând perioada de eşantionare T_s și metoda zero order hold sistemul neliniar (3.5) poate fi scris în timp discret astfel

$$\begin{cases} x_r[k+1] = x_r[k] + v_r[k] \cdot \cos \theta_r[k] \cdot T_s \\ y_r[k+1] = y_r[k] + v_r[k] \cdot \sin \theta_r[k] \cdot T_s \\ \theta_r[k+1] = \theta_r[k] + \frac{v_r[k]}{L} \cdot \tan \delta_r[k] \cdot T_s \end{cases} \quad (3.37)$$

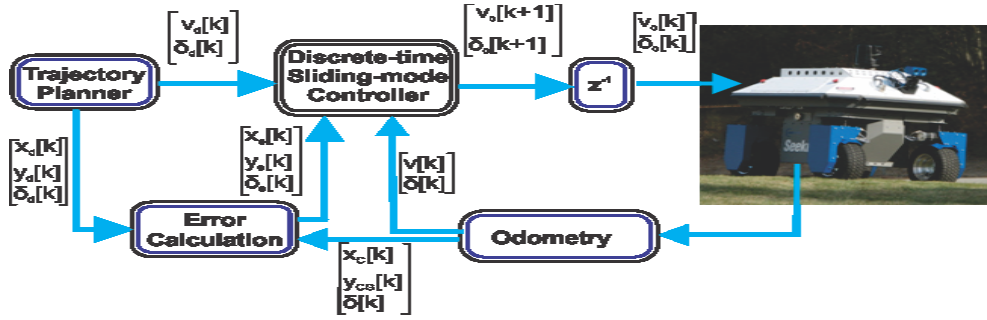


Fig. 3.8 Arhitectura conducerii sliding-mode cu timp discret a vehiculului autonom

Conducerea își propune rezolvarea problemei urmării traiectoriei. Rezolvarea acestei probleme necesită proiectarea unui regulator 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 următorul model cinematic

$$\begin{cases} x_d[k+1] = x_d[k] + v_d[k] \cdot \cos \theta_d[k] \cdot T_s \\ y_d[k+1] = y_d[k] + v_d[k] \cdot \sin \theta_d[k] \cdot T_s \\ \theta_d[k+1] = \theta_d[k] + \frac{v_d[k]}{L} \cdot \tan \delta_d[k] \cdot T_s \end{cases} \quad (3.38)$$

în care: x_d - reprezintă poziția dorită pe axa Ox, y_d - reprezintă poziția dorită pe axa Oy, θ_d - reprezintă orientarea dorită, v_d - reprezintă viteza liniară dorită, ω_d - reprezintă viteza unghiulară dorită.

Erorile de urmărire din Fig. 3.4 sunt

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos \theta_d & \sin \theta_d & 0 \\ -\sin \theta_d & \cos \theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x_d \\ y_r - y_d \\ \theta_r - \theta_d \end{bmatrix} \quad (3.39)$$

Aceste erori pot fi scrise ca

$$\begin{cases} x_e[k] = x_r[k] - x_d[k] \cdot \cos \theta_e[k] + y_r[k] - y_d[k] \cdot \sin \theta_e[k] \\ y_e[k] = -x_r[k] - x_d[k] \cdot \sin \theta_e[k] + y_r[k] - y_d[k] \cdot \cos \theta_e[k] \\ \theta_e[k] = \theta_r[k] - \theta_d[k] \end{cases} \quad (3.40)$$

Dinamica erorilor poate fi exprimată astfel

$$\begin{cases} \dot{x}_e = -v_d + v_r \cdot \cos \theta_e + \omega_d \cdot y_e \\ \dot{y}_e = v_r \cdot \sin \theta_e - \omega_d \cdot x_e \\ \dot{\theta}_e = \dot{\theta}_r - \dot{\theta}_d \end{cases} \quad (3.41)$$

În timp timp discret dinamica erorilor se exprimă

$$\begin{cases} x_e[k+1] = x_e[k] + (v_r[k] \cdot \cos \theta_e[k] - v_d[k] + y_e[k] \cdot \frac{v_d[k]}{L} \cdot \tan \delta_d[k]) \cdot T_s \\ y_e[k+1] = y_e[k] + (v_r[k] \cdot \sin \theta_e[k] - x_e[k] \cdot \frac{v_d[k]}{L} \cdot \tan \delta_d[k] \cdot T) \cdot T_s \\ \theta_e[k+1] = \theta_e[k] + (\frac{v_r[k]}{L} \tan \delta_r[k] - \frac{v_d[k]}{L} \tan \delta_d[k]) \cdot T_s \end{cases} \quad (3.42)$$

Regulatorul sliding-mode cu timp discret este un regulator 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 cu timp discret este controlul discontinuu.

Gao et al. [23] a propus următoarea condiție pentru existență sliding-mode

$$s[k] \cdot (s[k+1] - s[k]) < 0 \quad (3.43)$$

Legea de conducere propusă de Gao et al. [23] este:

$$s[k+1] = (1 - q \cdot T_s) \cdot s[k] - \varepsilon \cdot T_s \cdot \text{sgn}(s[k]); \quad (3.44)$$

$$0 < 1 - q \cdot T < 1; \quad (3.45)$$

$$0 < \varepsilon \cdot T_s < 1; \quad (3.46)$$

în care: $T_s > 0$ este perioada de eșantionare, $\varepsilon > 0$ este viteza de aducere, $q > 0$ este convergența exponențială.

Se definesc suprafețele de comutație

$$\begin{cases} s_1[k] = x_e[k+1] + k_1 \cdot x_e[k] \\ s_2[k] = y_e[k+1] + k_2 \cdot y_e[k] + k_0 \cdot \text{sign}(y_e[k]) \cdot \theta_e[k] \end{cases}, \quad (3.47)$$

în care: k_0, k_1, k_2 sunt constante pozitive x_e, y_e și θ_e sunt erorile de urmărire.

Din (3.44) și (3.47) obținem:

$$s_1[k+1] = x_e[k+2] + k_1 \cdot x_e[k+1] = (1 - q \cdot T_s) \cdot s_1[k] - \varepsilon \cdot T_s \cdot \text{sgn}(s_1[k]) \quad (3.48)$$

$$\begin{aligned} s_2[k+1] &= y_e[k+2] + k_2 \cdot y_e[k+1] + k_0 \cdot \text{sgn}(y_e[k]) \cdot \theta_e[k] = \\ &= (1 - q \cdot T_s) \cdot s_1[k] - \varepsilon \cdot T_s \cdot \text{sgn}(s_1[k]) \end{aligned} \quad (3.49)$$

Din (3.48) și (3.49) se obțin comenzile pentru viteza liniară și viteza unghiulară astfel:

$$\begin{aligned} v[k+1] &= \frac{1}{\cos \theta_e[k] \cdot T_s} \cdot [-(1 - q_1 \cdot T_s) \cdot s_1[k] + \varepsilon_1 \cdot T_s \cdot \text{sgn}(s_1[k] - x_e[k+1]) \cdot (1 + k_1) - \\ &- (v_d[k+1] - v_r[k] \cdot \theta_e[k+1] \cdot \sin \theta_e[k] - w_d[k+1] \cdot y_e[k] - w_d[k] \cdot y_e[k+1]) \cdot T_s] \end{aligned} \quad (3.50)$$

$$\delta[k] = \arctan\left(\frac{L}{v_r[k] \cdot \cos \theta_e + k_0 \cdot \operatorname{sgn} y_e[k+1]} \cdot [-(1 - q_2 \cdot T_s) \cdot s_2[k] + \varepsilon_2 \cdot T_s \cdot \operatorname{sgn}(s_2[k]) - \theta_e[k] - y_e[k+1] \cdot (k_2 + 1) - (v_r[k+1] \cdot \sin \theta_e[k] + w_d[k+1] \cdot x_e[k] - w_d[k] \cdot x_e[k+1]) \cdot T_s]\right) + \frac{L}{v_r} w_d[k] \quad (3.51)$$

3.4. Conducerea backstepping

Conducerea backstepping este una dintre cele mai importante tehnici de conducere a proceselor neliniare și poate fi aplicată la numeroase aplicații. Acest tip de conducere este necesar în cazul sistemelor non-holonomiche datorită faptului că nu se poate aplica condiția de stabilitate a lui Brockett, [4].

Problema conducerii backstepping a mai multor tipuri de roboți nonholonomici a fost prezentată într-un număr mare de lucrări, cum ar fi [5], [29] și [30].

Backstepping este o procedură recursivă care descompune o problemă de proiectare a întregului sistem în probleme de proiectare pentru sisteme de ordin inferior. Proiectarea backstepping are la bază integratorul backstepping, acesta fiind aplicat recursiv pentru rezolvarea problemei sistemelor complexe, procedură descrisă de Chen [5].

3.4.1 Conducerea backstepping aWMR cu 2DW/2FW

Arhitectura de conducere backstepping a roboților mobili cu 2 roți motoare și 2 libere este prezentată în Fig. 3.9.

Pentru a calcula comanda în cazul roboților mobili pentru problema urmăririi globale[30]. considerăm modelul cinematic al robotului (3.2), modelul cinematic al robotului virtual (3.6) erorile de urmărire (3.7) și dinamica erorilor de urmărire(3.8).

Conducerea backstepping presupune calculul legilor de conducere ale vitezelor liniare și unghiulare având forma

$$v = v(x_e, y_e, \theta_e, v_d, \omega_d, \dot{v}_d, \dot{\omega}_d); \quad (3.52)$$

$$\omega = \omega(x_e, y_e, \theta_e, v_d, \omega_d, \dot{v}_d, \dot{\omega}_d), \quad (3.53)$$

astfel încât traiectoria în buclă închisă definită de de (3.8), (3.52),(3.53) să fie mărginită și să convergă spre zero.

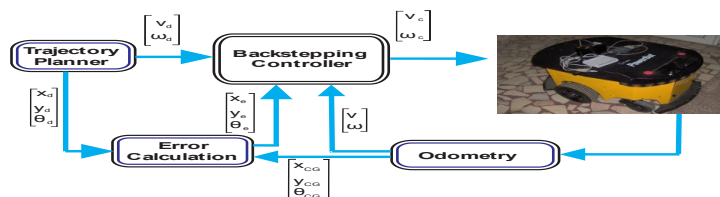


Fig. 3.9 Arhitectura de conducere backstepping a WMR cu 2DW/2FW

Integratorul backstepping se folosește pentru că y_e nu este controlat direct în (3.11).

Funcțiile $x_e = k_1 \cdot \omega_d \cdot y_e$ și $\theta_e = 0$ sunt funcții stabilizatoare pentru y_e în (3.11).

Folosim schimbarea de variabilă:

$$\bar{x}_e = x_e - k_1 \cdot \omega_d \cdot y_e, \quad (3.54)$$

unde $k_1 > 0$ este o constantă.

Utilizând (3.54) în (3.8) se obține ecuația

$$\dot{\bar{x}}_e = -v_d + v_r \cdot \cos \theta_e + \omega_d \cdot y_e - k_1 \cdot \dot{\omega}_d \cdot y_e - k_1 \cdot \omega_d \cdot (-\omega_d \cdot x_e + v_r \cdot \sin \theta_e). \quad (3.55)$$

Folosind schimbarea de variabilă

$$u = v_r \cdot \cos \theta_e + \omega_d \cdot y_e - k_1 \cdot \omega_d \cdot (-\omega_d \cdot x_e + v_r \cdot \sin \theta_e). \quad (3.56)$$

Rezultă comanda backstepping a roboților mobili:

$$v = u - y_e \cdot \omega_d + k_2 \cdot \bar{x}_e; \quad (3.57)$$

$$\omega = \omega_r + \lambda \cdot y_e \cdot v_r \int_0^1 \cos(s \cdot \theta_e) ds + k_3 \cdot \lambda \cdot \theta_e. \quad (3.58)$$

3.4.2 Conducerea backstepping a vehiculului autonom cu 4 DW/SW

Pentru a calcula comanda în cazul vehiculului autonom pentru problema urmăririi globale[30]. Considerăm modelul cinematic al robotului (3.5), modelul cinematic al robotului virtual (3.15), erorile de urmărire (3.16) și dinamica erorilor de urmărire(3.17).

Conducerea backstepping presupune calculul legilor de conducere ale vitezelor liniare și unghiulare având forma:

$$v = v(x_e, y_e, \theta_e, v_d, \omega_d, \dot{v}_d, \dot{\omega}_d) \quad (3.59)$$

$$\delta = \delta(x_e, y_e, \theta_e, v_d, \omega_d, \dot{v}_d, \dot{\omega}_d) \quad (3.60)$$

astfel încât traiectoria în buclă închisă definită de (3.17), (3.59),(3.60) să fie mărginită și să convergă spre zero.

Integratorul backstepping se folosește pentru că y_e nu este controlat direct în (3.17).

Funcțiile $x_e = k_1 \cdot \frac{v_d}{L} \cdot \tan \delta_d \cdot y_e$ și $\theta_e = 0$ sunt funcții stabilizatoare pentru y_e în (3.17).

Folosim schimbarea de variabilă:

$$\bar{x}_e = x_e - k_1 \cdot \frac{v_d}{L} \cdot \tan \delta_d \cdot y_e \quad (3.61)$$

unde $k_1 > 0$ este o constantă și notăm $\frac{v_d}{L} \cdot \tan \delta_d$ cu ω_d .

Utilizând (3.61) în (3.17) se obține ecuația:

$$\dot{\bar{x}}_e = -v_d + v_r \cdot \cos \theta_e + \omega_d \cdot y_e - k_1 \cdot \dot{\omega}_d \cdot y_e - k_1 \cdot \omega_d \cdot (-\omega_d \cdot x_e + v_r \cdot \sin \theta_e) \quad (3.62)$$

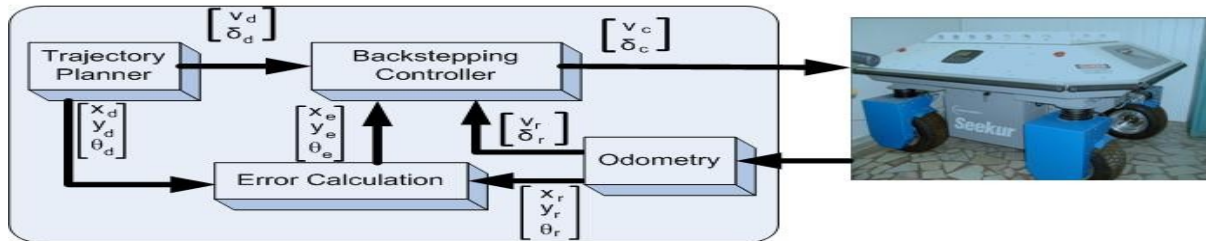


Fig. 3.10 Arhitectura de conducere backstepping a AV cu 4DW/SW

Folosind schimbarea de variabilă:

$$u = v_r \cdot \cos \theta_e + \omega_d \cdot y_e - k_1 \cdot \omega_d \cdot (-\omega_d \cdot x_e + v_r \cdot \sin \theta_e) \quad (3.63)$$

Rezultă comanda backstepping a roboților mobili:

$$v = u - y_e \cdot \omega_d + k_2 \cdot \bar{x}_e \quad (3.64)$$

$$\delta = a \tan\left(\frac{L}{v_r} (\omega_r + \lambda \cdot y_e \cdot v_r \int_0^1 \cos(s \cdot \theta_e) ds + k_3 \cdot \lambda \cdot \theta_e)\right) \quad (3.65)$$

Arhitectura de conducere este prezentată în Fig. 3.10.

3.5 Concluzii

În acest capitol am determinat modelul cinematic al WMR cu 2DW/2FW și modelul cinematic al vehiculului autonom electric SEEKUR (4DW/SW). 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 și AV.

Am propus trei metode pentru rezolvarea problemei conducerii roboților mobili și vehiculelor autonome: conducerea sliding-mode cu timp continuu, conducerea sliding-mode cu timp discret și conducerea backstepping.

De la cazul general și folosind modelul cinematic al WMR cu 2 DW/2SW, modelul erorilor de urmărire și dinamica erorilor de urmărire am calculat comenzile pentru viteza liniară și viteza unghiulară a WMR. În calcul am folosit suprafețe de comutație și legi de conducere bazate pe modelul propus de Gao pentru timp continuu în [22].

Utilizând cazul general și modelul cinematic simplificat al vehiculului autonom SEEKUR, suprafețe de comutație, legi de conducere similare celor propuse de Gao pentru

timp continuu [22], modelul erorilor de urmărire și dinamica erorilor de urmărire am calculat comenzile pentru viteza liniară și unghiul de direcție.

Pornind de la modelul cinematic al WMR în timp discret, erorile de urmărire în timp discret, dinamica erorilor de urmărire în timp discret, suprafețele sliding pentru timp discret și legea de conducere pentru timp discret am calculat comenzile pentru viteza liniară și viteza unghiulară a WMR.

Utilizând modelul cinematic simplificat al vehiculului autonom în timp discret, erorile de urmărire în timp discret, dinamica erorilor de urmărire în timp discret, suprafețele sliding pentru timp discret și legea de conducere pentru timp discret am calculat comenzile pentru viteza liniară și unghiul de direcție a vehiculului autonom.

Am calculat comanda pentru conducerea backstepping în cazul WMR utilizând modelul cinematic al robotului și dinamica erorilor de urmărire a robotului.

Am calculat comenzile pentru viteza liniară și unghiul de direcție al vehiculului autonom SEEKUR utilizând modelul cinematic și dinamica de urmărire a erorilor și procedura de conducere backstepping.

Capitolul 4.

Algoritmi de conducere a vehiculelor autonome și a roboților mobili, destinați evitării obstacolelor

În acest capitol am propus o soluție pentru evitarea obstacolelor de către roboții mobili și vehiculele autonome. Am consideră că roboții mobili sunt echipați cu sonare, iar vehiculele autonome sunt echipate cu senzori laser pentru depistarea obstacolelor și se propun două soluții în funcție de senzorii utilizați.

4.1 Algoritmi de evitare a obstacolelor implementați în conducerea roboților mobili

Am considerat situația în care se dorește urmărirea unei traiectorii impuse cu abateri minore în cazul apariției unui obstacol și revenirea la traiectoria inițială imediat după ce obstacolul a fost depășit. Obstacolele sunt detectate cu ajutorul a 14 sonare dispuse frontal și numerotate de la 0 la 14. Sonarul cu numărul 7 nu a fost implementat. Sonarele acoperă o zonă de 180 de grade. Pentru detectarea obstacolelor se folosește bula de sensibilitate prezentată în [48]. Această bulă este proporțională cu viteza de deplasare și permite o bună detecție a obstacolelor periculoase. Bula de sensibilitate este prezentată în Fig. 4.1. Bula de sensibilitate ($Boundary = [b_1 \ b_2 \ \dots \ b_{14}]$) se calculează utilizând formula

$$b_i = k_i \cdot v \cdot T_s ; \quad (4.1)$$

Unde b_i - reprezintă componenta bulei de sensibilitate corespunzătoare sonarului i , k_i - reprezintă coeficientul de siguranță a bulei de sensibilitate, v - reprezintă viteza robotului, T_s - reprezintă constanta de eșantionare

După calcularea bulei de sensibilitate se verifică dacă un obstacol a pătruns în zona sensibilă. Un obstacol este detectat dacă distanța măsurată de sonar este mai mică decât valoarea componentei bulei de sensibilitate corespunzătoare sonarului respectiv. Cât timp nu sunt detectate obstacole în bula de sensibilitate traiectoria globală calculată off-line este impusă modulului de conducere și robotul urmărește această traiectorie.

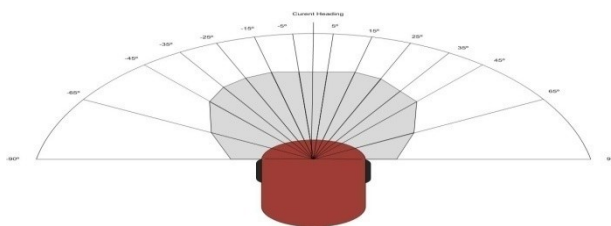


Fig. 4.1 Bula de sensibilitate la PowerBot

Dacă un obstacol a fost detectat în zona de sensibilitate, modulului de conducere îi este impusă o traiectorie locală generată pentru evitarea obstacolului. Traiectoria locală urmărește deplasarea robotului la un punct aflat perpendiculara la direcția de mers dusă din punctul în care este detectat cel mai apropiat obstacol. Pentru determinarea acestui punct se caută o zonă liberă suficient de mare pe perpendiculara la direcția de mers dusă din punctul în care este detectat cel mai apropiat obstacol pentru a permite trecerea robotului. Dacă distanța minimă este măsurată de sonarele aflate în partea stângă(0-6) atunci punctul dorit se caută întâi în partea dreaptă și dacă nu se găsește un astfel de punct se caută în partea stângă. Dacă distanța minimă este măsurată de sonarele aflate în partea dreaptă(8-14) atunci punctul dorit se caută întâi în partea stângă și dacă nu se găsește un astfel de punct se caută în partea dreaptă.

Algorimul este următorul: se determină sonarul care a depistat un obstacol în zona de sensibilitate și se știe că sonarele cu numărul mai mic decât 7 sunt în partea stângă, iar cele mai mari decât 7 sunt în partea dreaptă. În funcție de partea pe care se află obstacolul se încearcă căutarea în partea opusă a unei zone libere suficient de mari pentru a permite trecerea robotului. În Fig. 4.2 se observă că traiectoria dorită(linia neagră) este blocată și se dorește găsirea unui punct cu proprietatea că robotul poate ocoli obstacolul dacă centrul său de greutate se deplasează spre acel punct. Este cunoscută lățimea robotului care este considerată distanța minimă(h_{min}) pentru ca robotul să poată trece și începând cu sonarul 8, și incrementând indicele sonarului ales se verifică, care este primul sonar care nu detectează un obstacol apropiat în cazul ocolirii prin partea dreaptă. În cazul ocolirii prin partea stângă se pornește de la sonarul 6 și se scade indicele sonarului până se găsește primul sonar care nu depistează un obstacol. Se consideră zona de la intersecția direcției de înaintare cu perpendiculara dusă din punctul în care s-a detectat cel mai apropiat obstacol până la punctul determinat de această perpendiculară și unghiul sonarului ales o zonă de excludere.

Se consideră că un obstacol nu depistează un obstacol dacă:

$$range[k] \cdot \cos(alfa[k]) > min_range + c \quad (4.2)$$

Unde $range[k]$ -reprezintă distanța măsurată de sonarul k,

$alfa[k]$ -reprezintă unghiul sonarului k

min_range - reprezintă proiecția distanței până la cel mai apropiat obstacol pe direcția de înaintare,

c- reprezintă o constanta pentru validarea absenței obstacolelor

Se calculează proiecția distanței până la cel mai apropiat obstacol(min_range) și pe baza unghiului sonarului selectat se calculează zona de excludere:

$$Ze = |\min_range \cdot \tan(\text{angle})| \quad (4.3)$$

Urmează determinarea zonei în care poate naviga robotul astfel: se crește(sau scade în funcție de direcția de căutare) numărul sonarului analizat și se verifică dacă zona delimitată de sonarul ales, min_range și sonarul curent este suficient de mare pentru a trece robotul. În caz afirmativ se alege destinația punctul aflat la jumătatea distanței. Dacă nu se găsește un loc de ocolire se încearcă schimbarea direcției de ocolire.

Zona delimitată pe perpendiculara la direcția de mers aleasă are proprietatea:

$$h = \min_range \cdot \tan(\text{alfa}(j)) - Ze > h_{\min} \quad (4.4)$$

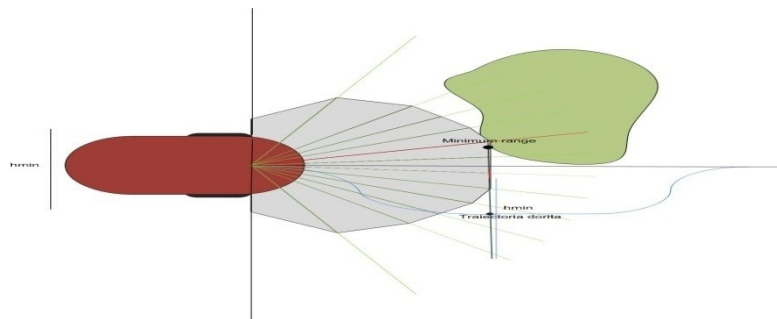


Fig. 4.2 Traiectoria dorita pentru evitarea obstacolului

Unde min_range - reprezintă proiecția pe direcția de înaintare a distanței până la obstacol, $\text{alfa}(j)$ - reprezintă unghiul sonarului j , Ze – reprezintă zona de excludere(4.3).

După determinarea zonei h prin care este posibilă evitarea obstacolului se determină punctul(new_x, new_y) în care se dorește deplasarea centrului de greutate al robotului:

$$\begin{cases} new_x = x + \min_range \cdot \cos(Th) - H \cdot \sin(Th) \\ new_y = y - \min_range \cdot \sin(Th) - H \cdot \cos(Th) \end{cases} \quad (4.5)$$

Unde: (x,y) reprezintă coordonatele curente ale robotului, H - reprezintă distanța de la direcția de înaintare la mijlocului segmentului delimitat de h , Th - reprezintă orientarea robotului.

După selecția noii destinații trebuie calculată o nouă traiectorie care să permită deplasarea robotului de la punctul curent la punctul (new_x, new_y). Traseul va fi generat folosind Quintic equations[47]. Se consideră punctul curent (x_c, y_c) și unghiul curent(θ_c) și se dorește calcularea punctelor intermediare care le unesc de punctul calculat și $unghi_final = unghi_curent + \pi / 4$ se determină punctele intermediare care descriu traseul pentru evitarea obstacolului($P(u) = \{p_1(u), p_2(u), \dots, p_n(u)\}, u \in [0,1]$). Fiecare punct p_i este compus din coordonatele x_i, y_i și θ_i .

O curbă bazată pe quintic equations se calculează astfel[47]:

$$p_{i,i+1}(u) = \begin{bmatrix} x_{i,i+1}(u) \\ y_{i,i+1}(u) \\ \phi_{i,i+1}(u) \end{bmatrix} = \begin{bmatrix} \alpha_{i0} + \alpha_{i1} \cdot u + \alpha_{i2} \cdot u^2 + \alpha_{i3} \cdot u^3 + \alpha_{i4} \cdot u^4 + \alpha_{i5} \cdot u^5 \\ \beta_{i0} + \beta_{i1} \cdot u + \beta_{i2} \cdot u^2 + \beta_{i3} \cdot u^3 + \beta_{i4} \cdot u^4 + \beta_{i5} \cdot u^5 \\ \phi_i(u) \end{bmatrix} \quad (4.6)$$

unde:

$$\alpha_{i0} = x_i(0)$$

$$\alpha_{i1} = g_1 \cdot \cos(\theta_i(0))$$

$$\alpha_{i2} = \frac{1}{2}(g_3 \cdot \cos(\theta_i(0)) - g_1^2 \cdot k_i \cdot \sin(\theta_i(0)))$$

$$\alpha_{i3} = 10 \cdot (x_{i+1} - x_i) - \left(6 \cdot g_1 + \frac{3}{2} \cdot g_3\right) \cos(\theta(0)) - \left(4 \cdot g_2 - \frac{1}{2} g_4\right) \cos(\theta_{i+1}(0)) + \frac{3}{2} \cdot g_1^2 \cdot k_i \cdot \sin(\theta_i(0)) - \frac{1}{2} g_2^2 \cdot k_{i+1} \cdot \sin(\theta_{i+1}(0))$$

$$\alpha_{i4} = -15 \cdot (x_{i+1} - x_i) + \left(8 \cdot g_1 + \frac{3}{2} \cdot g_3\right) \cos(\theta(0)) + (7 \cdot g_2 - g_4) \cos(\theta_{i+1}(0)) - \frac{3}{2} \cdot g_1^2 \cdot k_i \cdot \sin(\theta_i(0)) - g_2^2 \cdot k_{i+1} \cdot \sin(\theta_{i+1}(0))$$

$$\alpha_{i5} = 6 \cdot (x_{i+1} - x_i) - \left(3 \cdot g_1 + \frac{1}{2} \cdot g_3\right) \cos(\theta(0)) - \left(3 \cdot g_2 - \frac{1}{2} g_4\right) \cos(\theta_{i+1}(0)) + \frac{1}{2} \cdot g_1^2 \cdot k_i \cdot \sin(\theta_i(0)) - \frac{1}{2} g_2^2 \cdot k_{i+1} \cdot \sin(\theta_{i+1}(0))$$

$$\beta_{i0} = y_i(0)$$

$$\beta_{i2} = \frac{1}{2}(g_3 \cdot \sin(\theta_i(0)) - g_1^2 \cdot k_i \cdot \cos(\theta_i(0)))$$

$$\beta_{i3} = 10 \cdot (y_{i+1} - y_i) - \left(6 \cdot g_1 + \frac{3}{2} \cdot g_3\right) \sin(\theta(0)) - \left(4 \cdot g_2 - \frac{1}{2} g_4\right) \sin(\theta_{i+1}(0)) + \frac{3}{2} \cdot g_1^2 \cdot k_i \cdot \cos(\theta_i(0)) - \frac{1}{2} g_2^2 \cdot k_{i+1} \cdot \cos(\theta_{i+1}(0))$$

$$\beta_{i4} = -15 \cdot (y_{i+1} - y_i) + \left(8 \cdot g_1 + \frac{3}{2} \cdot g_3\right) \sin(\theta(0)) + (7 \cdot g_2 - g_4) \sin(\theta_{i+1}(0)) - \frac{3}{2} \cdot g_1^2 \cdot k_i \cdot \cos(\theta_i(0)) - g_2^2 \cdot k_{i+1} \cdot \cos(\theta_{i+1}(0))$$

$$\beta_{i5} = 6 \cdot (y_{i+1} - y_i) - \left(3 \cdot g_1 + \frac{1}{2} \cdot g_3\right) \sin(\theta(0)) - \left(3 \cdot g_2 - \frac{1}{2} g_4\right) \sin(\theta_{i+1}(0)) + \frac{1}{2} \cdot g_1^2 \cdot k_i \cdot \cos(\theta_i(0)) - \frac{1}{2} g_2^2 \cdot k_{i+1} \cdot \cos(\theta_{i+1}(0))$$

Parametrii k_i, k_{i+1} reprezintă curbura scalară și pot primi valori aleatoare, în acest caz

se consideră valoarea 0. Parametrii q_1, q_2, g_3, g_4 influențează forma curbei. În acest caz se consideră valorile: $g_1 = g_2 = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$, $g_3 = g_4 = 0$.

Regulatorul sliding-mode primește la intrare viteza, accelerația, viteza unghiulară și accelerația unghiulară, iar utilizând Quintic equations am obținut punctele noului traseu și nu traiectoria reprezentată de profilul de viteză. Pentru a utiliza același regulator trebuie calculate vitezele, accelerațiile, vitezele unghiulare și accelerațiile unghiulare care permit robotului să urmărească traseul generat mai sus.

Pentru fiecare punct al traseului calculăm lungimea segmentului până la punctul precedent. Obținem viteza împărțind această lungime la durată de timp în care se dorește parcurgerea segmentului și accelerația împărțind viteza la durată de timp.

Viteza unghiulară se obține astfel:

$$\omega[n] = a \tan 2(y[n] - y[n-1], x[n] - x[n-1]) / Ts$$

În final obținem accelerația unghiulară împărțind viteza unghiulară la durată de timp.

Se parcurge traiectoria nou generată prin impunerea acestei traiectorii la modulul de conducere. Se continuă deplasarea cu viteză constantă în ultima direcție până când obstacolul este evitat. Se estimează pozițiile în care s-ar fi aflat robotul dacă urma traiectoria inițială și apoi se calculează traiectoria de revenire la traiectoria inițială în locul în cel mai apropiat de poziția curentă a robotului și pozițiile estimate utilizând același algoritm folosit la determinarea traiectoriei locale, și se impune această traiectorie modulului de conducere, apoi se revine la traiectoria impusă începând cu punctul selectat anterior. Fig. 4.3 prezintă schema bloc a algoritmului prezentat pentru evitarea de obstacole.

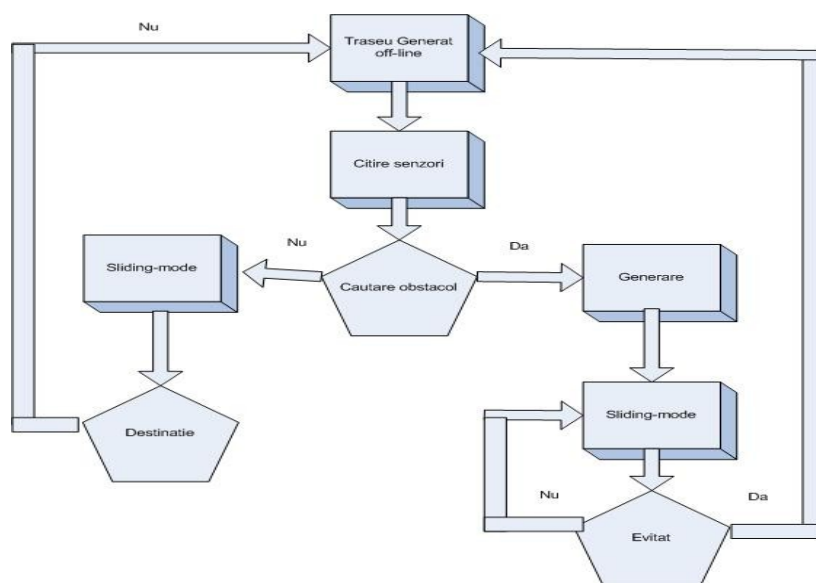


Fig. 4.3 Schema bloc a algoritmului de conducere cu evitarea de obstacole

4.2. Algoritmi de evitare a obstacolelor implementați în conducerea vehiculelor autonome

Se consideră un mediu cunoscut în care se dorește urmărirea unei traiectorii impuse de către un vehicul autonom. În acest mediu este posibilă apariția unor obstacole neașteptate care trebuie evitate. Se dorește evitarea acestor obstacole abandonând traiectoria impusă doar cât timp este blocată de un obstacol și revenirea la traiectoria impusă după ce obstacolul a fost evitat.

Obstacolele sunt detectate cu ajutorul unui laser SICK LMS111. Acest laser permite detectarea obstacolelor aflate într-un arc de la -90° la 90° de centrul robotului. Laserul este programat să ofere date cu o rezoluție de 1° , rezultând un număr total de 181 distanțe măsurate. Pentru a reduce volumul de date procesate programul folosește o rezoluție de 5° , obținută din media aritmetică a 5 distanțe consecutive. Această aproximare garantează că nici un obstacol nu va trece neobservat și că influența unor date eronate este limitată. Micșorând rezoluția de la 1 grad la 5 grade obținem 37 distanțe. La aceste 37 distanțe se poate defini o bulă de sensibilitate[48], care acoperă 180 de grade și permite evitarea oricărui obstacol. Bula de sensibilitate definește o zonă de siguranță proporțională cu viteza vehiculului, în care vehiculul se poate deplasa în siguranță. Calculul bulei de siguranță se realizează multiplicând coeficienții stabiliți pentru fiecare citire cu viteza vehiculului și perioada de eșantionare(4.1).

Procedura de evitare a obstacolelor începe dacă este detectat un obstacol în interiorul bulei de sensibilitate. În lipsa obstacolelor conducerea se realizează utilizând conducerea sliding-mode sau backstepping și traiectoria globală generată off-line.

Dacă un obstacol este detectat se calculează o traiectorie locală de ocolire care este impusă modulului de conducere până când obstacolul este evitat. În Fig. 4.4 este prezentată traiectoria dorită pentru evitarea unui obstacol.

Traectoria locală urmărește deplasarea robotului la un punct aflat perpendiculara la direcția de mers dusă din punctul în care este detectat cel mai apropiat obstacol. Pentru determinarea acestui punct se caută o zonă liberă suficient de mare pe perpendiculara la direcția de mers dusă din punctul în care este detectat cel mai apropiat obstacol pentru a permite trecerea robotului. Dacă distanța minimă este se află în partea stângă($i > 17$) atunci punctul dorit se caută întâi în partea dreaptă și dacă nu se găsește un astfel de punct se caută în partea stângă. Dacă distanța minimă este măsurată se află în partea dreaptă($i < 17$) atunci punctul dorit se caută întâi în partea stângă și dacă nu se găsește un astfel de punct se caută în partea dreaptă.

Pentru a evita obstacolul trebuie găsită o zonă fără obstacole suficient de mare pentru a permite trecerea vehiculului. Se definesc 2 zone de evitare, aflate la stânga și la dreapta direcției de deplasare. Citirile de la 0 la 17 se repartizează în zona din dreapta, iar citirile de la 19 la 36 în zona din partea stângă. Se determină la care citire s-a detectat cel mai apropiat obstacol în zona de sensibilitate. Dacă obstacolul se află în partea stângă se încearcă ocolirea prin partea dreaptă, dacă nu se obține o soluție se caută o soluție și în partea stângă. Dacă obstacolul se află în partea dreaptă se încearcă ocolirea prin partea stângă, dacă nu se obține o soluție se caută o soluție și în partea dreaptă. Evitarea prin ocolire la dreapta este prezentată în Fig. 4.4. Se observă că traiectoria dorită (linia roșie) este blocată și se calculează traiectoria de ocolire (linia albastră). Punctul spre care se deplasează vehiculul pentru evitarea traiectoriei se calculează în funcție de lățimea robotului plus o marjă de siguranță.

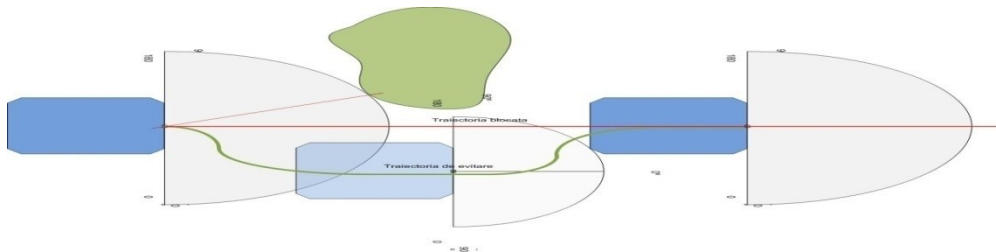


Fig. 4.4 Traiectoria dorită pentru evitarea obstacolului în cazul utilizării detecției cu ajutorul senzorilor laser

Se determină primul index a cărui distanță măsurată este mai mare decât distanța minimă. Se calculează proiecția distanței până la cel mai apropiat obstacol (min_range) și pe baza unghiului selectat se calculează zona de excludere folosind ecuația (4.4).

Condiția pentru selectarea primului index este

$$\text{range}[k] \cdot \cos(\text{alfa}[k]) > \text{min_range} + c, \quad (4.7)$$

unde $\text{range}[k]$ este distanța măsurată la indexul k , $\text{alfa}[k]$ unghiul la care se măsoară distanța.

Urmează determinarea zonei în care poate naviga vehiculul astfel: se crește (sau scade în funcție de direcția de căutare) indexul citirilor analizate și se verifică dacă zona delimitată de unghiul ales, min_range și unghiul curent este suficient de mare pentru a trece vehiculul aplicând condiția (4.4). În caz afirmativ se alege ca destinație punctul aflat la jumătatea distanței. Dacă nu se găsește un loc de ocolire se încearcă schimbarea direcției de ocolire. Coordonatele acestui punct sunt date de (4.5).

După selecția noii destinații trebuie calculată o nouă traiectorie care să ducă robotul de la punctul curent la punctul ($\text{new_x}, \text{new_y}$). Traseul va fi generat folosind Quintic

equations. Se consideră punctul curent și unghiul curent și se dorește calcularea punctelor intermediare care le unesc de punctul calculat și $unghi_final = unghi_curent + \pi / 4$.

Regulatorul sliding-mode primește la intrare viteza, accelerația, viteza unghiulară și accelerația unghiulară. Având punctele traseului calculăm vitezele, accelerațiile, vitezele unghiulare și accelerațiile unghiulare care descriu traiectoria necesară modulului de conducere conform cazului WMR.

Se parcurge traiectoria nou generată prin impunerea acestei traiectorii la modulul de conducere. Se continuă deplasarea cu viteză constantă în ultima direcție până când obstacolul este evitat. Se estimează pozițiile în care s-ar fi aflat robotul dacă urma traiectoria inițială și apoi se calculează traiectoria de revenire la traiectoria inițială în locul în cel mai apropiat de poziția curentă a robotului și pozițiile estimate utilizând același algoritm folosit la determinarea traiectoriei locale, și se impune această traiectorie modulului de conducere, apoi se revine la traiectoria impusă începând cu punctul selectat anterior. Fig. 4.3 prezintă schema bloc a algoritmului prezentat pentru evitarea de obstacole.

4.3. Concluzii

Am propus o soluție pentru problema evitării obstacolelor în cazul în care se dorește urmărirea unei traiectorii impuse și abaterea de la această traiectorie doar pentru a evita un obstacol și întoarcerea la traiectoria impusă. Am fost considerat cazul în care robotul este echipat cu sonare și cazul în care vehiculul este echipat cu senzori laser.

Detecția obstacolelor folosește o bulă de sensibilitate pentru a determina dacă un obstacol trebuie evitat. Am folosit aceeași metodă de conducere în cazul în care nu există obstacole și în cazul în care un obstacol este prezent. Cât timp nu este detectat nici un obstacol care blochează traiectoria dorită platforma urmărește traiectoria impusă folosind metoda de conducere aleasă din cele 3 prezentate în capitolul anterior.

Dacă un obstacol a fost detectat se caută coordonatele care să permită conducerea în siguranță a platformei și se generează o traiectorie bazată pe Quintic equations. Această traiectorie este impusă modulului de conducere. Se continuă deplasarea pe această traiectorie până când obstacolul a fost evitat. După ce obstacolul a fost evitat se calculează traiectoria pentru a reveni la traiectoria impusă și modul de conducere urmărește această traiectorie apoi se revine la traiectoria impusă inițial.

Capitolul 5.

Implementarea conducerii în timp real. Rezultate experimentale

Activitatea de cercetare s-a desfășurat în Laboratorul de Robotică al Facultății de Automatică, Calculatoare, Inginerie Electrică și Electronică din Universitatea „Dunărea de Jos” din Galați. Cercetarea s-a realizat pe roboții mobili Pioneer P3-DX, PatrolBot, PowerBot și vehiculul autonom SEEKUR de la Adept Mobile Robots.

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.

Programul este scris în C++ și rulat pe un PC cu o frecvență de eșantionare de 100 ms. În continuare sunt prezentate simulări și implementări în timp real pentru a valida metodele propuse. Pentru a trimite comenzile calculate de algoritmi de conducere am folosit funcțiile din ARIA.

5.1. Simularea conducerii roboților mobili și vehiculelor autonome

Regulatorul 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. 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 simularea evitării obstacolelor a fost creată o hartă în Mapper3basic în care a fost introdus obstacolul și această hartă a fost încărcată în MobileSim. Programele sunt scrise în C++ și compilate în Visual Studio. 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.

- Simularea 5.1

În acest test s-a dorit urmărirea unei traiectorii liniare cu viteza de 0.5m/s, urmată de o traiectorie circulară, cu viteza unghiulara 0.2 m/s, de către vehiculul autonom 4DW/SW SEEKUR utilizând conducerea sliding-mode cu timp continuu. Parametrii constanți folosiți în acest experiment sunt: $Q_1 = 0.05$, $Q_2 = 0.5$, $P_1 = 0.5$, $P_2 = 0.75$, $k_0 = 30$, $k_1 = 1.25$, $k_2 = 100$. Acești parametri au fost obținuți prin identificare în urma unor simulări succesive utilizând diferite valori ale parametrilor.

În continuare sunt prezentate graficele rezultate. În Fig. 5.1 este prezentată simularea în MobileSim a traiectoriei vehiculului autonom SEEKUR utilizând conducerea sliding-mode cu timp continuu. Fig. 5.2 prezintă eroarea pe axa Ox și eroarea pe axa Oy obținute prin simularea conducerii sliding-mode cu timp continuu

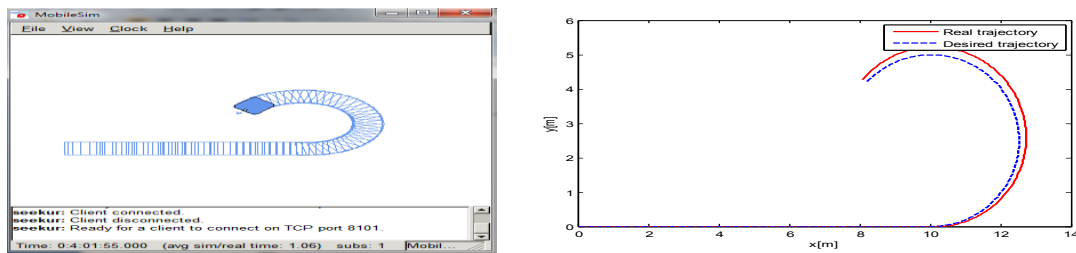


Fig. 5.1 Traiectoria obținută prin simulare în MobileSim la conducerea sliding-mode cu timp continuu a vehiculului autonom 4DW/SW SEEKUR.

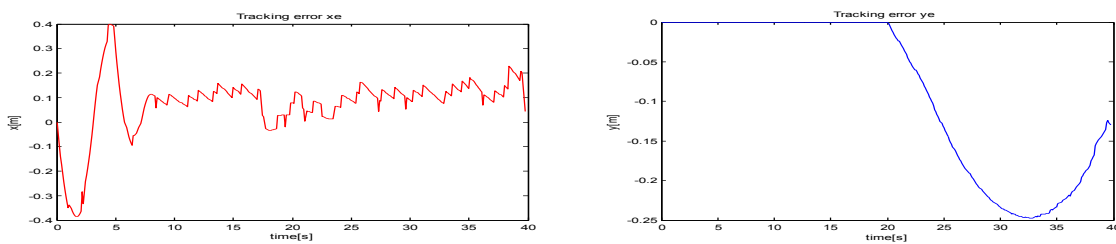


Fig. 5.2 Eroarea de urmărire pe axa X(stânga) și eroarea de urmărire pe axa Y obținute prin simulare în MobileSim la conducerea sliding-mode cu timp continuu a vehiculului autonom 4DW/SW SEEKUR.

Fig. 5.3 prezintă eroarea de orientare a robotului în cazul simulării în conducerii sliding-mode cu timp continuu a vehiculului autonom SEEKUR.

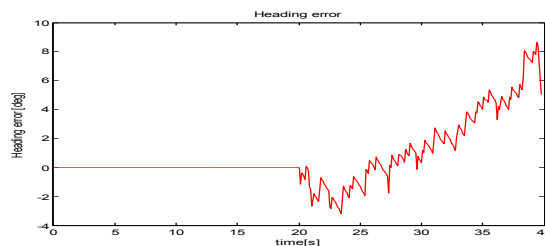


Fig. 5.3 Eroarea de urmărire a direcției obținută prin simulare în MobileSim la conducerea sliding-mode cu timp continuu a vehiculului autonom 4DW/SW SEEKUR.

Se observă că erorile de poziționare sunt mici, performanțele de urmărire a traiectoriei sunt destul de bune în cazul folosirii conducerii sliding-mode cu timp continuu a vehiculului autonom SEEKUR.

- Simularea 5.2.

Urmărirea unei traiectorii liniare urmată de o traiectorie în formă de S a fost testată pentru robotul mobil PatrolBot utilizând conducerea sliding-mode cu timp discret. Parametrii constanți folosiți în acest experiment sunt: $q_1 = 0.9$, $q_2 = 0.9$, $\varepsilon_1 = 0.01$, $\varepsilon_2 = 0.75$, $k_0 = 9.5$, $k_1 = 0.75$, $k_2 = 15$. Acești parametri au fost obținuți prin identificare în urma unor simulări succesive utilizând diferite valori ale parametrilor. În Fig. 5.4 este prezentată simularea în MobileSim a traiectoriei robotului PatrolBot utilizând conducerea sliding-mode cu timp discret. Fig. 5.5 prezintă eroarea de urmărire a traiectoriei pe axa Ox și eroarea de urmărire pe axa Oy a robotului PatrolBot utilizând conducerea sliding-mode cu timp discret. Fig. 5.6 prezintă eroarea de urmărire a orientării robotului PatrolBot utilizând conducerea sliding-mode cu timp discret.

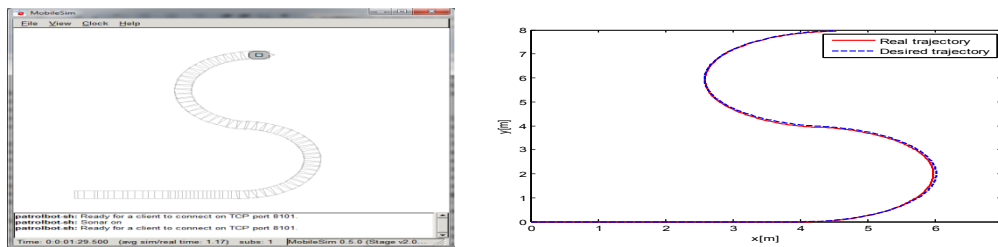


Fig. 5.4. Traiectoria obținută prin simulare în MobileSim la conducerea sliding-mode cu timp discret a robotului 2DW/2FW PatrolBot

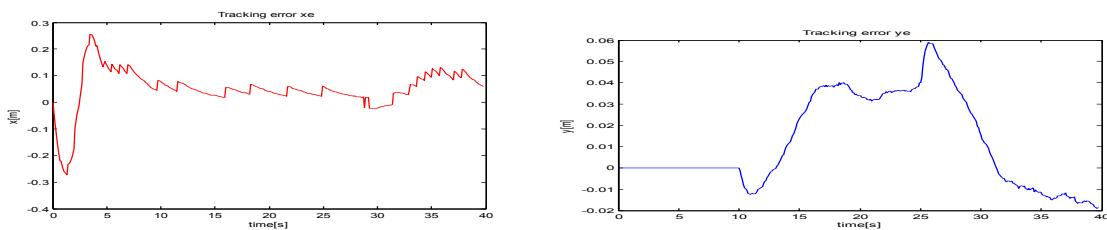


Fig. 5.5 Eroarea de urmărire pe axa X(stânga) și eroarea de urmărire pe axa Y(dreapta) obținute prin simulare în MobileSim la conducerea sliding-mode cu timp discret a robotului 2DW/2FW PatrolBot

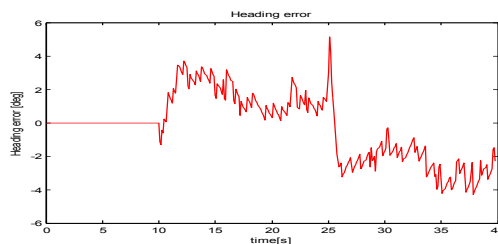


Fig. 5.6 Eroarea de urmărire a direcției obținută prin simulare în MobileSim la conducerea sliding-mode cu timp discret a robotului 2DW/2FW PatrolBot.

Din figurile prezentate se observă că robotul urmărește traiectoria dorită cu erori mici.

- Simularea 5.3

Simularea a dorit urmărirea de către robotul PowerBot a traiectoriei prezentate în Fig. 5.7 utilizând conducerea backstepping, această traiectorie a fost generată de un planificator, iar în partea stângă este prezentată simularea conducerii backstepping a WMR în MobileSim. Parametrii constanți folosiți în acest experiment sunt: $k_1 = 0.5$, $k_2 = 1.25$, $k_3 = 0.3$. Acești parametri au fost obținuți prin identificare în urma unor simulări succesive utilizând diferite valori ale parametrilor.

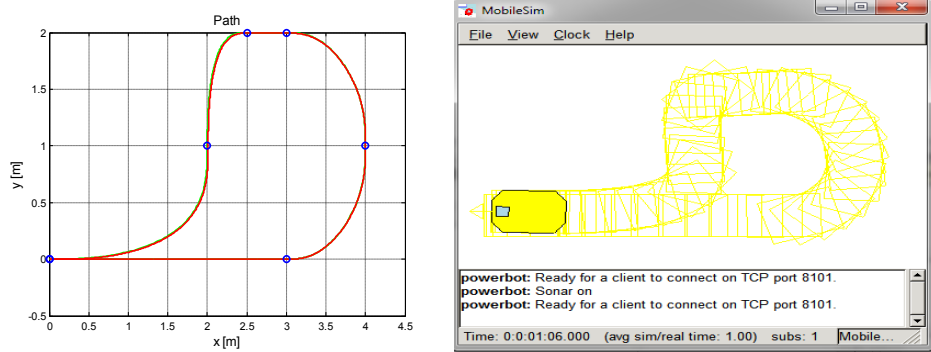


Fig. 5.7. Traiectoria generată de planificator(stânga) pentru testarea conducerii backstepping a robotului 2DW/2FW Powerbot și traiectoria obținută prin simulare în MobileSim la conducerea backstepping a robotului 2DW/2FW Powerbot

În

Fig. 5.8 este prezentată traiectoria simulată cu o linie continuă roșie și traiectoria dorită cu o linie întreruptă albastră a robotului PowerBot utilizând conducerea backstepping și eroarea de urmărire a robotului PowerBot simulată pe axa Ox utilizând conducerea backstepping. Fig. 5.9 prezintă eroarea de urmărire simulată pe axa Oy a robotului PowerBot utilizând conducerea backstepping și eroarea de urmărire a direcției.

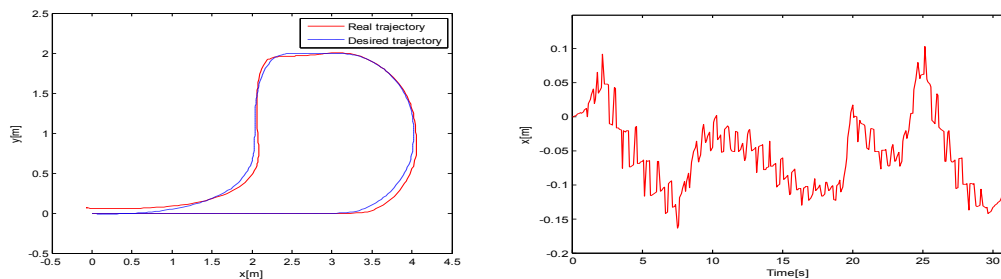


Fig. 5.8. Traiectoria obținută prin simulare în MobileSim și traiectoria impusă la conducerea backstepping a robotului 2DW/2FW Powerbot(stânga) și eroarea de urmărire pe axa X obținută

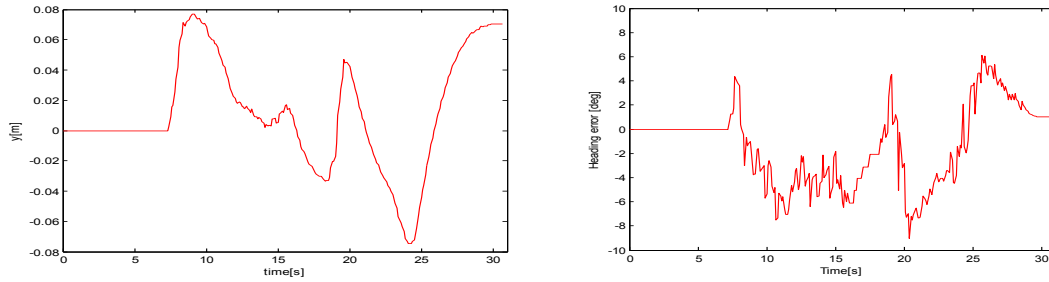


Fig. 5.9. Eroarea de urmărire pe axa Y și eroarea de urmărire a direcției obținute prin simulare în MobileSim la conducerea backstepping a robotului 2DW/2FW Powerbot.

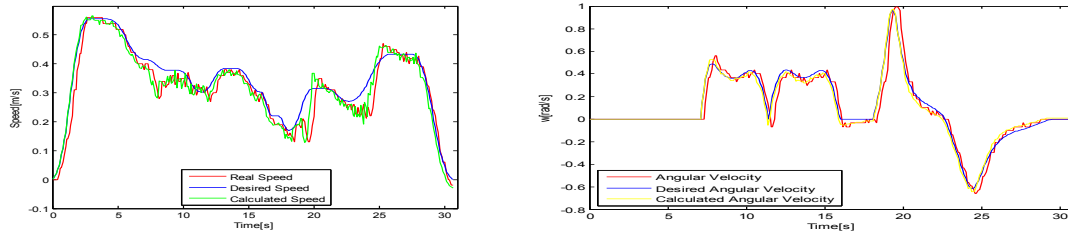


Fig. 5.10 Viteza liniară și viteza unghiulară obținute prin simulare în MobileSim la conducerea backstepping a robotului 2DW/2FW Powerbot

În Fig. 5.10 este prezentată viteza simulată, viteza dorită și viteza calculată utilizând conducerea backstepping a robotului PowerBot și viteza unghiulară dorită, viteza unghiulară reală, viteza unghiulară calculată utilizând conducerea backstepping a robotului PowerBot.

Se observă că traiectoria propusă este urmărită și erorile de urmărire sunt mici. Din graficele vitezelor liniare și unghiulare se observă că vitezele dorite, vitezele calculate și vitezele reale au valori apropiate și această metodă de urmărire este eficientă.

Simularea 5.5

Folosind MobileSim a fost simulată funcționarea arhitecturii de evitare a obstacolelor propuse atât în lipsa obstacolelor cât și în cazul în care e prezent unul sau două obstacole. O hartă simulând holul de la intrarea în corpul Y cu și fără obstacole a fost creată utilizând Mapper3basic. Modulul de conducere folosit pentru testarea arhitecturii de evitare a obstacolelor utilizează conducerea sliding-mode cu timp discret.

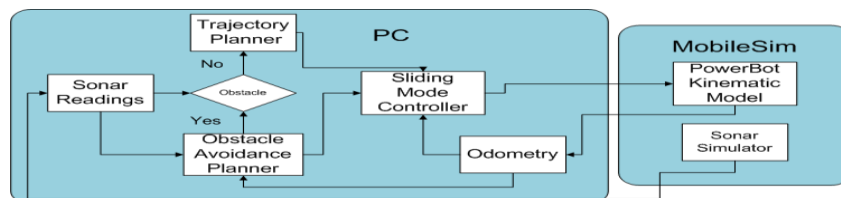


Fig. 5.11 Arhitectura algoritmului pentru evitarea obstacolelor a robotului 2DW/2FW PowerBot în MobileSim

Parametrii constanți sunt $q_1 = 0.9$, $q_2 = 0.9$, $\varepsilon_1 = 0.01$, $\varepsilon_2 = 0.5$. Acești parametri au fost obținuți prin identificare în urma unor simulări succesive utilizând diferite valori ale

parametrilor. Fig. 5.11 prezintă arhitectura folosită pentru evitarea obstacolelor a WMR 2DW/2FW PowerBot.

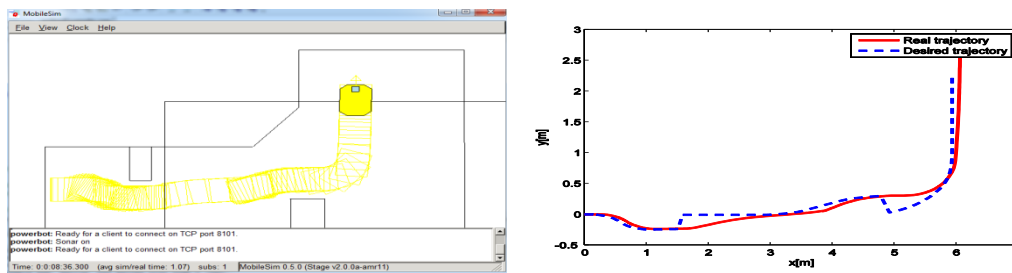


Fig. 5.12. Traiectoria obținută prin simularea în MobileSim a algoritmului de evitarea obstacolelor, în prezența a două obstacole, a robotului 2DW/2FW PowerBot

Fig. 5.12 prezintă traiectoria obținută prin simulare în MobileSim a algoritmului. Fig. 5.13 prezintă eroarea pe axa Ox și eroarea pe axa Oy. Fig. 5.14 prezintă eroarea de orientare a robotului.

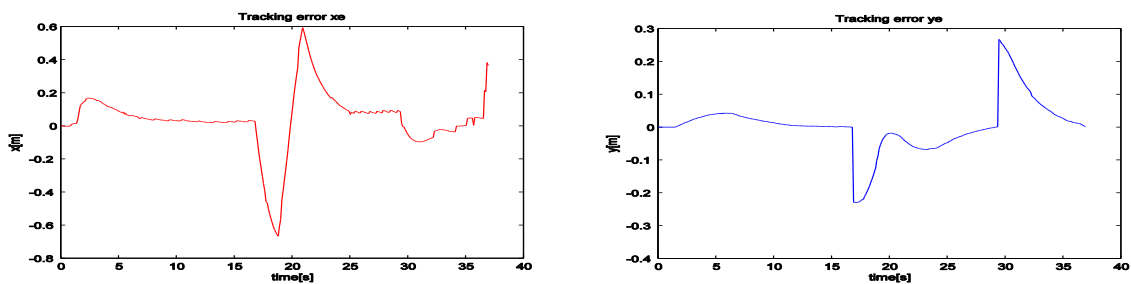


Fig. 5.13. Eroarea de urmărire pe axa X și eroarea de urmărire pe axa Y obținute prin simularea în MobileSim a algoritmului de evitarea obstacolelor, în prezența a două obstacole, a robotului 2DW/2FW PowerBot.

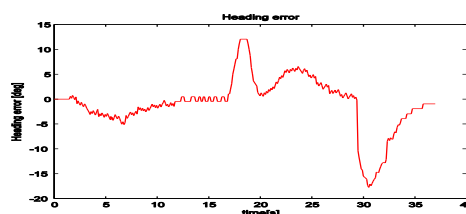


Fig. 5.14. Eroarea de urmărire a direcției obținută prin simularea în MobileSim a algoritmului de evitarea obstacolelor, în prezența a două obstacole, a robotului 2DW/2FW PowerBot.

Se observă că algoritmul de evitarea obstacolelor propus evită cele două obstacole și revine la traiectoria dorită. Evitarea obstacolelor introduce erori semnificative de urmărire și analizând graficele se poate determina începutul fiecărei proceduri de evitarea obstacolelor.

Simularea 5.8.

Se testează urmărirea traiectoriei generate de planificator în prezența unui obstacol. Parametrii constanți sunt $q_1 = 0.9$, $q_2 = 0.9$, $\varepsilon_1 = 0.02$, $\varepsilon_2 = 0.4$. Acești parametri au fost

obținuți prin identificare în urma unor simulări succesive utilizând diferite valori ale parametrilor.

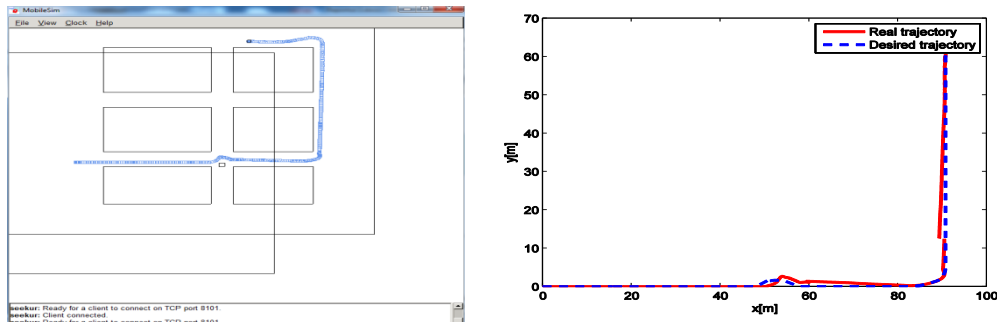


Fig. 5.15. Traiectoria obținută prin simularea în MobileSim a algoritmului pentru evitarea obstacolelor, în cazul prezenței unui obstacol, a vehiculului autonom 4DW/SW SEEKUR

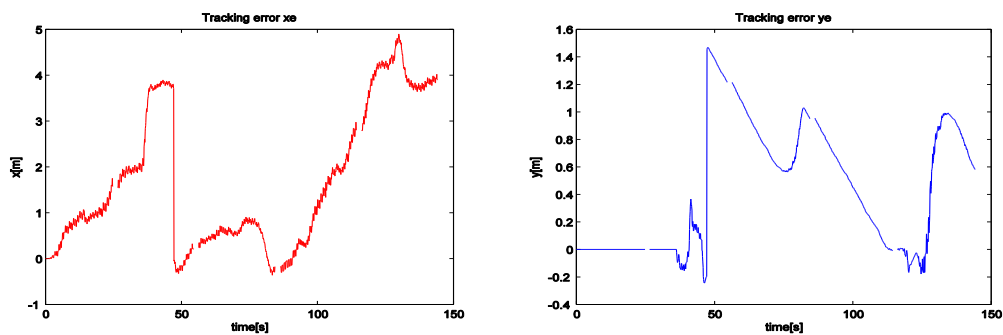


Fig. 5.16. Eroarea de urmărire pe axa X obținută prin simularea în MobileSim a algoritmului pentru evitarea obstacolelor, în cazul prezenței unui obstacol, a vehiculului autonom 4DW/SW SEEKUR.

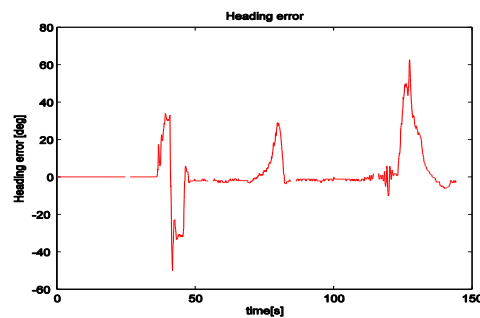


Fig. 5.17. Eroarea de urmărire a direcției obținută prin simularea în MobileSim a algoritmului pentru evitarea obstacolelor, în cazul prezenței unui obstacol, a vehiculului autonom 4DW/SW SEEKUR.

În Fig. 5.15 este prezentat traseul simulat în MobileSim. Fig. 5.16 prezintă eroarea de urmărire x_e și eroarea de urmărire y_e . În Fig. 5.17 este prezentată eroarea de urmărire a direcției.

Se observă că vehiculul urmărește traiectoria chiar și în prezența obstacolului, dar prezența obstacolului introduce erori de urmărire semnificative. Se observă o întârziere între momentul în care este trimisă comanda pentru începerea evitării obstacolului și începerea evitării de către vehicul.

5.2 Rezultatele experimentale ale conducerii în timp real ale roboților mobili și vehiculelor autonome

Experimente în timp real au fost efectuate pe robotul mobil PowerBot pentru a testa eficiența regulatorului sliding-mode cu timp discret. Programul, scris în C++, pentru conducerea robotului rulează pe un PC și trimite comenzi robotului și primește datele despre odometrie utilizând comenzile ARIA. Comunicarea dintre calculator și robot este gestionată de softul ARIA. Conectarea la calculator se realizează utilizând o rețea wireless. Pentru conectarea la robot se folosește un router wireless pentru conexiunea la rețea și pentru că robotul are o conexiune RS232 la router se conectează un Universal Data Server de la Lantronix pentru conversia din protocolul RS232 la protocolul Ethernet. Programul scris în C++ și compilat în Visual Studio este rulat utilizând CMD. Arhitectura conducerii sliding-mode cu timp real a WMR 2DW/2FW PowerBot este prezentată în Fig. 5.18.

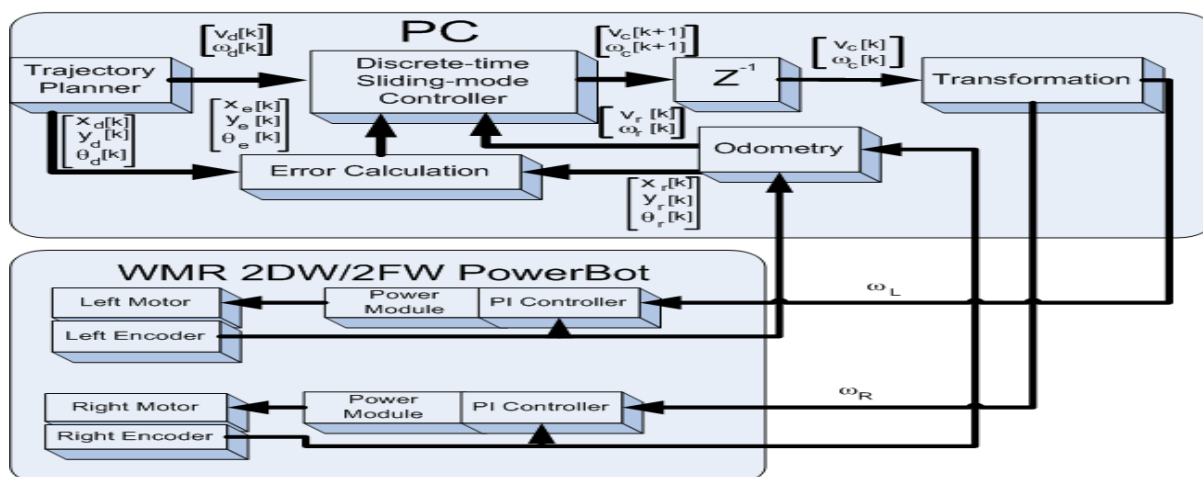


Fig. 5.18. Arhitectura conducerii sliding-mode cu timp discret a WMR 2DW/2FW Powerbot în timp real

Traectoria urmărită a fost compusă dintr-o traiectorie liniară și o traiectorie circulară. Parametrii constanți sunt $q_1 = 0.9$, $q_2 = 0.85$, $\varepsilon_1 = 0.02$, $\varepsilon_2 = 0.5$. Acești parametri au fost identificați experimental. Fig. 5.19 este prezentată traiectoria reală cu o linie continuă roșie și traiectoria dorită cu o linie întreruptă albastră și eroarea de urmărire pe axa X.

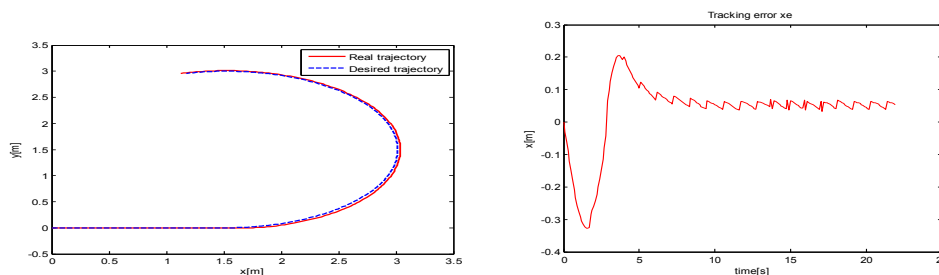


Fig. 5.19. Traectoria obținută la conducerea în timp real prin metoda sliding-mode cu timp discret a robotului 2DW/2FW PowerBot și eroarea de urmărire pe axa X obținută

Fig. 5.20 prezintă eroarea pe axa Oy și eroarea de orientare a robotului.

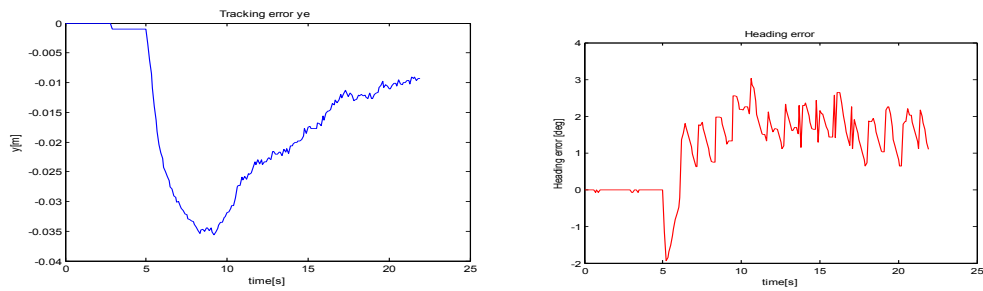


Fig. 5.20. Eroarea de urmărire pe axa Y și eroarea de urmărire a direcției obținute la conducerea în timp real prin metoda sliding-mode cu timp discret a robotului 2DW/2FW PowerBot.

Se observă o eroare inițială pe axa X la pornirea robotului care este redusă după aproximativ 5s. Erorile de urmărire obținute sunt mici și dovedesc eficiența conducerii propuse.

Capitolul 6

Concluzii

În această lucrare am tratat problema conducerii și evitării de obstacole utilizând roboții mobili și vehiculele autonome. Pentru o mai bună calculare a odometriei am analizat senzorul IMU Xsens Mti de la Xsens și am proiectat un filtru Kalman pentru a elimina erorile apărute în calculul atitudinii furnizată de IMU, datorită acumulării erorilor de integrare a semnalului afectat de zgomot. Cercetarea senzorilor IMU a fost realizată cu suportul Institutului de Sisteme și Robotică , din cadrul Universității din Coimbra, sub supervizarea profesorului Urbano Nunes. Am determinat modelele cinematice ale roboților mobili și vehiculelor autonome. Am folosit aceste modele pentru a sintetiza comenzile necesare pentru urmărirea traiectoriilor.

Am folosit trei metode de conducere diferite pentru a realiza conducerea roboților mobili și a vehiculelor autonome: conducerea sliding-mode cu timp continuu, conducerea sliding-mode cu timp discret și conducerea backstepping. Pentru fiecare tip de conducere am calculat comenzile specifice pentru roboții mobili și vehiculele autonome. Metodele de conducere obținute pot fi implementate pe toți roboții din laboratorul de robotică.

Am propus o metodă de evitare a obstacolelor care utilizează sonarele în cazul WMR și senzori laser în cazul AV. Am considerat cazul în care se dorește urmărirea traiectoriei impuse și abateră doar pentru a evita un obstacol și întoarcerea la traiectoria inițială. Am folosit un generator de traiectorii online care stabilește o nouă traiectorie de urmărit și care este impusă modulului de conducere, fără am fi nevoie de elaborarea unui nou modul de conducere pentru a evita obstacolul. Metoda propusă permite utilizarea oricărei metode de conducere propuse în această lucrare.

Am realizat simulări și implementări în timp real pentru a valida algoritmi propuși în această lucrare.

6.1. Contribuții privind conducerea și evitarea obstacolelor utilizând roboți mobili și vehicule autonome

Principalele contribuții gravitează în jurul domeniilor:

- Estimarea atitudinii
- Conducerea trajectory-tracking a roboților mobili și vehiculelor autonome

- Evitarea obstacolelor
- Testarea algoritmilor propuși

6.1.1 Estimarea atitudinii

Am pornit de la nevoia de a avea date corecte despre orientarea vehiculului și faptul că IMU poate oferi date despre orientare dacă se filtrează erorile. Pentru că am constatat erori semnificative la integrarea semnalelor de la senzori pentru obținerea atitudinii am analizat senzorul IMU Xsens Mti pentru a determina erorile caracteristice. Am analizat senzorul utilizând metoda Allan. Am ales cuaternionii ca mod de reprezentare a atitudinii pentru că nu prezintă singularitatea specifică reprezentării cu ajutorul unghiurilor Euler, dar pentru validarea rezultatelor, am trecut de la reprezentarea atitudinii în cuaternioni la reprezentarea Euler pentru o mai bună înțelegere a rezultatelor. Utilizând caracteristica rezultată am elaborat un filtru Kalman indirect în două etape pentru a elimina erorile care se acumulează la calculul orientării. Am constatat că este necesară folosirea mai multor senzori pentru o estimare corectă a atitudinii. În acest caz am folosit datele de la accelerometre și senzorii magnetici pentru a obține corecțiile necesare. Filtrul folosește datele de la giroscopie pentru a calcula atitudinea estimată apoi aplică corecții în două etape. În prima etapă sunt aplicate corecții rezultate de la accelerometre, starea eroare rezultată este apoi folosită la corecția atitudinii estimate. Etapa a doua aduce corecțiile magnetice, după actualizarea stării eroare se corectează estimarea atitudinii cu noua valoare obținută.

6.1.2 Conducerea trajectory-tracking a roboților mobili și vehiculelor autonome

Am tratat problema urmării traiectoriei în capitolul 3. Am analizat trei metode de conducere pentru conducerea vehiculelor și roboților mobili: conducerea sliding-mode cu timp continuu, conducerea sliding-mode cu timp discret și conducerea backstepping.

Am propus rezolvarea problemei conducerii roboților mobili și vehiculelor autonome utilizând conducerea sliding-mode cu timp continuu, conducerea sliding-mode cu timp discret și conducerea backstepping bazate pentru WMR și AV.

De la cazul general și folosind modelul cinematic al WMR cu $2 DW/2FW$, modelul erorilor de urmărire și dinamica erorilor de urmărire am calculat comenzile pentru viteza liniară și viteza unghiulară a WMR. În calcul am folosit suprafețe de comutație și legi de conducere bazate pe modelul propus de Gao pentru timp continuu [22].

Utilizând cazul general și modelul cinematic simplificat al vehiculului autonom SEEKUR, suprafețe de comutație, legi de conducere similare celor propuse de Gao pentru timp continuu [22], modelul erorilor de urmărire și dinamica erorilor de urmărire am calculat comenzile pentru viteza liniară și unghiul de direcție.

Am sintetizat modul de calcul al comenzii utilizând sliding-mode cu timp discret pentru cazul general. Utilizând cazul general am trecut la calculul comenzilor pentru WMR și AV.

Pornind de la modelul cinematic al WMR în timp discret, erorile de urmărire în timp discret, dinamica erorilor de urmărire în timp discret, suprafețele de comutație pentru timp discret și legea de conducere pentru timp discret am calculat comenzile pentru viteza liniară și viteza unghiulară a WMR. Conducerea în timp discret a WMR și AV a fost prezentată în publicațiile: [12], [13], [14], [14], [16], [17], și [44].

Utilizând modelul cinematic simplificat al vehiculului autonom în timp discret, erorile de urmărire în timp discret, dinamica erorilor de urmărire în timp discret, suprafețele de comutație pentru timp discret și legea de conducere pentru timp discret am calculat comenzile pentru viteza liniară și unghiul de direcție a vehiculului autonom.

Am prezentat cazul general al conducerii backstepping. În continuare am calculat comanda pentru conducerea backstepping în cazul WMR utilizând modelul cinematic al robotului și dinamica erorilor de urmărire a robotului. Conducerea backstepping a WMR a fost publicată în [15].

Am calculat comenzile pentru viteza liniară și unghiul de direcție al vehiculului autonom SEEKUR utilizând modelul cinematic și dinamica de urmărire a erorilor și procedura de conducere backstepping.

6.1.3 Evitarea obstacolelor

O soluție pentru problema evitării obstacolelor în cazul în care se dorește urmărirea unei traiectorii impuse și abaterea de la această traiectorie doar pentru a evita un obstacol și întoarcerea la traiectoria impusă este propusă. Am considerat cazul în care robotul este echipat cu sonare și cazul în care vehiculul este echipat cu senzori laser.

Detecția obstacolelor folosește o bulă de sensibilitate pentru a determina dacă un obstacol trebuie evitat. Atât sonarele cât și laserele au permis definirea unei bule de sensibilitate care acoperă 180° . Este folosită aceeași metodă de conducere în cazul în care nu există obstacole și în cazul în care un obstacol este prezent. Cât timp nu este detectat nici un

obstacol care blochează traiectoria dorită platforma urmărește traiectoria impusă folosind metoda de conducere aleasă din cele 3 prezentate în capitolul anterior.

Dacă un obstacol a fost detectat se caută coordonatele care să permită conducerea în siguranță a platformei și se generează o traiectorie bazată pe Quintic equations. Nu se dorește identificarea formei obstacolului ci doar o zonă în care acesta poate fi evitat. Această traiectorie este impusă modulului de conducere. Se continuă deplasarea pe această traiectorie până când obstacolul a fost evitat. După ce obstacolul a fost evitat se calculează traiectoria pentru a reveni la traiectoria impusă și modul de conducere urmărește această traiectorie apoi se revine la traiectoria impusă inițială.

6.1.4 Testarea algoritmilor propuși

Problema conducerii prezentată în capitolul 3 și problema evitării obstacolelor prezentată în capitolul 4 sunt testate în capitolul 5 cu ajutorul simularilor și testelor în timp real. Am realizat simulările în MobileSim, un simulator oferit de Mobile Robots care permite simularea comportamentului tuturor roboților comercializați de Mobile Robots și a senzorilor laser sau a sonarelor. MobileSim primește comenzile generate de ARIA, iar utilizatorul trebuie să folosească funcțiile implementate în ARIA pentru conducerea robotului.

Pentru a simula evitarea obstacolelor a fost nevoie să creez hărți în care să fie simulate obstacolele dorite. Toate hărțile folosite au fost create în Mapper3Basic. În cazul testelor în timp real am rulat programul pe un PC cu frecvența de eșantionare de 100ms și am folosit o rețea wireless pentru a comunica cu robotul. Pentru că robotul nu are o conexiune Ethernet a trebuit să folosesc un Universal Data Server de la Lantronix care a realizat conversia de la comunicația RS232 a robotului și comunicația Ethernet a routerului cu ajutorul căruia am implementat rețeaua wireless.

Simulările și testele efectuate au dovedit o bună urmărire a traiectoriei dorite, iar erorile au fost mici, astfel controllerele fiind validate.

6.2. Direcții de cercetare deschise

O direcție de cercetare poate fi implementarea în timp real a algoritmului de evitare a obstacolelor utilizând vehiculul autonom SEEKUR.

O altă direcție de cercetare poate fi conducerea vehiculelor autonome utilizând atitudinea furnizată de IMU, în timp real și dezvoltarea unui filtru Kalman care să furnizeze și odometria vehiculului, corectată cu date de la encodere sau GPS.

Bibliografie

- [1] Arambula Cosio F. and Padilla Castaneda M.A., Autonomous robot navigation using adaptive potential fields, *Mathematical and Computer Modelling*, vol. 40, pag. 1141-1156, 2004.
- [2] Borenstein J., Koren Y., Real-time obstacle avoidance for fast mobile robots, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, nr. 5, pag. 1179-1187, 1989.
- [3] Borenstein J., Koren Y., The vector field histogram-fast obstacle avoidance for mobile robots, *IEEE Journal of Robotics and Automation*, pag. 278-288, vol. 7, 1991.
- [4] Brockett R., Control theory and singular Riemannian geometry, *New Directions in Applied Mathematics*, Springer-Verlag, pag. 11-27, 1981.
- [5] Chen Chieh, Backstepping control design and its applications to vehicle lateral control in automated highway systems, University of California at Berkeley, 1996.
- [6] Choukroun D., Novel methods for attitude determination using vector observations for attitude determination, PhD Thesis, Haifa, Mai, 2003.
- [7] Chu-qing C., Lian-zheng G. and Rui-feng L., Mobile robots target tracking using finite-time convergence sliding mode controller, *2010 8th IEEE International Conference on Control and Automation*, Xiamen, China, June 9-11, pag. 460-464, 2010.
- [8] Chwa D., Seo J. H., Kim P., and Choi J. Y., Sliding mode tracking control of nonholonomic wheeled mobile robots, *Proceedings of the American Control Conference*, Anchorage, pag. 3991-3996, mai, 2002.
- [9] Chwa D., Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates, *IEEE Transactions on Control Systems Technology*, vol. 12, nr. 4, iulie, pag. 637-644, 2004.
- [10] Corradini M. L., Leo T., Orlando G., Variable structure control of robotic assistance system, *Proceedings of the 5th IEEE Mediterranean Conference on Control and Systems*, <http://med.ee.nd.edu/MED5/PAPERS/052/052.PDF>.
- [11] Danwei W. and Feng Qi, Trajectory planning for a four-wheel steering vehicle, *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, pag. 3320-3325, 2001.
- [12] Dumitrascu B. and Filipescu A., Discrete-time sliding-mode controller for wheeled mobile robots, *Proceedings of the 18th International Conference on Control Systems and Computer Science*, vol. 1, Bucuresti, pag. 397-403, mai, 2011.
- [13] Dumitrascu B. and Filipescu A., Sliding mode control of lateral motion for four driving-steering wheels autonomous vehicle, *Annals of the University of Craiova*, vol. 7 (34), pag. 20-25, 2010.

- [14] Dumitrascu B. and Filipescu A., Sliding mode controller for steering of four-wheels driving and steering vehicle, *Proceedings of the 14th International Conference on System Theory and Control*, pag. 202-206, Sinaia, 2010.
- [15] Dumitrascu B., Filipescu A., Backstepping control of wheeled mobile robots, *Proceedings of the 15th International Conference on System Theory and Control*, Sinaia, pag. 206-211, 2011.
- [16] Dumitrascu B., Filipescu A., Radaschin A., Filipescu A. Jr., Minca E., Discrete-time sliding mode control of wheeled mobile robots, *Proceedings of 2011 8th Asian Control Conference (ASCC)*, Kaohiung, Taiwan, pag. 771-776, mai, 2011.
- [17] Dumitrascu B., Filipescu A., Vasilache C., Minca E., Filipescu A. Jr., Discrete-time sliding-mode control of four driving/steering wheels mobile platform, *Proceedings of the 19th Mediterranean Conference on Control and Automation*, Corfu, Grecia, iunie, 2011, pag. 1076-1081.
- [18] Dumitrascu B., Filipescu A., Minzu V., Voda A., Minca E., Discrete-Time Sliding-Mode Control of Four Driving-Steering Wheels Autonomous Vehicle, *Proceedings of the 30th Chinese Control Conference*, pag. 3620-3625, 2011.
- [19] El-Sheimy N., Hou H., Niu Xiaoji, Analysis and modelling of inertial sensors using Allan variance, *IEEE Transactions on Instrumentation and Measurement*, vol. 57, nr. 1, pag. 140-149, ianuarie, 2008.
- [20] Filipescu A., Minzu V., Dumitrascu B. and Filipescu A., Trajectory-tracking and discrete-time sliding-mode control of wheeled mobile robots, *The 2011 IEEE International Conference on Information and Automation*, Shenzhen, pag. 27-32, iunie, 2011.
- [21] Flenniken W.S., Wall, J. H., Bevely D.M., Characterization of various IMU error sources and the effect on navigation performance, http://www.eng.auburn.edu/~dmbevly/gavlab/pub-pre/Flenniken_ION_GNSS_2005.pdf.
- [22] Gao W. and Hung J. C., Variable structure control of nonlinear systems: A new approach, *IEEE Transactions on Industrial Electronics*, 40(1), pag. 45 - 55, 1993.
- [23] Gao W., Wang Y. and Homaifa A., "Discrete-time variable structure control systems", *IEEE Transaction on Industrial Electronics*, vol. 42, 117-122, 1995.
- [24] Hou H., Modeling inertial sensors errors using Allan variance. Phd Thesis, University of Calgary, 2004.
- [25] <http://www.cs.iastate.edu/~cs577/handouts/quaternion.pdf>
- [26] Hung J. Y., Gao W., Hung J. C., Variable structure control: A survey, *IEEE Transactions on Industrial Electronics*, vol. 40(1), pag. 2-22, 1993.

- [27] IEEE standard, Specification format guide and test procedure for single-axis interferometric fiber optic gyros, *IEEE-SA Standards Board*.
- [28] Jalili-Kharaajoo M. and Moshiri B., Discrete-time sliding mode control for linear time-varying systems with application to steering control on highway vehicles, University of Tehran, Iran.
- [29] Jiang J.P. and Nijmeijer H., Backstepping-based tracking control of nonholonomic chained systems, *European Control Conference*, Brussels, iulie, 1997.
- [30] Jiang J.P. and Nijmeijer H., Tracking control of mobile robots: a case study in backstepping, *Automatica*, vol. 33, nr. 7, pag. 1393-1399, 1997.
- [31] Joan Sola, Quaternion Kinematics for the error-state Kalman filter, <http://www.joansola.eu/JoanSola/objectes/notes/kinematics.pdf>, august, 2012.
- [32] Joao Luis Marins, Xiaoping Yun, Eric R. Bachman, Robert B. McGhee, Michael J. Zyda, An extended Kalman filter for quaternion-based orientation estimation using MARG sensors, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, USA, octombrie, 2001.
- [33] Kanarat A., Motion Planning and robust control for nonholonomic mobile robots under uncertainties, PhD Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2004.
- [34] Kanayama Y., Kimura Y., Miyazaki F. and Noguchi T., A stable tracking control scheme for an autonomous mobile robot, *Proceedings of the IEEE International Conference on Robotics and Automation*, pag. 384-389, 1990.
- [35] Kimberly T., Tilt sensing using linear accelerometers, Freescale Semiconductors Application Note, 2007.
- [36] Lee H., Chattering suppression in sliding mode control system, PhD Thesis M. S. Ohio State University, 2007.
- [37] Lee J. H, Lin C., Lim H. and Jang Myung Lee, Sliding mode control of mobile robot in the RFID sensor space, *International Journal of Control, Automation and Systems*, pag 429-435, 2009.
- [38] Lucet E. , Grand C., Salle D., Bidaud P., Dynamic sliding mode control of a four-wheel skid-steering vehicle in presence of sliding, http://www.doc-center.robosoft.com/@api/deki/files/3002/=Dynamic_sliding_mode_control_of_a_four-wheel_skid-steering_vehicle_in_presence_of_sliding.pdf.
- [39] Lumelsky V., Skewis T., Incorporating range sensing in the robot navigation function, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, nr. 5, pag. 1058-1068, 1990.

- [40] Nassar S., Improving the inertial navigation system (INS) error model for INS and INS/DGPS applications, PhD Thesis, The University of Calgary, 2003.
- [41] Philips W. F., Hailey C.E., and Gebert G.A., Review of attitude representations used for aircraft kinematics, *Journal of Aircraft*, 38(4), pag. 718-737, 2001.
- [42] Solea R., Filipeascu A., Stamatescu G., Trajectory-tracking sliding-mode control for wheeled mobile robots, *Energy, Transport and Environment Control Applications-International Workshop*, pag 121-134, mai, 2009.
- [43] Solea R., Filipescu A. and Stamatescu G., Sliding-mode real-time mobile platform control in the presence of uncertainties, *Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai, pag. 7747-7752, 2009.
- [44] Solea R., Filipescu A., Filipescu S, and Dumitrascu B., Sliding-mode controller for four-wheel-steering vehicle: trajectory-tracking problem, *Proceedings of the 8th World Congress on Intelligent Control and Automation*, Jinan, China, pag. 1185-1190, 2010.
- [45] Solea R., Filipescu A., Manzu V., Filipescu S., Sliding-mode trajectory-tracking control of a four wheel steering vehicle, *2010 8th IEEE International Conference on Control and Automation*, Xiamen, China, pag. 382-387, 2010.
- [46] Solea R., Filipescu A., Nunes U., Sliding-mode control for trajectory-tracking of a wheeled mobile robot in presence of uncertainties, *Proceedings of the 7th Asian Control Conference*, Hong Kong, pag. 1701-1706, 2009.
- [47] Solea R., Sliding mode control applied in trajectory-tracking of WMRs and autonomous vehicles, PhD Thesis, University of Coimbra, Portugal, 2009.
- [48] Susnea I., Filipescu A., Vasiliu G., Coman G., Radaschin A., "The bubble rebound obstacle avoidance algorithm for mobile robots", in *Proceedings. 8th IEEE International Conference on Control and Automation*, Xiamen, pag. 540-545, 2010.
- [49] Utkin V.I., Guldner J., and Shi J., Sliding mode control in electromechanical systems, *Taylor & Francis*, London, 1999.
- [50] Utkin V.I., Sliding modes in optimization and control, *Springer-Verlag*, New York, 1992.