

METODOLOGIE DE LOCALIZARE RELATIVĂ ÎN MEDII COLABORATIVE ROBOTIZATE

Teză destinată obținerii
titlului științific de doctor inginer
la
Universitatea Politehnica Timișoara
în domeniul
CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
de către

Ing. Andrei Stancovici

Conducător științific: prof. univ. emerit. dr. ing. Vladimir I. Crețu
Referenți științifici: prof. univ. dr. ing. Sergiu Nedevschi
prof. univ. dr. ing. Nicolae Țăpuș
prof. univ. dr. habil. ing. Mihai V. Micea

Ziua susținerii tezei: 09.11.2018

Seriile Teze de doctorat ale UPT sunt:

- | | |
|---|--|
| 1. Automatică | 10. Știința Calculatoarelor |
| 2. Chimie | 11. Știința și Ingineria Materialelor |
| 3. Energetică | 12. Ingineria sistemelor |
| 4. Ingineria Chimică | 13. Inginerie energetică |
| 5. Inginerie Civilă | 14. Calculatoare și tehnologia informației |
| 6. Inginerie Electrică | 15. Ingineria materialelor |
| 7. Inginerie Electronică și Telecomunicații | 16. Inginerie și Management |
| 8. Inginerie Industrială | 17. Arhitectură |
| 9. Inginerie Mecanică | 18. Inginerie civilă și instalații |

Universitatea Politehnica din Timișoara a inițiat seriile de mai sus în scopul diseminării expertizei, cunoștințelor și rezultatelor cercetărilor întreprinse în cadrul școlii doctorale a universității. Seriile conțin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susținute în universitate începând cu 1 octombrie 2006.

Copyright © Editura Politehnica – Timișoara, 2018

Această publicație este supusă prevederilor legii dreptului de autor. Multiplicarea acestei publicații, în mod integral sau în parte, traducerea, tipărirea, reutilizarea ilustrațiilor, expunerea, radiodifuzarea, reproducerea pe microfilme sau în orice altă formă este permisă numai cu respectarea prevederilor Legii române a dreptului de autor în vigoare și permisiunea pentru utilizare obținută în scris din partea Universității Politehnica din Timișoara. Toate încălcările acestor drepturi vor fi penalizate potrivit Legii române a drepturilor de autor.

România, 300159 Timișoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Lucrarea de față reprezintă un moment important din viața mea, fapt pentru care doresc înainte de toate să arunc o privire retrospectivă peste anii de studiu ce s-au scurs și să-mi amintesc de cei care m-au însoțit, m-au încurajat și m-au ajutat să ajung la acest popas.

Un dicton latin spune: NIHIL SINE DEO! (nimic fără Dumnezeu!), fapt pentru care mulțumesc mai întâi lui Dumnezeu care mi-a luminat mintea, mi-a întărit voința și mi-a încălzit inima ca să fac cu drag ceea ce fac.

Doresc să aduc aici un buchet de recunoștință și alese sentimente de prețuire profesorilor mei fără de care nu aș fi reușit să fac mare lucru. Ei sunt cei care m-au ajutat, m-au încurajat și mi-au condus pașii în știința calculatoarelor. Teza mea de doctorat este rodul colaborării mele strânse cu mentorul meu prof. emerit. dr. ing. Vladimir I. Crețu care m-a îndrumat în direcția cea bună a cercetării în domeniu. Mulțumiri deosebite datorez celui care a propus tema cercetării de față, prof. dr. habil. ing. Mihai V. Micea, cel care m-a primit în echipa de cercetare a laboratorului DSPLabs și mi-a oferit șansa și condiții favorabile pentru cercetare și experimentare în domeniul roboților colaborativi autonomi. Îndrumările și instrucțiunile sale m-au ajutat să înțeleg mai bine problematica subiectului pe care mi l-a propus ca să-l tratez. Mulțumiri datorez întregii echipe de la DSPLabs (îndeosebi as. ing. Dan Chiciudean, ș.l. dr. ing. Răzvan Cioargă, apoi și colegilor ș.l. dr. ing. Valentin Stângaciu, ș.l. dr. ing. Cristina Stângaciu, ing. Ramona Cioargă, ing. Sînziana Indreica, ing. Gabriel Cârstoiu, ing. Lucian Ungurean), pentru cunoștințele împărtășite, pentru suport și pentru deschiderea și prietenia lor sinceră și binevoitoare. Mulțumesc domnilor prof. dr. ing. Mircea Stratulat, prof. dr. ing. Ionel Jian, prof. dr. habil. ing. Marius Marcu, ș.l. dr. ing. Sebastian Fuicu, și de asemenea tuturor cadrelor didactice de la Universitatea Politehnică pentru experiența lor bogată în domeniu, evidentă prin varietatea cunoștințelor împărtășite la cursuri și consultări, pentru încrederea pe care mi-au arătat-o în timpul studenției a masteratului și în perioada studiilor doctorale. În calitatea lor de formatori și moderatori ai tinerilor informaticieni, domniile lor rămân pentru mine adevărate repere și modele în ascensiunea pe drumul anevoios al cercetării științifice. Admirația mea pentru ei este mare și simt o satisfacție sufletească pentru privilegiu de a fi ucenicul lor. Mă simt împlinit prin faptul de a fi ucenicul unor profesori de prestigiu a Universității Politehnice a orașului de pe Bega, care a devenit primul oraș din Europa cu străzile iluminate electric.

Mă adresez de asemenea cu deosebită recunoștință și mulțumiri tuturor cadrelor didactice de la Liceul Teoretic „Dositei Obradovici” care m-au format. Îndeosebi vreau să mulțumesc învățătoarei Țvetanica Golub, profesorului de matematică Dușan Mișici, și în mod special domnului inspector Jelco Stancov, cel care în calitate de profesor de informatică m-a introdus în tainele informaticii.

Folosesc acest prilej ca să mulțumesc și unui bun prieten al nostru Gabriel-Miro Muntean, profesor la Universitatea din Dublin, de altfel timișorean, care mi-a facilitat participarea la un stagiu de practică la facultatea unde activează, pentru îmbogățirea cunoștințelor mele în domeniu.

Datorită faptului că munca mea în toată această perioadă de studiu doctoral s-a desfășurat în familie, țin să mulțumesc soției mele Maria care m-a secundat pe parcursul realizării acestei teze de doctorat, dar și fiicei mele Anastasia pentru înțelegere creându-mi atmosfera propice pentru studiu. Nu în ultimul rând mulțumesc părinților, socrilor și fraților care m-au încurajat și susținut moral în acest demers.

Acest popas reprezintă consolidarea unei munci și formarea unei baze pentru un început de drum pe tărâmul cercetării științifice pentru aprofundarea și descoperirea unor noi orizonturi în lumea roboților colaborativi autonomi.

Timișoara, noiembrie 2018

Andrei Stancovici

Stancovici, Andrei

Metodologie de localizare relativă în medii colaborative robotizate

Teze de doctorat ale UPT, Seria 14, Nr. 38, Editura Politehnica, 2018, 176 pagini, 67 figuri, 23 tabele.

ISSN: 2069-8216

ISSN-L: 2069-8216

ISBN: 978-606-35-0231-6

Cuvinte cheie: least squares, kalman filter, monte carlo, las vegas, particle filter, line of sight, indoor, autonomous, colaborative, cooperative, positioning, localization, trilateration, multilateration, confidence, omnidirectional, turret, ultrasonic, ultrasound, sonar, odometry, mobile robots

Rezumat,

Lucrarea de față abordează problema localizării în contextul explorării roboților autonomi în medii necunoscute.

În vederea soluționării acestei probleme și din lipsa unui standard care să guverneze cercetarea localizării în mediul interior, autorul propune o metodologie de localizare relativă, destinat sistemelor de roboți autonomi cu rol de cooperare.

Metodologia este aplicabilă pentru o gamă largă de sisteme. În lucrarea de față s-a luat ca exemplu sistemul CORE-TX (COLlaborative Robotic Environment – the Timisoara eXperiment). Există trei niveluri de localizare pe care le oferă metodologia: PREDICȚIE, COOPERARE și CENTRALIZARE.

Pentru nivelul COOPERARE se propune un model matematic bazat pe metoda ILS (Iterative Least Squares) care poate rezolva problema de localizare fără a fi nevoie de un echipament de calcul central. Nivelul CENTRALIZARE oferă o confiență superioară în localizare având la bază un echipament central cu multă putere de calcul și un algoritm destinat sistemelor cu procesare distribuită. Algoritmul BPF (Backtracking Particle Filter) propus, cu o tehnică originală în domeniul localizării, este un algoritm de tip „filtru de particule” sub forma de „backtracking” și care se bazează pe modelul algoritmilor probabiliști Las Vegas.

A fost dezvoltat un dispozitiv hardware de localizare, IRULT (Inter-Robot Ultrasonic Localization Turret), cu un concept original, care este poziționat pe o turelă și este echipat cu două traductoare ultrasonice cu rolul de a obține orientarea și poziția robotului în task-ul de navigare.

Deoarece experimentele făcute în practică acoperă un caz prea izolat pentru a putea valida metodologia propusă, se execută mai multe seturi de experimente prin simulare pentru a cuprinde toate cazurile izolate ce pot apărea în practică.

Prin analiza rezultatelor experimentale și prin introducerea conceptului „bază” cu rolul de a reseta propagarea erorilor în localizare, se arată că metodologia de localizare relativă propusă este aplicabilă în contextul explorării roboților autonomi în medii necunoscute.

CUPRINS

Lista de abrevieri	8
Lista de tabele	10
Lista de figuri.....	11
1. Introducere	13
1.1 Tema tezei.....	13
1.2 Actualitatea tezei și aria de aplicabilitate	13
1.3 Motivațiile tezei	15
1.4 Scopul și principalele obiective	16
1.5 Structura tezei	17
2. Localizarea, abordări curente	19
2.1 Clasificarea sistemelor de localizare	19
2.2 Sisteme de localizare	20
2.2.1 CAT I - Sistemul GPS.....	20
2.2.2 CAT I - Sistemul GLONASS	22
2.2.3 CAT II - Localizarea în telefonie mobilă	23
2.2.4 CAT II - Cricket Indoor Location System și sisteme asemănătoare.....	24
2.2.5 CAT III - Sisteme bazate pe camere omnidirecționale	26
2.2.6 CAT IV - PhotoBeacon (sistem optic de localizare)	27
2.2.7 CAT IV - Localizarea prin trilaterare pentru echipă muti-robot	29
2.2.8 CAT V - Localizare folosind senzori de la mouse-ul optic.....	31
2.3 Tehnici în localizare.....	32
2.3.1 TOF (Time Of Flight), TOA (Time Of Arrival)	32
2.3.2 TDOA (Time Diference Of Arrival)	32
2.3.3 AOA (Angle Of Arrival)	33
2.3.4 RTT (Round Trip Time), RTD (Round Trip Delay Time)	33
2.3.5 PDOA (Phase Difference Of Arrival)	33
2.3.6 Corelația semnalului	35
2.3.7 RSSI (Indicatorul de nivel a semnalului recepționat).....	36
2.4 Tehnologii în localizare	37
2.4.1 WLAN (Wireless Local Area Network)	37
2.4.2 CSS (Chirp Spread Spectrum).....	37
2.4.3 UWB (Ultra-WideBand)	38
2.4.4 Zigbee	38
2.4.5 Ultrasunete	39
2.4.6 IR (Infrared).....	39
2.4 Concluzii.....	40
3. Sistemul hardware de localizare propus	43
3.1 Introducere.....	43

3.2	Problemele utilizării SLR actuale bazate pe ultrasunete.....	44
3.2.1	Module cu un emițător de ultrasunete și un set de mai multe receptoare	44
3.2.2	Module cu un traductor și un con acustic reflector	44
3.2.3	Module cu un emițător de ultrasunete și două receptoare	44
3.2.4	Alte module ultrasonice.....	45
3.3	Proiectarea dispozitivului pentru orientarea și localizarea roboților mobili....	46
3.3.1	IRULT (Inter-Robot Ultrasonic Localization Turret) ca și parte a sistemului CORE-TX	46
3.3.2	Proiectarea dispozitivului IRULT	48
3.3.3	Problema atenuării semnalului	49
3.3.4	Problema alinierii	50
3.3.5	Performanța de aliniere.....	52
3.3.6	Reper pentru aliniere	52
3.4	Algoritmul de aliniere a roboților.....	53
3.5	Măsurarea distanței folosind metoda MTDOA	54
3.6	Măsurarea distanței folosind metoda CTOF	54
3.7	Rezultate experimentale	56
3.8	Concluzii.....	60
4.	Metodologia de localizare propusă	62
4.1	Introducere.....	62
4.2	Definirea unor concepte de bază.....	63
4.3	Măsurarea unghiului.....	65
4.4	Inițializarea sistemului	66
4.5	Orientarea și localizarea folosind metoda LOAD	67
4.6	Localizarea bilaterală	70
4.7	Rezultate experimentale	71
4.8	Concluzii.....	75
5.	Aplicarea metodologiei de localizare relativă pentru stratul de percepție și operare CORE-TX	76
5.1	Introducere.....	76
5.2	Nivelul I de localizare - PREDICȚIE.....	77
5.2.1	Gestionarea mobilității	77
5.2.2	Filtre Kalman	77
5.3	Tehnici și metode de localizare relativă.....	79
5.3.1	Trilaterația	79
5.3.2	Localizarea multilaterală	81
5.3.3	Metoda liniară pentru cele mai mici pătrate	82
5.3.4	ILS (Iterative Least Squares).....	82
5.3.5	AILS (Adaptive Iterative Least Squares).....	83
5.4	Nivelul II de localizare - COOPERARE	85
5.4.1	Tehnici și reguli.....	85
5.4.2	Modelul matematic	86
5.5	Simulatorul SiMuLoC	90
5.6	Exemplificarea metodologiei pe un studiu de caz	92
5.7	Reguli de formare a pânzelor	95
5.7.1	Regula PRP (poziționare PeRpendiculară)	96
5.7.2	Regula DST (DiSTanță mare).....	97
5.7.3	Regula SRT (drum ScuRT).....	99

5.7.4 Regula PNR (configurație PlaNaRă)	101
5.8 Concluzii.....	106
6. Aplicarea metodologiei de localizare relativă pentru stratul de control și supraveghere CORE-TX	107
6.1 Introducere.....	107
6.2 Filtre de particule	108
6.3 Nivelul III de localizare - CENTRALIZARE	110
6.3.1 Algoritmi de calcul.....	110
6.3.2 Algoritmul BPF (Backtracking Particle Filter)	111
6.4 Simulatorul Simon-TX	118
6.5 Exemplificarea metodologiei pe un studiu de caz	119
6.5.1 Experiment de bază.....	119
6.5.2 Experiment cu metoda incrementului.....	122
6.5.3 Experiment cu metoda soluțiilor multiple.....	125
6.5.4 Experiment cu metoda incrementului și metoda soluțiilor multiple	126
6.5.5 Experiment cu variația toleranței maxime la măsurarea distanței	128
6.6 Localizare relativă bazată pe măsurarea unghiului și distanței între roboți.	129
6.6.1 Sisteme bazate pe măsurarea unghiului între roboți	129
6.6.2 Sisteme bazate pe măsurarea distanței între roboți	130
6.6.3 Sisteme bazate pe măsurarea unghiului și distanței între roboți	130
6.6.4 Algoritmul BPF pentru măsurarea unghiului și distanței între roboți ...	131
6.6.5 Experiment cu variația toleranței maxime la măsurarea unghiului	143
6.7 Localizarea relativă prin construirea bazelor.....	147
6.8 Concluzii.....	155
7. Concluzii și perspective	156
7.1 Concluzii.....	156
7.2 Sinteza contribuțiilor	160
7.3 Perspective de cercetare și dezvoltare	161
Referințe bibliografice	163
Publicații.....	173

LISTA DE ABREVIERI

- 2MARS**– Sisteme mobile multi robot autonome (engleză: Multi Mobile Autonomous Robotic Systems)
- AILS**– Metoda iterativă pentru cele mai mici pătrate adaptată (engleză: Adaptive Iterative Least Squares)
- AOA**– Unghiul de sosire, din engleză: Angle of Arrival
- AP**– Din engleză: Access Point (se referă la WLAN)
- BEMF**– Din engleză: back electromotive force (back EMF)
- BPF**– Din engleză: Backtracking Particle Filter
- BPS**– Din engleză: Building Positioning System
- BRAIN**– Entitate de inducție a activității roboților mobili (engleză: Background Robotic Activity Induction Node)
- BTS**– Din engleză: Base Transceiver Station
- CID**– Din engleză: Cell ID, reprezintă un ID unic pentru identificarea fiecărui BTS
- CORE-TX**– Mediu colaborativ robotizat - Experimentul Timișoara (engleză: COllaborative Robotic Environment - the Timisoara eXperiment)
- CSS**– Din engleză: Chirp Spread Spectrum
- CTOF**– Durata de propagare combinată (engleză: Combined Time of Flight)
- EKF**– Filtru Kalman extins (engleză: Extended Kalman Filter)
- FCC**– comisia de comunicare federală (engleză: Federal Communications Commission)
- FD-PDOA**– Din engleză: Frequency Domain Phase difference of arrival
- FMCW**– Din engleză: Frequency-Modulated Continuous-Wave Radar
- GLONASS**– Din rusă: ГЛОНАСС - ГЛОбальная НАвигационная Спутниковая Система
- GNSS**– Din engleză: Global Navigation Satellite System
- GPS**– Din engleză: Global Positioning System
- I-UWB**– Din engleză: Impulse Ultra-wideband
- ILS**– Metoda iterativă pentru cele mai mici pătrate (engleză: Iterative Least Squares)
- IoT**– Din engleză: Internet of Things
- IR**– Infraroșu (engleză: Infrared)
- IRULT**– Turelă ultrasonică de localizare între - roboți (engleză: Inter-Robot Ultrasonic Localization Turret)
- ISR**– Roboți de supraveghere inteligenți (engleză: Intelligent Security Robots)
- LIDAR**– Din engleză: LIght Detection And Ranging
- LKF**– Din engleză: Linear Kalman Filter
- LOAD**– Localizare folosind un singur unghi și o singură distanță (engleză: Localization using One Angle and One Distance)
- LOS**– Din engleză: Line of Sight
- LR-WPAN**– Din engleză: Low Rate Wireless Personal Area Network
- MAC**– Din engleză: Medium Access Control
- MC-UWB**– UWB multi-purtător (engleză: MultiCarrier Ultra-WideBand)
- MCR**– Medii colaborative robotizate
- MEO**– Din engleză: Medium Earth Orbit
- MS**– Stația mobilă (engleză: Mobile Station)

MTDOA– Diferența de timp de semnal de sosire modificată (engleză: Modified Time Difference of Arrival)
NLOS– Din Engleză: Non Line of Sight
OFDM– multiplexarea cu diviziune în frecvență ortogonală (engleză: Orthogonal Frequency Division Multiplexing)
PDF– Din Engleză: Probability Density Function
PDOA– Din Engleză: Phase difference of arrival
PHY– Nivelul fizic, din Engleză: Physical layer
PNT– Din Engleză: Position, Navigation and Timing
RBPF– Filtru de particule Rao-Blackwellised (engleză: Rao-Blackwellised Particle Filter)
RDS– Din Engleză: Resource Discovery Services
RF– radio frecvență, din Engleză: Radio Frequency
RFID– Identificare prin frecvență radio, din Engleză: Radio-Frequency Identification
RSSI– Indicatorul de nivel a semnalului recepționat (engleză: Received Signal Strength Indicator)
RTD– Din Engleză: Round Trip Delay Time
RTT– Din Engleză: Round Trip Time
SD-PDOA– Din engleză: Spatial Domain Phase Difference Of Arrival
SDTWR– Din engleză: Symmetric Double Sided Two-Way Ranging
SLAM– Localizare și construire de hartă în mod simultan (engleză: Simultaneous Localization and Map Building)
SPS– Din engleză: Standard Positioning Service
SLR– Sistem de localizare relativă
TDOA– Din engleză: Time Diference Of Arrival
TD-PDOA– Din Engleză: Time Domain Phase Difference Of Arrival
TOA– Timpul de propagare a unui semnal de la un punct la altul (engleză: Time Of Arrival)
TOF– Din engleză: Time Of Flight
UWB– Din engleză: Ultra-WideBand
WiFi– Este numele comercial pentru WLAN
WIT– Terminal inteligent fără fir (engleză: Wireless Intelligent Terminal)
WLAN– Din Engleză: Wireless Local Area Network
WSN– Rețea de senzori fără fir (engleză: Wireless Sensor Network)

LISTA DE TABELE

Tabelul 2-1 Tipuri de celule [47]	24
Tabelul 3-1 Măsurarea distanței aplicând praguri diferite de tensine [96]	50
Tabelul 3-2 Rezultatele măsurării distanței pentru metoda MTDOA [104]	57
Tabelul 3-3 Rezultatele măsurării distanței pentru metoda CTOF [104]	57
Tabelul 4-1 Măsurătorile cu metoda LOAD [101]	74
Tabelul 4-2 Măsurătorile cu metoda de localizare bilaterală [101]	74
Tabelul 5-1 Simulare - metoda ILS.....	95
Tabelul 5-2 Regula PRP (poziționare PeRpendiculară)	97
Tabelul 5-3 Regula DST (DiSTanță mare).....	99
Tabelul 5-4 Regula SRT (drum ScuRT).....	101
Tabelul 5-5 Configurație coliniară.....	103
Tabelul 5-6 Regula PNR (configurație PlaNaRă).....	105
Tabelul 6-1 Metoda BPF-B100K (CONF2).....	121
Tabelul 6-2 Metoda BPF-b100n10i10s1 (CONF2).....	124
Tabelul 6-3 Metoda BPF-b100kn1i1s3 (CONF2).....	126
Tabelul 6-4 Metoda BPF-b100n10i10s3 (CONF2).....	128
Tabelul 6-5 Determinarea unghiurilor – pas cu pas.....	134
Tabelul 6-6 Metoda BPF-1grad (CONF2).....	144
Tabelul 6-7 Metoda BPF-10minute (CONF2)	146
Tabelul 6-8 Atenuarea prin materialele de construcție la 1,5 GHz [141]	148
Tabelul 6-9 Metoda BPF (fără UWB)	150
Tabelul 6-10 Metoda BPF (doar UWB).....	152
Tabelul 6-11 Metoda BPF (combinat)	153

LISTA DE FIGURI

Figura 2-1 Clasificarea sistemelor de localizare	19
Figura 2-2 Constelație de sateliți expansibilă, așa cum este definită în standardul de performanță SPS (Standard Positioning Service) [43]	21
Figura 2-3 Sateliții: Glonass(1982-2005), Glonass-M(2003-2016), Glonass-K(2011-2018) și Glonass-K2(2017-) [44]	23
Figura 2-4 Acoperirea cu celule a teritoriului [46].....	24
Figura 2-5 Sistemul de localizare Cricket [48]	25
Figura 2-6 Exemplu de imagine obținută prin sistemul Omnicam [57].....	27
Figura 2-7 Sistemul optic de localizare PhotoBeacon [58].....	28
Figura 2-8 Localizarea roboților 1 și 2 față de roboții A și B [58]	28
Figura 2-9 Conul parabolic [59]	30
Figura 2-10 Modelul de trilateratie [59]	30
Figura 2-11 Senzorul optic al mouse-ului echipat cu o lentilă suplimentară [62]....	31
Figura 2-12 Tehnica TD-PDOA [68]	34
Figura 2-13 Tehnica FD-PDOA [68]	34
Figura 2-14 Tehnica SD-PDOA [68]	35
Figura 2-15 Semnalul recepționat compus [70]	36
Figura 3-1 Sistemul CORE-TX [16].....	46
Figura 3-2 Modulul IRULT ca și parte a modului de percepție [96].....	47
Figura 3-3 IRULT, privire de sus [97]	47
Figura 3-4 Schema dispozitivului IRULT [96]	48
Figura 3-5 Obținerea timpului TOF aplicând praguri diferite de tensiune [96].....	49
Figura 3-6 Alinierea roboților, privire de sus [98]	51
Figura 3-7 Efect combinat al unghiurilor, rezultat experimental [96].....	51
Figura 3-8 Procedura de aliniere: timpul de aliniere în raport cu unghiurile φ_1 și φ_2 [101].....	52
Figura 3-9 Algoritmul de aliniere [97].....	53
Figura 3-10 Calculul distanței cu metoda CTOF [97].....	55
Figura 3-11 Semnalul ultrasonic recepționat [104].....	56
Figura 3-12 Distanța măsurată respectiv distanță reală și durata procedurii pentru metoda MTDOA [104]	58
Figura 3-13 Distanța măsurată respectiv distanță reală și durata procedurii pentru metoda CTOF [104].....	59
Figura 3-14 Compararea erorilor maxime (absolute și relative) pentru tehnicile MTDOA și CTOF [104]	60
Figura 4-1 Modelul CORE-TX [98]	63
Figura 4-2 Zona roșie, galbenă și verde	64
Figura 4-3 Măsurarea unghiului	66
Figura 4-4 Inițializarea sistemului	67
Figura 4-5 Localizarea folosind metoda LOAD	69
Figura 4-6 Metoda de localizare bilaterală	71
Figura 4-7 Mediu robotic – studiu de caz.....	72

Figura 4-8 Exemplu de un sistem robotic format din 7 roboți [98].....	73
Figura 5-1 Modelul teoretic al trilaterăției.....	79
Figura 5-2 Inconsistențele pozițiilor și distanțelor generând un sistem cu un număr infinit de soluții	80
Figura 5-3 Valoare reziduală ca și sumă de diferențe de pătrate între distanțele estimate și cele calculate.....	81
Figura 5-4 Localizarea multilaterală se bazează pe mai mult de trei repere și necesită rezolvarea unui sistem supradeterminat	81
Figura 5-5 Exemplu de minim local [119].....	85
Figura 5-6 Nivelul II de localizare – COOPERARE 2:2 [121]	87
Figura 5-7 Interfața simulatorului SiMuLoC	91
Figura 5-8 a) Cerințele pentru supraveghere b) Configurarea sistemului [121]	93
Figura 5-9 Simulare - metoda ILS	94
Figura 5-10 Regula PRP (poziționare PeRpendiculară)	96
Figura 5-11 Regula DST (DiSTanță mare)	98
Figura 5-12 Regula SRT (drum ScuRT)	100
Figura 5-13 Configurație coliniară	102
Figura 5-14 Regula PNR (configurație PlaNaRă)	104
Figura 6-1 Aplicarea filtrului de particule pentru estimarea locației [114]	109
Figura 6-2 Reprezentare vizuală a unui pas al algoritmului BPF.....	112
Figura 6-3 Schema bloc a algoritmului BPF.....	113
Figura 6-4 Schema logică a unui proces copil	116
Figura 6-5 Interfața simulatorului Simon-TX.....	118
Figura 6-6 Simulare - metoda BPF-B100K (CONF2)	120
Figura 6-7 Simulare - metoda BPF-b100n10i10s1 (CONF2)	123
Figura 6-8 Simulare - metoda BPF-b100kn1i1s3 (CONF2)	125
Figura 6-9 Simulare - metoda BPF-b100n10i10s3 (CONF2)	127
Figura 6-10 Simularea propagării erorilor utilizând metodele ILS și BPF (CONF2) [121].....	129
Figura 6-11 Exemplu pentru elaborarea calculul unghiurilor	133
Figura 6-12 Simulare - metoda BPF-B100K-1grad (CONF2).....	143
Figura 6-13 Simulare - metoda BPF-B100K-10minute (CONF2).....	145
Figura 6-14 Simulare – fără UWB.....	149
Figura 6-15 Simulare – doar UWB.....	152
Figura 6-16 Simulare – UWB și IRULT.....	153

1. INTRODUCERE

„Am muncit ani de zile la conceperea unor mecanisme cu comandă automată și cred că se pot produce mecanisme care să acționeze ca și când ar fi înzestrate cu rațiune, într-o măsură limitată, și care vor revoluționa multe sectoare comerciale și industriale”.

Nikola Tesla

1.1 Tema tezei

Teza de față prezintă contribuții proprii și rezultate din activitatea de cercetare a autorului tezei cu tema „Metodologie de localizare relativă în medii colaborative robotizate”, sub coordonarea științifică a domnului prof. emerit. dr. ing. Vladimir Crețu.

Tema programului de doctorat se referă la problema de localizare în contextul explorării roboților autonomi în medii necunoscute. Ideea acestei cercetări propuse de domnul prof. dr. habil. ing. Mihai Micea vine în urma dezvoltării unor serii de proiecte în domeniul rețelelor de senzori din cadrul Centrului de Cercetare în Calculatoare și Tehnologia Informației al UPT [1]. Multe proiecte se intersectează în aceeași necesitate de a avea un sistem de localizare care să ajute acolo unde nu există acces la resurse externe cu caracter tehnologic care să ofere posibilitatea de a localiza sau măcar de a ajuta în cea mai mică măsură procedura de localizare. Problemele de localizare apărute în proiectele de cercetare din cadrul laboratorului DSPLabs [2] se regăsesc și la alte grupuri de cercetare. Astfel metodologia propusă de autorul tezei este construită luând în considerare necesitățile în dezvoltarea proiectelor din cadrul laboratorului DSPLabs și problemele existente și evidențiate în literatura de specialitate.

În alegerea acestei teme s-a ținut cont de actualitatea domeniului de cercetare, de problemele menționate încă nerezolvate în ceea ce privește localizarea roboților autonomi, de experiența practică și competențele în domeniu, de suportul matematic necesar cât și de posibilitatea de a obține date experimentale folosind baza materială existentă în cadrul laboratorului DSPLabs.

1.2 Actualitatea tezei și aria de aplicabilitate

Dezvoltarea de roboți autonomi este un subiect foarte important și de mare actualitate, datorită avantajelor pe care le aduce. Dintre acestea se pot evidenția: reducerea costurilor în timp, facilitarea activităților umane, îndeplinirea sarcinilor care sunt foarte dificil sau chiar imposibil de efectuat de către o persoană (supravegherea în medii dificile spre exemplu în zone de radiații [3]).

Cunoașterea originii datelor și obținerea locației de unde provin anumite date reprezintă o informație importantă în analiza mulțimilor de date acumulate pe întregul proces de monitorizare a unui mediu [4]. Chiar dacă există progrese importante în această ramură de cercetare (cum ar fi localizarea roboților mobili prin sistemul GPS

[5]), localizarea existentă în literatura de specialitate este inconsistentă în anumite cazuri: atunci când nu există conexiune la sistemul GPS [6], de asemenea când este nevoie de informații odometrice și nu există repere fixe [7].

Pentru a evidenția aria de aplicabilitate a unui sistem de localizare relativă, autorul tezei redă câteva exemple de aplicații în domeniul roboților autonomi:

1) Roboți pentru pază

Securitatea este un subiect foarte important, fie că vorbim despre securitatea clădirilor importante sau securitatea unui individ. Securitatea poate fi îmbunătățită cu ajutorul roboților colaborativi care ar putea să ia diferite măsuri, cu scopul de a proteja persoane sau bunuri. De exemplu, în [8] se propune o arhitectură pentru un sistem inteligent de supraveghere cu roboți mobili autonomi, în care se urmărește să se reducă sarcinile oamenilor, reprezentând un avantaj semnificativ față de sistemele de supraveghere convenționale.

2) Roboți salvatori

În cazul unui dezastru natural, cum ar fi un cutremur, un grup de roboți autonomi ar putea fi utilizați cu scopul de a căuta oameni care ar putea fi prinși sub dărâmături. Activitatea realizată de roboți ar fi mult mai rapidă și mai eficientă decât activitatea echipelor de salvare. În acest tip de aplicație fiecare secundă irosită ar conta pentru viața unei persoane. De exemplu, în [9] se prezintă rezultate din experimente efectuate cu o echipă de roboți mixtă: roboți de sol și un robot aerian, implicate în construirea în mod colaborativ a unei hărți. Harta reprezintă o clădire avariata de cutremur. Scopul exercițiului este de a obține o hartă utilă pentru operațiuni de salvare în urma cutremurului.

3) Roboți subacvatici

După sute de ani, cu scopul de a căuta diferite lucruri (de exemplu nave scufundate sau noi specii de animale subacvatice) sub apă, este nevoie de o implicare directă a forței umane. Datorită faptului că, în căutarea unui vas vechi scufundat de exemplu, sunt necesari mulți ani de căutare, acest lucru ar putea fi efectuat și eficientizat cu ajutorul roboților subacvatici. Un alt exemplu de aplicație unde sunt implicați roboții subacvatici este monitorizarea piscinelor de depozitare a deșeurilor nucleare. Deșeurile radioactive generate de centralele nucleare trebuie să fie depozitate timp de aproximativ 60 de ani în bazine de stocare subacvatice înainte de eliminarea permanentă. În lucrarea [10] se propune o rețea de roboți subacvatici pentru a monitoriza bazinele de depozitare a deșeurilor radioactive. Sistemul este propus pentru a construi hărți ale structurii interne a bazinelor mai vechi, hărți care apoi pot fi utilizate pentru eliminarea deșeurilor și dezafectarea bazinelor.

4) Roboți pentru operațiuni miniere

Un robot care se ghidează cu ajutorul sateliților GPS nu ar putea să fie localizat în peșteri sau în mine. Pe de altă parte, un grup de roboți colaborativi ar putea să funcționeze într-un astfel de mediu, indiferent dacă unul sau mai mulți roboți s-ar defecta, obiectivul s-ar putea atinge cu succes. Acest tip de aplicație este și mai potrivit atunci când vorbim despre peșteri sau mine care conțin urme de elemente radioactive. De exemplu, în [11] se prezintă un sistem robotizat pentru cartografierea minelor abandonate. Sistemul a fost experimentat în trei mine, două dintre acestea

sunt inaccesibile oamenilor. După operațiunea de explorare s-au dobândit hărți cu detalii și acuratețe fără precedent.

5) Roboți de curățenie

Dacă ne-am putea imagina o clădire foarte mare în care curățenia nu mai poate fi întreținută economic, implementarea unui sistem robotic de curățat ar putea fi foarte util. Pentru ca o echipă de roboți să îndeplinească o astfel de sarcină, ar putea fi distribuite sectoarele de curățenie eficient astfel încât sarcina să fie efectuată într-un timp cât mai scurt eficientizând totodată și consumul de resurse. De exemplu în [12] se discută despre un algoritm de partiționare a unei regiuni ce trebuie curățată, în părți mai mici, pentru a fi alocate unui grup mare de roboți. Studiul realizat nu se referă doar la curățenia clădirilor de dimensiuni mici, dimpotrivă, se focusează pe suprafețe mari precum sunt stațiile de tren sau aeroporturile. În [13] este prezentată o abordare nouă pentru navigarea roboților de curățenie într-un spațiu de lucru necunoscut, care include o nouă metodă de reprezentare a hărților și un mecanism de navigare cu acoperire completă. Un alt exemplu de sistem de curățenie testat pentru coridoare lungi se discută în [14]. Sistemul expus este compus din 4 roboți, trei dintre aceștia au rolul de a poziționa pe al patrulea. Al patrulea robot este un robot special construit pentru a face curățenie.

6) Roboți pentru monitorizarea fermelor

Dăunătorii culturilor sunt o problemă foarte importantă pentru proprietarii de ferme. Dăunătorii au un impact important din punct de vedere economic pentru fermieri, datorită faptului că provoacă pagube. O echipă de roboți autonomi ar putea fi soluția pentru a elimina această problemă. În [15] se propune un robot autonom care monitorizează continuu exploatarea, pentru detectarea rapidă a diverselor boli asupra culturilor, care provin de exemplu din gestionarea incorectă a pesticidelor. De asemenea rolul robotului este și de a pulveriza culturile cu o cantitate optimă de pesticide.

În concluzie, luând în considerare faptul că în viitor roboții autonomi vor primi tot mai multe sarcini de realizat în diverse domenii, putem spune că localizarea acestora reprezintă în zilele noastre un subiect de actualitate. De asemenea, deoarece sistemul GPS nu are acuratețea pentru localizarea la scară umană, sau la scara roboților mobili, mai mult, deoarece tehnologia GPS nu funcționează în zone acoperite (acolo unde semnalul GPS este atenuat în totalitate), sau în lipsa acestuia, nu există acces la alte repere active de localizare, sau datele obținute pe baza reperelor naturale sunt dificil de manipulat, putem spune că un sistem de localizare relativă este necesar.

1.3 Motivațiile tezei

Proiectele laboratorului DSPLabs reprezintă o motivație importantă pentru alegerea acestei teme. Unul dintre proiectele care stă la baza acestei cercetări este CORE-TX (Collaborative Robotic Environment – the Timișoara eXperiment).

Proiectul CORE-TX [16] își propune studiul teoretic și aplicativ al mediilor colaborative de micro sisteme timp-real autonome cu inteligență încorporată, precum și utilizarea acestuia în dezvoltarea aplicațiilor de anvergură, cu impact direct asupra

societății informaționale moderne, introducerii tehnologiilor de ultimă oră în industrie și protecția mediului.

O altă motivație o reprezintă problemele existente în localizarea roboților autonomi. Pentru a scoate în evidență problemele existente, autorul tezei redă factorii de influență în localizare:

1) independența de repere fixe: cele mai uzuale sisteme de localizare în cazul roboților autonomi sunt construite să fie dependente de repere fixe;

2) dinamica mediului: dacă apar modificări în mediul în care se utilizează sistemul de localizare, cum ar fi numărul de persoane, prezența diferitelor obiecte (cum ar fi dulapurile, rafturile), localizarea se realizează mult mai dificil;

3) viteza de execuție: depinde foarte mult de algoritmul care se aplică, în unele situații calculele se fac în mod repetat cu scopul de a se obține o acuratețe mai bună;

4) acuratețea: este dată de acuratețea senzorilor și de modul în care sunt folosite informațiile de la senzori, de asemenea, de exemplu, erorile de calcul în cazul măsurării distanțelor, care provin din reflexii, refracții și difracții, pot să ducă la erori mari în localizare;

5) scalabilitatea: într-o rețea de roboți, numărul acestora poate fi de ordinul sutelor sau miilor, ceea ce duce la creșterea complexității protocolului de localizare;

6) conectivitatea: densitatea roboților trebuie să fie destul de mare pentru a nu se pierde legăturile, dar în același timp o densitate mare duce la îngreunarea comunicării din cauza caracteristicii de evitare a coliziunilor a mediului fără fir;

7) eficiența energetică: algoritmi complecși ar putea să aducă un consum considerabil robotului mobil autonom;

8) costul: prețul modulelor adiționale care aduc o îmbunătățire majoră în localizare este foarte ridicat;

9) robustețe: trebuie analizată situația când anumiți senzori se defectează, măsurătorile eșuează, sau, un robot mobil devine complet nefuncțional;

10) standardizare: în prezent, nu există niciun standard care să guverneze cercetarea localizării în mediul interior al clădirilor.

De asemenea o altă motivație o reprezintă aria vastă de aplicabilitate așa cum s-a prezentat în subcapitolul 1.2. menționând pe scurt, monitorizarea mediilor radioactive [3], monitorizarea și optimizarea traficului [17], explorarea mării pentru studiul creaturilor maritime [18][19], misiuni în spațiu pentru cercetarea și descoperirea unor zone încă neexplorate [20], aparate casnice robotizate cum ar fi un aspirator mobil inteligent [21] sau un scaun inteligent cu roți [22][23], roboți pentru distribuirea alimentelor sau robot de companie la cumpărături [24]; acestea fiind doar câteva exemple unde localizarea joacă un rol important.

1.4 Scopul și principalele obiective

Principalul scop al activității de cercetare este de a propune o metodologie de localizare relativă, bazată pe modele matematice, algoritmi, metode și tehnologii de localizare, care poate fi utilă în aplicații de explorare cu roboți mobili autonomi în medii necunoscute.

Pentru a atinge principalul scop, au fost fixate următoarele obiective:

- 1) Analiza și structurarea soluțiilor existente, evidențierea celor mai importante tehnici și tehnologii existente.

- 2) Identificarea factorilor de influență și a problemelor existente în localizarea roboților mobili.
- 3) Dezvoltarea unor metode de calcul de distanță între doi roboți folosind tehnologii bazate pe semnale mecanice.
- 4) Propunerea unui sistem hardware pentru roboți mobili care să ofere facilități de localizare având la bază metodele de calcul a distanței.
- 5) Elaborarea unei metodologii de localizare aplicabilă pe o gamă largă de sisteme luând ca exemplu sistemul CORE-TX.
- 6) Concepția și definirea unui model matematic de localizare bazat pe cooperare și localizare multilaterală și validarea acestuia.
- 7) Definirea unor reguli, pe baza cărora să se păstreze la un nivel superior confiența de localizare a roboților mobili.
- 8) Exemplificarea metodologiei pe un studiu de caz concret.
- 9) Elaborarea unui algoritm de localizare de confiență superioară, destinat sistemelor cu procesare distribuită.

1.5 Structura tezei

Teza de față este structurată în șapte capitole după cum urmează:

- Primul capitol constituie o prezentare scurtă a temei de cercetare, motivând alegerea temei prin actualitatea studiului, aria vastă de aplicabilitate precum și punctul de pornire: laboratorul DSPLabs. De asemenea se prezintă și obiectivele tezei.
- O clasificare a sistemelor de localizare este prezentată în capitolul 2. Ca și o principală categorisire, sistemele de localizare pot fi grupate în: globale și locale. Tot aici sunt evidențiați și factorii de influență asupra sistemelor de localizare. De asemenea sunt expuse problemele și soluțiile propuse în funcție de factorii de influență evidențiați.
- Un nou concept hardware pentru orientarea și localizarea roboților mobili în plan 2D, denumit IRULT (Inter-Robot Ultrasonic Localization Turret), se prezintă în capitolul 3. De asemenea, tot aici se prezintă o serie de metode care stau la baza modelului de orientare și localizare a roboților mobili. Una dintre aceste metode este algoritmul de aliniere. Alinierea reprezintă o etapă esențială în cadrul modelului de orientare și localizare a roboților mobili, fiind tocmai punctul de pornire al acestuia. Tot în capitolul 3 sunt descrise două metode de calcul a distanței și rezultate experimentale.
- Metodologia de localizare relativă propusă se prezintă în capitolul 4. Aceasta se bazează pe trei niveluri: 1-PREDICȚIE, 2-COOPERARE și 3-CENTRALIZARE. Cele trei niveluri sunt detaliate în capitolele 5 și 6. Tot în cadrul acestui capitol sunt redată următoarele noțiuni ce fac parte din metodologie: concepte de bază, măsurarea unghiului, inițializarea sistemului, metoda LOAD și localizarea bilaterală. De asemenea este prezentat un set extins de experimente într-un studiu de caz efectuat în cadrul laboratorului DSPLabs, pentru a evalua metodologia de localizare propusă. În urma experimentelor se constată că este nevoie de mai multe seturi de experimente care trebuie

să fie efectuate prin simulare pentru a cuprinde toate cazurile izolate ce pot apărea în practică.

- În Capitolul 5 se evidențiază primul nivel (PREDICȚIA) din cadrul metodologiei de localizare relativă propuse. La acest nivel fiecare robot mobil se auto-localizează pe baza resurselor proprii de mobilitate și navigare, prin procesare locală. În cadrul acestui nivel se prezintă conceptul dead reckoning. Acesta reprezintă un proces de subordonare față de următoarele două niveluri de localizare care sunt prezentate în continuare. Nivelul doi (COOPERARE), expus în cadrul aceluiași capitol, are la bază metoda de localizare multilaterală. Tehnici și metode existente care se referă la localizare multilaterală precum și modelul matematic propus bazat pe metoda ILS (Iterative Least Squares) sunt redată în acest capitol. De asemenea se prezintă și reguli de formare a pânelor care reduc din efectul de propagare a erorilor. Pentru validarea modelului matematic se țin cont de măsurătorile de calcul a distanței descrise în capitolul 3 și simulări de propagare a erorilor în cele mai defavorabile cazuri prezentate în acest capitol. Tot aici este descris simulatorul dezvoltat, SiMuLoC, precum și un exemplu de aplicație unde poate fi folosit modelul matematic.
- Tot din cadrul metodologiei de localizare, al treilea nivel (CENTRALIZARE) prezentat în capitolul 6, cuprinde un algoritm destinat sistemelor cu procesare distribuită. Algoritmul BPF (Backtracking Particle Filter) propus, cu o tehnică originală în domeniul localizării, este un algoritm de tip „filtru de particule” sub forma de „backtracking” și care se bazează pe modelul algoritmilor probabiliști Las Vegas. Deoarece experimentele făcute în practică acoperă un caz prea izolat pentru a putea valida metodologia propusă, se execută mai multe seturi de experimente prin simulare pentru a cuprinde toate cazurile izolate ce pot apărea în practică. Prin analiza rezultatelor experimentale și prin introducerea conceptului „bază” cu rolul de a reseta propagarea erorilor în localizare, se arată că metodologia de localizare relativă propusă este aplicabilă în contextul explorării roboților autonomi în medii necunoscute.
- În ultimul capitol sunt prezentate concluziile acestei cercetări, o sinteză a contribuțiilor aduse în cadrul tezei de doctorat și perspective de cercetare și dezvoltare în viitor.

2. LOCALIZAREA, ABORDĂRI CURENTE

În acest capitol se prezintă o clasificare a sistemelor de localizare. Ca și o principală categorisire, sistemele de localizare pot fi grupate în: globale și locale. Principalul scop al acestui capitol este cel de a defini noțiuni de clasificare care vor fi utilizați în celelalte capitole. Pentru fiecare categorie se prezintă cel puțin un exemplu de sistem de localizare din literatura de specialitate. Tot aici se prezintă și câteva tehnici reprezentative de localizare precum și tehnologii.

La finalul capitolului sunt evidențiați factorii de influență asupra sistemelor de localizare. De asemenea sunt expuse problemele și soluțiile propuse în funcție de factorii de influență evidențiați.

2.1 Clasificarea sistemelor de localizare

Sistemele de localizare pot fi împărțite în două mari categorii: sisteme de localizare globală (CAT I) și sisteme de localizare locală (CAT II-V) (Figura 2-1). De obicei în practică se cere ca un sistem local să aibă o acuratețe mai mare față de cel global [25]. Acuratețea unui sistem global poate să ajungă la ordinul metrilor, iar acuratețea unui sistem local poate să ajungă la ordinul centimetrilor depinzând de cerințele sistemului și tehnologia aplicată.

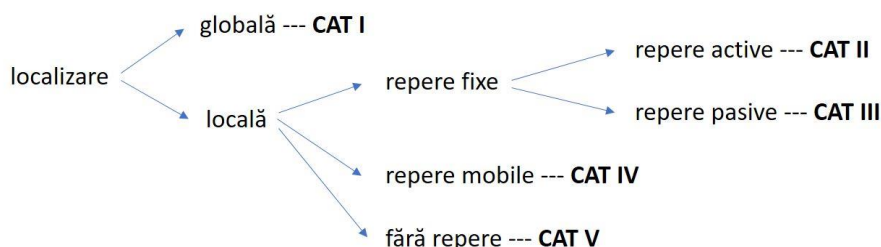


Figura 2-1 Clasificarea sistemelor de localizare

Cunoașterea locației unui nod mobil înseamnă a cunoaște poziția nodului respectiv în task-ul de navigare unde poziția inițială este în mod aprioric cunoscută (specific categoriilor CAT IV-V) [26] sau cunoașterea unei poziții, fără a fi nevoie de orice cunoaștere în mod aprioric de locație (specific categoriilor CAT I-III) [27][28]. Există două abordări diferite în managementul locației: abordare topologică (specific categoriei CAT III) [29] și abordare metrică (specific tuturor categoriilor în afară de CAT III) [30].

Reperele active (CAT II) sunt prezente deja în foarte multe aplicații în domeniul rețelelor de senzori. De obicei acestea sunt folosite pentru a menține comunicarea în sistem astfel ca resursele să poată fi folosite într-un mod optim iar consumul să fie cât mai redus. Reperele active uneori sunt denumite balize [31].

În cazul reperelor pasive (CAT III), localizarea într-un mediu în care formele geometrice sunt altele decât cele ideale, adică cele de bază (pătrate, cercuri, etc.),

recunoașterea reperelor este dificilă [32], este nevoie de o bază de date imensă unde reperatele pasive sunt introduse pe măsură ce sunt descoperite [33], iar în cazul unei reveniri la un reper anterior, reperul poate fi găsit în baza de date. Dezavantajul acestei abordări este numărul mare de observări înainte ca să se poată stabili o locație precisă. Acest proces de observare duce la creșterea timpului de execuție și a consumului de resurse. Una din tehnicile de localizare folosind această abordare se numește tehnica mobility-aware. Un exemplu al acestei tehnici este Localizarea Monte Carlo [34], care depinde în mod direct de mobilitate. Dacă nodurile (roboții mobili) sunt în repaus, sistemul nu poate colecta informații utile de la reperatele naturale. În [35], se prezintă o metodă care permite roboților să se localizeze cu ajutorul imaginilor prelucrate și analizate, la o precizie rezonabilă. Există și aplicații reale unde este dificilă extragerea caracteristicilor din mediu pe baza imaginilor și apoi să fie procesate într-un timp real pe platforma robotului mobil [36]. Localizarea și construirea hărții în mod simultan SLAM (Simultaneous Localization and Map Building) reprezintă un mod eficient și consistent de a construi harta și de a folosi această hartă pentru a obține estimările de locație. Această abordare a primit o mare atenție în ultimii ani iar unele rezultate pot fi văzute în [37], [38] și [39]. În SLAM atât traiectoria robotului mobil cât și amplasarea tuturor reperelor sunt estimate "online" fără a fi nevoie de vreo cunoaștere în mod aprioric de locație [40].

Anumite sisteme de localizare care fac parte din categoria CAT IV mai folosesc și termenul de "localizare relativă" [41]. În aceste sisteme, există un număr mic de noduri, denumite noduri de referință, care au în mod aprioric informația despre locația lor. Toate nodurile, indiferent de cunoștințele lor de coordonate absolute, estimează distanța dintre acestea și nodurile vecine. Estimările de distanță colectate sunt predominant obținute între perechi de noduri care nu au cunoștințe absolute de coordonate, de aceea s-a ajuns la folosirea termenului de "localizare relativă".

Performanța sistemelor de localizare de tip dead-reckoning din CAT V depind în mare măsură de specificațiile senzorilor montați pe platforma robotului (de exemplu giroscopul, accelerometrul, encoder-ul, etc). Măsurătorile de la acești senzori conțin diferite tipuri de zgomote. Mai mult, deoarece eroarea de măsurare de regulă se acumulează, performanța localizării este în mod substanțial degradată pe măsură ce robotul se deplasează.

Există sisteme de localizare care combină (au proprietăți) cel puțin două categorii din cele 5 [42]. Aceste sisteme de localizare pot fi numite sisteme de *localizare hibridă*.

În continuare se prezintă câteva exemple de sisteme de localizare din categorii diferite, pentru a avea o vedere de ansamblu asupra avantajelor și dezavantajelor fiecărei categorii în parte. De asemenea se prezintă și câteva tehnici și tehnologii în localizare care se aplică în majoritatea categoriilor prezentate.

2.2 Sisteme de localizare

2.2.1 CAT I - Sistemul GPS

Sistemul de localizare GPS (Global Positioning System) [43], inițiat de departamentul de apărare a Statelor Unite ale Americii, oferă utilizatorilor servicii de poziționare, navigare și sincronizare. Acest sistem este format din trei segmente: segmentul spațial, segmentul de control și segmentul de utilizator. În anul 1995, sistemul GPS a fost declarat operațional.

Segmentul spațial constă dintr-o constelație de sateliți (Figura 2-2) care transmit semnale radio către utilizatori. Statele Unite se angajează să mențină disponibilitatea a cel puțin 24 de sateliți operaționali, 95% din timp. Pentru a asigura acest angajament, forțele aeriene au urcat pe orbită 31 de sateliți GPS operaționali. (Începând cu data de 17 octombrie 2017, în constelația GPS s-au numărat un număr total de 31 de sateliți operaționali).

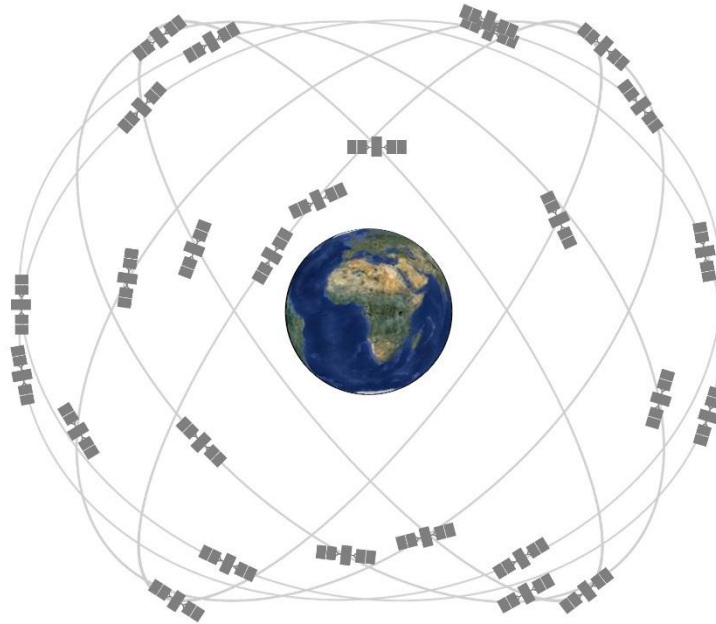


Figura 2-2 Constelație de sateliți expansibilă, așa cum este definită în standardul de performanță SPS (Standard Positioning Service) [43]

Sateliții GPS se deplasează pe orbita medie a pământului MEO (Medium Earth Orbit) la o altitudine de aproximativ 20.200 km. Fiecare satelit înconjoară Pământul de două ori într-o zi. Sateliții sunt aranjați pe șase planuri. Fiecare plan ocupă patru sloturi ocupate de sateliți. Acest aranjament cu 24 de sloturi permite utilizatorilor să vizualizeze cel puțin patru sateliți din aproape orice punct de pe planetă.

Segmentul de control GPS constă într-o rețea globală de baze la sol care urmăresc sateliții GPS, monitorizează transmisiile, efectuează analize și trimit date și comenzi către constelație.

Sateliții GPS difuzează semnalele în spațiu cu o anumită precizie, dar ceea ce recepționează utilizatorul depinde de factori suplimentari, cum ar fi geometria de poziționare față de sateliți, blocajele semnalului, condițiile atmosferice și modul în care este realizat receptorul. De exemplu, smartphone-urile cu funcții GPS au de obicei acuratețea de aproximativ 5m sub cerul liber. Totuși acuratețea se înrăutățește în apropierea clădirilor, podurilor și copacilor. Utilizatorii privilegiați, au la dispoziție receptoare cu dublă frecvență și/sau augmentare. Acestea permit poziționarea în timp real la câțiva centimetri, iar măsurătorile pe termen lung la nivelul milimetrilor.

Dezavantajele sistemului GPS sunt: semnalele directe pot fi blocate de clădiri, poduri, copaci etc; din această cauză, receptorul ar putea să obțină semnalele reflectate de pereți, clădiri etc., care duc la o poziționare eronată; semnalul este

atenuat în totalitate în interior sau subteran; cazuri mai puțin frecvente pot include interferențe radio, furtuni solare majore, întreținerea și manevrarea sateliților pot crea lacune temporare în acoperire, de asemenea aparatele proiectate incorect care nu respectă specificațiile interfeței GPS, pot duce la funcționări incorecte.

2.2.2 CAT I - Sistemul GLONASS

Sistemul GLONASS (ГЛОБАЛЬНАЯ НАВИГАЦИОННАЯ СПУТНИКОВАЯ СИСТЕМА) [44] a fost declarat oficial operațional în 1993. În 1995 a fost adus la o constelație complet operațională (24 de sateliți GLONASS din prima generație).

Prima propunere de folosire a sateliților pentru navigație a fost făcută de V. S. Shebashevich în 1957. Această idee s-a născut în timpul cercetării posibilității aplicării tehnologiilor de radio-astronomie pentru aeronavigație. Pe baza cercetării globale, s-a decis alegerea constelației orbitale formate din 24 de sateliți distribuiți în mod egal în trei planuri orbitale înclinate la $64,8^\circ$ față de ecuator. Sateliții GLONASS sunt plasați în orbite aproximativ circulare cu o altitudine de 19,100 km și o perioadă orbitală de 11 ore, 15 minute, 44 de secunde. Datorită valorii perioadei, a devenit posibil să se creeze un sistem orbital sustenabil, care spre deosebire de GPS, nu necesită susținerea impulsurilor corective în timpul vieții sale active. Înclinația nominală asigură disponibilitatea globală pe teritoriul Federației Ruse chiar și atunci când mai mulți sateliți nu sunt operaționali. Un mare dezavantaj pe care ar fi trebuit să se concentreze a constatat în lipsa echipamentelor de navigație civilă și a utilizatorilor civili.

Reducerea finanțării pentru industria spațială în 1990 a dus la degradarea constelației GLONASS. În 2002, constelația GLONASS a constatat din 7 sateliți insuficienți pentru sprijinirea navigației pe teritoriul Rusiei. GLONASS se afla în spatele GPS-ului în caracteristicile de precizie iar durata de viață activă a sateliților a fost 3-4 ani. Lucrurile s-au îmbunătățit atunci când programul federal "Sistemul global de navigație pentru perioada 2002-2011" a fost adoptat și lansat în 2002. Pe parcursul acestui program federal, au fost obținute următoarele rezultate:

- 1) sistemul GLONASS a fost conservat, modernizat și a devenit operațional. Astfel în prezent există două sisteme existente de navigație globală: GPS și GLONASS.
- 2) segmentul de control al solului a fost modernizat încât împreună cu constelația orbitală asigură caracteristicile de precizie la un nivel comparabil cu GPS.
- 3) standardele de stat pentru facilitățile de timp și frecvență și facilitățile de definire a parametrilor de rotație a Pământului au fost modernizate
- 4) au fost proiectate: prototipuri de augmentare GNSS (Global Navigation Satellite System), numeroase modele de recepție și măsurare, echipamente PNT (Position, Navigation and Timing) pentru utilizarea civilă și specială

Prima generație de sateliți Glonass între anii 1982 și 2005 au avut durata de viață între 3-4 ani și o masă de 1500 kg. De asemenea, a doua generație Glonass-M între 2003 și 2016 cu durata de viață 7 ani, o masă de 1415 kg și o putere de 1400 W; a treia generație Glonass-K între 2011 și 2018 cu durata de viață 10 ani, o masă de 935 kg și o putere de 1270 W; și generația Glonass-K2 în dezvoltare începând cu anul 2017, cu o durată de viață 10 ani, 1600 kg, 4370 W (Figura 2-3).



Figura 2-3 Sateliții: Glonass(1982-2005), Glonass-M(2003-2016), Glonass-K(2011-2018) și Glonass-K2(2017-) [44]

Dezavantajele sistemului GLONASS sunt aceleași ca și în cazul sistemului GPS: Semnalele directe pot fi blocate, semnalul este atenuat în totalitate în interior sau subteran iar în cazuri mai puțin frecvente pot exista interferențe radio, furtuni solare majore, mentenanță la nivel de satelit, de asemenea aparatele proiectate incorect care nu respectă specificațiile interfeței GLONASS, pot duce la funcționări incorecte.

2.2.3 CAT II - Localizarea în telefonia mobilă

Poziția unui telefon mobil prin identificarea celulei CID (Cell ID) este cea mai primitivă metodă de localizare în telefonia mobilă. În telefonia mobilă se folosesc și alte tehnici de localizare care duc la rezultate de poziționare mult mai bune [45].

Această tehnică nu face nici o încercare explicită de a determina poziția telefonului mobil MS (Mobile Station), în afară de a indica celula cu care este înregistrat (sau a fost înregistrat) dispozitivul mobil. Diferite cazuri pot fi observate în Figura 2-4.

Acuratețea se bazează pe raza de acoperire a celulei de servire și, prin urmare, este mult mai bună în mediul urban și variază de la 1m până la 30km. Prin această metodă, nivelul de acuratețe depinde de factori cum ar fi, dimensiunea celulei, tipul de celule (omnidirecțional sau sectorial). Cu toate acestea, această metodă nu determină dacă utilizatorul este de exemplu, 5 kilometri nord sau sud față de stația BTS (Base Transceiver Station). În zonele în care BTS-urile sunt apropiate, utilizatorii pot fi localizați mai precis. În zonele rurale, unde BTS-urile sunt mai depărtate, gradul de precizie este mult mai slab.

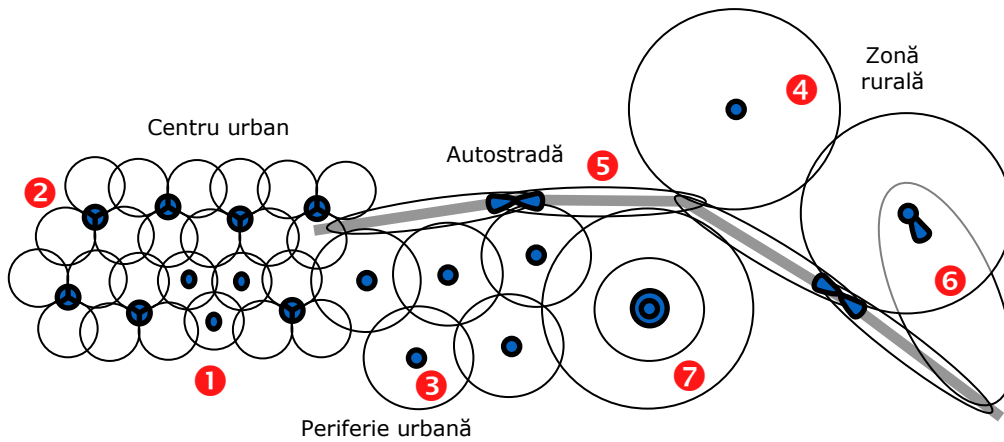


Figura 2-4 Acoperirea cu celule a teritoriului [46]

În lucrarea [47] sunt exemplificate tipuri de celule:

Tabelul 2-1 Tipuri de celule [47]

Tip celulă	Dimensiune în km
Celulă macro mare	3-30
Celulă macro mică	1-3
Celulă micro	0,1-1
Celulă pico	0,01-0,1
Celulă nano	0,001-0,01

Tot în această lucrare se prezintă câteva experimente pentru a testa acuratețea poziționării prin identificarea celulei CID. Rezultatele arată că acuratețea localizării este adesea inadecvată pentru a implementa chiar și servicii foarte simple bazate pe locație cum ar fi RDS (Resource Discovery Services).

2.2.4 CAT II - Cricket Indoor Location System și sisteme asemănătoare

Cricket Indoor Location System descris în [48] este un sistem de localizare care se bazează pe calculul a mai multor distanțe și este destinat laboratoarelor de cercetare în domeniul sistemelor mobile. Sistemul este alcătuit din mai multe dispozitive de transmisie, în funcție de dimensiunea suprafeței de acoperire dorite (cel puțin trei module) și un dispozitiv sau mai multe dispozitive mobile care joacă și rolul de recepție. În cele mai multe cazuri dispozitivele de transmisie se atașează pe partea superioară a încăperii în așa fel încât să acopere o porțiune cât mai mare sau chiar întreaga încăpere (Figura 2-5). Dispozitivul (sau dispozitivele) de recepție se atașează robotului (sau mai mulți roboți) aflat pe podeaua încăperii. Așa cum se prezintă în [49], pentru calcularea distanțelor, sistemul se bazează pe două tipuri de semnale: semnalul RF (Radio Frequency) și semnalul de ultrasunete. Semnalul radio este de 10^6 ori mai rapid decât semnalul ultrasonic, iar distanța se obține calculând diferența

celor doi timpi aplicând principiul TDOA (Time Difference Of Arrival). Localizarea roboților mobili prin acest sistem se realizează la o acuratețe de $1 \div 3$ cm.

Asemănător cu acest sistem, este sistemul prezentat în [50], un sistem de localizare pe baza a patru transmițătoare ca și repere fixe și un receptor mobil. În [51] este prezentat un sistem GPS (pentru interior) care este format din două receptoare ca și repere fixe și un transmițător mobil. Ambele receptoare și emițătorul folosesc un semnal radio prin care se sincronizează pentru a obține date de la semnalul de ultrasunete.



Figura 2-5 Sistemul de localizare Cricket [48]

Spre deosebire de soluțiile în care se măsoară distanța față de un obiect fizic, în care intervin zgomote în funcție de forma geometrică a obiectului [50], [52], [53], sistemele de măsurare a distanței între două transductoare (unul de transmisie și altul de recepție) nu necesită o corelație a semnalului. Semnalul care este emis ajunge la receptor fără a exista zgomot în semnal. Aceasta constituie un avantaj în cazul calcului distanței între doi roboți mobili echipați cu module radio și transductoare de ultrasunete.

Sistemul BPS (Building Positioning System) descris în [54], determină poziția a unui dispozitiv portabil de recepție prin recepționarea a semnalelor radio de la dispozitivele fixe care au rolul de a transmite semnale radio asemănător funcționării modulelor Cricket sau a sistemului GPS. Comparativ cu GPS acest sistem folosește o frecvență mult mai joasă, ceea ce face ca undele să se propage cu o atenuare relativ mică. Sistemul necesită numai 4 antene fixe de transmisie atașate în patru colțuri diferite a clădirii. Acuratețea unui astfel de sistem este de 5cm.

Hagisonic StarGazer reprezintă un sistem de localizare destinat roboților mobili. Se bazează pe analiza razelor infraroșii care sunt reflectate de un reper pasiv cu ID unic. Sistemul funcționează în felul următor: Transmițătorul se află pe robot. Acesta transmite raze infraroșii către reperul fix care este atașat pe partea superioară a încăperii; razele infraroșii sunt reflectate și ajung la StarGazer care se află pe robot; StarGazer conține o cameră CMOS care poate să proceseze imaginea aflând unghiul

și distanța la care se află robotul față de reperul fix; prin calcule matematice se obține poziția robotului în încăperea. Avantajul unui astfel de sistem este acuratețea acestuia, conform [55] ajunge la 2 cm. Sistemul poate să efectueze 20 de măsurători pe secundă. Dezavantajul acestui sistem este prețul ridicat și aria de acoperire care variază de la 2,5m la 5m.

BATSY [56], este un sistem de localizare cu ultrasunete, care este format din 6 până la 8 balize montate pe pereți pentru a putea recepționa semnale ultrasonice și un robot mobil care emite un semnal radio și semnal ultrasonic. Acuratețea unui astfel de sistem este de 3 cm.

Dezavantajul sistemelor prezentate este costul ridicat al modulelor fixe preinstalate (balize). Un alt dezavantaj a acestei abordări este acela că un robot poate fi localizat într-un spațiu limitat (acolo unde există acoperire cu modulele preinstalate).

2.2.5 CAT III - Sisteme bazate pe camere omnidirecționale

Un sistem bazat pe localizare vizuală trebuie să fie antrenat înainte de a fi folosit. În timpul antrenării, sunt capturate mai multe imagini reprezentative din mediu și sunt asociate locațiilor corespunzătoare. Pe durata unei operațiuni concrete de localizare, imaginile preluate sunt comparate cu cele originale aflate deja în baza de date din faza de antrenare, iar atunci când se găsește cea mai potrivită imagine din baza de date care să corespundă cu imaginea actuală, se poate deduce locația. Prin urmare scopul este ca în faza de antrenare să fie luate seturi reprezentative de imagini pentru fiecare locație.

De exemplu în [57], antrenarea se realizează în două etape. În prima etapă, robotul este plimbat prin mediu, sistemul de localizare achiziționând imagini cu rate cuprinse între 1 și 5 Hz. Acestea sunt salvate pe hard-disk. În a doua etapă, utilizatorul creează tabele cu imaginile capturate, indicând pentru fiecare imagine locația corespunzătoare. Datorită naturii secvențiale a imaginilor și cu ajutorul unui editor calitativ, maparea pozelor este un proces care durează doar câteva minute.

Imaginile capturate în faza de antrenare trebuie să redea cu exactitate mediul. Pentru a avea un număr cât mai mic de poze se folosește o cameră omnidirecțională ce oferă imagini panoramice ale mediului (Figura 2-6).

Unul dintre cele mai importante avantaje ale unui sistem cu cameră omnidirecțională este acela că folosește obiective de larg consum și generează imagini de ansamblu.

Un alt avantaj este reprezentat de unghiul mare de vedere, acest lucru înseamnă că numărul minim de poze necesar pentru redarea unui mediu este redus. Astfel sistemul observă chiar și cele mai mici schimbări.

Imaginile achiziționate în faza normală de operare sunt clasificate de funcția de recunoaștere a locației. Această funcție este formată din 4 pași:

- Generarea unei histograme color a imaginii achiziționate;
- Pentru fiecare bandă de culoare se determină cea mai bună potrivire pentru fiecare locație posibilă;
- Pentru fiecare bandă de culoare se determină un vot și o confiență;
- Clasificarea imaginilor bazată pe un vot unanim.



Figura 2-6 Exemplu de imagine obținută prin sistemul Omnicam [57]

Primul pas transformă imaginea în 6 histograme de o singură dimensiune. În al doilea pas, aceste histograme sunt comparate cu cele de referință. Pentru a reduce numărul căutărilor, prima dată se caută în locațiile cu histograme apropiate și în imediata vecinătate. După această căutare, algoritmul determină cea mai apropiată histogramă de cea de referință. Pentru a determina cât de bine se potrivesc aceste histograme, trebuie folosit un sistem metric. Sistemul metric divergent Jeffrey a obținut cele mai bune rezultate în cazul comparării histogramei.

Avantajul unui astfel de sistem este că nu se axează pe informații odometrice și pe modele de odometrie de incertitudine. În timp ce odometria este foarte utilă pentru navigările pe distanțe scurte, un astfel de sistem poate localiza indiferent de distanțele parcurse și de obstacolele întâlnite. Astfel, erorile odometrice nu vor afecta sistemul de localizare bazat pe camere omnidirecționale.

În cazul unei rețele de roboți, ideal ar fi ca aceștia să poată comunica între ei locația lor actuală, ceea ce ar ajuta sistemul de localizare pentru a crea cât mai puține confuzii și erori. Dacă roboții nu comunică între ei, numărul de clasificări crește semnificativ. Timpul de execuție a clasificărilor ar trebui diminuat, în acest fel, rețeaua de roboți ar funcționa eficient și ar fi o variantă viabilă și practică pentru implementări într-un mediu real.

2.2.6 CAT IV - PhotoBeacon (sistem optic de localizare)

În cazul sistemelor cu roboți există metode indirecte de calcul a distanței. Dacă avem 3 roboți care formează un triunghi iar unghiurile triunghiului sunt cunoscute, atunci prin formule matematice poate rezulta distanța dintre fiecare robot. Un astfel de sistem este descris în [58] (Figura 2-7). La baza acestui sistem stau 4 componente care fac sistemul să fie funcțional. În primul rând un LED de putere care servește pentru transmiterea unei raze de lumină, lentile de dimensiuni foarte mici pentru capturarea luminii transmise de la alți roboți pe plan orizontal, un senzor CMOS

sofisticat pentru detectarea semnalului de la alt robot și în final o placă PCB cu un microcontroler pentru controlul altor componente.

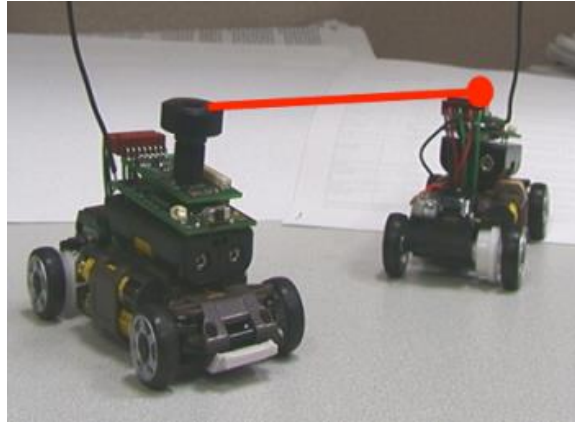


Figura 2-7 Sistemul optic de localizare PhotoBeacon [58]

Trebuie remarcat faptul că tehnologia folosită oferă doar unghiul nu și distanța dintre emițător și receptor. Distanța poate fi dedusă. Sistemul este unul de tip colaborativ, localizarea unui anumit robot se poate face prin colaborarea cu alți roboți. Pentru ca un robot să-și afle locația este necesar ca minim doi roboți să-și cunoască poziția.

În Figura 2-8 sunt reprezentați 4 roboți. Roboții A și B își cunosc poziția, astfel distanța d dintre aceștia este cunoscută. Robotul 1 nu știe poziția și comunică cu roboții A și B pentru a o stabili. Astfel se formează triunghiul AB1. Cu ajutorul acestui sistem optic de localizare se determină unghiurile α și β . Aceste informații sunt suficiente pentru a determina poziția robotului 1. La fel se determină și poziția robotului 2.

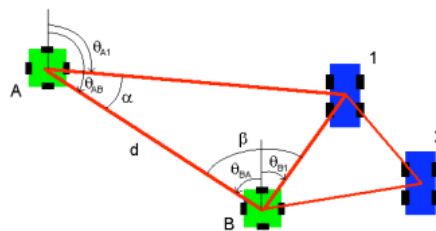


Figura 2-8 Localizarea roboților 1 și 2 față de roboții A și B [58]

Din [58] aflăm că eroarea de localizare unui astfel de sistem obținută dintr-o simulare este de 3,3 cm față de valoarea așteptată. Dezavantajul unui astfel de sistem este prețul costisitor prin introducerea senzorilor optici.

2.2.7 CAT IV - Localizarea prin trilateratie pentru echipă multi-robot

Proiectul "Localizarea prin trilateratie pentru echipă multi-robot" [59] a fost dezvoltat de o echipă de studenți de la Universitatea Wyoming Laramie, USA. Abilitatea roboților pentru a localiza rapid și exact roboții din jurul lor este extrem de importantă pentru echipele de roboți. Fiecare robot are propriul său cadru local de coordonate.

Scopul proiectului a fost de a crea un modul hardware pentru a localiza roboții din vecinătate. În trilateratia 2D, locațiile celor trei puncte de bază sunt cunoscute la fel de bine ca și distanțele de la fiecare din aceste trei puncte de bază către robotul care trebuie localizat. De regulă în mod vizual, trilateratia 2D implică găsirea locației unde cele trei cercuri se intersectează. Astfel, pentru a localiza un robot folosind trilateratia 2D, robotul care trebuie să facă detecția trebuie să cunoască locațiile celor trei puncte în propriul său sistem de coordonate și să fie capabil să măsoare distanțele de la aceste trei puncte spre robotul ce trebuie localizat.

Metoda de măsurare a distanței se bazează pe faptul că sunetul se propagă mult mai încet decât lumina. Fiecare robot are un transceiver de radiofrecvență și trei traductoare de ultrasunete. Robotul 2 emite simultan un puls de radiofrecvență și un puls ultrasonic. Când robotul 1 recepționează pulsul de radiofrecvență (aproape instantaneu), pornește un ceas. Atunci când pulsul ultrasonic este recepționat de fiecare din cele trei traductoare ultrasonice ale robotului 1, sunt calculați timpii scurși. Acești trei timpi sunt convertiți în distanțe, ținând cont de viteza sunetului. Pentru că locațiile traductoarelor acustice sunt cunoscute, robotul 1 este acum capabil să folosească trilateratia pentru a calcula locația robotului 2, mai precis locația traductorului acustic de pe robotul 2. Toate cele trei traductoare acustice au capabilitatea de a recepționa, dar numai unul dintre acestea este capabil să și transmită.

Traductoarele de ultrasunete produc un con de energie de-a lungul unei linii perpendiculare pe suprafața traductorului. Lățimea lobului principal (pentru traductoare de 40 KHz folosite în implementare) este de 30 grade. Pentru a produce energia acustică într-un plan 2D este nevoie de 12 traductoare acustice amplasate pe un cerc imaginar. Pentru a obține cele trei puncte de bază, este nevoie de 36 de traductoare. Acest lucru este scump și necesită multă putere electrică. Din acest motiv s-a adoptat o alternativă mai accesibilă. Fiecare punct de bază este alcătuit dintr-un traductor acustic orientat în jos. Un con parabolic este poziționat sub traductor, cu vârful orientat către traductor așa cum se poate observa în Figura 2-9.

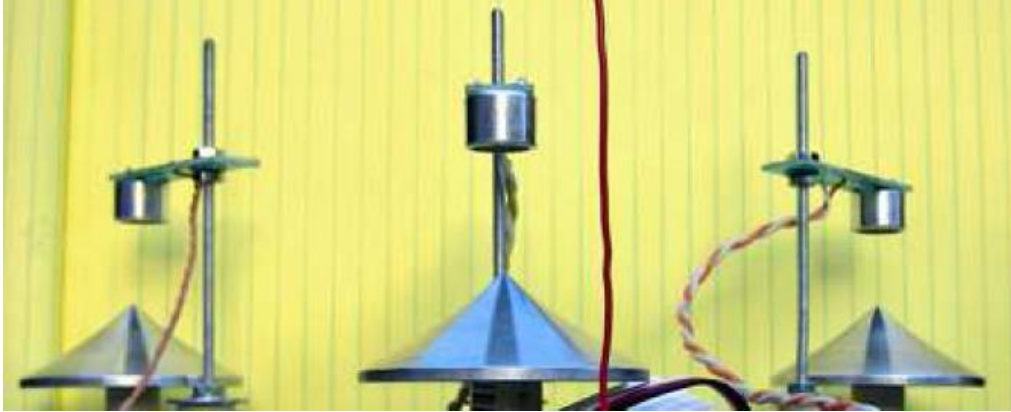


Figura 2-9 Conul parabolic [59]

Conul parabolic acționează ca o lentilă. Când traductorul este plasat în punctul focal virtual, conul colectează energia acustică în plan orizontal, și concentrează această energie spre traductorul acustic de recepție. În mod similar, un con funcționează în mod invers, reflectând energia acustică transmisă în plan orizontal. Acest lucru funcționează foarte bine, energia acustică fiind recepționată și la o distanță de 3.5m. O detecție de la o distanță mai mare poate fi obținută crescând energia în traductorul de emisie.

Modelul de trilaterare este prezentat în Figura 2-10. Presupunem că cei doi roboți sunt în formă de cerc. Un transceiver de radiofrecvență se află în centrul fiecărui robot. Fiecare robot are trei traductoare acustice (puncte de bază) denumite A, B, și C.

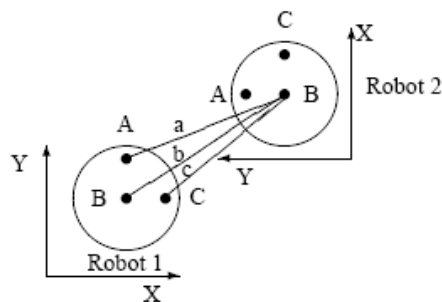


Figura 2-10 Modelul de trilaterare [59]

Robotul 2 emite simultan un puls RF și un puls acustic folosind traductorul B. Robotul 1 măsoară distanțele a , b și c . Presupunem că transceiverul B al robotului 1 este localizat la:

$$(x_{1B}, y_{1B}) = (0,0) \quad (2-1)$$

Cu alte cuvinte, fie $A(0, d)$, $B(0, 0)$, și $C(d, 0)$, unde d este distanța dintre A și B, respectiv B și C. Pentru ca robotul 1 să determine poziția traductorului B al robotului 2 în sistemul de coordonate robotului 1, are nevoie să găsească soluția la cele trei ecuații neliniare ca fiind intersecția cercurilor cu centrele localizate în A, B și C (punctele robotului 1) și razele a , b și c :

$$\begin{cases} (x_{2B} - x_{1A})^2 + (y_{2B} - y_{1A})^2 = a^2 \\ (x_{2B} - x_{1B})^2 + (y_{2B} - y_{1B})^2 = b^2 \\ (x_{2B} - x_{1C})^2 + (y_{2B} - y_{1C})^2 = c^2 \end{cases} \quad (2-2)$$

Modul de rezolvare a sistemului va fi discutat în subcapitolul 5.3.1 Trilaterația.

Un sistem de localizare asemănător este prezentat în [60] cu o acuratețe de 8mm și 3°, pentru distanțe de 6.7m și cu o rată de actualizare de 75 ms.

2.2.8 CAT V - Localizare folosind senzori de la mouse-ul optic

Senzorul mouse-ului optic este alcătuit dintr-o unitate care realizează în mod repetat imagini și le procesează. Unitatea de procesare calculează cantitatea de deplasare utilizând o prelucrare a imaginii pentru a distinge între două cadre succesive. Cele două cadre trebuie să se suprapună, astfel încât multe caracteristici, care au fost prezente în primul cadru, să poată fi recunoscute în cel de-al doilea. Este nevoie de o rată a cadrelor suficient de ridicată și de caracteristici optice bune pentru imagini clare.

În lucrarea [61] se propune un sistem de localizare care folosește senzorul optic ADNS-2610. Calea de navigare este determinată prin colectarea datelor de la senzorul optic și compunerea cu datele de rotație obținute de la busola digitală încorporată în telefonul mobil. Prin urmare, poziția utilizatorului este actualizată pe o hartă. În urma experimentelor efectuate, s-a constatat că eroarea medie este mai mică de 2% din lungimea parcursă.

În lucrarea [62] este folosit senzorul optic ADNS-2051. Față de mouse-ul optic, senzorul a fost adaptat pentru a putea fi utilizat în scopul localizării. Astfel, a fost montată o lentilă pe chip-ul senzorului (Figura 2-11), pentru a permite o capturare a cadrelor de la o distanță aprox. 5 cm față de sol. Pe de altă parte a fost redusă rezoluția cu factorul 10, ajungând la aproximativ 80 cpi. Acest lucru a permis operarea la viteze mai mari în deplasare, dar a scăzut calitatea de recunoaștere a trăsăturilor suprafețelor. De asemenea datele obținute de la senzorul optic au fost combinate cu datele de la o busolă digitală. Mai multe teste au fost efectuate, până la o viteză maximă de aproximativ 2 m/s, pe o lungime parcursă de 24 m, în urma cărora s-a constatat că s-au acumulat erori de aproximativ 1% din lungimea parcursă.

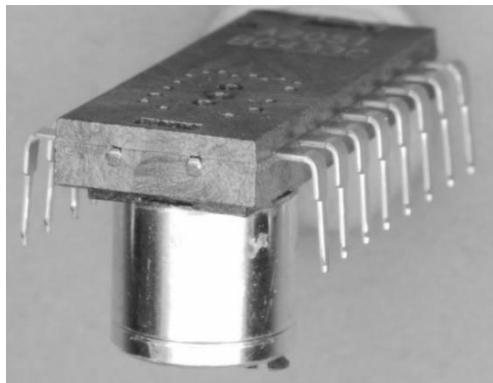


Figura 2-11 Senzorul optic al mouse-ului echipat cu o lentilă suplimentară [62]

Deși sistemele bazate pe senzorii folosiți de la mouse-ul optic, necesită costuri de instalare foarte scăzute și oferă o precizie destul de bună, au și anumite dezavantaje. De exemplu, nu pot funcționa pe suprafețe netede. În plus, sistemul nu poate funcționa la viteze mari.

2.3 Tehnici în localizare

2.3.1 TOF (Time Of Flight), TOA (Time Of Arrival)

Metoda TOF (Time Of Flight) sau altfel mai spus TOA (Time Of Arrival) constă în măsurarea timpului t în care un semnal traversează calea între două puncte (punctul de referință și obiect). Dacă viteza semnalului este v , distanța d este dată de ecuația:

$$d = v \times t \quad (2-3)$$

De exemplu, GPS utilizează TOF pentru a estima distanța dintre sateliții GPS și receptorul GPS.

Prin metoda TOF pot fi obținute valori mai precise la calculul distanței. Cu toate acestea, TOF este mai mult sau mai puțin precis sub efectul timer-ului microprocesor-ului, temperaturii ambiente, sau de alte influențe din mediul înconjurător, care aduce o eroare de calcul. În studiile din [63] și [64], un simplu LKF (Linear Kalman Filter) reprezintă o metodă pentru a estima TOF mai precis, care la rândul său este folosit pentru a calcula valori precise pentru calculul distanțelor.

Avantajele tehnicii TOF sunt: oferă acuratețe ridicată și achiziție rapidă.

Dezavantajele tehnicii TOF sunt: necesită sincronizarea timpului între emițător și receptor [65], necesită eșantioane de timp, linia de vizibilitate (alinieră) dintre emițător și receptor este obligatorie pentru o acuratețe mai bună.

2.3.2 TDOA (Time Difference Of Arrival)

TDOA (Time Difference Of Arrival) [66] este o metodă care constă în a măsura distanța dintre două puncte cu ajutorul a două semnale cu viteze diferite, care parcurg aceeași cale, între cele două puncte. Considerând două semnale A și B, cu vitezele v_A și v_B , trimise simultan de către un emițător. În cazul în care $v_A > v_B$, atunci semnalul B se situează în urma semnalului A ca și propagare. Acest decalaj de timp este sesizat de către un receptor situat la o distanță d de emițător. De exemplu, Cricket Indoor Location System [48], folosește TDOA pentru a măsura distanța față de punctele de referință.

Avantajele tehnicii TDOA sunt: achiziție rapidă și nu necesită sincronizarea timpului între emițător și receptor.

Dezavantajele tehnicii TDOA sunt: necesită sincronizarea timpului între nodurile de referință (balize), necesită eșantioane de timp, linia de vizibilitate (alinieră) dintre emițător și receptor este obligatorie pentru o acuratețe mai bună.

2.3.3 AOA (Angle Of Arrival)

AOA (Angle of Arrival) [67] măsoară în mod direct direcția de propagare a undelor semnalului care sosește. Aceste unde sunt în mod obișnuit recepționate de mai multe antene sau microfoane poziționate într-un șir. De asemenea, este posibilă colectarea informației AOA din metode de comunicare optică.

De exemplu, 4 receptoare în șir, depărtate unul față de celălalt la o distanță prestabilită, ascultă semnalul transmis. Analizând faza, sau diferența de timp între recepționarea semnalului prin cele 4 receptoare, este posibilă determinare unghiului de sosire al semnalului.

Această metodă poate să determine unghiul la o acuratețe de câteva grade. Un dezavantaj este necesitatea de mai multe antene care fac sistemul mai costisitor comparând cu metoda TDOA, deoarece fiecare nod trebuie să fie alcătuit dintr-un transmițător și mai multe receptoare. Mai mult, necesitatea de separare spațială între receptoare este dificil de adaptat atunci când factorul de formă a senzorilor trebuie micșorat.

AOA este uneori augmentat cu busole digitale. O busolă digitală indică pur și simplu orientarea globală a nodului care poate fi destul de utilă în combinație cu informațiile AOA.

Avantajele tehnicii AOA sunt: achiziția este rapidă și poate oferi o precizie de localizare ridicată.

Dezavantajele tehnicii AOA sunt: poate necesita antene direcționale și hardware complex, necesită algoritmi relativ complecși, performanța se deteriorează cu creșterea distanței dintre emițător și receptor, linia de vizibilitate (alinieră) dintre emițător și receptor este obligatorie pentru o acuratețe mai bună.

2.3.4 RTT (Round Trip Time), RTD (Round Trip Delay Time)

RTT (Round Trip Time) sau altfel mai spus RTD (Round Trip Delay-time) este o tehnică de a măsura distanța dintre nodul sursă și țintă. Se măsoară timpul de propagare acumulat a semnalului de la sursă la țintă și înapoi. Distanța se calculează după cum urmează:

$$d = \frac{(t_{RT} - \Delta t) \cdot v}{2} \quad (2-4)$$

unde, t_{RT} este timpul total de propagare a semnalului de la sursă la țintă și înapoi, Δt este timpul de procesare, necesar pentru nodul sursă și nodul țintă să proceseze pachetul de comunicare, v este o constantă predeterminată care reprezintă viteza semnalului.

Avantajele tehnicii RTT sunt: poate oferi o precizie de localizare ridicată, nu este nevoie de sincronizarea ceasurilor dintre emițător și receptor.

Dezavantajele tehnicii RTT sunt: întârzierea procesării poate afecta performanța în cazul măsurătorilor pe distanță scurtă, linia de vizibilitate (alinieră) dintre emițător și receptor este obligatorie pentru o acuratețe mai bună.

2.3.5 PDOA (Phase Difference Of Arrival)

Tehnica PDOA (Phase Difference Of Arrival) [68] poate fi utilizată pentru identificarea spațială a tag-urilor RFID (Radio-Frequency Identification). Există trei

mari tehnici care se bazează pe PDOA: TD-PDOA (Time Domain Phase Difference Of Arrival), FD-PDOA (Frequency Domain Phase Difference Of Arrival), SD-PDOA (Spatial Domain Phase Difference Of Arrival).

TD-PDOA permite estimarea proiecției vectorului de viteză a tag-ului, pe linia imaginară ce unește tag-ul și reader-ul (antena cititoare), prin măsurarea fazelor tag-ului la diferite momente de timp. Aceasta mai poate fi privită ca o formă de măsurare a efectului Doppler pentru a determina viteza nodului mobil. Această tehnică este ilustrată în Figura 2-12, unde tag-ul se îndepărtează cu o viteză constantă față de antena cititoare.

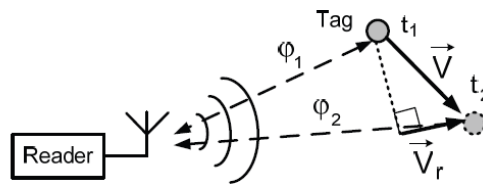


Figura 2-12 Tehnica TD-PDOA [68]

Prin măsurarea fazei semnalului de la tag la două momente de timp diferite (la o frecvență fixată), presupunem că alte două componente ale fazei tag-ului (faza de offset și faza de retro difuzie) nu se modifică, și luând derivata fazei în raport cu timpul, se poate calcula proiecția vitezei radiale a tag-ului cu formula:

$$V_r = -\frac{c}{4\pi f} \frac{\partial \phi}{\partial t} \quad (2-5)$$

FD-PDOA permite estimarea distanței până la tag prin măsurarea fazei tag-ului la diferite frecvențe. Această tehnică este ilustrată în Figura 2-13.

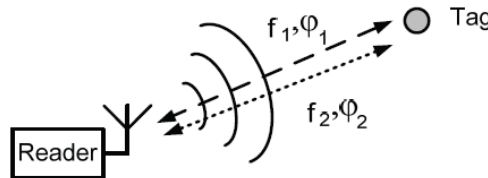


Figura 2-13 Tehnica FD-PDOA [68]

Prin măsurarea fazei de semnal a tag-ului la câteva (cel puțin două) frecvențe, luând derivata fazei în raport cu frecvența, și presupunând că alte două componente ale fazei tag-ului (faza de offset și faza de retro difuzie) nu se modifică cu frecvența și tag-ul este deplasat puțin (mult mai puțin decât lungimea de undă) pe parcursul măsurărilor, se poate calcula distanța față de tag cu formula:

$$d = -\frac{c}{4\pi} \frac{\partial \phi}{\partial f} \quad (2-6)$$

Deoarece tehnica FD-PDOA este similară cu radarul FMCW (Frequency-Modulated Continuous-Wave Radar), poate funcționa pentru tag-urile staționare cât și pentru cele în mișcare.

SD-PDOA permite estimarea direcției (direcția spre tag), sau a unghiului de sosire AOA, prin măsurarea fazelor de semnal a tag-ului de la mai multe antene receptoare.

Tehnica este ilustrată în Figura 2-14, pentru o configurație cu o antenă de transmisie separată și mai multe antene de recepție.

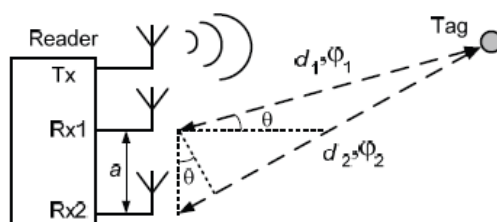


Figura 2-14 Tehnica SD-PDOA [68]

Prin măsurarea diferenței de fază $\varphi_2 - \varphi_1$, a semnalului de tag recepționat cu două antene diferite (la o frecvență fixă) și atribuind la diferența de cale $d_2 - d_1$, putem calcula direcția tag-ului:

$$\theta \approx \arcsin \left[-\frac{c}{2\pi f} \frac{(\varphi_2 - \varphi_1)}{a} \right] \quad (2-7)$$

unde a este spațiul dintre două antene receptoare.

2.3.6 Corelația semnalului

Corelația este o tehnică care urmărește gradul de asemănare dintre două semnale. Pentru un anumit deplasament de timp discret d , corelația se definește ca fiind suma produselor eșantioanelor ce se suprapun în timp ale primului semnal și celui de-al doilea care este întârziat cu d eșantioane. Determinând corelația pentru toate deplasamentele posibile, se va obține secvența de corelație a celor două semnale studiate. Dacă cele două semnale sunt identice atunci rezultatul obținut se numește autocorelație, iar dacă cele două semnale sunt distincte atunci se obține intercorelația.

Intercorelația a două semnale numerice de energie finită, $x[n]$ și $y[n]$, este definită în [69] astfel:

$$r_{xy}[d] = \sum_{n=-\infty}^{\infty} x[n]y[n-d], d \in Z \quad (2-8)$$

Parametrul d se numește deplasamentul corelației și reprezintă numărul de unități de timp discret cu care este deplasat semnalul $y[n]$ pentru calculul valorii $r_{xy}[d]$.

Semnalul purtător poate fi, de exemplu, un semnal de ultrasunete iar semnalul recepționat conține și zgomot. Algoritmul de intercorelație se aplică pentru semnalul recepționat $y[n]$ și semnalul purtător $x[n]$, așa cum este ilustrat în Figura 2-15.

Intercorelația se poate aplica, de exemplu, în cazul sistemelor de localizare cu ultrasunete. Rolul intercorelației este de a găsi deplasamentul care reprezintă distanța măsurată de la un emițător la un receptor de ultrasunete [71]. Obținând trei astfel de distanțe și aplicând principiul trilaterăției se poate obține poziția în spațiu.

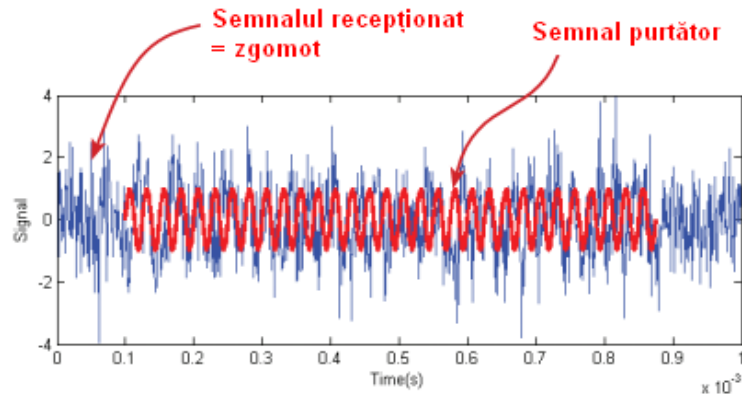


Figura 2-15 Semnalul recepționat compus [70]

2.3.7 RSSI (Indicatorul de nivel a semnalului recepționat)

Unele sisteme se bazează pe calculul RSSI (Received Signal Strength Indicator). Dacă se recepționează semnalul emis, putem să deducem distanța în funcție de intensitatea acestuia. Modelul matematic al puterii semnalului emis este prezentat în [72] astfel:

$$\frac{P_t}{P_r} = \frac{4\pi d}{\lambda^2} = \frac{(4\pi f d)^2}{c^2} \quad (2-9)$$

unde,

- P_t – puterea semnalului emis
- P_r – puterea semnalului recepționat
- d – distanța dintre transmițător și receptor
- λ – lungimea de undă a semnalului
- f – frecvența semnalului
- c – viteza luminii

Acuratețea unui sistem de localizare în care se folosește tehnologia WLAN (Wireless Local Area Network) pe baza RSSI este de 2 ÷ 4m.

În [73] este descris un sistem de noduri bazat pe protocolul Zigbee care pentru a determina distanța între două noduri, se bazează pe modelul matematic al puterii semnalului.

Din păcate, cea mai mare parte a lucrărilor bazate pe RSSI este susținută doar de rezultate din simulări, în timp ce unele studii experimentale demonstrează o posibilă lipsă de acuratețe a modelelor folosite. Acest lucru este valabil mai ales în mediu interior în care propagarea semnalelor de radiofrecvență este supusă unor efecte precum reflecții, interferențe multiple, care perturbă relația dintre distanța reală și RSSI [74].

Avantajele tehnicii RSSI sunt: ușor de implementat, eficiență din punct de vedere al costurilor, comunicarea și localizarea este dată de același modul hardware, nu necesită senzori direcționali precum senzorii de ultrasunete.

Dezavantajele tehnicii RSSI sunt: acuratețe redusă a localizării, predispunere la mai multe căi de propagare și zgomot ambiental, necesită combinarea cu tehnica fingerprinting pentru o acuratețe mai bună.

2.4 Tehnologii în localizare

2.4.1 WLAN (Wireless Local Area Network)

Tehnologia WLAN (Wireless Local Area Network), bazată pe standardele IEEE 802.11 și comercializată sub numele de brand WiFi, are avantajul utilizării unei zone radio fără licență, precum și a unei operațiuni de putere redusă. Există două tehnici de localizare prin WiFi: tehnica Fingerprint și tehnica trilaterăției/localizării multilaterale [75].

Inițial, WiFi avea o rază de recepție până la 100 metri, dar odată cu apariția IEEE 802.11ah, în principal optimizat pentru serviciile IoT (Internet of Things), raza de comunicare s-a mărit până la 1 km.

Localizarea prin tehnica Fingerprint necesită o hartă în care coordonatele carteziene le sunt asociate valori ale puterii semnalului (RSSI) măsurat în acel loc. Această tehnică constă în două etape: etapa de pregătire offline în care sunt acumulate măsurătorile (sunt adăugate într-o bază de date) și etapa de poziționare online, atunci când sistemul este funcțional (se fac măsurători și se stabilește locația prin potrivirea similarității cu baza de date).

Tehnica Fingerprint necesită un număr rezonabil de dispozitive de referință și un mediu stabil înainte de calibrare, deoarece rezultatul este sensibil la modificări în mediu, de exemplu, mutatul obiectelor pot afecta proprietățile semnalului sau modificarea orientării antenelor. Prin urmare, la astfel de modificări în mediu, este necesară repetarea etapei de pregătire offline.

Tehnica trilaterăției/localizării multilaterale în WiFi se bazează pe indicatorul de nivel a semnalului (RSSI) și necesită cel puțin trei stații WiFi AP (Access Point) de coordonate cunoscute. Modul de determinare a poziției va fi discutat în subcapitolul 5.3.1 Trilaterația. Cu cât sistemul are mai multe stații AP, acuratețea va fi îmbunătățită. Un minim de $n+1$ stații AP sunt necesare pentru a furniza o estimare a poziției în n dimensiuni. De exemplu, pentru o poziție 2D, este nevoie de minim 3 stații AP. Localizarea prin WiFi în interiorul clădirilor este foarte nefavorabilă, cauzată de propagarea semnalelor radio prin căi indirecte NLOS (Non Line of Sight).

Avantajele tehnologiei WiFi sunt: disponibilitate pe scară largă, nu necesită hardware suplimentar complex, simplu de integrat.

Dezavantajele tehnologiei WiFi sunt: predispusă la zgomote și interferențe, căile de propagare a undelor sunt multiple.

2.4.2 CSS (Chirp Spread Spectrum)

Tehnologia CSS [36] utilizează 80MHz de lățimea de bandă, cu pulsul centrat în 2,44GHz de durată 1μs și este certificată pentru funcționare în UE, Statele Unite și

Japonia. Avantajul major al tehnologiei CSS este acela de a oferi o performanță robustă pentru LR-WPAN (Low Rate Wireless Personal Area Network), chiar și dacă apar pierderi de căi, căi multiple sau reflexii în cazul undelor. Deoarece efectul Doppler al semnalului chirp este ignorabil, este deosebit de util pentru aplicațiile mobile. Semnalul up-chirp sau down-chirp este detectat prin autocorelarea aceluiași semnal. Dacă același semnal este auto-corelat cu un semnal care a fost sincronizat cu timpul, va apărea un impuls. Mărimea impulsului depinde de acuratețea sincronizării timpului dintre convoluția a celor două semnale. Tehnica de măsurare a distanței descrisă în IEEE 802.15.4a se numește SDTWR (Symmetric Double Sided Two-Way Ranging). SDTWR se bazează pe durata totală de timp necesară semnalului de a parcurge distanța între transmițător și receptor în ambele sensuri (tur-retur).

2.4.3 UWB (Ultra-WideBand)

Tehnologia radio UWB (Ultra-WideBand) primește o atenție impresionantă [76] și generează tot mai mare interes în industria de cercetare, care promite o rată de date ridicată, consum redus de energie și o soluție economică. Începând cu anii 1950, tehnologia UWB a fost folosită de armată iar în anul 2002, comisia de comunicare federală FCC (Federal Communications Commission) a aprobat utilizarea benzii 3,1-10,6 GHz pentru aplicațiile UWB fără licență [77]. UWB este o alegere potrivită pentru detectarea și poziționarea obiectelor datorită lățimii de undă vaste și rezoluției de timp.

În UWB [36] există două tipuri diferite de semnale: I-UWB (Impulse Ultra-wideband) și MC-UWB (MultiCarrier Ultra-WideBand). MC-UWB este pentru comunicarea de date datorită caracteristicilor sale de rată de date ridicată. Modularea MC-UWB se bazează pe multiplexarea cu diviziune în frecvență ortogonală OFDM (Orthogonal Frequency Division Multiplexing). Pentru modularea I-UWB nu se folosesc unde sinusoidale. În schimb, se utilizează o serie de semnale de impulsuri a căror durată este extrem de scurtă. I-UWB este adecvată pentru localizarea în interior.

Avantajele tehnologiei UWB sunt: acuratețe bună și imunitate la interferențe cu semnalele radio existente.

Dezavantajele tehnologiei UWB sunt: necesită hardware suplimentar, costuri ridicate, interferențe cu materiale metalice și disponibilitate la distanțe scurte 10-20m.

2.4.4 Zigbee

Protocoalele de comunicare ZigBee [36] sunt definite în IEEE 802.15.4a pentru MAC (Medium Access Control) și PHY (Physical Layer). ZigBee Alliance, a publicat specificația ZigBee cu informațiile legate de stratul de aplicație bazat pe IEEE 802.15.4 MAC și PHY.

Comunicarea ZigBee este eficientă pentru schimbul de informații de scurtă durată. Caracteristica cheie este puterea scăzută și faptul că nu este necesară nici o infrastructură. Spre deosebire de UWB, ZigBee nu este conceput pentru localizare. De aceea există puține sisteme ZigBee pentru localizarea în interior. Chiar dacă puterea semnalului ZigBee poate fi utilizată pentru scopul de detectare a distanței, este dificil să se mențină constantă. De asemenea, deoarece intensitatea semnalului depinde de

variația mediului, un model de propagare a semnalului nu poate fi utilizat pentru scopul de a determina distanțe.

Avantajele tehnologiei Zigbee sunt: necesită energie redusă, cost redus iar banda de frecvență în comunicare nu necesită licență.

Dezavantajele tehnologiei Zigbee sunt: interferențe cu banda de frecvență utilizată de o gamă largă de echipamente; nu este concepută pentru localizare, este concepută pentru rețelele în care conversația dintre două dispozitive durează câteva milisecunde, ceea ce permite transceiverului să treacă repede în modul sleep.

2.4.5 Ultrasunete

Măsurarea liniară a distanței este una dintre cele mai mari probleme pe care o au majoritatea aplicațiilor. Tehnologia ultrasunetelor este una din soluțiile folosite. Sunetul este o vibrație mecanică care se transmite și se propagă în mediu elastic. Gama de frecvențe pe care oamenii pot auzi sunt cuprinse în intervalului 20 Hz și 20.000 Hz. Sunetele mai sus de 20.000 Hz sunt cunoscute sub numele de ultrasunete, iar sunetele mai jos de 20 Hz ca infrasunete. Viteza sunetului în aer depinde de temperatură. La temperatura camerei viteza este de 345 m/s. Formula de calcul a vitezei este dată de următoarea ecuație:

$$c_{air} = (331.3 + 0.606 \cdot \vartheta) \left[\frac{m}{s} \right] \quad (2-10)$$

unde, ϑ este temperatura în grade Celsius (°C).

O altă denumire folosită pentru ultrasunete este sonarul. Sonarul funcționează în mod similar cu radarul. Un semnal de ultrasunete este generat într-o anumită direcție. Dacă există un obiect în calea acestui semnal, semnalul este reflectat înapoi la expeditor ca un ecou și este detectat. Dacă se măsoară diferența de timp între semnalul transmis și semnalul recepționat, este posibil să se determine distanța la care se află obiectul. În natură, liliecii folosesc o varietate de ultrasunete pentru a detecta prada lor.

Avantajele tehnologiei semnalelor ultrasonice sunt: absorbție mică, acuratețe la ordinul milimetrilor și nu interferă cu undele electromagnetice.

Dezavantajele tehnologiei semnalelor ultrasonice sunt: înaltă dependență de amplasare a senzorilor, necesită o compensare în funcție de temperatură și umiditate, interferență cu alte semnale ultrasonice sau sunete de înaltă frecvență, undele nu penetrează pereții solizi.

2.4.6 IR (Infrared)

O categorie de senzori de calcul a distanței folosiți în robotică sunt senzorii IR. Există mai multe tipuri, ele variind în funcție de parametrii lor. Parametrii importanți sunt: distanța maximă ce poate fi măsurată și acuratețea. Senzorii IR sunt mai ieftini ca și preț iar semnalul IR se propagă mai repede comparativ cu semnalul ultrasonic.

Prin deosebire, față de semnalul ultrasonic care face parte din categoria undelor mecanice, semnalul IR face parte din categoria undelor electromagnetice. Frecvența semnalului IR este cuprinsă în intervalul 1 ÷ 430THz.

Senzorii IR au caracteristici non-liniare iar aceștia depind de proprietățile de reflexie a obiectelor și suprafețele acestora. Acești senzori măsoară intensitatea luminii reflectate de un obiect, bazându-se pe acest concept pot calcula

distanța față de acesta. În [78], [79], [80] sunt descrise metode de determinare a proprietăților IR reflectate de o suprafață și de asemenea utilizând aceste proprietăți poate fi determinată distanța față de obstacol.

Dezavantajele tehnologiei semnalelor infraroșii sunt: nu penetrează pereții, linia de vizibilitate (aliniera) dintre emițător și receptor este obligatorie, performanță slabă în locații cu lumină directă a soarelui sau iluminare fluorescentă.

2.4 Concluzii

În continuare, autorul tezei sugerează câteva soluții pentru problemele existente în localizare. Problemele și propunerile sunt discutate în funcție de factorii de influență în localizare pentru explorarea roboților autonomi în medii necunoscute:

1) independența de repere fixe:

- Problema: Cele mai uzuale sisteme de localizare în cazul roboților autonomi sunt construite să fie dependente de repere fixe. Aici putem referi sistemele de localizare din categoriile CAT I, CAT II și CAT III.
- Propunere: Dacă se dorește o independență de repere fixe, atunci se poate merge pe una din cele două variante: CAT IV sau CAT V. În lucrarea de față se propune un sistem de localizare aparținând categoriei CAT IV, de asemenea, se propune un sistem hibrid care combină categoriile CAT II și CAT IV.

2) dinamica mediului:

- Problema: Dacă apar modificări în mediul în care se utilizează sistemul de localizare, cum ar fi numărul de persoane, prezența diferitelor obiecte (cum ar fi dulapurile, rafturile), localizarea se realizează mult mai dificil. Anumite sisteme de localizare din categoria III ar putea avea dificultăți la acest capitol. De asemenea, sistemele de localizare din CAT II care folosesc abordări precum fingerprinting (2.4.1 W), au mari probleme atunci când mediul devine dinamic.
- Propunere: Pentru aplicațiile unde mediul este dinamic, oricare categorie ar fi potrivită în afară de categoria III. De asemenea, trebuie evitate abordări precum fingerprinting. În lucrarea de față se propune o abordare metrică în care roboții mobili își conservă pozițiile prin așa numitele "pânze de localizare". Avantajul acestei abordări, prezentate în capitolul 4, este faptul că poziționarea pânzelor este destul de flexibilă astfel încât sistemul ar putea să funcționeze și în medii dinamice.

3) viteza de execuție:

- Problema: În sistemele de localizare, în funcție de algoritmul care se aplică, calculele se fac în mod repetat cu scopul de a se obține o acuratețe mai bună. Acest lucru duce la creșterea timpului în calcularea poziției.
- Propunere: Se propune ca algoritmul să fie descompus în mai multe fire de execuție și procese, în așa fel încât, calculul să fie transmis la un nod central (server/cloud) unde urmează să fie executat pe mai

multe procesoare. În lucrarea de față acest lucru se propune să se realizeze cu ajutorul nodului central, BRAIN (Background Robotic Activity Induction Node), definit în proiectul de cercetare Core-TX [16].

4) acuratețe:

- Problema: Acuratețea localizării depinde în mod direct de acuratețea senzorilor și de modul în care sunt folosite informațiile de la senzori. De asemenea, de exemplu, erorile de calcul în cazul măsurării distanțelor, care provin din reflexii, refracții și difracții, pot să ducă la erori mari în localizare.
- Propunere: Pentru a obține o estimare precisă a locației, este nevoie de tehnici complexe de procesare a semnalului care să identifice semnalul în linie directă (dacă există) și să minimalizeze/elimine efectele semnalelor pe căi multiple. De exemplu, tehnologiile prezentate în 2.4.2 CSS și 2.4.3 UWB sunt mai robuste la căi multiple și zgomot, comparativ cu tehnologia prezentată în 2.4.1 W. În lucrarea de față se propune aplicarea tehnologiilor robuste, CSS sau UWB, sub forma unui sistem de localizare de categorie CAT II pentru a compensa propagarea erorilor în sistemul de localizare de categorie CAT IV propus în această lucrare. De asemenea, în capitolul 6 se arată cum prin combinarea celor două sisteme se ajunge la o acuratețe mai bună în localizare.

5) scalabilitate:

- Problema: Într-o rețea de roboți, numărul acestora poate fi de ordinul sutelor sau miilor, ceea ce duce la creșterea complexității protocolului de localizare. Această problemă este specifică categoriei CAT IV.
- Propunere: Se rezolvă prin introducerea *bazelor*. La nivel de *bază*, se propune ca protocolul să fie descompus în mai multe fire de execuție și procese, astfel încât localizarea să fie gestionată de un nod central (server/cloud) unde protocolul se execută pe mai multe procesoare. Ca și în cazul vitezei de execuție, în lucrarea de față, scalabilitatea se propune să se realizeze cu ajutorul nodului central, BRAIN.

6) conectivitate:

- Problema: Densitatea roboților trebuie să fie destul de mare pentru a nu se pierde legăturile, dar în același timp o densitate mare duce la îngreunarea comunicării din cauza caracteristicii de evitare a coliziunilor a mediului fără fir. Această problemă este specifică categoriei CAT IV.
- Propunere: Se propun trei zone de vizibilitate: zona verde pentru calculul distanței sau a unghiului între doi roboți mobili, zona galbenă pentru comunicare între doi roboți și zona roșie în care nu se poate comunica. Aceste definiții fac parte din metodologia propusă expusă în capitolul 4. De asemenea gestionarea conectivității se realizează central (server/cloud) în funcție de regulile de mobilitate prezentate în capitolul 6.

7) eficiență energetică:

- Problema: Algoritmii complecși ar putea să aducă un consum considerabil robotului mobil autonom.
- Propunere: Se propune ca algoritmii complecși să fie descompuși în mai multe fire de execuție și procese, astfel încât calculele să se realizeze într-un nod central (server/cloud) unde algoritmii se execută pe mai multe procesoare. Ca și în cazul vitezei de execuție și a scalabilității, în lucrarea de față, algoritmii complecși care duc la un consum semnificativ de energie, se propune să se realizeze cu ajutorul nodului central, BRAIN.

8) costul:

- Problema: Prețul modulelor adiționale care aduc o îmbunătățire majoră în localizare este foarte ridicat.
- Propunere: Se propune un modul hardware simplu și de cost redus, care se bazează pe calculul distanțelor între două module de același fel folosind unde ultrasonice la o acuratețe de ordinul milimetrilor. Modulul este prezentat în capitolul 3.

9) robustețe:

- Problema: Trebuie analizată situația când anumiți senzori se defectează, măsurătorile eșuează, sau, un robot mobil devine complet nefuncțional.
- Propunere: În capitolul 3, se propune un modul hardware cu doi senzori funcționali, dar cu un rol dublu, fiecare senzor ar putea să acopere cazul în care celălalt senzor s-ar defecta. Astfel, cu anumite constrângeri, modulul hardware ar fi funcțional, dacă maxim un senzor se defectează. Tot în capitolul 3, se propun două metode pentru calcul distanței: una unidirecțională și cea de a doua, bidirecțională. Astfel, în cazul în care metoda bidirecțională eșuează, ar putea să se apeleze la metoda unidirecțională. Metodologia prezentată în capitolul 4 definește o pânză care este compusă din cel puțin doi roboți. Pentru a crește robustețea, numărul roboților mobili dintr-o pânză poate fi mărit.

10) standardizare:

- Problema: În prezent, nu există niciun standard care să guverneze cercetarea localizării în mediul interior al clădirilor [81].
- Propunere: În lipsa unui standard, în prezenta lucrare se propune o metodologie de localizare relativă care poate fi utilă la proiectarea soluțiilor de explorare a roboților autonomi în medii necunoscute.

3. SISTEMUL HARDWARE DE LOCALIZARE PROPUȘ

În acest capitol se prezintă un nou concept hardware pentru orientarea și localizarea roboților mobili în plan 2D, denumit IRULT (Inter-Robot Ultrasonic Localization Turret). Se prezintă și alte module hardware de localizare existente în literatura de specialitate. Tot aici se prezintă o serie de metode care stau la baza modelului de orientare și localizare a roboților mobili. Una dintre aceste metode este algoritmul de aliniere. Alinierea reprezintă o etapă esențială în cadrul modelului de orientare și localizare a roboților mobili, fiind tocmai punctul de pornire al acestuia. La final sunt descrise două metode de calcul a distanței și rezultate experimentale. Aceste rezultate sunt expuse pentru a fi referite și utilizate în cadrul altor capitole.

3.1 Introducere

Calculul distanței între doi roboți în sistemele cu mai mulți roboți mobili autonomi 2MARS (Multi Mobile Autonomous Robotic Systems) este un subiect tot mai intens întâlnit în literatura de specialitate. Există două tehnologii pentru a realiza calculul distanței între doi roboți: folosind unde electromagnetice și folosind unde mecanice.

Undele electromagnetice (de exemplu semnalele radio RF) sunt utilizate în localizare, dar pentru că propagarea este omnidirecțională, căile de propagare sunt complexe și viteza de propagare este prea mare, locul exact de unde încep ele să se propage nu poate fi determinat cu o acuratețe bună. Un alt exemplu este laserul cu un spectru foarte îngust. LIDAR (LIght Detection And Ranging) este o tehnică bazată pe laser și este folosită în multe aplicații, de exemplu vehicule autonome, robotică, zbor spațial, topografie și multe altele. Problema folosirii undelor electromagnetice în localizare este costul prohibitiv al echipamentelor necesare pentru multe categorii de aplicații.

Undele mecanice (de exemplu semnalele ultrasonice) se propagă cu un raport de viteză de 10^6 ori mai lent decât undele electromagnetice, ceea ce avantajează măsurarea timpului de propagare a semnalului (TOF) [82] și, prin urmare, coordonatele de origine ale undelor propagate pot fi determinate [83].

Semnalele ultrasonice sunt utilizate într-o gamă largă de aplicații: acasă, industrie, medicină, oceanografie și multe altele. În aplicațiile 2MARS, tehnologia ultrasunetelor este avantajată din două motive:

- 1) toate resursele necesare pentru a lucra cu semnale ultrasonice sunt de cost redus, acesta fiind un factor important pentru proiectarea unui sistem cu mai mulți roboți;

- 2) undele ultrasonice sunt utilizate pentru a determina cu mai multă acuratețe coordonatele de origine ale undelor propagate.

În lucrarea de față, semnalele ultrasonice sunt utilizate pentru a obține localizarea relativă a unui robot mobil în task-ul de navigare.

3.2 Problemele utilizării SLR actuale bazate pe ultrasunete

3.2.1 Module cu un emițător de ultrasunete și un set de mai multe receptoare

În [84] și [85] este descris un operator - sonar pentru distanțe mari, cu un traductor compus central omnidirecțional care emite un impuls sonar în toate direcțiile, și opt receptoare ultrasonice montate în cerc, la 45 de grade una față de alta, acoperind astfel 360 de grade. Problemele acestor module hardware sunt:

- Emițătorul nu este tipic, un traductor compus este greu de găsit pe piața senzorilor.
- În acest caz 8 canale ADC sunt utilizate pentru achiziția de semnale.
- Sunt necesare 8 amplificatoare pentru fiecare canal, ceea ce duce la creșterea costului pe lângă costul celor 8 traductoare. Un multiplexor analogic poate fi folosit, dar partea de achiziție devine mai complexă.
- În [86] se arată că unghiul de aliniere este unul dintre factorii care influențează TOF. În acest caz, opt receptoare nu sunt o acoperire consistentă pentru o mai mare precizie pentru localizarea pe distanțe mai mari.

3.2.2 Module cu un traductor și un con acustic reflector

În [87] este prezentat un modul compus dintr-un traductor ultrasonic cu două funcții: emițător și receptor. Traductorul este poziționat (îndreptat) drept în sus și toate undele ultrasonice de intrare și ieșire sunt reflectate de un con acustic reflector. Aceste module hardware au anumite probleme:

- Nu se discută deloc fenomenul de atenuare a semnalelor ultrasonice folosind un con acustic reflector, nu se știe dacă există pierderi de propagare datorate reflecției. Dacă intensitatea fasciculului este mai mică după reflecție atunci distanța maximă de propagare este mai mică.
- Unghiul de incidență pentru fasciculul reflectat prin conul acustic reflector este foarte dificil să se mențină la fel pe toate modulele deoarece conurile au o anumită toleranță de producție.
- Cu aceste module obstacolele nu pot fi identificate în mod corect.
- Uneori, există o interferență distructivă. Obstacolele pot influența o distanță de măsurare incorectă, deoarece la anumite distanțe între roboți, undele propagate în calea directă interferează distructiv cu undele care se reflectă de podea sau obstacole.

3.2.3 Module cu un emițător de ultrasunete și două receptoare

Sisteme de localizare bazate pe achiziție stereo cu ultrasunete sunt discutate frecvent în literatura de specialitate. În [88] se discută despre poziția unghiulară a unui robot mobil în raport cu o sursă de ultrasunete cunoscută. În lucrarea extinsă [89], fiecare robot este echipat cu un transmițător ultrasonic și două receptoare. Metoda folosită se bazează pe măsurarea diferenței de fază între cele două receptoare

ultrasonice și măsurare TOF. Măsurarea diferenței de fază este mai precisă decât măsurarea TOF datorită rezoluției mai mari [90]. Comentariile pentru această abordare sunt :

- Intervalul maxim de măsurare folosind metoda diferență de fază este limitat la o perioadă de semnal, de exemplu $25\mu\text{s}$ pentru frecvența de 40 kHz, care este echivalent cu 8mm.
- Metoda de diferență de fază este utilizată pentru determinarea poziției unghiulare cu acuratețe mare, dar cu mai multe constrângeri.
- Atunci când cele două receptoare au fost la o distanță de 450mm, doar o distanță mai mare de 88 cm putea fi măsurată deoarece prin propagarea semnalului ultrasonic, unda conică putea să treacă între cele două receptoare fără a putea fi detectat un semnal. Când receptorii au fost poziționați la o distanță de 350mm, intervalul de măsurare acceptabil obținut în rezultatele experimentale a fost [0,5m; 5,5m].

3.2.4 Alte module ultrasonice

În [91] este prezentată o turelă cu trei traductoare ca o placă de extensie pentru robotul Khepera [92]. Utilizarea acestei extensii favorizează detectarea diferitelor tipuri de obiecte. Distanța maximă pentru detectarea obiectelor este de 20 de centimetri . Acest modul nu este conceput pentru comunicarea cu alți roboți, dar este interesant faptul că semnalul de ultrasunete este utilizat pentru a îmbunătăți achiziția undelor infraroșii de pe robotul Khepera. Unele dintre problemele legate de senzorul infraroșu sunt că unele materiale nu reflectă lumina infraroșie (de exemplu, panou de sticlă, folia de hârtie) [93].

SpiderBat [94] este un modul cu patru emițătoare ultrasonice și patru receptoare care sunt montate alternativ într-un cerc. Abordarea utilizată pentru măsurarea distanței se bazează pe diferența de timp de sosire (TDOA), similar cu Cricket [49], dar cu o ameliorare: transmisia de ultrasunete este pornită la un interval de timp fix (20ms), după ce procedura este inițiată de către nodul expeditor. În acest interval de timp fix (20ms), nodul expeditor comunică timpul de transmisie prin trimiterea unui pachet radio. În acest fel nodul receptor poate stabili exact când transmisia cu ultrasunete a fost efectuată cu ajutorul unui algoritm de sincronizare de timp expeditor - receptor [95]. După momentul de detectare cu ultrasunete, nodul receptor poate stabili distanța mai precis decât sistemul de localizare Cricket. Prin urmare, prin executarea unui protocol de sincronizare de timp dedicat, în sistem se introduce o eroare de măsurare a distanței de numai 1,4 mm, în cel mai rău caz. Deviația standard a erorii este de 0,31 mm pentru măsurători de până la 1 m și 5,39 mm pentru distanțe între 1m și 14m. Cu patru receptoare poziționate în cruce, SpiderBat poate măsura unghiul de sosire, folosind informații de la maxim trei receptoare atunci când este posibil, dacă nu, de la doar două. Eroarea medie măsurată este mai mică de 5° pentru distanțe scurte.

Există multe alte module în literatura de specialitate utilizate în aplicații 2MARS care combină unele dintre tehnologiile menționate mai sus. De exemplu, în [59] și [60], sistemul prezentat este format din mai multe module cu doar un traductor și un con acustic reflector așa cum au fost prezentate mai sus. Modulele cu prea mulți senzori și actuatoare sunt prea complexe și costisitoare. Scopul este de a implementa un modul simplu și de cost redus cu număr minim de senzori și actuatoare cu scopul de a cerceta noi protocoale pentru problema localizării relative în aplicațiile 2MARS.

3.3 Proiectarea dispozitivului pentru orientarea și localizarea roboților mobili

3.3.1 IRULT (Inter-Robot Ultrasonic Localization Turret) ca și parte a sistemului CORE-TX

Modulul hardware propus are un set comun de cerințe pentru sistemul robotizat țintă: CORE-TX (Figura 3-1).

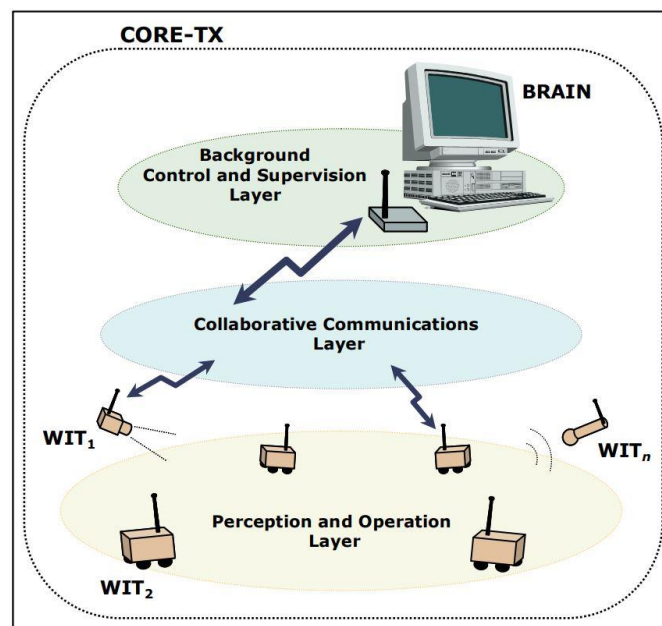


Figura 3-1 Sistemul CORE-TX [16]

CORE-TX (COLlaborative Robotic Environment – the Timisoara eXperiment) [16] își propune studiul teoretic și aplicativ al mediilor colaborative de micro sisteme timp-real autonome cu inteligență încorporată, precum și utilizarea acestuia în dezvoltarea aplicațiilor de anvergură, cu impact direct asupra societății informaționale moderne, introducerii tehnologiilor de ultimă oră în industrie și protecția mediului. Acesta este conceput ca o platformă complexă și este format la nivelul arhitecturii de un set eterogen de micro sisteme autonome cu inteligență încorporată, numite WIT (terminale inteligente fără fir, din engleză: Wireless Intelligent Terminals), un mediu de comunicare colaborativ și o entitate centrală, cu rol de configurare, control și supraveghere a întregului sistem numit BRAIN (Background Robotic Activity Induction Node). Terminalul WIT este format din diferite module: modulul de comunicare, modulul de percepție (placă de achiziție), modulul de bază, modulul de gestionare a energiei și modulul de acționare și operare.

Modulul de percepție (Figura 3-2), este compus dintr-o unitate de procesare, un driver pentru controlul motorului pas cu pas, accelerometrul, senzorul de temperatură, modulul de extensie (IRULT) și alte circuite de interfațare.

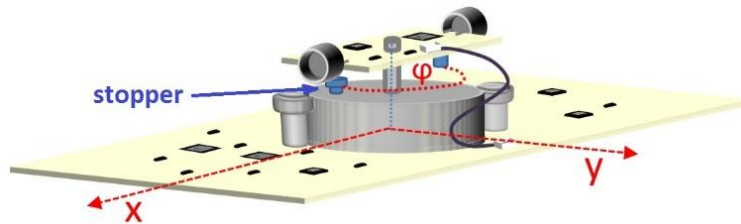


Figura 3-2 Modulul IRULT ca și parte a modului de percepție [96]

Modulul IRULT (Inter-Robot Ultrasonic Localization Turret) face parte din modulul de percepție (placa de achiziție) și este folosit pentru a obține orientarea și poziția robotului în task-ul de navigare. IRULT constă dintr-un motor pas cu pas și două traductoare ultrasonice (Figura 3-2). Fiecare traductor are un interval de directivitate de formă conică de aproximativ 50 de grade și poate să emită sau să recepționeze semnale ultrasonice la frecvența de 40 KHz. Aceste două traductoare sunt montate spate în spate la 180 de grade. Prin rotirea motorului pas cu pas, fiecare traductor poate acoperi un unghi de vizibilitate de 240 de grade (Figura 3-3).

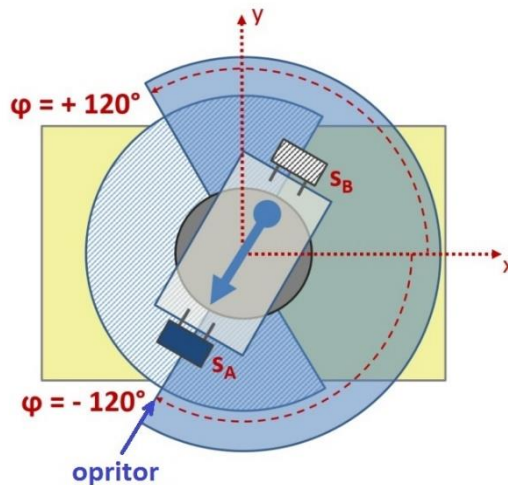


Figura 3-3 IRULT, privire de sus [97]

Proiectarea schematică a dispozitivului IRULT este prezentată în Figura 3-4. Microcontrolerul folosit este MSP430 (Atmel Corporation, 2010), utilizat pentru rapiditate, achiziție periodică de date și procesare. Două traductoare similare sunt utilizate atât pentru transmiterea și pentru recepționarea semnalelor ultrasonice. Senzorul BPU-1640IOAH12 (Bestar Electronics, 2006) a fost selectat, datorită

caracteristicilor sale convenabile, care includ costuri reduse, operare bidirecțională, frecvență nominală de 40 kHz și tensiune maximă de intrare de 120 Vpp. Partea de comutare la nivelul traductor (operație bidirecțională), a fost implementată cu ajutorul circuitelor MOSFET Si4894DY.

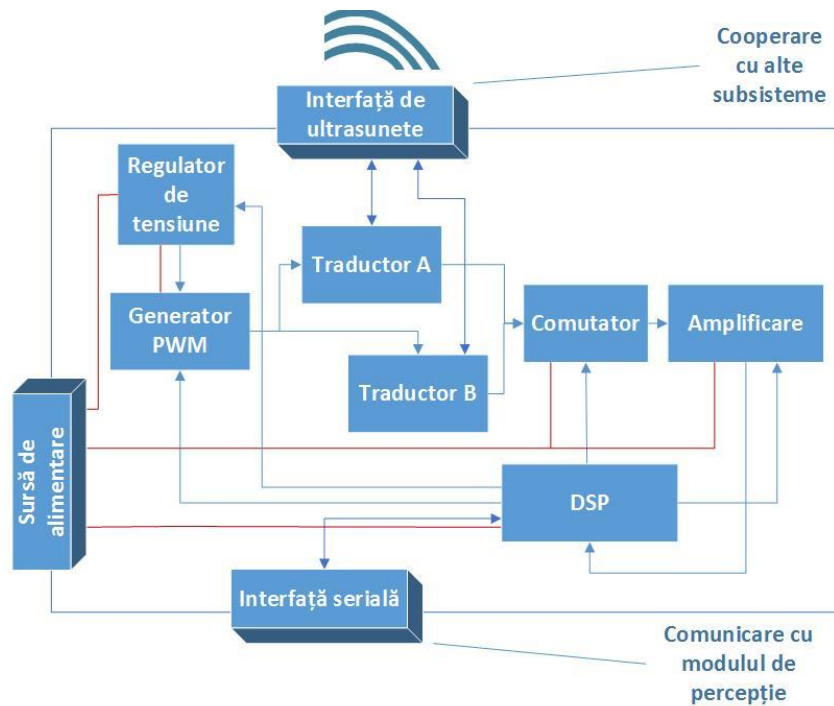


Figura 3-4 Schema dispozitivului IRULT [96]

3.3.2 Proiectarea dispozitivului IRULT

Sistemul de poziționare relativă propus se bazează pe precizia de măsurare a distanței dintre doi roboți.

Presupunem că fiecare robot este echipat cu un modul IRULT. Conform lucrării publicate [98], pentru a măsura distanța dintre doi roboți, roboții trebuie să efectueze cu succes algoritmul de aliniere. Alinierea corectă înseamnă traductoarele ultrasonice de roboți se află față în față, cât mai aproape posibil de linia dreaptă dintre ele. Există câteva surse de erori și probleme în măsurarea distanței pe care s-au luat în considerare în planificarea și efectuarea design-ului IRULT.

3.3.3 Problema atenuării semnalului

Să presupunem că două traductoare sunt aliniat corect și sunt situate la o distanță de 20 cm unul de altul. Primul traductor transmite o undă acustică la o frecvență de 40 KHz, prin aplicarea unui semnal de tip BURST care este format din 8 impulsuri cu amplitudinea $\pm 10V$. Prin urmare, cristalul piezoelectric este contractat cu o tensiune în timpul $8 \times 25\mu s = 200\mu s$. Cel de-al doilea traductorul recepționează undele și convertește înapoi în tensiune. Semnalul convertit se măsoară cu ajutorul unui osciloscop (Figura 3-5). Testul a fost executat într-o cameră la temperatura de $25^\circ C$ și umiditate 36,3%, măsurată cu un termometru digital.

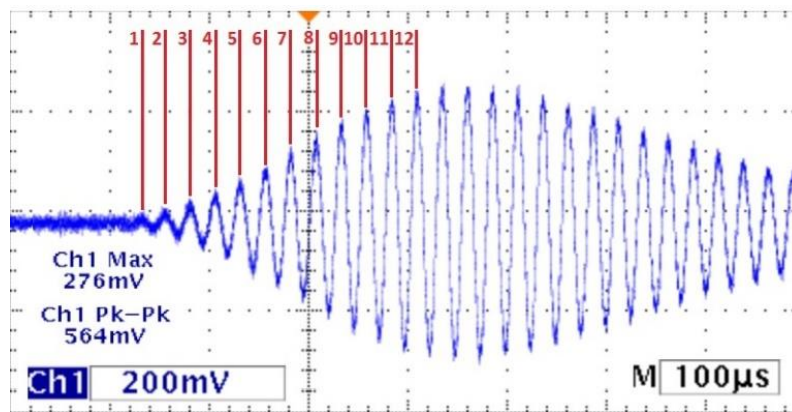


Figura 3-5 Obținerea timpului TOF aplicând praguri diferite de tensiune [96]

În Figura 3-5 se observă că semnalul recepționat are o durată mai mare de $500\mu s$. Distanța este obținută prin aplicarea formulei:

$$d = c_{air} \cdot t \quad (3-1)$$

unde $c_{air} = 345$ m/s este viteza undelor acustice în aer, la temperatura camerei și la presiune normală. Aplicând praguri diferite pentru detectarea semnalului așa cum se arată în Figura 3-5, se pot obține valori diferite pentru măsurarea distanței între două traductoare ultrasonice (Tabelul 3-1).

Semnalele ultrasonice se atenuază mult mai rapid decât sunetul audibil în propagarea prin aer. Prin urmare, prin detectarea semnalului de ultrasunete atenuat, obiectivul pare puțin mai departe decât este de fapt. Există în literatura de specialitate o metodă de a cuantifica această eroare [99] dar oricum, folosind metoda pragului de măsurare a distanței, momentul potrivit nu poate fi detectat exact. Prin aplicarea metodei de cuantificare de eroare cu metoda pragului, este posibilă detectarea momentul potrivit aproximativ într-un interval. În cazul nostru, folosind frecvența 40 KHz, intervalul posibil este de $\pm 25\mu s$ ceea ce este echivalent, în exemplul de mai sus, cu $\pm 8,7$ mm. Măsurarea distanței în acest caz (folosind metoda pragului) poate fi îmbunătățită prin măsurători repetate. În lucrarea [98] prin aplicarea filtrului Kalman pentru măsurători repetate, a fost atinsă o acuratețe de 1 cm la distanțe de până la 3m.

Tabelul 3-1 Măsurarea distanței aplicând praguri diferite de tensiune [96]

Nr.	Prag [mV]	Durăță [μs]	Distanță (măsurată) [mm]
1	14	57.7	20
2	26	82.7	29
3	48	107.7	37
4	67	132.7	46
5	92	157.7	55
6	113	182.7	63
7	173	207.7	72
8	188	232.7	81
9	207	257.7	89
10	225	282.7	98
11	253	307.7	107
12	262	332.7	115

Distanța poate fi obținută cu o acuratețe mai bună dacă maximul (vârful semnalului) ar putea fi localizat. De exemplu, în Figura 3-5 Obținerea timpului TOF aplicând praguri diferite de tensiune, vârful nu poate fi detectat prin metoda pragului deoarece variația de amplitudine în această zonă este relativ scăzută. Mai mult decât atât, un vârf mic de zgomot ar putea produce un maxim fals. O metodă bună de detecție a maximului este intercorelația, definită în [69], după cum urmează:

$$r_{xy}[d] = \sum_{n=-\infty}^{\infty} x[n] \cdot y[n-d], \quad d \in Z \quad (3-2)$$

În [100], folosind intercorelația, distanțele au fost obținute cu o acuratețe mai mică de 0,5 mm în unele cazuri.

3.3.4 Problema alinierii

Erorile de măsurare a distanței pot fi cauzate de o aliniere incorectă a traductoarelor ultrasonice. Rezultatele experimentale din [86] arată că, prin creșterea unghiului de aliniere, crește și eroarea de măsurare a distanței. Toate rezultatele au fost obținute folosind metoda pragului de detecție a semnalului ultrasonic. Recepționare atenuată a ultrasunetelor poate fi un efect al alinierii incorecte.

Autorul tezei a realizat un experiment de efect combinat al orientării emițătorului respectiv orientării receptorului (a se vedea Figura 3-6) și a măsurat atenuarea semnalului. Experimentul a fost executat într-o cameră la temperatura de 26°C și de umiditate 38,3%, măsurată cu un termometru digital. În experiment, au fost utilizați doi roboți: w_1 și w_2 . Distanța dintre roboți a fost 3m. În Figura 3-7 este prezentat un rezultat al acestui experiment. A fost notată cu 100%, amplitudinea maximă recepționată de către robotul w_1 când roboții au fost aliniați într-o situație ideală ($\alpha_{12} = \varphi_1$ și $\alpha_{21} = \varphi_2$). De asemenea, s-a observat că, odată cu creșterea

distanței, unghiul de vizibilitate scade, prin urmare, turela este introdusă în sistem pentru a găsi maximul atunci când vizibilitatea (între două traductoare) este redusă.

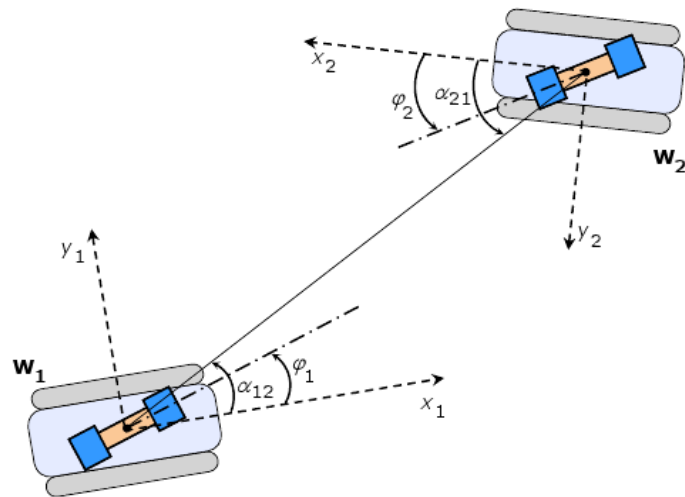


Figura 3-6 Alinierea roboților, privire de sus [98]

$\alpha_{21} - \varphi_2$	+40°	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	+35°	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	+30°	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	+25°	0%	0%	0%	0%	48%	48%	52%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	+20°	0%	0%	56%	62%	76%	58%	56%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	+15°	0%	58%	68%	66%	78%	72%	68%	56%	38%	0%	0%	0%	0%	0%	0%	0%	0%
	+10°	50%	66%	62%	80%	90%	76%	68%	66%	54%	0%	0%	0%	0%	0%	0%	0%	0%
	+5°	58%	60%	70%	94%	88%	82%	94%	60%	50%	0%	0%	0%	0%	0%	0%	0%	0%
	0°	60%	68%	82%	86%	100%	96%	90%	66%	62%	0%	0%	0%	0%	0%	0%	0%	0%
	-5°	56%	60%	86%	86%	96%	94%	76%	68%	56%	0%	0%	0%	0%	0%	0%	0%	0%
	-10°	50%	58%	70%	86%	90%	84%	82%	64%	54%	0%	0%	0%	0%	0%	0%	0%	0%
	-15°	48%	50%	70%	76%	94%	90%	70%	52%	48%	0%	0%	0%	0%	0%	0%	0%	0%
	-20°	0%	0%	62%	70%	72%	68%	74%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	-25°	0%	0%	48%	68%	74%	52%	54%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	-30°	0%	0%	0%	52%	64%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	-35°	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	-40°	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
		-40°	-30°	-20°	-10°	0°	+10°	+20°	+30°	+40°								
		$\alpha_{12} - \varphi_1$																

Figura 3-7 Efect combinat al unghiurilor, rezultat experimental [96]

3.3.5 Performanța de aliniere

Au fost efectuate simulări pentru procedura de aliniere, în care au fost considerate toate valorile între 0° - 180° pentru unghiul φ_1 (definit de axa Ox_1 și axa de senzori cu ultrasunete - IRULT a robotului w_1). Procedura de aliniere a fost simulată în două moduri. În primul rând, w_1 și w_2 au avut aceeași direcție de rotație a turelei. Pe de altă parte, a fost simulată procedura de aliniere cu direcții opuse de rotație pentru roboții w_1 și w_2 . De asemenea, s-au luat în considerare toate valorile între 0° - 180° pentru unghiul φ_2 și au fost generate toate combinațiile posibile. φ_2 este unghiul definit de axa Ox_2 și axa sensorului ultrasonic w_2 . A fost considerat un raport de turație a motorului (1/3), cel mai bun raport de viteză obținut în studiul publicat în [98]. Rezultatele sunt reprezentate în Figura 3-8.

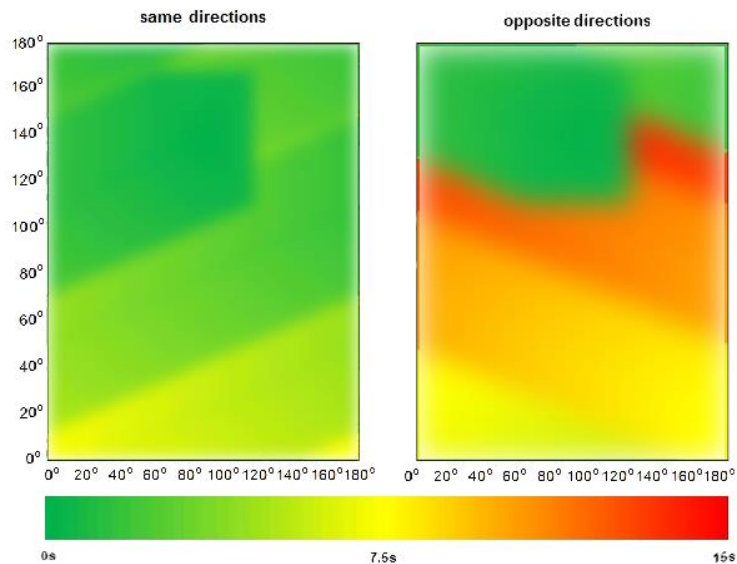


Figura 3-8 Procedura de aliniere: timpul de aliniere în raport cu unghiurile φ_1 și φ_2 [101]

În Figura 3-8 se observă că atunci când motoarele pas cu pas sunt rotite în aceeași direcție, performanța de aliniere este mai bună. În continuare, simulările au fost extinse și publicate în lucrarea „Simulator Based Study of Robot Alignment and Localization” [102].

3.3.6 Reper pentru aliniere

IRULT conține un opritor mecanic pentru calibrarea unghiului φ în raport cu axa Ox (de orientare) a robotului. Acest opritor (Figura 3-3) este situat la -120° în

raport cu axa Ox. Opritorul este detectat prin metoda Back-EMF. BEMF (back electromotive force) este dată în [103] astfel:

$$\text{bemf} = -N \cdot B \cdot A \cdot \omega \cdot \sin(\omega t) \quad (3-3)$$

unde, N este numărul de spire ale bobinei motorului pas cu pas, B reprezintă câmpul magnetic, A este zona cuprinsă în câmpul magnetic a motorului, ω este viteza unghiulară. Parametrii N, B și A sunt constante specifice construcției motorului. BEMF este direct proporțională cu viteza motorului.

3.4 Algoritmul de aliniere a roboților

Înainte de a calcula distanța dintre doi roboți, aceștia vor efectua în prealabil procedura de aliniere LOS (Line of Sight), care se bazează pe directivitatea undelor mecanice (Figura 3-9). În timp ce cei doi roboți își rotesc turela cu viteze diferite, cu un raport între 1/2 și 1/3 (stabilit la inițializarea sistemului, în așa fel încât performanța procedurii de aliniere să fie optimă), robotul receptor determină puterea semnalului de ultrasunete care se propagă de la robotul transmițător. Turelele celor doi roboți se pot roti în aceeași direcție sau în direcții opuse.

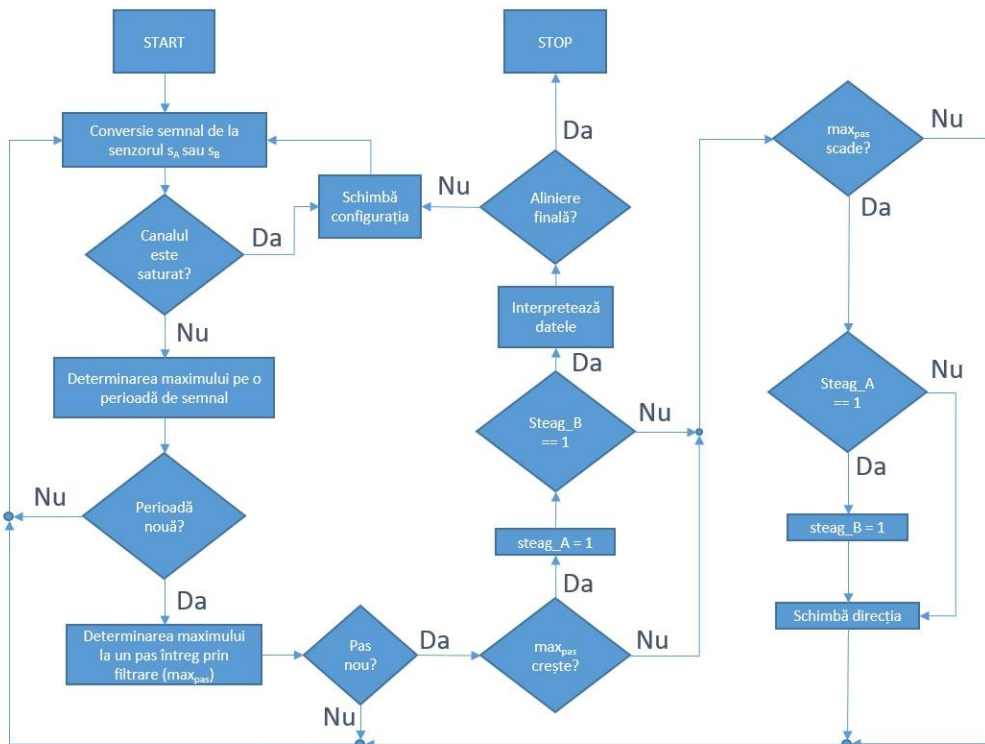


Figura 3-9 Algoritmul de aliniere [97]

La inițializarea sistemului, sensul de rotație este ales aleatoriu, iar apoi se stabilește după fiecare ciclu de interpretare a datelor (semnalelor recepționate). Pentru fiecare perioadă a semnalului ultrasonic, se determină valoarea maximă recepționată iar apoi, pentru fiecare pas întreg de rotație a motorului pas cu pas, se obține puterea semnalului mecanic prin filtrarea maximelor obținute. Dacă se simte creșterea puterii semnalului față de pasul de rotație anterior, se va continua rotirea până va fi întâlnită o scădere a puterii semnalului. Apoi, turelele celor doi roboți vor schimba direcțiile de rotație pentru a reveni la poziția anterioară. Un caz special este considerat atunci când turelele celor doi roboți sunt deja aliniate înainte de începerea procedurii de aliniere. În acest caz, se simte de la început o scădere a puterii semnalului recepționat și se vor modifica direcțiile de rotație a turelelor pentru a găsi din nou maximul. Deciziile care se iau în urma interpretării datelor sunt următoarele: sensul de rotație, viteza de rotație, inversarea rolurilor de transmițător și receptor, stabilirea granițelor de căutare a puterii semnalului (rotația este redusă la un interval mai restrâns), schimbarea amplitudinii semnalului generat la transmisie, schimbarea factorului de amplificare la recepție și în final decizia de a opri procedura de aliniere.

3.5 Măsurarea distanței folosind metoda MTDOA

Metoda MTDOA (Modified Time Difference of Arrival) aplicată în cazul modelului WIT, inițial, constă în transmiterea unui mesaj prin modulul de comunicare Xbee de către un WIT C, care are rolul de coordonator temporar. Acest mesaj se transmite simultan către alte două WIT-uri: A și B. Se presupune că aceștia se află la o anumită distanță depărtare unul de celălalt și că au efectuat un procedura de aliniere. WIT-ul A, după ce recepționează semnalul de la coordonator, transmite un semnal ultrasonic către WIT-ul B. WIT-ul B, când recepționează semnalul de la coordonator, dă drumul la un contor, iar în momentul în care recepționează semnalul ultrasonic de la WIT-ul A, oprește contorul. Pe baza acestui timp se calculează distanța între WIT-ul A și WIT-ul B, aplicând formula:

$$d = c_{\text{aer}} (\Delta t - \text{offset}) \quad (3-4)$$

unde,

d – distanța între WIT-ul A și WIT-ul B

c_{aer} – 345 m/s, reprezintă viteza de deplasare a sunetului în aer

Δt – timpul de recepție a semnalului de ultrasunete (acest timp reprezintă diferența de timp între momentul în care a fost recepționat semnalul ultrasonic și momentul în care a fost recepționat semnalul radio prin modulul Xbee)

offset – reprezintă o valoare de 0,352ms obținută experimental, care se datorează întârzierilor provocate de protocoalele modulului Xbee

3.6 Măsurarea distanței folosind metoda CTOF

Metoda CTOF (Combined Time of Flight) este necesară în cazul calculării distanței între doi roboți mobili, fără a mai fi nevoie de un robot coordonator, aceasta constituie un avantaj față de metoda MTDOA. Calculul distanței cu metoda CTOF, nu depinde de întârzierile provocate de modulele Xbee. În Figura 3-10 este prezentată secvența în timp a procedurii de calcul a distanței cu metoda CTOF.

Distanța între doi roboți se poate calcula prin două metode, robotul A (rol de transmițător) inițiază, sau robotul B (rol de transmițător) inițiază procedura de calcul a distanței. Ambele metode sunt necesare pentru ca ulterior să poată fi comparate rezultatele. Dacă rezultatele diferă foarte mult, se repetă procedura de calcul. În Figura 3-10, robotul A inițiază procedura de calcul a distanței. Acesta transmite un semnal radio robotului B, cu intenția să calculeze distanța față de acesta (Xbee message 1). Robotul B confirmă că poate să înceapă procedura de calcul (Xbee message 2). Astfel robotul B pornește un task de $1\mu\text{s}$ în care ascultă semnalul ultrasonic care va fi transmis de robotul A (start ADC read task). Când ajunge confirmarea la robotul A, acesta pornește un contor (START counter prin care vrea să măsoare timpul total Δt de derulare a procedurii de calcul a distanței) după care transmite semnal ultrasonic către robotul B. Când robotul B recepționează semnalul ultrasonic, acesta va executa *delay1* (valoare experimentală = 13ms), cu o durată suficient de mare astfel încât să cuprindă recepționarea întregului semnal ultrasonic transmis de către robotul A. După acest timp acesta transmite un mesaj către robotul A (message 3) după care mai execută un *delay2* (valoare experimentală = 7ms) timp maxim în care mesajul prin modulele Xbee ajunge la robotul A. Robotul A, când recepționează mesajul 3, pornește un task de $1\mu\text{s}$ în care ascultă semnalul de ultrasunete. Robotul B, după ce a efectuat cele două *delay*-uri, transmite un semnal ultrasonic către robotul A. În momentul în care robotul A recepționează semnalul ultrasonic, acesta oprește contorul și stabilește valoarea Δt .

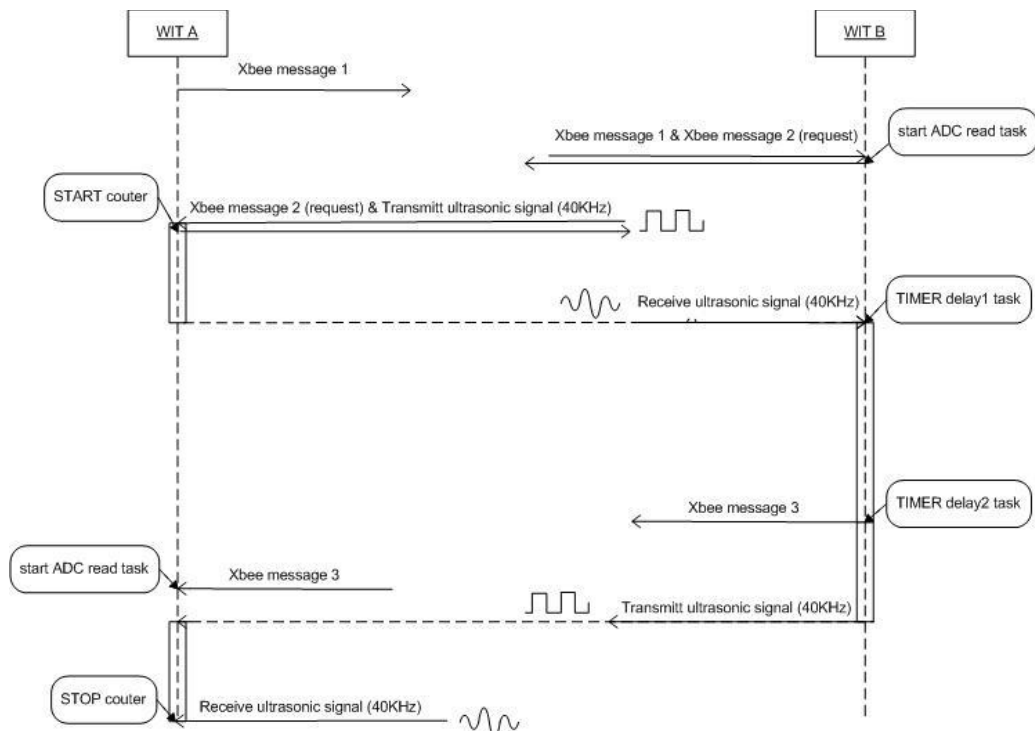


Figura 3-10 Calculul distanței cu metoda CTOF [97]

Ca și rezultat, intervalul Δt conține două valori predefinite $delay1$ și $delay2$ și dublul timpului de propagare a semnalului ultrasonic între robotul A și robotul B. Pe baza acestui timp se poate calcula distanța între A și B, aplicând formula:

$$d = c_{aer} [\Delta t - (delay1 + delay2)] / 2 \quad (3-5)$$

unde,

d – distanța între WIT-ul A și WIT-ul B

c_{aer} – 345 m/s, reprezintă viteza de deplasare a sunetului în aer

Δt – valoarea contorului executat de robotul care inițiază procedura de aliniere

$delay1$ – timpul asigurat în care semnalul ultrasonic se pierde

$delay2$ – timpul asigurat în care semnalul radio este recepționat de la momentul în care a fost transmis

3.7 Rezultate experimentale

Mai multe experimente au fost efectuate în laboratorul DSPLabs folosind în total trei module IRULT. De asemenea, experimentele au fost publicate în lucrarea "Indoor Inter-Robot Distance Measurement in Collaborative Systems" [104]. Configurarea experimentului a constat în trei roboți mobili, dintre care doi au fost aleși la întâmplare pentru a efectua calculul distanței pentru fiecare experiment. Roboții au fost plasați la o distanță variind de la 100 mm până la 3000 mm și, pentru fiecare 10 mm în acest interval, s-au efectuat o serie de 50 de perechi de măsurători, atât cu metoda MTDOA cât și cu metoda CTOF. Înainte de fiecare măsurătoare, roboții au fost poziționați în direcții aleatoare în raport unul cu altul, pentru a verifica dacă procedura de aliniere funcționează.

Deoarece tehnicile propuse se bazează pe sonar și sunt proiectate special pentru măsurători de interior, experimentele, evaluările și rezultatele sunt obținute și considerate la valorile normale ale parametrilor unei camere cum ar fi temperatura, umiditatea, presiunea, etc. Acești parametri ar putea influența viteza undelor ultrasonice utilizate în ecuațiile (3-4) și (3-5). Figura 3-11 exemplifică recepția semnalului de ultrasunete pentru al doilea canal al modulului sonar - receptor.

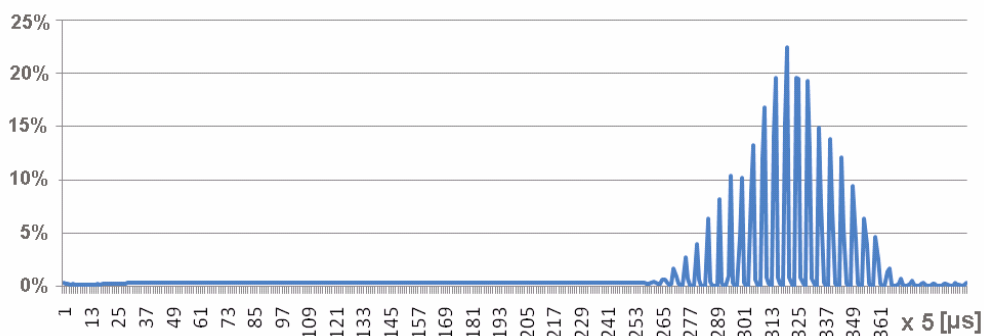


Figura 3-11 Semnalul ultrasonic recepționat [104]

Diferența dintre timpii de primire de pachete pentru modulele XBee ale celor doi roboți implicați în procedurile de măsurare a distanței a fost măsurată cu un analizor logic. Această diferență variază în mod aleatoriu de fiecare dată, în mare parte ca urmare a modului de funcționare a stivei de protocol a modulelor Xbee corespunzătoare. S-a obținut o valoare maximă de 86 μs , ceea ce corespunde cu măsurarea distanței de 30 mm.

Figura 3-12 și Tabelul 3-2 prezintă rezultatele experimentale obținute prin metoda MTDOA de măsurare a distanței, în timp ce Figura 3-13 și Tabelul 3-3 reprezintă rezultatele pentru metoda CTOF. De asemenea, s-a efectuat o analiză statistică a datelor. Erorile maxime absolute și relative cu ambele tehnici sunt descrise, în mod comparativ, în Figura 3-14.

Tabelul 3-2 Rezultatele măsurării distanței pentru metoda MTDOA [104]

Distanța reală [mm]	Distanța măsurată [mm]			Durata procedurii [μs]
	Min	Media	Max	
100	36	71	101	559
200	164	198	243	929
300	281	324	358	1296
400	394	423	470	1584
500	494	522	554	1872
600	546	602	661	2105
700	674	705	736	2405
800	789	817	855	2731
900	882	919	966	3028
1000	937	1003	1073	3273
2000	1931	1983	2046	6127
3000	2948	2978	3023	9024

Tabelul 3-3 Rezultatele măsurării distanței pentru metoda CTOF [104]

Distanța reală [mm]	Distanța măsurată [mm]			Durata procedurii [μs]
	Min	Media	Max	
100	92	96	99	20849
200	199	201	207	21461
300	298	300	303	22037
400	401	404	410	22643
500	504	508	515	23249
600	604	607	612	23825
700	700	706	710	24402
800	803	807	813	24990
900	906	911	916	25596
1000	1013	1019	1026	26225
2000	2024	2033	2043	32130
3000	3018	3031	3047	37943

Pentru metoda MTDOA, eroarea maximă absolută a fost obținută atunci când cei doi roboți au fost poziționați la o distanță de 100 cm unul de celălalt și are o valoare de 7,3 cm. În cazul metodei CTOF, eroarea absolută maximă este de 4,8 cm, și a fost obținută atunci când roboții au fost poziționați la o distanță de 300 cm.

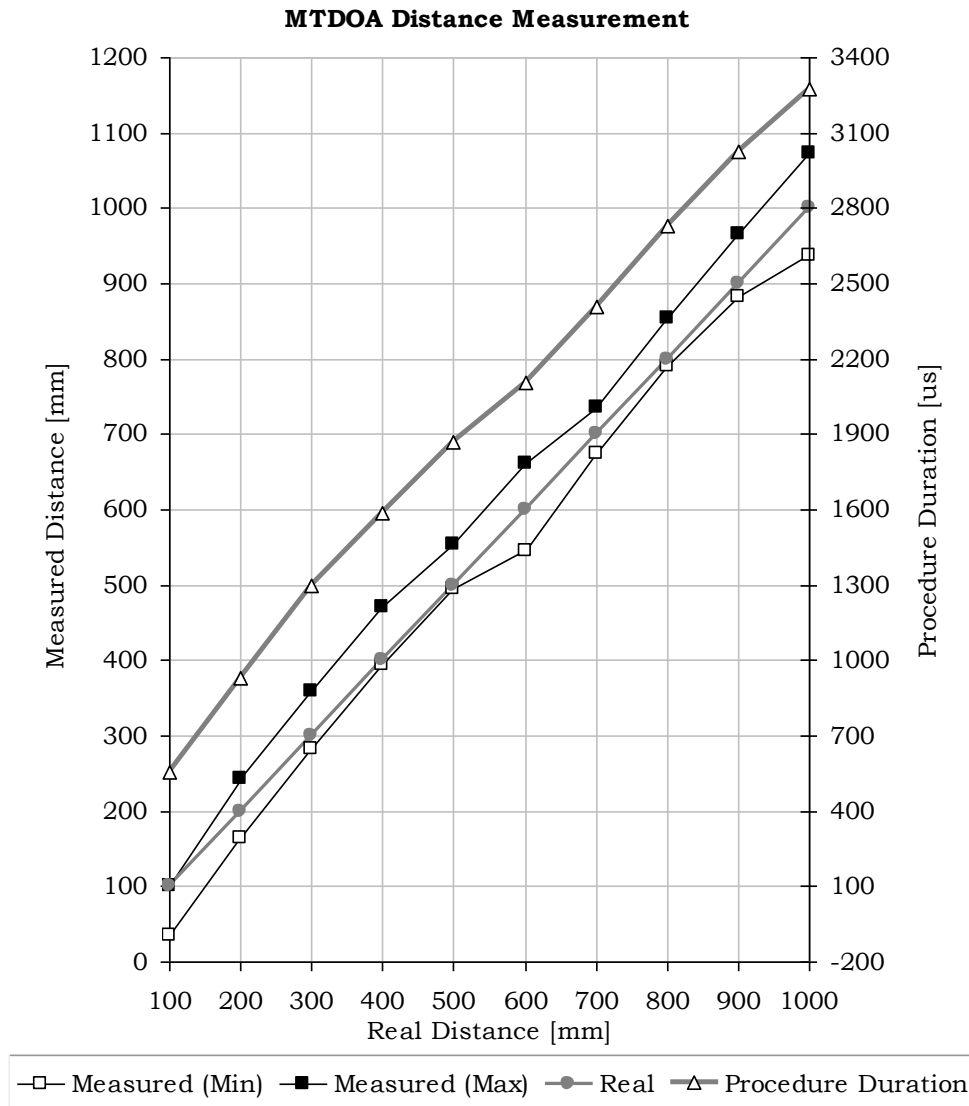


Figura 3-12 Distanța măsurată respectiv distanță reală și durata procedurii pentru metoda MTDOA [104]

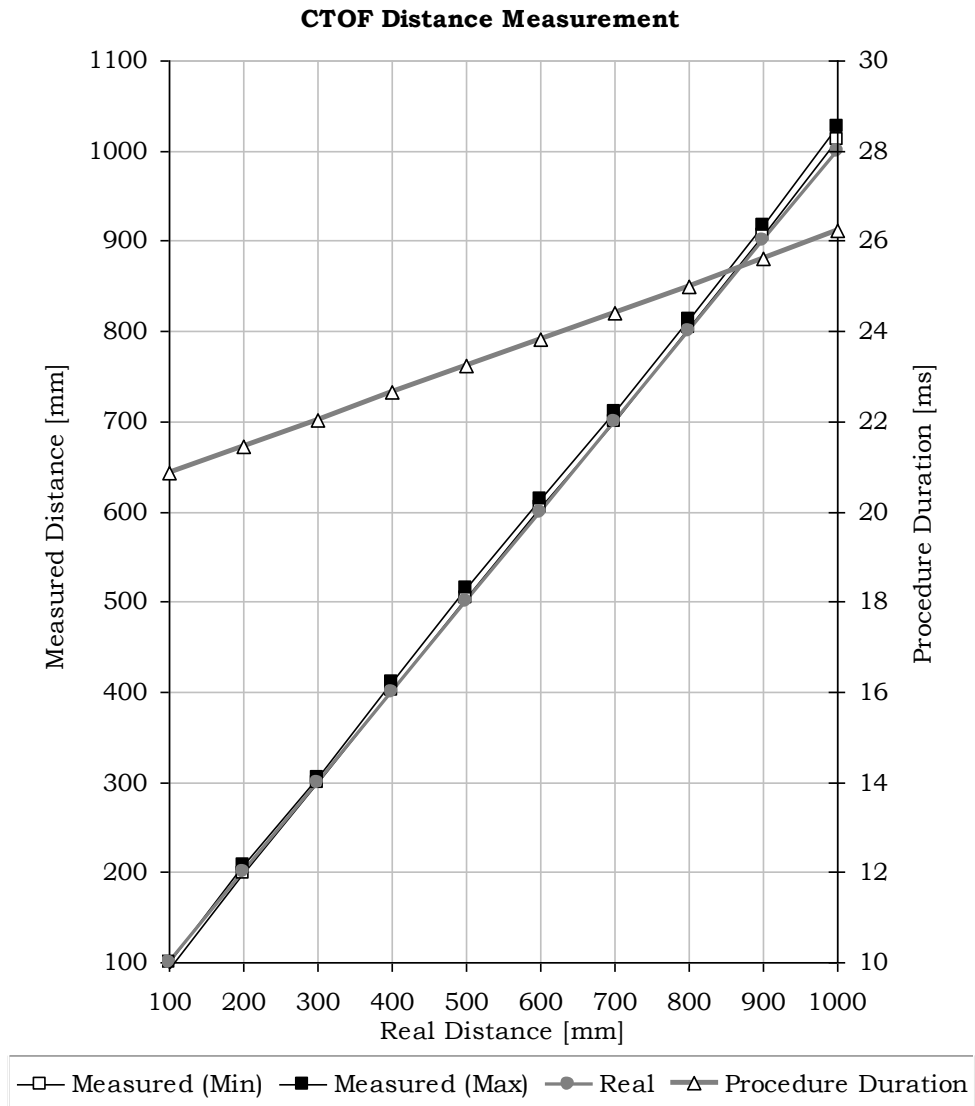


Figura 3-13 Distanța măsurată respectiv distanță reală și durata procedurii pentru metoda CTOF [104]

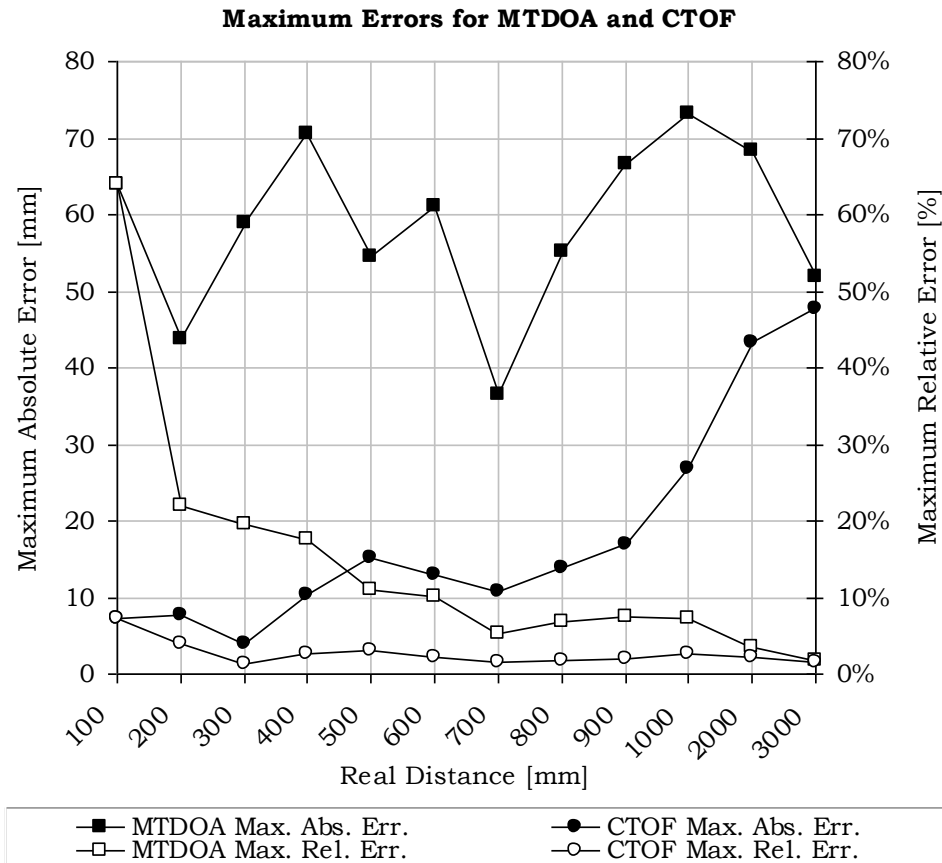


Figura 3-14 Compararea erorilor maxime (absolute și relative) pentru tehnicile MTDOA și CTOF [104]

3.8 Concluzii

În acest capitol autorul tezei a expus sistemul hardware pentru poziționare relativă, IRULT, cu un set comun de cerințe pentru sistemul robotic țintă CORE-TX. Analizând modulele destinate localizării în literatura de specialitate și observând problemele acestora, s-a ajuns la o propunere de modul nou destinat mediilor colaborative robotizate, care are ca scop principal calculul distanței între doi roboți mobili. De asemenea, modulul poate să deducă și unghiul dat de cei doi roboți. Acest lucru se realizează prin procedura de aliniere. Astfel, s-a prezentat algoritmul de aliniere și s-a discutat despre două tehnici de măsurare a distanței pentru MCR (Medii colaborative robotizate). S-a arătat cum MTDOA și metodele CTOF îndeplinesc cerințele specificate, cost redus, aplicații de măsurare a distanței în MCR, eficiență energetică, fără a fi nevoie de o preinstalare.

Rezultatele experimentale arată că, deși metoda MTDOA generează erori absolute relativ ridicate, valoarea medie este mai apropiată de distanța reală și are o

evoluție liniară, acestea fiind obținute după ajustările de calibrare corespunzătoare. Mai mult decât atât, erorile maxime relative tind să scadă odată cu distanța măsurată. Aceasta este o altă indicație care arată că întârzierile provocate de modul de operare a modulelor Xbee contribuie predominant la influențarea rezultatelor experimentale. O îmbunătățire a acestei tehnici ar putea fi să se efectueze măsurători în ambele direcții pentru o anumită poziționare a robotului, iar în cazul în care rezultatele diferă în mod semnificativ, procedura ar trebui reluată.

Pe de altă parte, metoda CTOF se comportă mult mai bine. După calibrările necesare, caracteristicile sale de măsurare sunt liniare iar rezultatele măsurătorilor sunt mult mai apropiate de distanța reală. Prin aceasta se arată independența acestei tehnici de întârzierile aleatorii introduse de către modulele de comunicare Xbee.

4. METODOLOGIA DE LOCALIZARE PROPUȘĂ

În acest capitol se prezintă metodologia de localizare relativă propusă. Aceasta se bazează pe trei niveluri: 1-PREDICȚIE, 2-COOPERARE și 3-CENTRALIZARE. Cele trei niveluri sunt detaliate în capitolele 5 și 6. Tot în cadrul acestui capitol sunt redate următoarele noțiuni ce fac parte din metodologie: concepte de bază, măsurarea unghiului, inițializarea sistemului, metoda LOAD și localizarea bilaterală. De asemenea este prezentat un set extins de experimente într-un studiu de caz efectuat în cadrul laboratorului DSPLabs, pentru a evalua metodologia de localizare propusă. În urma experimentelor se constată că este nevoie de mai multe seturi de experimente care trebuie să fie efectuate prin simulare pentru a cuprinde toate cazurile izolate ce pot apărea în practică.

4.1 Introducere

Metodologia de localizare propusă este alcătuită din trei niveluri:

- primul nivel (PREDICȚIE) constă în localizarea roboților mobili pe baza predicției de mobilitate, la acest nivel fiecare robot mobil se auto-localizează în funcție de resursele proprii de mobilitate și navigare prin procesare locală;
- al doilea nivel (COOPERARE) cuprinde o clasă de tehnici și reguli prin care un robot mobil se localizează prin procesare distribuită și cooperare cu roboții mobili din vecinătatea proprie, limitat de distanța maximă de cooperare;
- al treilea nivel (CENTRALIZARE) are la bază un nod central (BRAIN) cu rolul de a centraliza coordonatele și de a gestiona mobilitatea roboților mobili astfel încât să păstreze superior coeficientul de acuratețe a localizării.

Bazată pe arhitectura CORE-TX, această metodologie este asociată modelului CORE-TX [16], astfel: primul nivel (PREDICȚIE) și al doilea nivel (COOPERARE) sunt asociate stratului de percepție și operare al modelului CORE-TX, iar al treilea nivel (CENTRALIZARE) este asociat stratului de control și supraveghere (Figura 4-1).

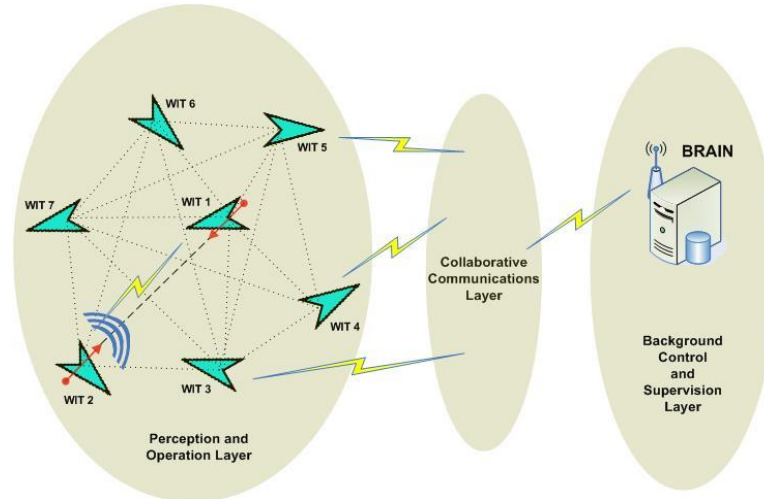


Figura 4-1 Modelul CORE-TX [98]

4.2 Definirea unor concepte de bază

Prin asocierea cu modelul CORE-TX, se consideră nodul central BRAIN. Primul rol al nodului BRAIN este de a inițializa sistemul. BRAIN este considerat și un centralizator de date.

Se consideră o rețea compusă din n noduri (WIT-uri), $W = \{w_1, w_2, \dots, w_n\}$, fiecare nod poate comunica prin semnale ultrasonice la o rază de distanță d sau prin unde radio la rază de distanță r . Există două tipuri de noduri: *necunoscute* și *stabilite* (noduri ancoră). Nodurile *necunoscute* sunt setul de noduri care nu își cunosc orientarea și localizarea. Scopul principal al sistemului de localizare este de a obține localizarea pentru fiecare nod *necunoscut*. Nodurile *stabilite* au fost inițial noduri *necunoscute*.

Fiecare robot are o zonă de vizibilitate de ultrasunete (de rază d). Aceasta este zona verde în care un robot se poate alinia cu alți roboți folosind algoritmul de aliniere (3.4 Algoritmul de aliniere a roboților). Fiecare robot gestionează un anumit set de liste. Lista verde descrie prezența altor roboți în zona verde. Cu alte cuvinte, doi roboți ar putea efectua procedura de aliniere dacă apar reciproc în listele lor verzi. Fiecare robot are o zonă de vizibilitate RF (semnal radio) de rază r . Aceasta este zona galbenă, în care un robot poate comunica cu alți roboți cu mesaje RF, dar nu poate efectua procedura de aliniere. Există, de asemenea, o listă galbenă care descrie prezența altor roboți în zona galbenă. În cele din urmă, orice robot care părăsește zona verde și zona galbenă este considerat în zona roșie și este înregistrat în lista roșie. (Figura 4-2)

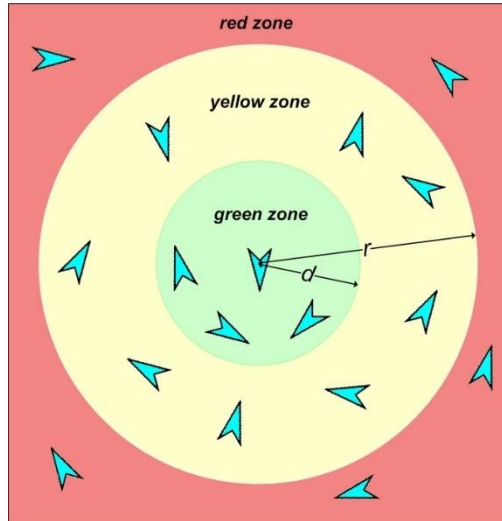


Figura 4-2 Zona roșie, galbenă și verde

Fiecare nod i are o pereche de coordonate, $(x_i, y_i) \in R^2$. Fiecare robot se consideră într-un sistem local de coordonate raportat la locația acestuia. De aceea, pentru n noduri, există în total $n + 1$ sisteme de coordonate. Sistemul de coordonate adițional este denumit axa de referință sau axa *home* și este determinată la inițializarea sistemului. Nodul BRAIN salvează toate coordonatele în raport cu axa *home*.

Fiecare nod *ancoră* are o axă de referință paralelă cu axa de origine (axa *home*). Fiecare nod *necunoscut* are axa de orientare, aliniată cu propria direcție. Notăm cu θ unghiul care translatează poziția nodului *necunoscut* în raport cu axa stabilită paralelă cu axa *home*. Orientarea nodurilor este dată de unghiul θ . Notăm cu α_{mn} unghiul dintre axele de orientare și linia care unește doi roboți, m și n . În literatura de specialitate, unghiul α este denumit „bearing angle” iar unghiul θ este denumit „heading angle” [105]. Abordări în spațiul 3D folosesc pentru α denumirea „azimuth angle” [106]. De asemenea, vom nota cu β_{mn} unghiul dintre axele de referință și linia de legătură a doi roboți. Se definește o proprietate specială pentru locația roboților, denumită *confiență* de localizare, $\xi \in \mathbb{N}$. Toate nodurile *necunoscute* au valoarea 0 asociată pentru această proprietate. În cazul în care un robotul are *confiența* mai mare, coordonatele sale sunt considerate mai importante în determinarea poziției.

Pentru a menține cât mai ridicată *confiența* sistemului, localizarea nodurilor necunoscute se realizează cu ajutorul *pânzelor*. O *pânză* este formată din cel puțin două noduri *stabilite*. Locul de conservare a *pânzei* se poate modifica prin mobilitatea *pânzei*. Ruta de mobilitate a unei *pânze*, densitatea *pânzelor* și numărul acestora sunt stabilite în funcție de aplicație, de nodul coordonator BRAIN (dinamic – prin managementul mobilității) sau de utilizator (static – la configurarea sistemului). Pentru a menține valoarea de *confiență* cât mai ridicată, o *pânză* își poate schimba poziția cu ajutorul altei *pânze*, în așa fel încât o *pânză* rămâne în repaos când cealaltă

pânză își modifică poziția. Pentru a păstra superior coeficientul de acuratețe a localizării, numărul de schimbări de poziție trebuie să fie cât mai mic.

Toate pânzele care au poziție fixă în sistem și nu își modifică poziția sunt numite *baze*. Rolul *bazelor* este de a reseta propagarea erorilor în localizare. Prima pânză din sistem care devine bază este chiar pânza cu axa *home*. Poziționarea *bazelor* se realizează printr-o abordare topologică iar localizarea *pânzelor* se realizează printr-o abordare metrică.

Toate nodurile care fac parte dintr-o *bază*, poartă denumirea de noduri *satelit*. *Bazele* ar putea să ofere *pânzelor* informații suplimentare care să îmbunătățească confiența de localizarea a *pânzelor*. De exemplu, un *satelit* ar putea comunica cu un nod *necunoscut* prin semnale UWB, CSS sau altele, astfel, nodurile pot obține distanțe față de sateliți. În acest caz, se poate defini zona albastră ca fiind o zonă în care o *bază* poate comunica cu pânzele din aceeași *clasă*. *Clasa* se referă la toate pânzele care aparțin unei *baze*. Limitarea zonei albastre este dată de distanța maximă de calcul de distanță cu semnale UWB, CSS sau altele. Nu numai nodurile din zona verde ci și celelalte noduri care au alte funcții în sistem pot să își reîmprospăteze poziția relativă cu ajutorul nodurilor *sateliți*.

Ca să fie distincția clară, zona galbenă este prevăzută pentru comunicare prin semnale wireless, dar zona albastră este tot pentru semnale wireless prevăzută, doar că se referă la calculul distanței. Câteodată, în funcție de tehnologiile folosite, cele două zone, galbenă și albastră, ar putea să coincidă.

4.3 Măsurarea unghiului

Măsurarea unghiului este obținută prin identificarea unghiului dintre două noduri. O astfel de măsurare va fi o componentă importantă a sistemului de localizare, deoarece este folosită pentru calculul poziției și, de asemenea, pentru algoritmul de localizare.

Măsurarea unghiului poate fi efectuată numai în zona verde. În Figura 4-3 este prezentat un exemplu în care doi roboți (robotul w_1 și w_2) pot măsura unghiul format de aceștia față de axa robotului w_2 . Robotul w_1 aflat în originea axei x_1Oy_1 , localizează robotul w_2 . Robotul w_2 este considerat în originea axei x_2Oy_2 și de asemenea poate localiza robotul w_1 aplicând aceeași procedură. Unghiul dintre w_1 și w_2 măsurat de primul robot w_1 este notat astfel: α_{12} . Unghiul dintre w_2 și w_1 măsurat de al doilea robot w_2 este notat astfel: α_{21} . Presupunând că w_1 are axele paralele cu axa *home*, acest nod este considerat nod *ancoră*. Al doilea nod w_2 este considerat un nod cu coordonate *necunoscute* (nod *necunoscut*). Aceasta înseamnă că, după ce al doilea robot obține o orientare corectă și localizare raportată la primul robot, nodul w_2 va fi considerat un nod *ancoră*. Al doilea robot va realiza o translație a coordonatelor pentru a obține axe paralele cu axa *home*.

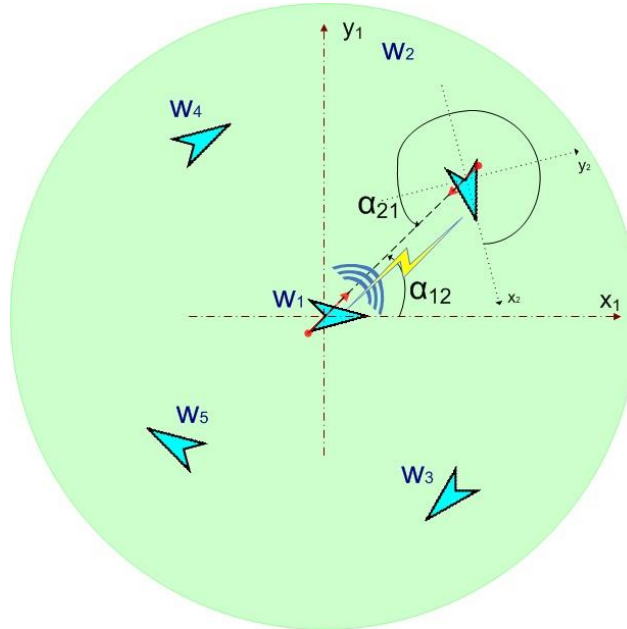


Figura 4-3 Măsurarea unghiului

4.4 Inițializarea sistemului

În general, atunci când vorbim despre aplicații de explorare a mediilor necunoscute, există unele cerințe pentru a inițializa sistemul de localizare. Astfel, este necesar să se stabilească, un punct de referință, numită *poziția de bază*, și un sistem de axe de referință, numite axe de origine (cum a fost discutat anterior). Aceste referințe sunt stabilite prin poziționarea primului robot în sistem. BRAIN are un rol important aici: descoperă primul robot care a fost introdus în sistem. Poziția inițială este poziția acestui robot iar axele de origine stabilite sunt date de orientarea primului robot în sistem.

În Figura 4-4, w_5 este primul robot din sistem. Acest robot trimite un mesaj pentru a descoperi alți roboți în sistem. După primirea mesajului de la robotul w_5 , BRAIN consideră că acesta este primul robot al sistemului și stabilește poziția *home* (O) și axa *home* (xOy). Oricare robot introdus în sistem, începe să transmită un mesaj pentru a descoperii alți roboți în sistem. De exemplu, al doilea robot care intră în sistem, w_1 (momentan *necunoscut*), trimite (broadcast) un mesaj și primește un răspuns de la w_5 . Astfel, w_1 trimite o cerere la w_5 pentru a obține o informație de localizare. Această primă metodă este denumită LOAD (Localization using One Angle and One Distance). Toți roboții încearcă inițial să se localizeze prin metoda LOAD în momentul intrării în sistem. Dacă procedura de poziționare este terminată cu succes, aceștia stabilesc propria confiență și își actualizează lista verde și lista galbenă.

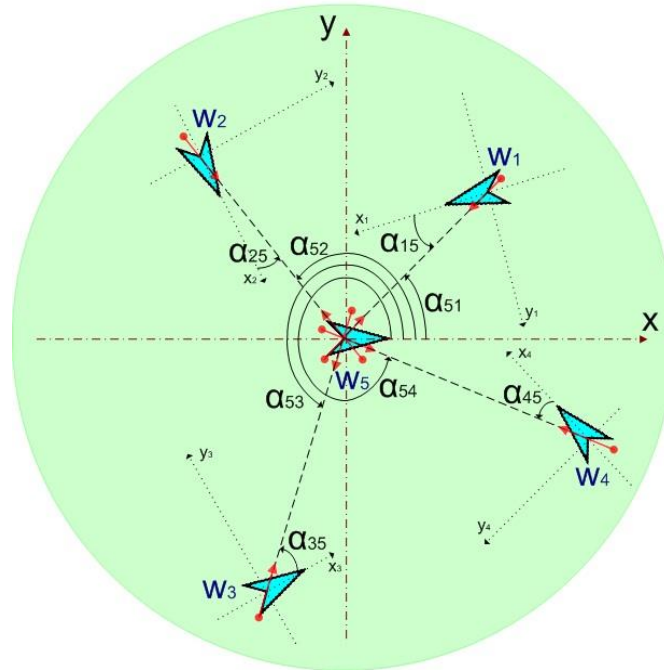


Figura 4-4 Inițializarea sistemului

4.5 Orientarea și localizarea folosind metoda LOAD

Orientarea și localizarea folosind metoda LOAD (Localization using One Angle and One Distance) este prima metodă de a obține orientarea și localizarea nodurilor *necunoscute*. Această metodă începe cu procedura de aliniere dintre doi roboți. Un nod *necunoscut* poate să obțină poziția prin efectuarea unei proceduri de colaborare cu un nod *ancoră* din lista verde. De exemplu, în Figura 4-5, robotul w_5 se află în poziția *home*. Există patru noduri *necunoscute*: w_1 , w_2 , w_3 și w_4 . Robotul w_1 , după ce intră în sistem, începe procedura de aliniere cu robotul w_5 , astfel calculează unghiul între axa Ox_1 și linia ce conectează roboții w_1 și w_5 . Unghiul este notat cu α_{15} . Asemănător, robotul w_5 stabilește unghiul α_{51} . Pentru robotul w_5 este considerat un caz particular unde axa de referință este aceeași cu axa de orientare, deci unghiul θ între aceste două axe este 0. În acest caz, unghiul β_{51} coincide cu α_{51} . După ce procedura de calculare a unghiului ia sfârșit, următoarea etapă care urmează este calculul distanței. În acest caz distanța dintre w_1 și w_5 este d_{15} . Următoarea etapă este obținerea unghiului θ_1 , unghi care translatează axa de orientare în axa de referință. Aceasta se obține folosind formula de mai jos:

$$\theta_1 = (\alpha_{15} - \alpha_{51} + 180^\circ) \pmod{360^\circ} \quad (4-1)$$

Următoarea etapă este obținerea locației robotului $w_1(x_1, y_1)$ raportat la w_5 . Aceasta se calculează folosind formula de mai jos:

$$\begin{cases} x_1 = d_{15} \cos \alpha_{51} \\ y_1 = d_{15} \sin \alpha_{51} \end{cases} \quad (4-2)$$

Etapă finală este actualizarea listei verzi pentru cei doi roboți. Aceasta este realizată la nivelul colaborativ. Robotul w_5 transmite informația $w_1(x_1, y_1)$ robotului w_1 . Această informație este utilă pentru a raporta propria axă (axa robotului w_1) la o axă de referință (axa robotului w_5). Aceasta se realizează calculând unghiul β , unghiul între axa de referință și linia care conectează w_1 cu w_5 . Unghiul β se calculează folosind formula de mai jos:

$$\beta_{15} = (\alpha_{15} + \theta_1) \pmod{360^\circ} \quad (4-3)$$

Se observă că unghiul β_{15} se află în cadranul III, $\beta_{15} \in \left[\pi, \frac{3\pi}{2} \right)$ aceasta înseamnă că, robotul w_1 poate să deducă locația robotului w_5 raportat la propria axă: $w_5(-x_1, -y_1)$. Aceasta reprezintă un caz particular pentru calculul unghiurilor. Pentru a generaliza considerăm cele două noduri: w_1 și w_2 . În acest caz, w_1 este un nod ancoră iar w_2 este un nod necunoscut. Pentru a determina unghiul θ_2 , autorul tezei propune următoarea ecuație bazată pe ecuația (4-1):

$$\theta_2 = (\alpha_{12} - \alpha_{21} + \theta_1 + 180^\circ) \pmod{360^\circ} \quad (4-4)$$

Pentru a determina locația robotului w_2 , robotul w_1 ar putea să calculeze unghiul β_{12} aplicând formula de mai jos bazată pe ecuația (4-3):

$$\beta_{12} = (\alpha_{12} + \theta_1) \pmod{360^\circ} \quad (4-5)$$

Locația robotului w_2 în coordonatele lui w_5 se calculează în funcție de unghiul β_{12} , luând în considerare sistemul de mai jos și ținând cont de faptul că β_{12} se află în cadranul II.

$$\begin{cases} I \begin{cases} x_j = x_i + d_{ij} \cos \beta_{ij} \\ y_j = y_i + d_{ij} \sin \beta_{ij} \end{cases}, \beta_{ij} \in \left[0, \frac{\pi}{2} \right) \\ II \begin{cases} x_j = x_i - d_{ij} \cos(\pi - \beta_{ij}) \\ y_j = y_i + d_{ij} \sin(\pi - \beta_{ij}) \end{cases}, \beta_{ij} \in \left[\frac{\pi}{2}, \pi \right) \\ III \begin{cases} x_j = x_i - d_{ij} \cos(\beta_{ij} - \pi) \\ y_j = y_i - d_{ij} \sin(\beta_{ij} - \pi) \end{cases}, \beta_{ij} \in \left[\pi, \frac{3\pi}{2} \right) \\ IV \begin{cases} x_j = x_i + d_{ij} \cos(2\pi - \beta_{ij}) \\ y_j = y_i - d_{ij} \sin(2\pi - \beta_{ij}) \end{cases}, \beta_{ij} \in \left[\frac{3\pi}{2}, 2\pi \right) \end{cases} \quad (4-6)$$

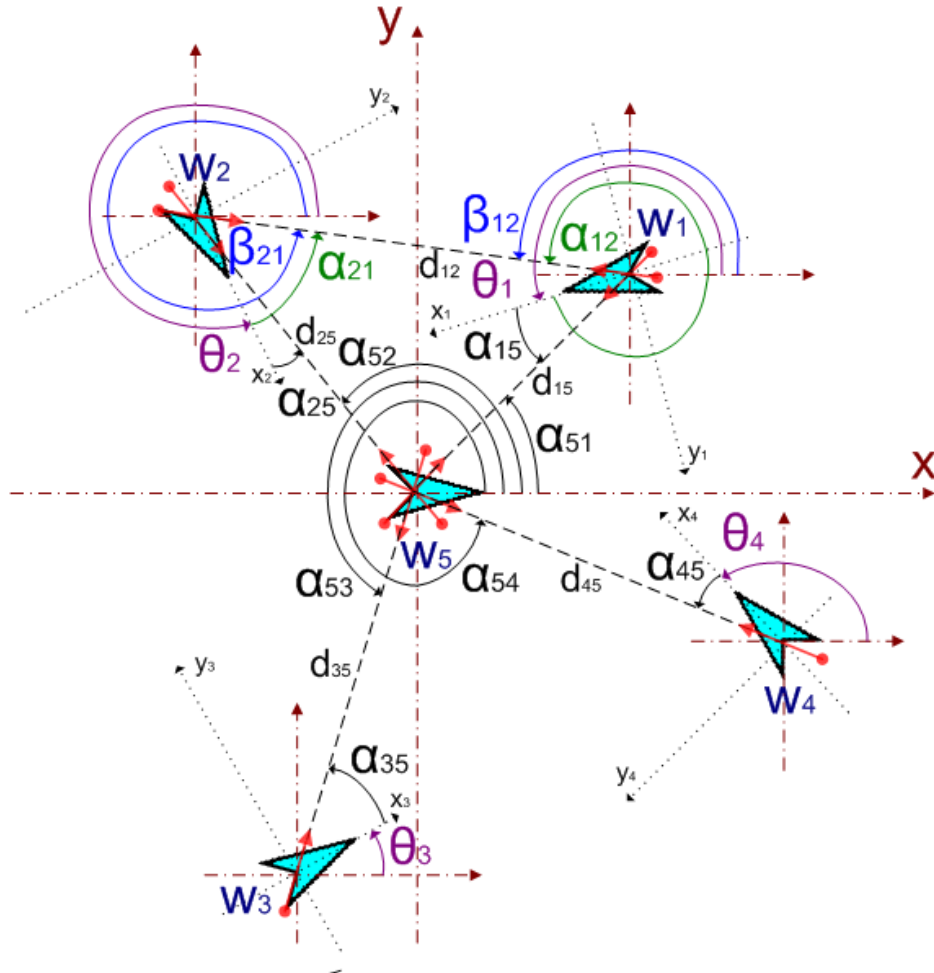


Figura 4-5 Localizarea folosind metoda LOAD

Pentru obținerea locației robotului w_1 în coordonatele lui w_2 folosind sistemul (4-6), robotul w_2 ar putea să calculeze unghiul β_{21} aplicând formula de mai jos bazată pe ecuația (4-3), astfel se deduce că β_{21} se află în cadranul IV și se ține cont că (x_i, y_i) , adică (x_2, y_2) este origine pentru w_2 , deci are coordonatele (0,0):

$$\beta_{21} = (\alpha_{21} + \theta_2) \pmod{360^\circ} \tag{4-7}$$

Sistemul (4-6) este folosit doar la nivel teoretic, în practică acest sistem este înlocuit astfel:

$$\begin{cases} x_j = x_i + d_{ij} \cos \beta_{ij} \\ y_j = y_i + d_{ij} \sin \beta_{ij} \end{cases}, \forall \beta_{ij} \tag{4-8}$$

Din sistemul (4-8) dacă scoatem pe d_{ij} din prima ecuație și înlocuim în a doua, atunci obținem următoarea relație:

$$\beta_{ij} = \tan^{-1} \frac{y_j - y_i}{x_j - x_i} + k\pi, \quad k \in Z \quad (4-9)$$

Sau, ecuația (4-9) o putem scrie sub forma unui sistem care să acopere toate cazurile, astfel:

$$\begin{cases} \beta_{ij} = \tan^{-1} \frac{y_j - y_i}{x_j - x_i}, & x_i > x_j \\ \beta_{ij} = \tan^{-1} \frac{y_j - y_i}{x_j - x_i} + \pi, & x_i < x_j \\ \beta_{ij} = \frac{\pi}{2}, & x_i = x_j \text{ și } y_i > y_j \\ \beta_{ij} = \frac{3\pi}{2}, & x_i = x_j \text{ și } y_i < y_j \\ \beta_{ij} = 0, & x_i = x_j \text{ și } y_i = y_j \end{cases} \quad (4-10)$$

4.6 Localizarea bilaterală

Un robot care are cel puțin doi roboți în lista verde ar putea să calculeze locația folosind metoda de localizare bilaterală. De exemplu, în Figura 4-6, se poate determina locația robotului w_2 raportat la roboții w_1 și w_5 . Considerăm ecuația de mai jos:

$$d(w_1, w_2) = d_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4-11)$$

Conform ecuației (4-11), robotul w_2 poate să obțină două distanțe: d_{12} (distanța dintre w_1 și w_2) și d_{25} (distanța dintre w_2 și w_5). d_{12} și d_{25} sunt obținute aplicând algoritmul de aliniere și procedura de calcul a distanței. După această etapă, robotul w_2 are doi roboți în propria listă verde: w_5 și w_1 . În acest caz, robotul w_2 ar putea să obțină o poziție aplicând metoda de localizare bilaterală.

Metoda de localizare bilaterală este realizată de robotul care are *confiența* cea mai mare. Presupunem că robotul w_5 are *confiența* cea mai mare și că robotul w_5 are două coordonate în lista verde: $w_5 (x_5, y_5)$ și $w_1 (x_1, y_1)$ și a obținut două distanțe comunicând cu w_2 : d_{12} și d_{25} . Aplicând metoda de localizare bilaterală, robotul w_5 poate să obțină coordonatele pentru robotul $w_2 (x_2, y_2)$. Coordonatele pentru robotul w_2 sunt obținute aplicând formulele:

$$y_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (4-12)$$

$$x_2 = s + \frac{y_2(y_1 - y_5)}{(x_5 - x_1)} \quad (4-13)$$

unde,

$$\begin{cases} a = \frac{(y_1 - y_5)^2}{(x_5 - x_1)^2} + 1 \\ b = \frac{2(s - x_5)(y_1 - y_5)}{x_5 - x_1} - 2y_j \\ c = (s - x_5)^2 + y_5^2 - d_{25}^2 \end{cases} \quad (4-14)$$

și

$$s = \frac{d_{12}^2 - d_{25}^2 + x_5^2 + y_5^2 - x_1^2 - y_1^2}{2(x_5 - x_1)} \quad (4-15)$$

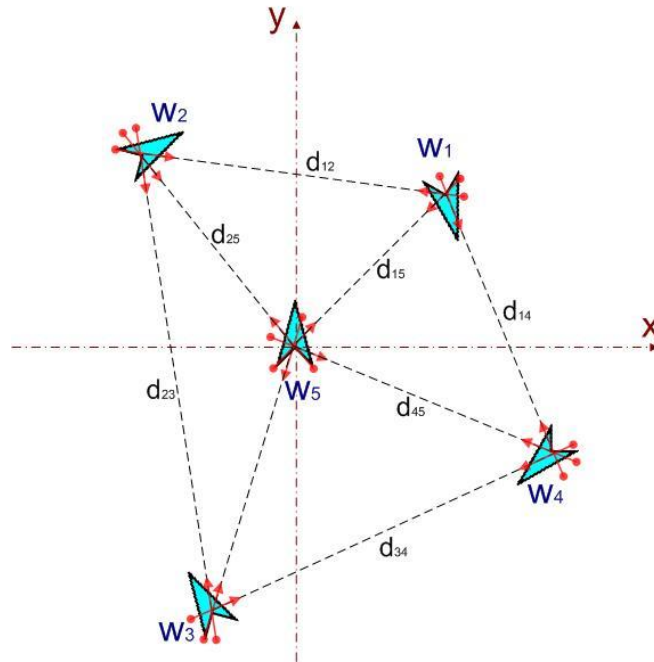


Figura 4-6 Metoda de localizare bilaterală

Există două soluții în ecuația (4-12). Soluția corectă este considerată prin compararea locației obținute prin metoda LOAD. În acest caz robotul w_5 actualizează lista verde adăugând coordonatele robotului w_2 (x_2, y_2) și comunică aceste coordonate roboților din lista verde.

Până când robotul w_5 nu își va modifica locația, acesta va avea cea mai mare *confiență* în sistem. Dacă robotul w_5 se va deplasa, acesta va reseta propria valoare de *confiență* la valoarea 0.

4.7 Rezultate experimentale

Un set extins de experimente au fost efectuate în DSPLabs, pentru a evalua metodologia de localizare propusă într-un studiu de caz. A fost efectuat un experiment cu șase roboți considerați noduri *stabilite* și un robot considerat nod *necunoscut* (Figura 4-7). Robotul cu poziție *necunoscută* (marcat cu roșu) la momentul t_1 a fost localizat în raport cu un alt robot *stabilit* prin metoda LOAD, apoi doi roboți prin metoda de localizare bilaterală cu două repere (momentul t_2) și în cele din urmă trei roboți folosind metoda de localizare bilaterală pentru 3 repere (momentul t_3).

Metoda de localizare bilaterală pentru 3 repere presupune, efectuarea metodei de localizare bilaterală, prima dată pentru 2 repere din cele 3, iar a doua oară alte două repere din cele 3. Se obțin două poziții, iar poziția finală se obține din media celor două.

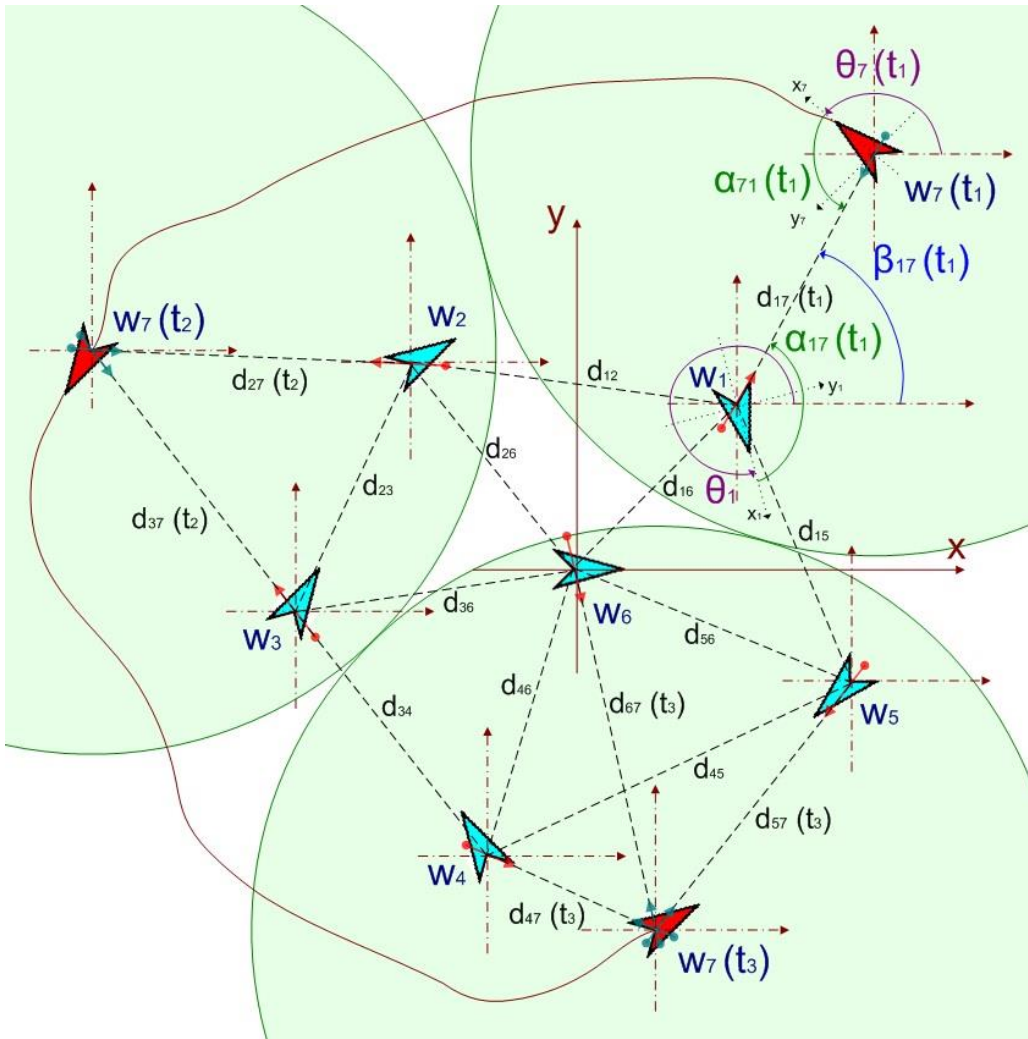


Figura 4-7 Mediu robotic – studiu de caz

În total au fost prevăzuți șapte roboți: $w_1 - w_7$ (Figura 4-8). Deoarece doar doi roboți pot să opereze în același moment din cauza evitării de interferență a semnalelor ultrasonice, și deoarece doar două module IRULT (prezentate în capitolul 3) au fost finalizate din punct de vedere hardware, acest lucru nu a influențat efectuarea experimentului. Experimentul a fost efectuat în felul următor:

Un modul IRULT funcțional a fost atașat robotului w_7 . Cel de-al doilea modul IRULT funcțional a fost atașat pe rând roboților $w_1 - w_6$, atunci când era nevoie pentru

efectuarea procedurii de aliniere și pentru calculul distanței. Robotul w_1 a fost utilizat la momentul t_1 pentru a localiza robotul w_7 prin metoda LOAD. De asemenea roboții w_2 și w_3 au fost folosiți la momentul t_2 pentru a localiza robotul w_7 prin metoda de localizare bilaterală. Și, în final, roboții w_4 , w_5 și w_6 au fost folosiți pentru a localiza robotul w_7 prin metoda de localizare bilaterală pentru 3 repere. Doar robotul w_7 trebuia să se deplaseze. Acest lucru s-a realizat static, prin plasarea acestuia la distanțe diferite față de roboții cu poziție *stabilită*, pe rând pentru cele trei momente: t_1 , t_2 și t_3 . La momentul t_1 , distanța d_{17} a fost variată de la 10cm până la 3m. Astfel, procedura de localizare prin metoda LOAD a fost repetată pentru fiecare distanță nouă a lui d_{17} . Asemănător, la momentul t_2 , a fost variată distanța d_{37} iar la momentul t_3 , distanța d_{67} .

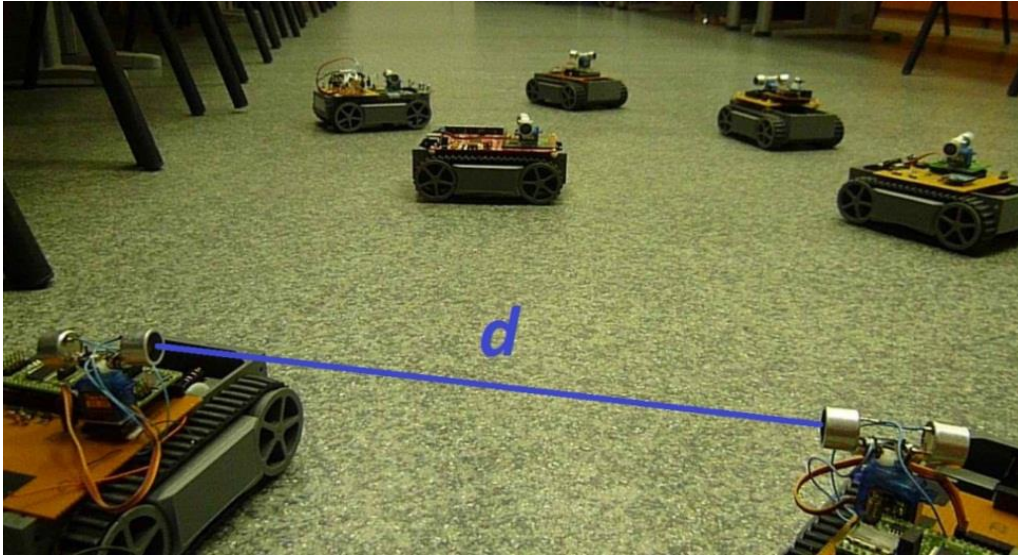


Figura 4-8 Exemplu de un sistem robotic format din 7 roboți [98]

Metoda LOAD are o acuratețe scăzută. În Tabelul 4-1 pot fi observate rezultatele experimentale pentru distanțe diferite. Se poate observa că eroarea crește cu distanța. Toate distanțele au fost obținute prin utilizarea metodei CTOF pentru măsurarea distanței și aplicarea filtrului Kalman pentru măsurători repetitive prezentate în (47).

Tabelul 4-1 Măsurătorile cu metoda LOAD [101]

Distanța reală [mm]	Metoda LOAD în studiul de caz la momentul t_1		
	Măsurarea distanței d_{17} [mm]	Durata maximă pentru calculul distanței d_{17} [ms]	Eroarea absolută a poziției [mm]
100	98	688	17
200	205	708	43
300	298	727	42
400	397	747	77
500	508	767	70
600	598	786	94
700	697	805	110
800	796	825	138
900	897	845	157
1000	1004	865	227
2000	2006	1060	488
3000	2993	1252	783

Obținerea locației cu metoda de localizare bilaterală se realizează cu o acuratețe mult mai bună față de metoda LOAD. Rezultatele experimentale, pentru distanțe diferite, sunt expuse în Tabelul 4-2. De asemenea sunt expuși și timpii totali de aliniere și de calcul a distanțelor.

Tabelul 4-2 Măsurătorile cu metoda de localizare bilaterală [101]

Distanța reală [mm]	în studiul de caz la momentul t_2 cu două repere		în studiul de caz la momentul t_3 cu trei repere	
	Eroarea absolută a poziției [mm]	Timpul de aliniere și de calcul a distanțelor [ms]	Eroarea absolută a poziției [mm]	Timpul de aliniere și de calcul a distanțelor [ms]
100	2,32	6212	2,01	9318
200	5,76	6252	4,98	9378
300	2,31	6290	2,00	9435
400	3,48	6330	3,01	9495
500	9,24	6370	7,98	9555
600	2,32	6408	2,10	9612
700	3,48	6446	3,08	9669
800	4,64	6486	4,01	9729
900	3,48	6526	3,00	9789
1000	4,63	6566	4,00	9849
2000	6,94	6956	6,00	10434
3000	8,11	7340	7,01	11010

La momentul t_3 , robotul w_7 se localizează în raport cu w_4 , w_5 și w_6 . Aplicând metoda de localizare bilaterală, robotul w_7 obține două poziții diferite. Prima poziție se obține prin localizarea față de reperele w_4 și w_6 , iar a doua față de reperele w_5 și w_6 . Poziția finală se calculează prin aplicarea mediei aritmetice. Această metodă este eficientă atunci când există două posibilități de localizare cu aceeași confiență. Altfel, metoda este aplicată o singură dată pentru opțiunea pentru care confiența este mai mare, pe lângă acuratețe, obținându-se astfel o performanță mai bună în ceea ce privește timpul de execuție. În studiul de caz prezentat, robotul w_6 are cea mai mare

confiență deoarece se află în origine (axa *home*). Roboții w_4 și w_5 au confiența egală dar mai mică decât confiența robotului w_6 . De aceea pentru localizarea robotului w_7 la momentul t_3 s-au ales reperele w_4 cu w_6 respectiv w_5 cu w_6 . Combinația reperelor w_4 cu w_5 nu are sens în acest caz, atâta timp cât w_6 are confiența cea mai mare.

4.8 Concluzii

Metodologia prezentată în acest capitol a fost dezvoltată în paralel cu dezvoltarea modului IRULT. Noțiunile au fost stabilite pe măsură ce s-au obținut rezultatele experimentale și s-a observat că definirea unei metodologii este absolut necesară, mai ales pentru că în prezent, nu există niciun standard care să guverneze cercetarea localizării în mediul interior al clădirilor [81]. Metodologia a fost patentată [97] împreună cu dispozitivul IRULT. În lucrarea de față, metodologia este actualizată. Anumite aspecte care țin de metodologie și nu sunt prezentate în acest capitol, apar în continuare în capitolele 5 și 6, cum ar fi de exemplu modelul matematic de localizare pentru nivelul COOPERARE și algoritmul BPF (Backtracking Particle Filter) pentru nivelul CENTRALIZARE. Aceste informații au fost introduse în capitole separate pentru că necesită o evaluare separată prin simulări. Experimentele hardware nu sunt suficiente pentru a valida metodologia prezentată. Este nevoie de o analiză a performanței și prin simulări.

După cum s-a observat în studiul de caz prezentat în acest capitol, experimentele făcute acoperă un caz prea izolat pentru a putea valida un model de localizare pentru un sistem foarte complex cum ar fi CORE-TX. Experimentele au fost făcute folosind doar două module IRULT. Roboții nu s-au deplasat, au fost poziționați static. Măsurătorile de distanță reală au fost făcute folosind o ruletă. Modulele IRULT au putut realiza procedura de aliniere și calculul distanței. Ca și informație de ieșire, folosind interfața Xbee, modulul IRULT putea să ofere distanța măsurată, timpul de durată a unei măsurători de distanță și timpul de aliniere. Toate datele au fost acumulate și adăugate într-o tabelă de tip Microsoft Excel, inclusiv măsurătorile repetate pentru distanțe diferite. Datele au fost procesate manual și rezultatele de poziționare au fost stabilite în urma calculelor făcute în tabel. Prin urmare, timpul pentru efectuarea unui astfel de studiu a crescut semnificativ.

Experimentele nu au fost făcute automatizat din două motive: modulele hardware necesare nu au fost disponibile și s-a dorit ca funcționalitatea de localizare să fie testată separat, izolând astfel erorile de localizare care puteau să provină din alte surse, altele decât modulul IRULT. Astfel erorile de localizare expuse în acest capitol, obținute prin experiment, includ doar erori care provin de la modulul de măsurare a distanței, și nu conțin alte surse de erori.

Scopul acestei cercetări a fost acela de a izola restul de componente ale sistemului CORE-TX iar atenția să fie pusă pe modulul IRULT și de asemenea de a găsi metode bune de a filtra erorile dobândite prin măsurarea distanței. Astfel, ținând cont de erorile pe care le poate genera modulul IRULT, autorul tezei a dezvoltat un simulator în care a implementat modelul matematic de localizare respectiv algoritmul BPF. Simulatorul are ca și intrare eroarea absolută de măsurare a distanței și poate genera inclusiv poziționări de cel mai defavorabil caz, ceea ce în practică ar fi greu de obținut sau și mai greu de reprodus, pentru că sunt cazuri rare care au o probabilitate foarte mică de apariție. De aceea, pentru a obține o validare corectă, autorul tezei folosește mediul de simulare pentru a cuprinde toate cazurile izolate ce pot apărea în practică. Simulările necesare au fost realizate și sunt expuse în capitolele 5 și 6.

5. APLICAREA METODOLOGIEI DE LOCALIZARE RELATIVĂ PENTRU STRATUL DE PERCEPȚIE ȘI OPERARE CORE-TX

În acest capitol este evidențiat primul nivel (PREDICȚIA) din cadrul metodologiei de localizare relativă propuse. La acest nivel fiecare robot mobil se auto-localizează pe baza resurselor proprii de mobilitate și navigare, prin procesare locală. În cadrul acestui nivel se prezintă conceptul dead reckoning. Acesta reprezintă un proces de subordonare față de următoarele două niveluri de localizare care sunt prezentate în continuare. Nivelul doi (COOPERARE), expus în cadrul acestui capitol, are la bază metoda de localizare multilaterală. Tehnici și metode existente care se referă la localizare multilaterală precum și modelul matematic propus bazat pe metoda ILS (Iterative Least Squares) sunt redată în acest capitol. De asemenea se prezintă și reguli de formare a pânzelor care reduc din efectul de propagare a erorilor. Pentru validarea modelului matematic se țin cont de măsurătorile de calcul a distanței descrise în capitolul 3 și simulări de propagare a erorilor în cele mai defavorabile cazuri prezentate în acest capitol. Tot aici este descris simulatorul dezvoltat, SiMuLoC, precum și un exemplu de aplicație unde poate fi folosit modelul matematic.

5.1 Introducere

Conform [16], sistemul CORE-TX interacționează cu mediu prin micro sisteme autonome cu inteligență încorporată numite WIT. Întrucât unele WIT-uri pot fi mobile, se pune problema localizării. Sistemul CORE-TX este proiectat în așa fel încât, deși WIT-urile pot fi abordate individual pentru scopuri specifice aplicării, la nivel superior de abstractizare, WIT-urile pot fi identificate după locația lor. Problema localizării este lăsată la un strat inferior de abstractizare care este responsabil pentru gestionarea mobilității.

Aplicarea metodologiei pentru stratul de percepție și operare CORE-TX presupune aplicarea unor clase de tehnici și reguli prin care un robot (WIT) ia deciziile sale bazate pe cunoștințele locale și limitate. Metodologia aplicată la acest strat oferă cele două niveluri de localizare: PREDICȚIE și COOPERARE.

Nivelul PREDICȚIE este un mod de bază de a localiza un robot mobil fără a implica și alți roboți în determinarea locației acestuia. Folosind senzorii interni pentru gestionarea mobilității, un robot reușește să-și păstreze confiența de localizare doar pentru deplasări scurte. În literatura de specialitate această procedură este denumită „dead reckoning” [107].

„Dead reckoning” este o procedură de modelare a poziției robotului mobil prin actualizarea poziției de la pasul următor prin măsurarea internă a vitezei, accelerației și a timpului de deplasare a robotului mobil. De exemplu, măsurătorile (estimările odometrice) se pot face pe baza unor encodere optice. De obicei măsurările efectuate sunt corupte datorită inegalității de diametru a roților, nealinierea roților, rezoluția encoderelor, alunecarea roților sau traversarea suprafețelor neregulate.

Nivelul COOPERARE este un nivel superior care oferă o confiență mai ridicată și pentru deplasări mai lungi. Prin urmare, cel de-al doilea nivel vine ca și o completare

peste primul nivel, bazându-se pe informațiile directe de la roboții din vecinătatea robotului în cauză.

Cele două niveluri se prezintă în acest capitol, cu accentul pe COOPERARE, deoarece acesta face parte din obiectivele tezei. Primul nivel, PREDICȚIE, este moștenit și întâlnit des în literatura de specialitate, de exemplu, sub forma de filtre Kalman pentru fuziunea datelor obținute de la senzorii interni ai robotului.

5.2 Nivelul I de localizare - PREDICȚIE

5.2.1 Gestionarea mobilității

Problema gestionării mobilității presupune ca un robot să-și determine poziția actuală iar pe măsură ce își modifică poziția să-și actualizeze această informație. Acest lucru poate fi realizat folosind coordonatele precedente unei mișcări și una din următoarele informații:

- predicția mișcării conform comenzilor date
- informațiile de la senzorii cu care robotul este dotat
- informația de la senzorii din mediu

Practic, toate aceste informații sunt afectate de erori, fie ele sistematice sau aleatoare. Folosirea unei singure surse de informații expune robotul pericolului reprezentat de cumularea erorilor și desincronizarea reprezentării interne cu poziția lui reală. Astfel, este necesară găsirea unei metode care să asimileze periodic datele disponibile în reprezentarea internă.

Problema de predicție este dată de faptul că fiecare algoritm își menține o reprezentare internă a coordonatelor posibile și a distribuției probabilităților de ocupare a fiecărei coordonate. Starea internă a sistemului la momentul k este adeseori notat cu x_k în literatura de specialitate. Predicția constă în estimarea stării ce va rezulta în urma unei mișcări, date fiind etapele precedente mișcării. Pentru estimarea modului în care se va face mișcarea se folosește așa numitul *model-mișcare*, $P(x_k | x_{k-1}, x_{k-2}, \dots, x_0)$.

Problema de actualizare apare după ce poziția curentă este modificată. Se rezolvă prin citirea informațiilor de la senzori (de ex.: cameră video, sonar, accelerometru, senzor Hall etc.) și se integrează aceste informații în setul de date pentru a determina poziția actuală. Setul de date obținut de la senzori la momentul k se notează cu z_k , iar mulțimea seturilor de date $\{z_1, z_2, \dots, z_k\}$ se notează cu Z_k . Informațiile primite de la senzori sunt susceptibile erorilor și astfel, algoritmi adeseori nu le folosesc direct, ci, pentru integrarea lor corectă folosesc un model statistic numit *model-senzor*, $P(Z_k | x_{k-1}, x_{k-2}, \dots, x_0)$.

Obiectivul algoritmilor de localizare este de a calcula distribuția probabilităților de poziționare, date fiind informațiile de la senzori până în momentul k , $P(x_k | Z_{k-1})$. O fuziune de date primite de la senzori poate fi realizată cu ajutorul filtrelor Kalman.

5.2.2 Filtre Kalman

Acest concept a fost realizat de Rudolph Kalman în anul 1960. Filtrele Kalman au devenit o abordare standard de reducere a erorilor în sensul celor mai mici pătrate și pot folosi măsurători chiar și de la mai multe surse [108][109]. Filtrele Kalman simple pot fi aplicate pentru un sistem dinamic liniar sau în situația în care, nu poate

fi sau este foarte greu de modelat un sistem dinamic neliniar. În [110] este dat un exemplu de aplicabilitate a unui filtru Kalman pe un model dinamic liniar și se arată prin punere în practică cum oscilațiile perturbatoare atenuază ajutând la o estimare optimă a stării sistemului. Un model liniar Gauss-Markov este prezentat în [111] sub următoarea formă:

$$x_{t+1} = Ax_t + w_t \quad (5-1)$$

$$y_t = Cx_t + v_t \quad (5-2)$$

unde x_t reprezintă starea sistemului, y_t reprezintă ieșirea, w_t reprezintă zgomotul de proces, iar v_t reprezintă zgomotul de măsurare. Despre un proces se poate spune că este de tip Markov doar dacă starea sa curentă depinde de starea anterioară.

În lucrarea [108] sunt prezentate două seturi de ecuații, de predicție și de actualizare. Ecuațiile de predicție sunt redată astfel:

$$\hat{X}_{k|k-1} = A\hat{X}_{k-1} + BU_{k-1} \quad (5-3)$$

$$P_{k|k-1} = AP_{k-1}A^T + Q \quad (5-4)$$

unde \hat{X}_{k-1} reprezintă estimarea optimă de la ultimul pas, $\hat{X}_{k|k-1}$ reprezintă predicția pentru pasul următor k , A și B sunt parametrii de sistem, iar U este o intrare de control opțională. Eroarea de covarianță este extrapolată de (5-4), unde $P_{k|k-1}$ și P_{k-1} sunt covarianțe pentru $X_{k|k+1}$ respectiv $X_{k-1|k+1}$ iar Q este covarianța zgomotului de proces. Ecuațiile de actualizare sunt redată astfel:

$$K_{gk} = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \quad (5-5)$$

$$\hat{X}_{(K|K)} = \hat{X}_{(K|K-1)} + K_{gk}(Z_{(K)} - HX_{(K|K-1)}) \quad (5-6)$$

$$P_{k|k} = (I - K_{gk}H)P_{k|k-1} \quad (5-7)$$

Folosind valoarea măsurată și valoarea de predicție, se generează o stare de estimare anterioară, încorporând măsurătoarea ca și în (5-6), altfel spus, se actualizează starea optimă de estimare curentă. H este un model de observare care mapează spațiul de stare adevărat în spațiu observat, adică măsurat. Eroarea covarianței de estimare este actualizată în ecuația (5-7).

Filtrele Kalman extinse EKF (Extended Kalman Filter) pot fi aplicate pentru un sistem dinamic neliniar atunci când există zgomot în măsurători [112]. În lucrarea [113] este exemplificat un model de localizare a unui robot mobil prin folosirea filtrului Kalman extins pentru filtrarea zgomotelor obținute din măsurători. Tot aici se explică pe scurt un model general de folosire a filtrului Kalman extins:

Se dau ecuațiile:

$$x_{i+1} = f_i(x_i, w_i) \quad (5-8)$$

$$y_i = h_i(x_i, v_i) \quad (5-9)$$

unde x_i , y_i , w_i și v_i reprezintă: vectorul de stare, vectorul de măsurare, zgomotul în proces, respectiv, vectorul zgomot de măsurare. Vectorul de stare la momentul $i + 1$ este determinat de funcția matrice de tranziție neliniară f_i . Funcția matrice de măsurare neliniară h_i face legătura între y_i , x_i și v_i . Ecuațiile (5-8) și (5-9) poartă denumirile: ecuație de proces respectiv ecuație de măsurare. Pentru EKF, un estimativ al lui x_i bazat pe vectorii de măsurare y_1, \dots, y_{i-1} este notat cu $\hat{x}_{i|i-1}$. Mai mult, matricea de corelație a erorii de estimare între valoarea adevărată și stările estimate este definită astfel:

$$P_{i|i-1} = E[\tilde{x}_{i|i-1}\tilde{x}_{i|i-1}^T] \quad (5-10)$$

unde $\tilde{x}_{i|i-1} = x_i - \hat{x}_{i|i-1}$.

Iterația EKF începe cu valorile inițiale date de $x_{1|0}$ și $P_{1|0}$. Pentru $i = 1, 2, \dots$, următoarele ecuații se repetă [112] :

$$e_i = y_i - h_i(\hat{x}_{i|i-1}, 0) \quad (5-11)$$

$$K_i = P_{i|i-1} H_i^T [H_i P_{i|i-1} H_i^T + S_i]^{-1} \quad (5-12)$$

$$\hat{x}_{i|i} = \hat{x}_{i|i-1} + K_i e_i \quad (5-13)$$

$$\hat{x}_{i+1|i} = f_i(\hat{x}_{i|i}, 0) \quad (5-14)$$

$$P_i = [I - K_i H_i] P_{i|i-1} \quad (5-15)$$

$$P_{i+1|i} = F_i P_i F_i^T + Q_i \quad (5-16)$$

unde e_i din ecuația (5-11) poartă denumirea de înnoire a EKF, ceea ce înseamnă, o informație nouă este implicată în măsurătoarea curentă y_i . În (5-12) este obținută amplificarea Kalman K_i . Matricele S_i și Q_i sunt matricele de corelație a măsurării respectiv a zgomotului în proces. Matricea H_i este matricea jacobiană a funcției $h_i(x_i, v_i)$ în raport cu x_i , care este definită astfel:

$$H_i = \left. \frac{\partial h_i(x_i, v_i)}{\partial x_i} \right|_{x_i = \hat{x}_{i|i-1}} \quad (5-17)$$

Măsurătoarea și actualizarea timpului sunt efectuate în ecuațiile (5-13) respectiv (5-14). În final, matricea de corelație a erorii estimate $P_{i|i-1}$ este actualizată la $P_{i+1|i}$ prin ecuațiile (5-15) și (5-16). Matricea Q_i este matricea de corelație a zgomotului în proces iar I este o matrice de identitate având aceeași dimensiunea ca și $K_i H_i$. Matricea F_i este matricea jacobiană a funcției $f_i(x_i, w_i)$ în raport cu x_i , care este definită astfel:

$$F_i = \left. \frac{\partial f_i(x_i, w_i)}{\partial x_i} \right|_{x_i = \hat{x}_{i|i-1}} \quad (5-18)$$

În ciuda limitărilor teoretice ale filtrelor Kalman, în practică s-a reușit aplicarea lor cu succes în diverse probleme de urmărire, unde sunt chiar eficiente, realizează estimări suficient de exacte, chiar și pentru sisteme neliniare [114].

5.3 Tehnici și metode de localizare relativă

5.3.1 Trilaterația

Trilaterația este abordată în lucrarea [115].

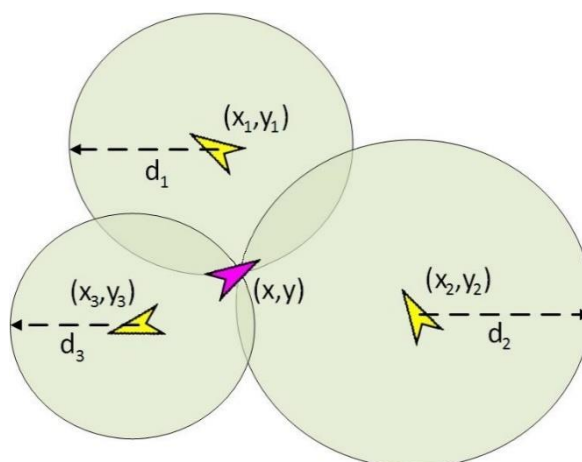


Figura 5-1 Modelul teoretic al trilaterației

Trilaterația este cea mai de bază și intuitivă metodă de localizare. Prin această metodă se determină poziția robotului mobil față de alți trei roboți mobili prin intersecția a trei cercuri așa cum se reprezintă în Figura 5-1. Pentru a estima poziția folosind trilaterația, un nod trebuie să cunoască pozițiile a trei repere și distanțele până la acestea.

Cercurile formate de pozițiile reperelor și distanțele până la acestea pot fi reprezentate sub forma:

$$\begin{cases} (\hat{x} - x_1)^2 + (\hat{y} - y_1)^2 = d_1^2 \\ (\hat{x} - x_2)^2 + (\hat{y} - y_2)^2 = d_2^2 \\ (\hat{x} - x_3)^2 + (\hat{y} - y_3)^2 = d_3^2 \end{cases} \quad (5-19)$$

unde (\hat{x}, \hat{y}) este poziția pe care dorim să o aflăm, (x_i, y_i) este poziția reperului i , iar d_i este distanța de la poziția pe care dorim să o aflăm până la reperul i . În acest caz, avem trei ecuații cu două necunoscute, care teoretic, poate fi rezolvată într-o singură soluție.

În aplicațiile reale, neacuratețea estimării distanțelor, la fel ca și neacuratețea estimării poziției fiecărui reper, face ca poziția să fie obținută cu dificultate. Așa cum este reprezentat în Figura 5-2, cercurile nu se intersectează doar într-un singur punct, rezultând un set de infinitate de posibile soluții.

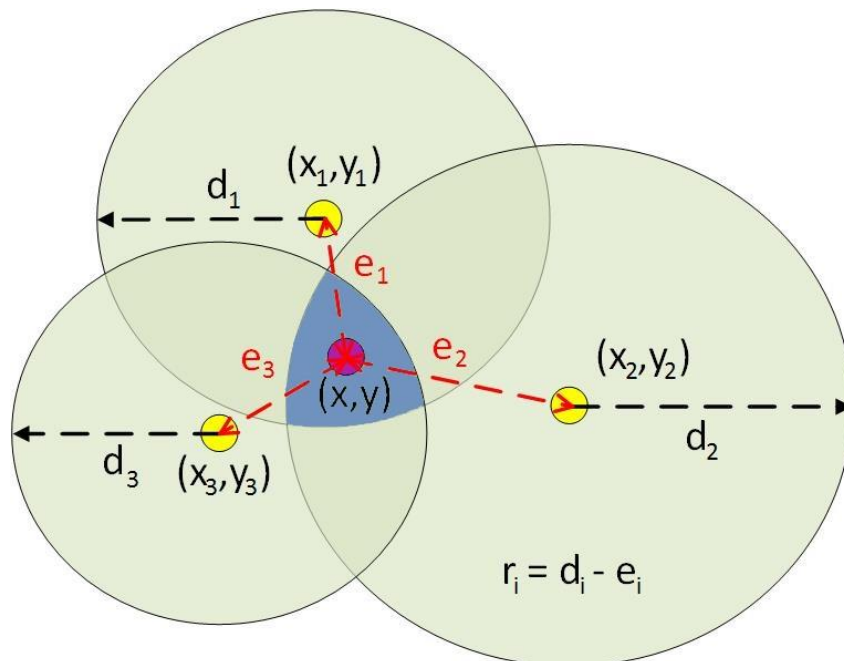


Figura 5-2 Inconsistențele pozițiilor și distanțelor generând un sistem cu un număr infinit de soluții

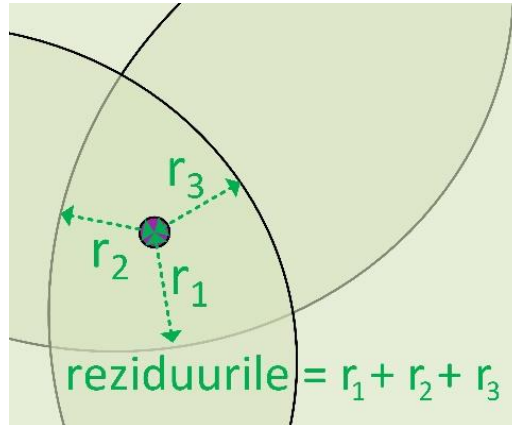


Figura 5-3 Valoare reziduală ca și sumă de diferențe de pătrate între distanțele estimate și cele calculate

5.3.2 Localizarea multilaterală

Localizarea multilaterală este abordată în lucrarea [115].

Atunci când un număr mai mare de repere sunt disponibile, poate fi folosită localizarea multilaterală pentru a calcula poziția nodului necunoscut. În acest caz, trebuie să fie soluționat un sistem supradeterminat unde avem mai multe ecuații decât necunoscute. În Figura 5-4 este reprezentat acest caz. De obicei, un sistem supradeterminat nu are o soluție unică.

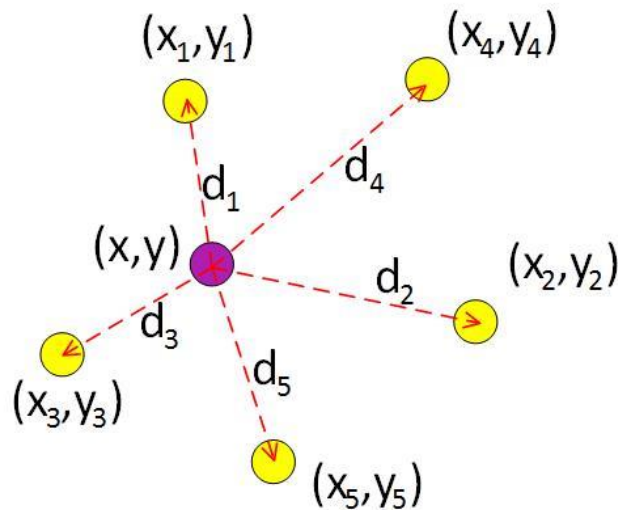


Figura 5-4 Localizarea multilaterală se bazează pe mai mult de trei repere și necesită rezolvarea unui sistem supradeterminat

Atunci când considerăm n puncte de referință și de asemenea eroarea estimării distanței $d_i = \hat{d}_i - \epsilon$, atunci sistemul de ecuații devine:

$$\begin{cases} (\hat{x} - x_1)^2 + (\hat{y} - y_1)^2 = \hat{d}_1^2 - \epsilon \\ \vdots \\ (\hat{x} - x_n)^2 + (\hat{y} - y_n)^2 = \hat{d}_n^2 - \epsilon \end{cases} \quad (5-20)$$

unde ϵ este o variabilă aleatoare normală, independentă, de medie zero. Acest sistem poate fi liniarizat, prin scăderea ultimei ecuații, sub forma $Ax \approx b$

$$\begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} \approx \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_n^2 - d_1^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 - d_{n-1}^2 + d_n^2 \end{bmatrix} \quad (5-21)$$

Acest sistem linear poate fi rezolvat folosind metode standard cum ar fi abordarea celor mai mici pătrate.

5.3.3 Metoda liniară pentru cele mai mici pătrate

Rezolvarea sistemului linear prin abordarea celor mai mici pătrate este discutată în [115], [116] și [117]. Aceasta poate fi realizată de următoarea estimare:

$$x = (A^T A)^{-1} (A^T b) \quad (5-22)$$

Idea principală a acestei metode este de a minimiza suma diferențelor de pătrate dintre distanța măsurată și distanța calculată (folosind poziția estimată). Această sumă de diferențe este cunoscută ca și sumă de reziduuri așa cum este ilustrat în Figura 5-3. În termeni matematici, poziția calculată este:

$$(\hat{x}, \hat{y}) = \min \left(\sum_{i=1}^n \left(\sqrt{(\hat{x} - x_i)^2 + (\hat{y} - y_i)^2} - d_i \right)^2 \right) \quad (5-23)$$

unde (x_i, y_i) este poziția reperului i , iar d_i este distanța estimată. Distanța dintre poziția calculată și poziția reperului, $\sqrt{(\hat{x} - x_i)^2 + (\hat{y} - y_i)^2}$, reprezintă distanța calculată.

5.3.4 ILS (Iterative Least Squares)

Un algoritm cu metoda iterativă pentru cele mai mici pătrate este redată în [118]. În respectiva lucrare se prezintă o ecuație generală care modelează comportamentul practic de măsurare a distanței:

$$R = f(S_1, S_2, \dots, S_n, X) + \vartheta \quad (5-24)$$

unde R este vectorul cu toate distanțele până la fiecare reper $[r_1, r_2, \dots, r_n]^T$. Vectorul ϑ reprezintă erorile de măsurare. Vectorul cu parametrii poziției necunoscute este notat cu X iar reperele sunt notate cu S_1, S_2, \dots, S_n . Prin linearizarea ecuației (5-24) cu privire la valorile nominale R^* și X^* se obține:

$$\delta R = F(S_1, S_2, \dots, S_n, X^*) \delta X + \vartheta \quad (5-25)$$

unde

$$\delta R = R - R^* = R - f(S_1, S_2, \dots, S_n, X^*) \quad (5-26)$$

$$\delta X = X - X^* \quad (5-27)$$

și

$$F(S_1, S_2, \dots, S_n, X^*) = \left. \frac{\partial f(S_1, S_2, \dots, S_n, X)}{\partial X} \right|_{X=X^*} \quad (5-28)$$

Această liniarizare duce la următorul algoritm:

- 1) Se folosește o estimare inițială X_0^* , a lui X pentru a calcula F conform ecuației (5-28).
- 2) Se calculează o matrice de pondere W bazată pe raportul dintre semnal și zgomot al reperelor
- 3) Se folosesc distanțele R și estimarea inițială X_0^* pentru a calcula δR conform ecuației (5-26).
- 4) Se calculează estimarea celor mai mici pătrate ponderate δX astfel:

$$\delta X = (F^T W^{-1} F)^{-1} F^T W^{-1} \delta R \quad (5-29)$$
- 5) Se actualizează X_0^* :

$$X_1^* = X_0^* + \delta X \quad (5-30)$$
- 6) Se verifică testul de convergență:

$$|\delta X| < \varepsilon \quad (5-31)$$
- 7) Dacă condiția din ecuația (5-31) este îndeplinită, atunci procedura se oprește. Altfel, se sare la pasul 1 și se incrementează indexul de iterație cu 1.

5.3.5 AILS (Adaptive Iterative Least Squares)

În lucrarea [119] este prezentată o metodă pentru cele mai mici pătrate adaptată. Se dă ecuația de calcul a distanței prin metoda TOA:

$$t_{i-MU}c = \rho_{i-MU} = \sqrt{(x_i - x_{MU})^2 + (y_i - y_{MU})^2} \quad (5-32)$$

unde MU reprezintă unitatea mobilă (robotul mobil), t_{i-MU} este timpul măsurat de propagare a semnalului pe distanța în cauză, c este viteza de propagare a semnalului, ρ_{i-MU} reprezintă distanța în cauză, (x_i, y_i) este locația reperului i iar (x_{MU}, y_{MU}) este locația unității mobile. Locația unității mobile poate fi estimată folosind ecuația (5-32). Această ecuație este neliniară și poate fi liniarizată folosind formula lui Taylor de ordinul întâi [120]:

$$\begin{aligned} \rho_{i-MU} &\cong \sqrt{(x_i - \bar{x}_{MU})^2 + (y_i - \bar{y}_{MU})^2} \\ &- \frac{x_i - \bar{x}_{MU}}{\sqrt{(x_i - \bar{x}_{MU})^2 + (y_i - \bar{y}_{MU})^2}} \delta x_{MU} \\ &- \frac{y_i - \bar{y}_{MU}}{\sqrt{(x_i - \bar{x}_{MU})^2 + (y_i - \bar{y}_{MU})^2}} \delta y_{MU} \end{aligned} \quad (5-33)$$

unde $(\bar{x}_{MU}, \bar{y}_{MU})$ reprezintă punctul nominal pentru liniarizarea ecuației neliniare (5-32), iar $(\delta x_{MU}, \delta y_{MU})$ reprezintă diferența dintre punctul nominal și locația reală.

Algoritmul de calcul constă în următorii pași:

$$1) \quad (\bar{x}_{MU}, \bar{y}_{MU}) = NP_{cand.}(k), \quad k = 1 \quad (5-34)$$

$$2) \quad \begin{bmatrix} \delta \hat{x}_{MU} \\ \delta \hat{y}_{MU} \end{bmatrix} = (H^T H)^{-1} H^T \delta D \quad (5-35)$$

$$3) \quad (\bar{x}_{MU}, \bar{y}_{MU}) = (\bar{x}_{MU}, \bar{y}_{MU}) + \begin{bmatrix} \delta \hat{x}_{MU} \\ \delta \hat{y}_{MU} \end{bmatrix} \quad (5-36)$$

$$4) \quad \delta D^T \delta D < \varepsilon \quad (5-37)$$

$$5) \quad (\bar{x}_{MU}, \bar{y}_{MU}) = NP_{cand.}(k), \quad k = k + 1 \quad (5-38)$$

unde,

$$H = \begin{bmatrix} \frac{x_1 - \bar{x}_{MU}}{\bar{\rho}_{1-MU}} & -\frac{y_1 - \bar{y}_{MU}}{\bar{\rho}_{1-MU}} \\ \vdots & \vdots \\ \frac{x_n - \bar{x}_{MU}}{\bar{\rho}_{n-MU}} & -\frac{y_n - \bar{y}_{MU}}{\bar{\rho}_{n-MU}} \end{bmatrix} \quad (5-39)$$

$$\bar{\rho}_{i-MU} = \sqrt{(x_i - \bar{x}_{MU})^2 + (y_i - \bar{y}_{MU})^2} \quad (5-40)$$

și

$$\delta D = \begin{bmatrix} \rho_{1-MU} - \bar{\rho}_{1-MU} \\ \vdots \\ \rho_{n-MU} - \bar{\rho}_{n-MU} \end{bmatrix} \quad (5-41)$$

La pasul 1 este selectată o valoare nominală inițială din vectorul cu posibili candidați. Posibilii candidați sunt determinați în funcție de poziționarea reperelor, adică se formează un dreptunghi care include toate reperatele curente. Se iau ca și posibili candidați extremele, adică toate colțurile dreptunghiului și se mai pot lua și alte valori din dreptunghiul format astfel:

$$NP_{cand.} = \left\{ \begin{bmatrix} x_{max} \\ y_{min} \end{bmatrix}, \begin{bmatrix} x_{max} \\ y_{max} \end{bmatrix}, \begin{bmatrix} x_{min} \\ y_{min} \end{bmatrix}, \begin{bmatrix} x_{min} \\ y_{max} \end{bmatrix}, \dots \right\} \quad (5-42)$$

La pasul 2 se calculează eroarea curentă de estimare. La pasul 3, punctul nominal se actualizează folosind eroarea curentă de estimare. Valoarea ε de la pasul 4 este o valoare de prag foarte mică care se determină în mod euristic. Dacă la pasul 4 condiția este îndeplinită atunci ultima valoare nominală calculată reprezintă soluția finală estimată.

Pașii 2-4 se repetă cât timp condiția de la pasul 4 nu este îndeplinită. Dacă condiția de la pasul 4 nu este îndeplinită și timpul de „timeout” expiră, la pasul 5 se stabilește o nouă valoare nominală, adică se ia un alt posibil candidat din vectorul (5-42). După fiecare revenire la pasul 5, timpul de „timeout” se resetează iar apoi se reiau pașii 2-4.

Valoarea nominală se ia uniform de pe suprafața dreptunghiulară stabilită la pasul 1, astfel încât să existe posibilitatea ca algoritmul să se oprească cât mai aproape de minimul global și nu de cel local (Figura 5-5). Minimul global și cel local se referă la ecuația (5-33), care prin aproximare, are două puncte în care tinde să se îndrepte soluția algoritmului. Ideea lui Seong Yun Cho [119] este de a influența algoritmul ILS să tindă spre minimul global, de aceea propune algoritmul AILS ca și o îmbunătățire față de algoritmul clasic ILS.

O informație foarte importantă pe care o oferă Seong Yun Cho este că atunci când reperatele nu formează o linie dreaptă, eroarea în punctul global este aproape de zero iar eroarea în punctul local este mai depărtată de zero. Acest aspect este simulat și discutat în subcapitolul 5.7 Reguli de formare a pânzelor.

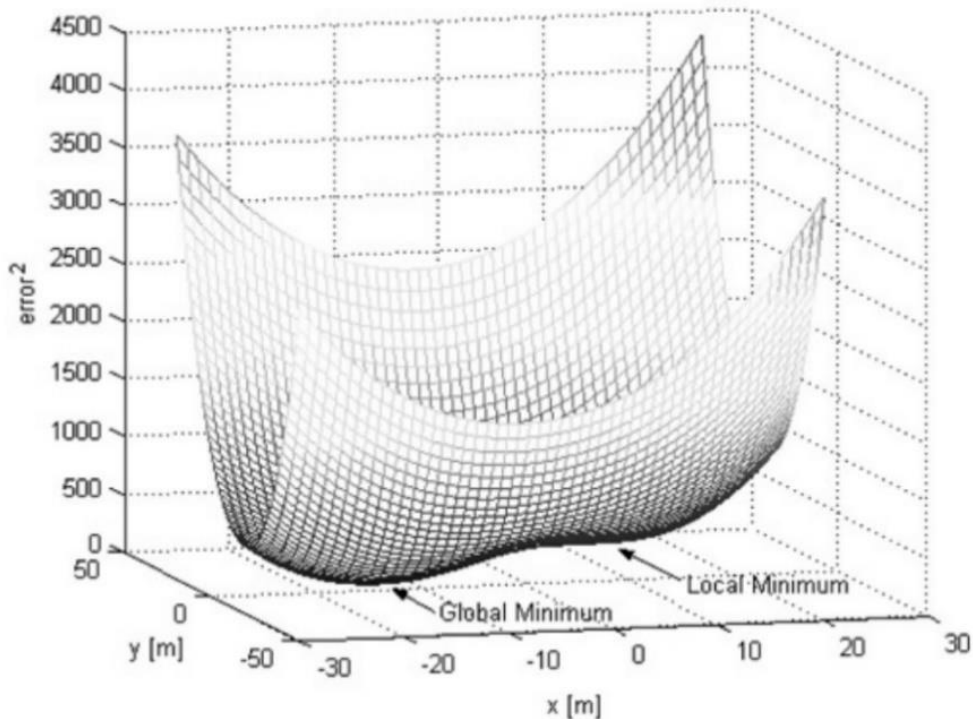


Figura 5-5 Exemplu de minim local [119]

5.4 Nivelul II de localizare - COOPERARE

5.4.1 Tehnici și reguli

Nivelul II de localizare – COOPERARE, cuprinde o clasă de tehnici și reguli prin care un robot mobil se localizează prin procesare distribuită și cooperare cu roboții mobili din vecinătatea proprie, limitat de distanța maximă de cooperare.

Fiecare robot din sistemul CORE-TX este construit pe baza modelului WIT. Echipat cu un modul IRULT, fiecare robot cooperează cu alți roboți pe raza de vizibilitate, cu scopul de a determina orientarea și poziția sa în mediul actual. Dispozitivul IRULT este alcătuit dintr-un motor pas cu pas, două traductoare ultrasonice și alte circuite de interfațare. Fiecare traductor are o directivitate în formă de con de aproximativ 50 de grade și poate să transmită sau să recepționeze semnale ultrasonice de frecvență 40KHz. Aceste două traductoare sunt montate spate în spate la 180 de grade. Astfel, prin rotirea a două traductoare, robotul poate avea o vizibilitate omnidirecțională.

Bazat pe modulul IRULT, fiecare robot poate calcula distanța față de un alt robot prin folosirea metodei „durata de propagare combinată” CTOF, metodă prezentată în subcapitolul 3.6 Măsurarea distanței folosind metoda CTOF.

Procedura de măsurare a distanței este precedată de procedura de aliniere. Procedura de aliniere constă în cooperarea dintre doi roboți prin interfață de

comunicare wireless, pentru a schimba comenzile necesare, și se bazează pe măsurarea continuă a intensității undelor ultrasonice. Doi roboți care efectuează procedura de aliniere au funcții diferite, unul dintre roboți este emițător și celălalt este receptor. Procedura de aliniere poate fi realizată în două moduri:

- În timp ce doi roboți rotesc turelele lor de la o limită la cealaltă, în mod repetat și cu viteze diferite, receptorul calculează indicatorul RSSI a semnalului ultrasonic transmis de către transmițător;

- În timp ce primul robot rotește turela, al doilea menține turela la un anumit unghi. Pentru fiecare rotație a turelei primului robot, de la o limită la cealaltă, turela celui de-al doilea robot incrementează unghiul cu un pas dat de intervalul de directivitate de formă conică a senzorului de ultrasunete.

Algoritmul de aliniere a fost expus în subcapitolul 3.4 Algoritmul de aliniere a roboților.

Acestea reprezintă tehnicile și regulile de bază pentru nivelul COOPERARE valabile pentru fiecare robot construit pe baza modelului WIT. Tot la nivelul COOPERARE dar pentru fiecare pânză se aplică modelul matematic de localizare care este prezentat în subcapitolul următor.

5.4.2 Modelul matematic

Modelul matematic de localizare pentru stratul de percepție și operare CORE-TX are la bază metoda ILS standard prezentată de Biton în lucrarea [118]. Nivelul II de localizare - COOPERARE ține cont de problemele ILS prezentate de Seong Yun Cho în lucrarea [119].

Ecuția pentru determinarea distanței între două WIT-uri a fost prezentată în [104] astfel:

$$d_{i,j} = c_{aer} \frac{(\Delta t_{i,j} - \delta U_{i,j} - \delta W_{i,j})}{2} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (5-43)$$

unde $d_{i,j}$ este distanța între WIT-ul i și j . Viteza sunetului prin aer este notată cu c_{aer} și are valoarea nominală $343,4 \frac{m}{s}$. $\Delta t_{i,j}$ reprezintă timpul total de derulare a procedurii de calcul a distanței între WIT-ul i și j . $\delta U_{i,j}$ reprezintă timpul maxim de recepție a semnalului ultrasonic iar $\delta W_{i,j}$ reprezintă timpul maxim de recepție a semnalului wireless (incluzând și întârzierile provenite din stiva de protocoale wireless). (x_i, y_i) respectiv (x_j, y_j) reprezintă coordonatele WIT-ului i respectiv j .

În continuare se prezintă modelul matematic pe un caz particular denumit COOPERARE 2:2, după aceea se va expune o formă generală.

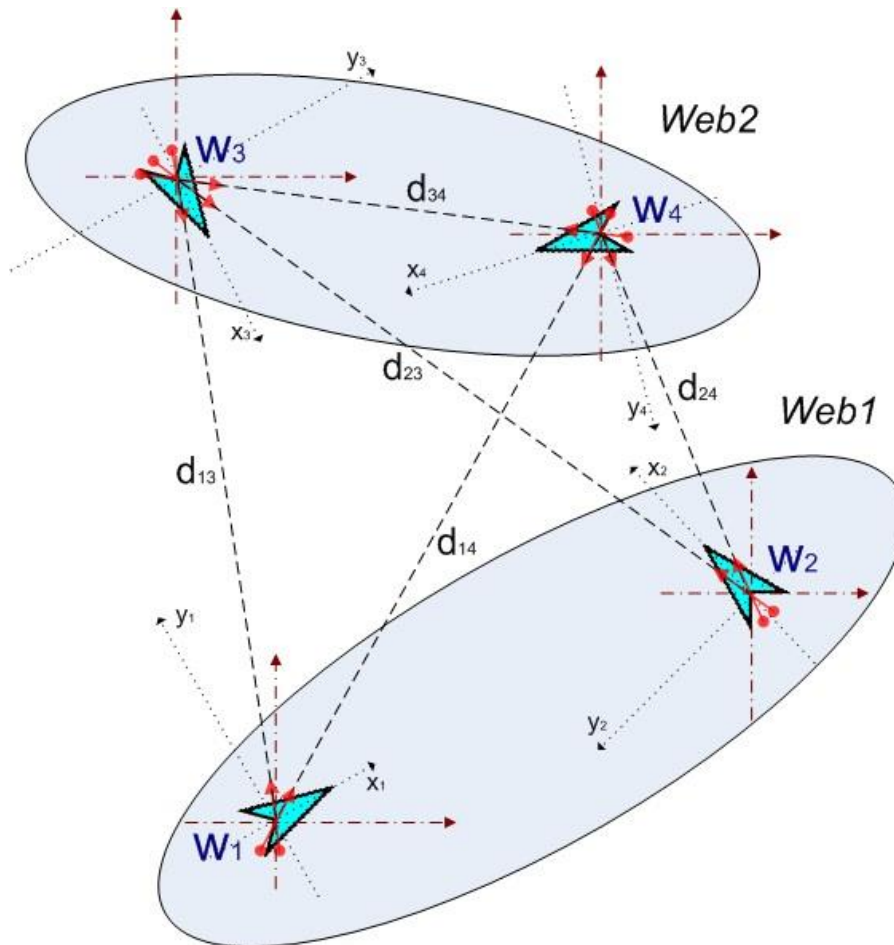


Figura 5-6 Nivelul II de localizare – COOPERARE 2:2 [121]

Presupunem că avem două pânze, fiecare pânză este formată din doi roboți (Figura 5-6). Prima pânză, Web1, este o pânză cu roboți *ancoră*, adică cu poziție cunoscută. A doua pânză, Web2, este formată din roboți cu poziție necunoscută. Problema se pune de a localiza roboții *necunoscuți* față de roboții *ancoră*. Acest lucru este realizat de stratul de percepție și operare CORE-TX. Coordonatorul acestei acțiuni poate să fie oricare robot din cele două pânze. Coordonatorul va executa algoritmul implementat pe baza modelului matematic prezentat în continuare.

Pe baza ecuației (5-43) obținem următorul sistem:

$$\begin{cases} (x_3 - x_1)^2 + (y_3 - y_1)^2 = d_{13}^2 \\ (x_4 - x_1)^2 + (y_4 - y_1)^2 = d_{14}^2 \\ (x_3 - x_2)^2 + (y_3 - y_2)^2 = d_{23}^2 \\ (x_4 - x_2)^2 + (y_4 - y_2)^2 = d_{24}^2 \\ (x_4 - x_3)^2 + (y_4 - y_3)^2 = d_{34}^2 \end{cases} \quad (5-44)$$

De asemenea, ecuația (5-43) este neliniară și poate fi liniarizată folosind formula lui Taylor de ordinul întâi [120]:

$$d_{i,j} \cong \sqrt{(x_i - \bar{x}_j)^2 + (y_i - \bar{y}_j)^2} - \frac{x_i - \bar{x}_j}{\sqrt{(x_i - \bar{x}_j)^2 + (y_i - \bar{y}_j)^2}} \delta x_j - \frac{y_i - \bar{y}_j}{\sqrt{(x_i - \bar{x}_j)^2 + (y_i - \bar{y}_j)^2}} \delta y_j \quad (5-45)$$

unde (\bar{x}_j, \bar{y}_j) reprezintă punctul nominal pentru liniarizarea ecuației neliniare (5-43), iar $(\delta x_j, \delta y_j)$ reprezintă diferența dintre punctul nominal și locația reală.

Având cei doi roboți cu poziție necunoscută, w_3 și w_4 , putem estima poziția acestora raportat la cei doi roboți cu poziție cunoscută, w_1 și w_2 , printr-un proces iterativ aplicând următorii pași:

$$1) \quad (\bar{x}_j, \bar{y}_j) = (x_{j,bilateral}, y_{j,bilateral}) \quad , \quad k = \{3; 4\} \quad (5-46)$$

$$2) \quad \begin{bmatrix} \delta \hat{x}_3 \\ \delta \hat{y}_3 \\ \delta \hat{x}_4 \\ \delta \hat{y}_4 \end{bmatrix} = (A^T A)^{-1} (A^T b) \quad (5-47)$$

$$3) \quad \begin{bmatrix} \bar{x}_3 \\ \bar{y}_3 \\ \bar{x}_4 \\ \bar{y}_4 \end{bmatrix} = \begin{bmatrix} \bar{x}_3 \\ \bar{y}_3 \\ \bar{x}_4 \\ \bar{y}_4 \end{bmatrix} + \begin{bmatrix} \delta \hat{x}_3 \\ \delta \hat{y}_3 \\ \delta \hat{x}_4 \\ \delta \hat{y}_4 \end{bmatrix} \quad (5-48)$$

$$4) \quad \begin{bmatrix} \delta \hat{x}_3 \\ \delta \hat{y}_3 \\ \delta \hat{x}_4 \\ \delta \hat{y}_4 \end{bmatrix}^T \begin{bmatrix} \delta \hat{x}_3 \\ \delta \hat{y}_3 \\ \delta \hat{x}_4 \\ \delta \hat{y}_4 \end{bmatrix} < \varepsilon \quad (5-49)$$

unde $[\delta \hat{x}_3, \delta \hat{y}_3, \delta \hat{x}_4, \delta \hat{y}_4]^T$ semnifică erorile estimate ale punctelor nominale

$[\bar{x}_3, \bar{y}_3, \bar{x}_4, \bar{y}_4]^T$,

$$A = \begin{pmatrix} a_{13} & b_{13} & 0 & 0 \\ a_{23} & b_{23} & 0 & 0 \\ 0 & 0 & a_{14} & b_{14} \\ 0 & 0 & a_{24} & b_{24} \\ -a_{34} & -b_{34} & a_{34} & b_{34} \end{pmatrix} \quad (5-50)$$

$$a_{ij} = -\frac{\hat{x}_i - \bar{x}_j}{\bar{d}_{ij}} \quad (5-51)$$

$$b_{ij} = -\frac{\hat{y}_i - \bar{y}_j}{\bar{d}_{ij}} \quad (5-52)$$

$$\begin{cases} \bar{d}_{ij} = \sqrt{(\bar{x}_i - \bar{x}_j)^2 + (\bar{y}_i - \bar{y}_j)^2}, & \text{dacă } i = 3 \text{ și } j = 4 \\ \bar{d}_{ij} = \sqrt{(\hat{x}_i - \bar{x}_j)^2 + (\hat{y}_i - \bar{y}_j)^2}, & \text{în rest} \end{cases} \quad (5-53)$$

și

$$b = \begin{pmatrix} \tilde{d}_{13} - \bar{d}_{13} \\ \tilde{d}_{23} - \bar{d}_{23} \\ \tilde{d}_{14} - \bar{d}_{14} \\ \tilde{d}_{24} - \bar{d}_{24} \\ \tilde{d}_{34} - \bar{d}_{34} \end{pmatrix} \quad (5-54)$$

unde ($\bar{\cdot}$) semnifică măsurătoarea lui (\cdot), ($\hat{\cdot}$) semnifică valoarea nominală a lui (\cdot) și($\tilde{\cdot}$) semnifică estimarea lui (\cdot).

La pasul 1 este selectată o valoare nominală ($x_{j_bilateral}, y_{j_bilateral}$) pentru $j = \{3; 4\}$ pe baza celor două repere (x_1, y_1) și (x_2, y_2) astfel:

$$y_{j_bilateral} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (5-55)$$

$$x_{j_bilateral} = s + \frac{y_j(y_1 - y_2)}{(x_2 - x_1)} \quad (5-56)$$

unde,

$$\begin{cases} a = \frac{(y_1 - y_2)^2}{(x_2 - x_1)^2} + 1 \\ b = \frac{2(s - x_2)(y_1 - y_2)}{x_2 - x_1} - 2y_2 \\ c = (s - x_2)^2 + y_2^2 - d_{2j}^2 \end{cases} \quad (5-57)$$

și

$$s = \frac{d_{1j}^2 - d_{2j}^2 + x_2^2 + y_2^2 - x_1^2 - y_1^2}{2(x_2 - x_1)} \quad (5-58)$$

După cum se observă, se obțin două locații simetrice față de dreapta ce unește WIT-urile 1 și 2. Soluția corectă este poziția mai apropiată de cea obținută prin metoda LOAD care folosește doar un singur reper din cele 2 posibile:

$$\begin{cases} x_{j_LOAD} = x_i + d_{ij} \cos \beta_{ij} \\ y_{j_LOAD} = y_i + d_{ij} \sin \beta_{ij} \end{cases} \quad (5-59)$$

unde, β_{ij} este unghiul dintre axa de referință Ox proiectată în punctul i și dreapta care unește cele două WIT-uri i și j iar $j = \{3; 4\}$. Valoarea $i = \{1; 2\}$ se selectează, în funcție de confiență, iar dacă confiența dintre w_1 și w_2 este egală, atunci se alege aleator.

La pasul 2 se calculează eroarea curentă de estimare. La pasul 3, punctul nominal se actualizează folosind eroarea curentă de estimare. Valoarea ε de la pasul 4 este o valoare de prag foarte mică care se determină în mod euristic. Dacă la pasul 4 condiția este îndeplinită atunci ultima valoare nominală calculată reprezintă soluția finală estimată. Pașii 2-4 se repetă cât timp condiția de la pasul 4 nu este îndeplinită.

Dacă dorim să generalizăm modelul matematic COOPERARE $m:n$, m WIT-uri pentru Web1 și n WIT-uri pentru Web2, atunci ecuația (5-47) o vom rescrie sub următoarea formă:

$$\begin{bmatrix} \delta \hat{x}_{Web2_1} \\ \delta \hat{y}_{Web2_1} \\ \vdots \\ \delta \hat{x}_{Web2_n} \\ \delta \hat{y}_{Web2_n} \end{bmatrix} = (A^T A)^{-1} A^T b \quad (5-60)$$

unde matricea b are următoarea formă:

$$b = \begin{bmatrix} \tilde{d}_{Web1_1,Web2_1} - \bar{d}_{Web1_1,Web2_1} \\ \tilde{d}_{Web1_1,Web2_2} - \bar{d}_{Web1_1,Web2_2} \\ \tilde{d}_{Web1_1,Web2_3} - \bar{d}_{Web1_1,Web2_3} \\ \vdots \\ \tilde{d}_{Web1_1,Web2_n} - \bar{d}_{Web1_1,Web2_n} \\ \tilde{d}_{Web1_2,Web2_1} - \bar{d}_{Web1_2,Web2_1} \\ \tilde{d}_{Web1_2,Web2_2} - \bar{d}_{Web1_2,Web2_2} \\ \vdots \\ \tilde{d}_{Web2_n-2,Web2_n-1} - \bar{d}_{Web2_n-2,Web2_n-1} \\ \tilde{d}_{Web2_n-2,Web2_n} - \bar{d}_{Web2_n-2,Web2_n} \\ \tilde{d}_{Web2_n-1,Web2_n} - \bar{d}_{Web2_n-1,Web2_n} \end{bmatrix} \quad (5-61)$$

iar matricea A are următoarea formă:

$$A = \begin{bmatrix} a_{Web1_1,Web2_1} & b_{Web1_1,Web2_1} & -a_{Web1_1,Web2_1} & -b_{Web1_1,Web2_1} & \cdots & 0 & 0 \\ a_{Web1_1,Web2_2} & b_{Web1_1,Web2_2} & 0 & 0 & \cdots & 0 & 0 \\ a_{Web1_1,Web2_3} & b_{Web1_1,Web2_3} & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{Web1_1,Web2_n} & b_{Web1_1,Web2_n} & 0 & 0 & \cdots & -a_{Web1_1,Web2_n} & -b_{Web1_1,Web2_n} \\ 0 & 0 & a_{Web1_2,Web2_1} & b_{Web1_2,Web2_1} & \cdots & 0 & 0 \\ 0 & 0 & a_{Web1_2,Web2_2} & b_{Web1_2,Web2_2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & -a_{Web2_n-2,Web2_n} & -b_{Web2_n-2,Web2_n} \\ 0 & 0 & 0 & 0 & \cdots & -a_{Web2_n-1,Web2_n} & -b_{Web2_n-1,Web2_n} \end{bmatrix} \quad (5-62)$$

5.5 Simulatorul SiMuLoC

Pentru a putea valida modelul matematic, autorul tezei a dezvoltat simulatorul SiMuLoC. Simulatorul a fost dezvoltat în Microsoft Visual Studio utilizând limbajul de programare Visual Basic, baza de date Microsoft Access și biblioteca pentru operații cu matrici cMatLib.vb, dezvoltată de Medeiros [122]. Au fost alese aceste tehnologii deoarece a fost prevăzut un proiect de dimensiuni mai mici pentru o dezvoltare cât mai rapidă punându-se accent doar pe algoritmi și validarea acestora. Planul principal este ca după testare și validare, modelul să fie inclus într-un proiectul de dimensiuni mai mari, Simon-TX. Acest proiect este descris în subcapitolul 6.4 .

Simulatorul poate fi executat pe orice calculator cu sistem de operare Windows. În bara de meniu al simulatorului se pot găsi următoarele opțiuni: File, View și Insert.

La opțiunea File, găsim următoarele funcții: creare proiect, deschidere proiect, ștergere proiect respectiv salvare proiect. La crearea unui proiect, utilizatorul introduce denumirea proiectului, tipul proiectului (de exemplu ILS), denumirea pânzei din „home position”, numărul de roboți care formează pânză din „home position”, ID-

ul roboților și coordonatele acestora. De asemenea, tot la crearea proiectului se mai specifică toleranța măsurării distanței cu ajutorul modulelor IRULT (în milimetri), valoarea ε (epsilon) conform condiției (5-49) din modelul matematic și numărul de generări folosind metoda Monte Carlo.

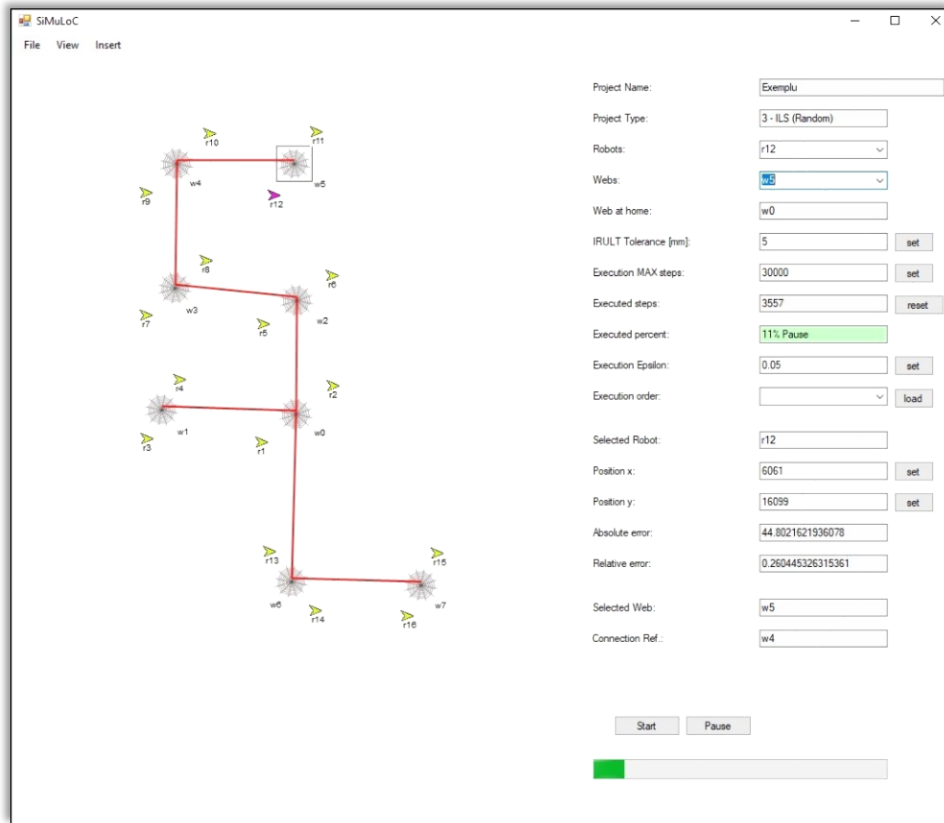


Figura 5-7 Interfața simulatorului SiMuLoC

La opțiunea Insert, găsim următoarele funcții: creare robot, creare pânză respectiv creare conexiune. Pentru crearea unui robot se specifică ID-ul robotului și coordonatele acestuia. La crearea unei pânze, se specifică denumirea pânzei, numărul de roboți care formează pânza și de asemenea se selectează în „drop down list”, roboții care formează pânza. Pentru crearea unei conexiuni, se selectează pânza sursă respectiv pânza de referință. Prin conexiune se înțelege localizarea unei pânze în raport cu altă pânză.

La opțiunea View, găsim următoarele funcții: vizualizare completă, vizualizează doar roboții, vizualizează doar pânzele, vizualizează doar ID-urile roboților, vizualizează doar pozițiile generate și o funcție de vizualizare a conexiunilor care poate fi activată în combinație cu oricare funcție deja enumerată.

În fereastra principală a simulatorului, pe partea stângă, se află porțiunea de vizualizare iar pe partea dreaptă se află porțiunea de proprietăți. Proprietățile pot fi observate în Figura 5-7. Informațiile din proprietăți apar în funcție de selecția făcută pe porțiunea de vizualizare. Se pot selecta atât pânze cât și roboți. De asemenea,

este posibilă și o selectare după ID, aceasta se poate realiza pe porțiunea de proprietăți prin intermediul „drop down list”-urilor. Tot în porțiunea de proprietăți se pot seta valorile proiectului, spre exemplu, valoarea epsilon. Pentru simulările efectuate s-a ales pentru epsilon valoarea 0.05, fiind valoarea standard a simulatorului. Pe porțiunea de vizualizare, folosind funcția „drag and drop” cu ajutorul mouse-ului, roboții pot fi mutați foarte simplu de pe o poziție pe alta.

Sub porțiunea de proprietăți se află două butoane. Primul reprezintă butonul de start al simulării, iar cel de-al doilea reprezintă intrarea în modul de pauză. Sub aceste două butoane se află o bară de progres prin care se poate observa progresul de generare a pozițiilor. Valoarea standard pentru numărul de generări este 30000. Aceasta reprezintă o valoare optimă de observare a toleranțelor în localizare prin metoda Monte Carlo.

Așa cum s-a specificat deja, în simulator se pot crea proiecte. Proiectele pot fi salvate, cu scopul de a reveni oricând la o configurație deja începută sau se poate continua o generare de poziții (prin ieșirea din modul de pauză).

Rezultatele se stochează în baza de date Microsoft Access, dar de asemenea rezultatele pot fi observate pe interfața grafică (porțiunea din stânga) prin simpla selectare a robotului dorit sau prin vizualizarea pozițiilor generate pe suprafața de lucru (porțiunea din stânga).

5.6 Exemplificarea metodologiei pe un studiu de caz

Există mai multe metode de monitorizare și supraveghere a mediului, dar cele mai interesante sunt cele bazate pe roboții mobili. În [123], autorii prezintă patru clase diferite de roboți de siguranță inteligenți ISR (Intelligent Security Robots), dezvoltati în ultimul deceniu: roboți comandați de la distanță, roboți distribuiți, roboți de supraveghere și roboți de procuratură (LERs). Roboții comandați de la distanță [124], sunt roboți controlați de un operator. Roboții distribuiți [125] sunt roboți cu capacitate autonomă care depind de membrii echipei din care fac parte, pentru a lua decizii. Roboții de supraveghere [126] sunt roboți autonomi care iau decizii în cazul în care o activitate anormală este percepută într-o zonă monitorizată. Roboții de procuratură (LERs) [127] sunt roboți autonomi care folosesc arhitecturi versatile pentru a gestiona armament. Toate aceste categorii au fost studiate și analizate în [123]. Unul din rezultatele comparației concluzionează că roboții distribuiți sunt considerați o soluție ideală deoarece probabilitatea de finalizarea cu succes a sarcinii este mai mare decât în cazul sistemelor cu un singur robot.

Un aspect crucial în aplicațiile de supraveghere este originea activităților anormale. Originea poate fi obținută cu exactitate dacă localizarea și mobilitatea sistemului robotic sunt gestionate în mod continuu în timpul întregului proces de supraveghere.

În cele ce urmează, considerăm o aplicație de supraveghere pentru a valida modelul matematic propus. Aplicația rulează într-un mediu fără senzori preinstalați. Ca studiu de caz, vom presupune că mediul este un birou cu 6 camere și care trebuie să fie monitorizat numai în timpul nopții (între orele 10pm și 7am). Inițial, toți roboții sunt poziționați în prima încăpăre. Când sistemul devine activ, roboții se îndreaptă spre locul țintă. Fiecare robot are un obiectiv predefinit de monitorizare (Figura 5-8.a).

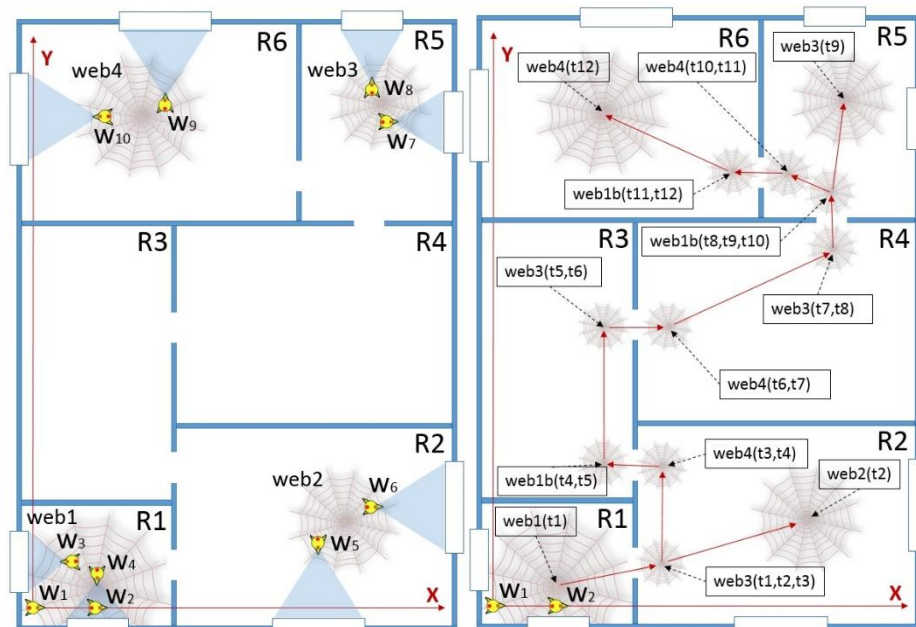


Figura 5-8 a) Cerințele pentru supraveghere b) Configurarea sistemului [121]

Pentru a menține confiența de localizare la valori ridicate, în timpul procedurii de supraveghere, roboții definesc un set de „pânze de localizare”. O pânză este compusă din cel puțin doi roboți.

Traseul roboților mobili, densitatea pânzelor și numărul roboților sunt specifice pentru fiecare aplicație în parte. Acestea pot fi determinate la nivelul III de localizare – CENTRALIZARE.

În acest studiu de caz se folosesc 10 roboți mobili. În conformitate cu cele discutate în capitolul 3, fiecare robot este construit pe baza modelului WIT și trebuie să fie echipat cu un modul IRULT. Deci, fiecare robot poate efectua procedura de aliniere și poate măsura distanța față de un alt robot.

Prima instalare (calibrare) se poate realiza prin poziționarea manuală a pânzelor. Sistemul global de referință xOy este definit de roboții w_1 și w_2 . Orientarea robotului w_1 , este aceeași cu orientarea robotului w_2 . Robotul w_1 este coordonatorul sistemului.

Pânza web_1 este prima pânză definită în sistem și inițial este formată din doi roboți: w_1 și w_2 . Pânza web_1 este poziționată în camera R1. Utilizatorul specifică care dintre camere vor fi monitorizate. Aici, regulile de mobilitate se stabilesc prin învățare, prin plasarea manuală a pânzelor.

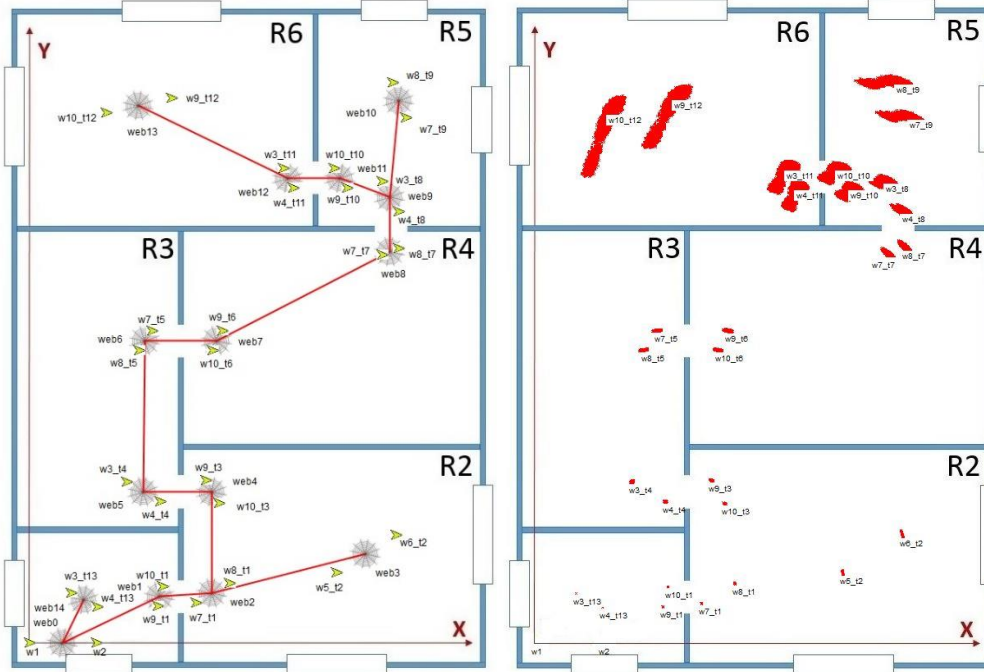


Figura 5-9 Simulare - metoda ILS

A doua pânză (web_2) este poziționată în camera R2. Pentru a propaga coordonatele din camera R1 în R2, două pânze temporare sunt necesare (web_3 și web_4), fiecare având doi roboți (w_7 și w_8 respectiv, w_9 și w_{10}). Pașii se repetă în mod asemănător (Figura 5-8.b) pentru a poziționa web_3 în R5 și web_4 în R6. Primul moment notat cu t_1 începe atunci când se realizează localizarea pentru pânza temporară web_3 . Pentru a abstractiza ceea ce se întâmplă înainte de momentul t_1 , se folosește aceeași notație t_1 . De asemenea, pentru studiul de caz expus, prin abstractizare s-a considerat pentru camera R1 doar o singură pânză, dar din punct de vedere tehnic conform metodologiei expuse în capitolul 4, această pânză este compusă din două pânze așa cum se va evidenția în continuare.

Un set experimente, folosind simulatorul SiMuLoC, au fost efectuate pentru a evalua procesul de propagare a erorilor și pentru a valida modelul matematic. Rezultatele sunt prezentate în Figura 5-9 și Tabelul 5-1. În Tabelul 5-1 se poate observa toleranța pozițiilor estimate. Toleranța calculului distanțelor (ϵ) folosind modulul IRULT a fost considerată $\pm 2\text{mm}$. Experimentele au fost repetate de 30.000, de fiecare dată generându-se alte toleranțe pentru valoarea ϵ . În simulare, pânzele au fost denumite de la web_0 la web_{14} pentru a reduce lungimea textului. Denumirile corecte ale pânzelor, pentru studiul de caz expus, apar în Figura 5-8.b.

Tabelul 5-1 Simulare - metoda ILS

CONF1 - ILS - 30000 - IRULT 2 - EPSILON 0.05				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
w1	0	0	0	0
w2	2000	0	0	0
w9_t1	3850	1100	12.2660593098919	0.378099707081656
w10_t1	4003	1700	13.8979151745481	0.428401455087096
w7_t1	5000	1200	20.5939921949971	0.42718964227028
w8_t1	6000	1800	30.8030802232352	0.638960950202906
w5_t2	9230	2110	99.2819727906486	1.0352267777002
w6_t2	11010	3260	137.91453818023	1.43805384749343
w9_t3	5300	4900	59.7482105115063	0.759111529136227
w10_t3	5700	4200	51.1653788608793	0.650065142594299
w3_t4	2930	4855	65.6112587801089	0.660018977506943
w4_t4	3930	4255	46.6073147168071	0.468848072353357
w7_t5	3679	9400	156.091553083165	1.076784756308
w8_t5	3279	8820	143.433294898397	0.98946280207318
w9_t6	5800	9400	160.44588673303	0.962651589424223
w10_t6	5500	8820	143.537127534941	0.861201535149439
w7_t7	10600	11700	282.057605414006	1.25238389992481
w8_t7	11100	11900	301.816155569578	1.34011523432501
w3_t8	10600	13900	633.040711610129	2.61893801235145
w4_t8	11100	13000	480.327171558965	1.98715037578146
w7_t9	11310	15820	1089.33473065191	4.0214261650783
w8_t9	10910	16890	1310.50859037069	4.83791931587702
w9_t10	9550	13700	593.107962524904	2.30160578165661
w10_t10	9150	14300	748.128590044587	2.90317648230346
w3_t11	7620	14300	916.231394453333	3.35010771914222
w4_t11	7920	13700	770.695757333835	2.81797133495379
w9_t12	4260	16410	1782.30034607128	5.51029286896435
w10_t12	2300	15970	2098.83471215083	6.48891415691673
w3_t13	1250	1500	3.5927494781895	0.248334542928851
w4_t13	2040	1090	3.51758043342968	0.243138781163063

5.7 Reguli de formare a pânzelor

În acest subcapitol va fi folosit studiul de caz din subcapitolul anterior pentru a scoate în evidență regulile de formare a pânzelor pentru nivelul II de localizare - COOPERARE. Pentru a putea face o comparație cât mai corectă, următorii parametri de configurare au fost utilizați pentru toate simulările efectuate: toleranța de măsurare a distanței $\pm 2\text{mm}$, valoarea epsilon = 0.05 și 30000 de poziții generate.

5.7.1 Regula PRP (poziționare PeRpendiculară)

Configurația 1 din subcapitolul anterior (Figura 5-9) a fost modificată, astfel încât, pozițiile pânzelor au fost păstrate, doar modul de poziționare a roboților în cadrul pânzelor a fost modificat.

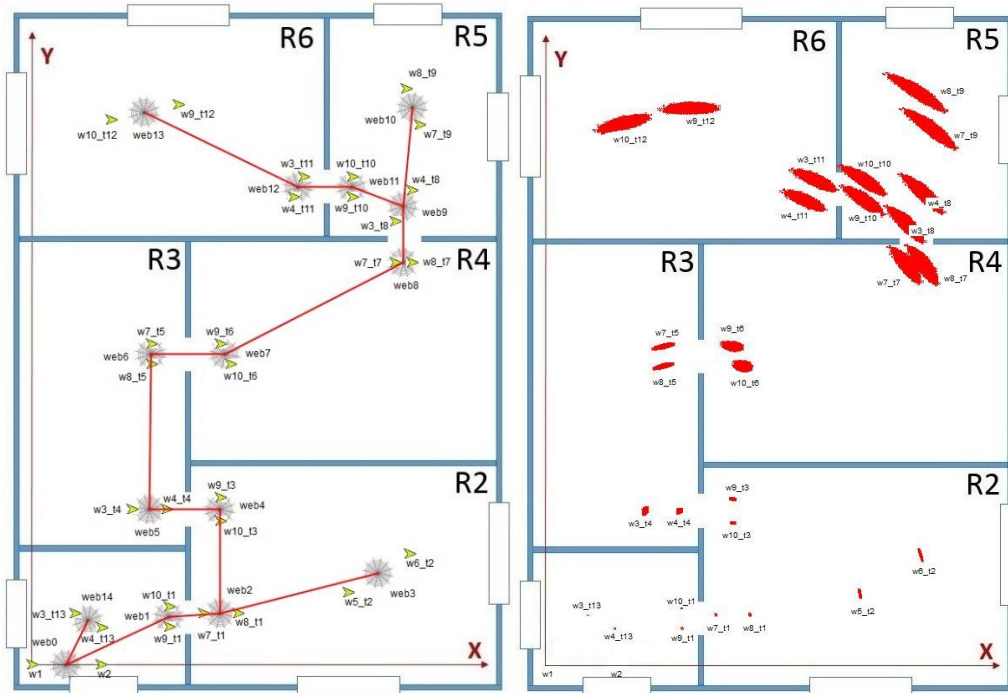


Figura 5-10 Regula PRP (poziționare PeRpendiculară)

Regula PRP constă în a poziționa roboții mobili pe o linie perpendiculară față de linia care unește cele două repere. După cum se observă față de configurația 1, noua configurație aduce o îmbunătățire din punct de vedere al acurateții de localizare. De exemplu, dacă roboții w_9 și w_{10} au avut la momentul t_{12} , în configurația 1, eroarea absolută 1782mm respectiv 2098mm, în noua configurație, eroarea absolută a devenit 941mm respectiv 918mm.

În Figura 5-10, în partea stângă, este prezentată configurația roboților, de asemenea se pot observa conexiunile între pânzele sistemului. Tot în Figura 5-10 dar în partea dreaptă, sunt reprezentate pozițiile generate ale roboților după efectuarea celor 30000 de generări. Fiecare poziție generată a fost marcată cu un punct. Astfel, prin generări multiple prin metoda Monte Carlo, s-au obținut zonele (suprafețele) de toleranță. Se observă că la distanță mai mare de origine, suprafața de generare a punctelor crește tot mai mult.

În Tabelul 5-2 pot fi observate erorile absolute ale fiecărui robot în parte și de asemenea eroarea relativă. Comparând valorile din Tabelul 5-2 cu valorile din Tabelul 5-1 se poate spune că regula PRP îmbunătățește confiența de localizare.

Tabelul 5-2 Regula PRP (poziționare PeRpendiculară)

CONF2 - ILS - 30000 - IRULT 2 - EPSILON 0.05				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
w1	0	0	0	0
w2	2000	0	0	0
w9_t1	4012	1100	13.0717287767802	0.394036901026871
w10_t1	4003	1700	14.0114268984777	0.42236335593559
w7_t1	5000	1500	21.6208569228478	0.449196114222397
w8_t1	6000	1500	30.709961383184	0.638031849083118
w5_t2	9230	2110	127.715814381989	1.33276299384483
w6_t2	11010	3260	189.327568667824	1.97570503274085
w9_t3	5500	4900	83.1064236146825	1.05689886733249
w10_t3	5500	4200	70.5429742942337	0.897124270146894
w3_t4	2930	4555	133.33680327678	1.34232951425436
w4_t4	3930	4555	87.0160328547151	0.876008620603793
w7_t5	3479	9400	345.833971370342	2.38695449909353
w8_t5	3479	8820	316.08705104702	2.18164053002745
w9_t6	5500	9400	358.322548138122	2.15085978673391
w10_t6	5800	8820	319.957120645336	1.92056823621926
w7_t7	10600	11800	774.488618172949	3.44001859974035
w8_t7	11100	11800	806.330286800562	3.58144860885289
w3_t8	10600	13000	853.467239073994	3.53196628367497
w4_t8	11100	13900	938.591814760417	3.88424357971131
w7_t9	11310	15820	1084.39598443941	4.0043138364619
w8_t9	10910	16890	1145.57175081172	4.23021560228909
w9_t10	9350	13700	837.759302483802	3.25195194898565
w10_t10	9350	14300	883.05891464768	3.4277926249744
w3_t11	7920	14300	824.610451479528	3.01594027118966
w4_t11	7620	13700	768.042782080341	2.80904899072839
w9_t12	4260	16410	941.010368553682	2.9099794079653
w10_t12	2300	15970	918.478551319096	2.84030204162825
w3_t13	1250	1500	3.67176330986553	0.253796060324773
w4_t13	2040	1090	3.5897717695139	0.248128720639426

5.7.2 Regula DST (DiSTanță mare)

Pentru a putea fi observată îmbunătățirea pe care o aduce regula DST, s-a moștenit configurația 2 de la regula PRP pentru a fi comparată cu configurația 3. Configurația 3 este asemănătoare cu 2, doar că distanțele între roboții care fac parte din aceeași pânză a fost mărită.

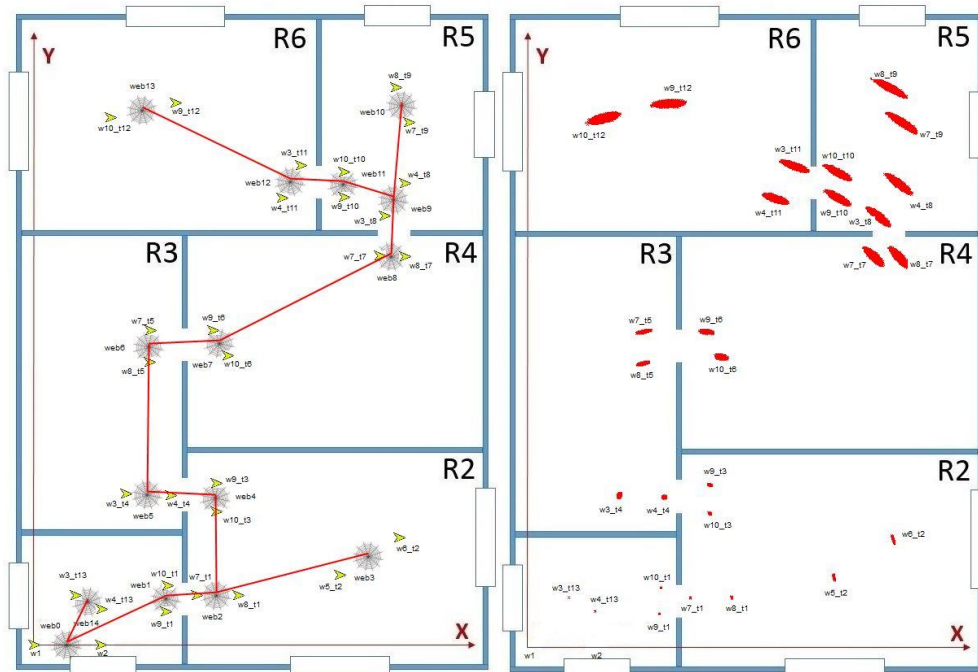


Figura 5-11 Regula DST (DiSTanță mare)

Prin mărirea distanței, se poate observa că eroarea absolută scade. De exemplu, dacă în configurația 2 eroarea absolută pentru roboții w_9 și w_{10} la momentul t_{12} a fost 941mm respectiv 918mm, acum, eroarea absolută este 549mm respectiv 543mm.

De asemenea există și o limitare în această regulă, distanța între roboți poate fi limitată de la caz la caz de obstacolele care apar pe suprafața de navigare (Figura 5-11).

În Tabelul 5-3 poate fi observată eroarea absolută și eroare relativă pentru fiecare robot în parte. Comparând valorile din Tabelul 5-3 cu valorile din Tabelul 5-2 se poate spune că regula DST îmbunătățește confiența de localizare.

Tabelul 5-3 Regula DST (DiSTanță mare)

CONF3 - ILS - 30000 - IRULT 2 - EPSILON 0.05				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
w1	0	0	0	0
w2	2000	0	0	0
w9_t1	3975	1009	12.65717029486	0.382229498670573
w10_t1	4024	1797	14.1426626835873	0.427089368435679
w7_t1	4907	1501	19.6315717858357	0.404938810583282
w8_t1	6159	1501	29.4298641835202	0.607047378992881
w5_t2	9230	2110	96.9394405442353	1.01132623575246
w6_t2	11010	3260	141.110374171955	1.47214201707532
w9_t3	5500	4900	61.1987953108129	0.783019792271466
w10_t3	5496	4037	46.8785380301251	0.59979649801788
w3_t4	2772	4578	85.8996802613549	0.870443031293217
w4_t4	4122	4529	54.8372275276353	0.555679397311388
w7_t5	3509	9526	228.399857672525	1.59038718049246
w8_t5	3484	8566	194.516825913626	1.35445384894532
w9_t6	5398	9526	225.930060257349	1.37043105233152
w10_t6	5840	8763	197.50879544865	1.19803529500727
w7_t7	10429	11791	415.297140174325	1.86178246011763
w8_t7	11189	11767	438.766859424889	1.96699751559873
w3_t8	10576	12997	468.566617108047	1.95085670172173
w4_t8	11189	13982	525.034355565326	2.18595767131354
w7_t9	11310	15820	598.089434038705	2.22395771193661
w8_t9	10910	16890	629.059146326073	2.33911662723938
w9_t10	9349	13564	461.763273264223	1.80246957878618
w10_t10	9349	14327	496.459904986595	1.93790612557741
w3_t11	8048	14524	480.949931032792	1.76795768845411
w4_t11	7484	13539	427.97642274659	1.57322864242252
w9_t12	4260	16410	549.652851346907	1.70795192563137
w10_t12	2300	15970	543.18780888106	1.68786291544653
w3_t13	1250	1500	3.56573150875665	0.246467033064691
w4_t13	2040	1090	3.38707771942675	0.234118299209164

5.7.3 Regula SRT (drum ScuRT)

Așa cum s-a observat în cazul configurațiilor 1, 2 și 3, conexiunile dintre pânzele de localizare formează un graf ponderat. Regula SRT spune că pe baza teoriei grafurilor ponderate în care nodurile reprezintă toate pozițiile pe care le poate lua o pânză într-un mediu, soluția SRT reprezintă drumul cel mai scurt de la nodul din originea sistemului până la nodul țintă.

Pentru a fi expusă această regulă s-a luat în considerare configurația 3 pentru comparație cu noua configurație. În configurația 4 redată în Figura 5-12, față de configurația 3, a fost exclusă pânza web₁₁ prin conectarea directă a pânzei web₁₂ la pânza web₉, astfel s-a obținut un graf cu un nod mai puțin între web₀ și web₁₃. Dacă eroarea absolută în cazul configurației 3 pentru roboții w₉ și w₁₀ la momentul t₁₂ a fost

549mm respectiv 543mm, acum eroare absolută devine 514mm respectiv 503mm. În Tabelul 5-4 poate fi observată eroarea absolută și eroare relativă pentru fiecare robot în parte.

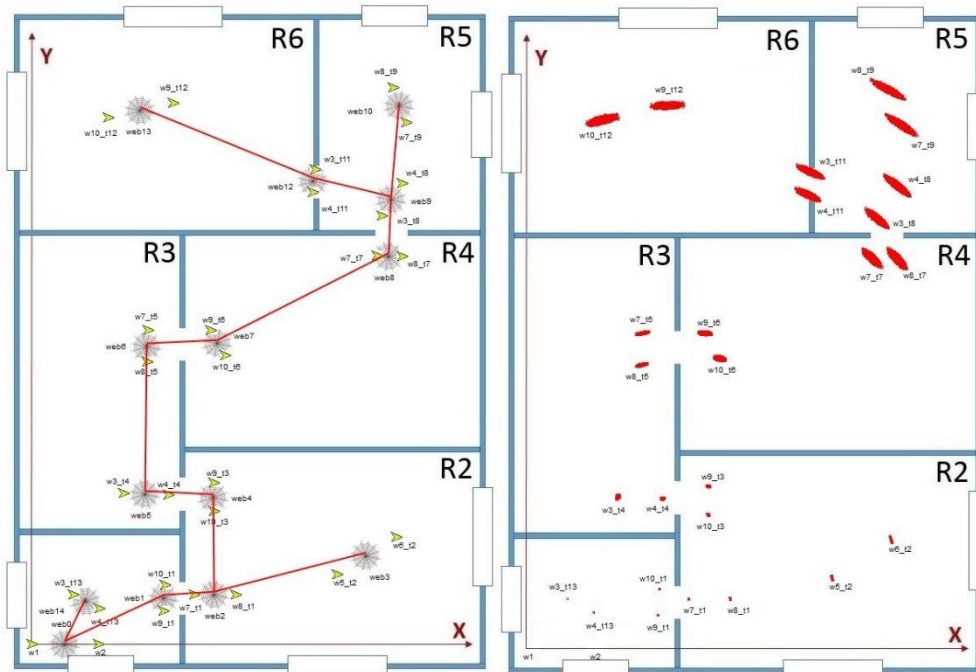


Figura 5-12 Regula SRT (drum ScuRT)

O altă regulă de care se ține cont în metodologia propusă este și vizibilitatea între roboții care efectuează procedura de calcul de distanță. Doi roboți care calculează distanța unul față de celălalt trebuie să fie vizibili adică să nu fie întreruptă vizibilitatea acestora de un alt robot mobil sau de un obstacol.

Comparând valorile din Tabelul 5-4 cu valorile din Tabelul 5-3, de asemenea, se poate spune, ca și în cazul celorlalte reguli, că și regula SRT îmbunătățește confianța de localizare.

Tabelul 5-4 Regula SRT (drum ScuRT)

CONF4 - ILS - 30000 - IRULT 2 - EPSILON 0.05				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
w1	0	0	0	0
w2	2000	0	0	0
w9_t1	3975	1009	12.9845527986884	0.392116009430619
w10_t1	4024	1797	14.4690326140503	0.436945300843628
w7_t1	4907	1501	18.8950972524017	0.389747611180245
w8_t1	6159	1501	29.9176336599842	0.617108559717122
w5_t2	9230	2110	101.787664519144	1.06190560855575
w6_t2	11010	3260	146.784864586161	1.53134146157587
w9_t3	5500	4900	63.0204896553284	0.806327811979701
w10_t3	5496	4037	46.9556218475213	0.600782761789552
w3_t4	2772	4578	91.5133995013653	0.927328258074463
w4_t4	4122	4529	58.4984750545725	0.592779737917613
w7_t5	3509	9526	217.251859848636	1.51276176947889
w8_t5	3484	8566	182.398543173009	1.27007217849833
w9_t6	5398	9526	215.058616193452	1.30448779311281
w10_t6	5840	8763	187.194106262514	1.13546916131206
w7_t7	10429	11791	426.190102097691	1.91061574955261
w8_t7	11189	11767	452.639345565041	2.02918805070118
w3_t8	10576	12997	471.567329953075	1.96335003895475
w4_t8	11189	13982	523.186514339175	2.17826426485192
w7_t9	11310	15820	592.334432491134	2.20255810287229
w8_t9	10910	16890	623.109931205077	2.31699484729259
w3_t11	8564	14401	470.557161833835	1.7796382352473
w4_t11	8490	13711	440.287125373982	1.66515753314441
w9_t12	4260	16410	514.0342402244	1.60107701581852
w10_t12	2300	15970	503.221471035287	1.56739817719003
w3_t13	1250	1500	3.71443605412842	0.256745644342912
w4_t13	2040	1090	3.60646552013655	0.249282609869892

5.7.4 Regula PNR (configurație PlaNaRă)

Următoarea regulă este valabilă pentru pânzele formate din cel puțin trei roboți. În continuare se consideră două configurații (5 și 6) pentru a compara două situații. Configurația 5 constă din pânze formate din 3 roboți poziționați coliniar (Figura 5-13). Complementar cu configurația 5, în configurația 6, roboții sunt poziționați astfel încât să formeze un plan (Figura 5-14).

Prin simulare, folosind configurațiile 5 și 6, s-a reușit să se reconstituie problema expusă de Seong Yun Cho în lucrarea [119] și anume, atunci când reperatele nu formează o linie dreaptă, eroarea în punctul „minim global” este aproape de zero iar eroarea în punctul „minim local” este mai depărtată de zero (Figura 5-5).

Metodologia propusă implică reguli de formare a pânzelor, prin urmare, un caz asemănător cu configurația 5 poate fi evitat implicând o configurație asemănătoare cu configurația 6.

Dacă eroarea absolută în cazul configurației 5 pentru roboții w_{40} , w_{41} și w_{42} a fost 1998mm, 2167mm respectiv 2350mm, pentru configurația 6, eroare absolută devine 389mm, 346mm respectiv 395mm. Comparând valorile din Tabelul 5-5 cu valorile din Tabelul 5-6, se poate spune, că și regula PNR îmbunătățește conștiența de localizare.

Dacă comparăm oricare configurație de la 1 la 4 cu configurația 6, putem spune că implicarea a trei roboți în formarea unei pânze îmbunătățește conștiența de localizare.

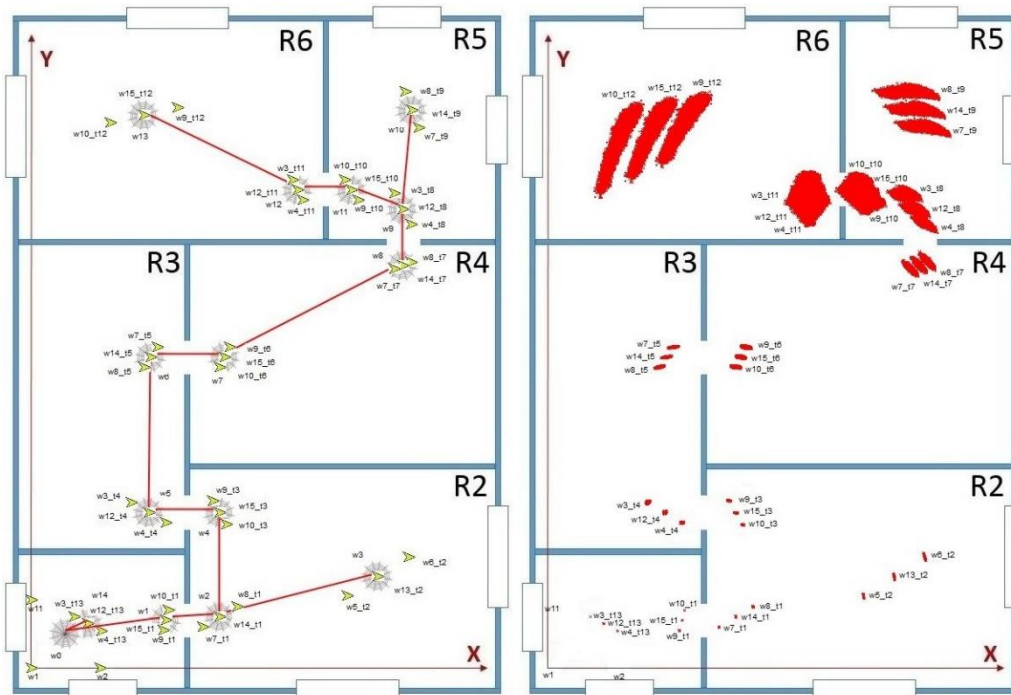


Figura 5-13 Configurație coliniară

Tabelul 5-5 Configurație coliniară

CONF5 - ILS - 30000 - IRULT 2 - EPSILON 0.05				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
w1	0	0	0	0
w2	2000	0	0	0
w11	0	2000	0	0
w9_t1	3850	1100	5.53095474146622	0.170490971451914
w10_t1	4003	1700	6.61381730330142	0.203870071217835
w15_t1	3927	1400	6.54950644429749	0.201887697226545
w7_t1	5000	1200	14.986963594584	0.328044410538952
w14_t1	5500	1500	20.6096861149725	0.451118219531956
w8_t1	6000	1800	27.2079715306664	0.595545783934749
w5_t2	9230	2110	82.2811404845192	0.902614090899248
w13_t2	10120	2685	107.059510327909	1.17442979056574
w6_t2	11010	3260	124.662592055955	1.36753345341512
w9_t3	5300	4900	65.6712429510207	0.826374634712289
w15_t3	5500	4550	58.5453520894115	0.736705927479904
w10_t3	5700	4200	53.7249632346779	0.676048524025966
w3_t4	2930	4855	72.6017486848163	0.714092188799358
w12_t4	3430	4555	64.2541760960546	0.631987604694251
w4_t4	3930	4255	52.017388623015	0.511629699977241
w7_t5	3679	9400	173.589485434753	1.17811904334398
w14_t5	3479	9110	162.901692473119	1.10558301163744
w8_t5	3279	8820	158.653352984312	1.07675033411772
w9_t6	5800	9400	177.363966232416	1.05070573116202
w15_t6	5650	9110	167.81520589221	0.994138789025643
w10_t6	5500	8820	157.940301281407	0.935639765296928
w7_t7	10600	11700	346.316305801187	1.54134858044361
w14_t7	10850	11800	344.262721216892	1.53220869984659
w8_t7	11100	11900	376.880985857408	1.67738267825306
w3_t8	10600	13900	690.080136398301	2.82896293275355
w12_t8	10850	13450	599.569731830909	2.45791822932813
w4_t8	11100	13000	535.520581102203	2.19535064662456
w7_t9	11310	15820	1185.83040617681	4.41581565793337
w14_t9	11110	16355	1304.44983439888	4.85753272451364
w8_t9	10910	16890	1426.22476028284	5.31100028754884
w9_t10	9550	13700	645.884665197254	2.51509292566301
w15_t10	9350	14000	728.193168975694	2.83560453823062
w10_t10	9150	14300	811.917917668243	3.16163104804911
w3_t11	7620	14300	1003.88343262967	3.65569539106607
w12_t11	7770	14000	930.123074546327	3.38709308892196
w4_t11	7920	13700	860.356801559331	3.13303546198978
w9_t12	4260	16410	1998.64755278226	6.25827339157638
w15_t12	3280	16190	2167.53663410675	6.78710802393036
w10_t12	2300	15970	2350.58033670857	7.36026437252917
w3_t13	1250	1500	2.69115309121424	0.183396023830488
w12_t13	1645	1295	3.03424579385767	0.206777019016336
w4_t13	2040	1090	2.93917558661581	0.200298198450602

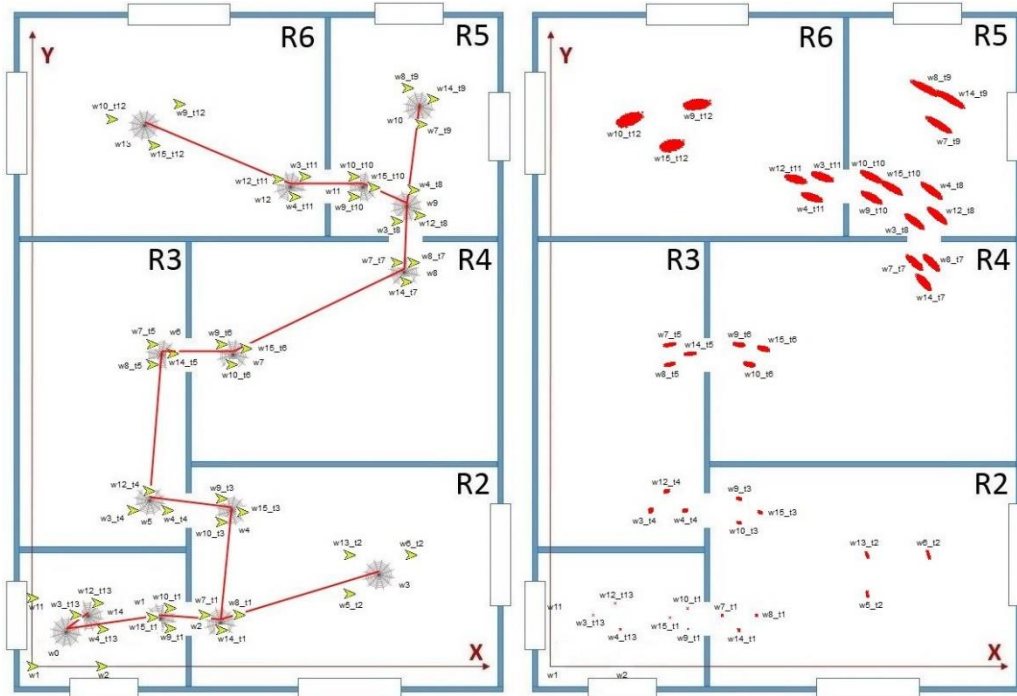


Figura 5-14 Regula PNR (configurație PlaNaRă)

Tabelul 5-6 Regula PNR (configurație PlaNaRă)

CONF6 - ILS - 30000 - IRULT 2 - EPSILON 0.05				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
w1	0	0	0	0
w2	2000	0	0	0
w11	0	2000	0	0
w9_t1	4012	1100	6.05152175517545	0.200079703064348
w15_t1	3484	1427	5.93353094778591	0.19617860732977
w10_t1	4003	1700	7.11074945552881	0.235100640332397
w7_t1	5000	1500	14.4018291219284	0.318298097548601
w14_t1	5496	1058	21.4323780810373	0.473681857448888
w8_t1	6000	1500	28.6739322684519	0.633729091840774
w5_t2	9230	2110	90.0181607928412	1.02950215060501
w13_t2	9226	3249	96.2989579349975	1.1013331467997
w6_t2	11010	3260	126.492560999414	1.4466454595103
w9_t3	5500	4900	70.3842234605288	0.880610992709839
w15_t3	6110	4504	68.6183848834656	0.858517733938788
w10_t3	5500	4200	56.2347342496686	0.703579903529742
w3_t4	2930	4555	70.523042476703	0.662616019151859
w12_t4	3386	5120	77.5582522745267	0.728717006097679
w4_t4	3930	4555	61.8087475510654	0.580738788525447
w7_t5	3479	9400	182.143769145452	1.20505049387233
w14_t5	4073	9133	171.989278951093	1.13786909381048
w8_t5	3479	8820	164.292436964828	1.08694732316567
w9_t6	5500	9400	181.884364232142	1.05781584721158
w15_t6	6208	9280	182.431015213853	1.06099509834616
w10_t6	5800	8820	167.640804492092	0.974977153092167
w7_t7	10600	11800	306.099915859925	1.35821163260407
w14_t7	10870	11225	296.4527769137	1.31540581770765
w8_t7	11100	11800	314.960504857478	1.39752740639118
w3_t8	10600	13000	334.327946539556	1.3860042134985
w12_t8	11263	13170	348.84209375934	1.44617468207636
w4_t8	11100	13900	365.103359110271	1.51358807819382
w7_t9	11310	15820	420.752445797154	1.54242296102697
w14_t9	11655	16567	445.104715871007	1.63169517058924
w8_t9	10910	16890	443.876173095342	1.62719149484185
w9_t10	9350	13700	334.387664658634	1.30544913661395
w15_t10	9938	13982	348.285618656137	1.35970673629324
w10_t10	9350	14300	348.508476561389	1.36057677335126
w3_t11	7920	14300	335.383452332381	1.20768967858089
w12_t11	7140	14228	328.982748357114	1.18464124231243
w4_t11	7620	13700	320.88077423305	1.15546666480207
w9_t12	4260	16410	389.593476515166	1.2282274307357
w15_t12	3533	15213	346.198544290727	1.0914211202458
w10_t12	2300	15970	395.048168189632	1.24542382221718
w3_t13	1250	1500	2.674542916769	0.151337150936527
w12_t13	1889	1846	3.36159003617433	0.190213234381678
w4_t13	2040	1090	3.09044654861087	0.174870768704433

5.8 Concluzii

În acest capitol au fost prezentate cele două niveluri de localizare: PREDICȚIE și COOPERARE. Cele două niveluri sunt aplicabile pe stratul de percepție și operare CORE-TX. Nivelul COOPERARE oferă confiență de localizare mai ridicată față de nivelul PREDICȚIE.

Nivelul PREDICȚIE a fost prezentat doar la nivel de idee în subcapitolul 5.2 Nivelul I de localizare - PREDICȚIE. Această metodă de localizare este des întâlnită în literatura de specialitate. În cercetarea de față s-a pus accentul pe nivelul COOPERARE, fiind considerat un obiect de studiu interesant care nu are încă o acoperire consistentă în literatura de specialitate.

Nivelul COOPERARE a fost expus în subcapitolul 5.4 Nivelul II de localizare - COOPERARE. S-a propus un model matematic, generalizat, oferind posibilitatea de a localiza o pânză în raport cu alta, indiferent de numărul de roboți care alcătuiesc o pânză. Trebuie remarcat faptul că localizarea prin COOPERARE se face secvențial din pânză în pânză, astfel o pânză care trebuie localizată este dependentă doar de pânza de referință. Acest lucru face posibilă implementarea modelului matematic la nivel de WIT. Avantajul acestei idei este că sistemul de localizare la nivelul COOPERARE poate fi funcțional și atunci când, din diverse motive, lipsește comunicarea cu entitatea BRAIN.

S-a ales ca validarea modelului să fie făcută prin simulări de tip Monte Carlo, deoarece, în practică, erorile de localizare cu propagare secvențială sunt foarte greu de reprodus. Atunci când distribuția zgomotului în cazul măsurării distanței nu poate fi modelată, singurul lucru ce se poate face este să fie găsit intervalul maxim și minim al distribuției. Cu un astfel de zgomot, găsirea celor mai defavorabile cazuri în practică este imposibilă. De aceea, aplicând metoda Monte Carlo în simulări, se poate evalua modelul rapid și corect. În simulările efectuate în simulatorul SiMuLoC se ține cont de acest aspect. De aceea s-a și reușit să se reproducă anumite probleme cunoscute în literatura de specialitate, cum ar fi problema minimelor locale și globale evidențiate de Seong Yun Cho în lucrarea [119]. În cazul metodologiei de față, aceste probleme sunt rezolvate prin folosirea regulilor de formare a pânzelor și prin apelarea la informației de poziție de la un nivel inferior, cum ar fi metoda LOAD sau nivelul I de localizare (PREDICȚIE).

În abordarea de față, s-a considerat că zgomotul aleator în cazul măsurării distanțelor poate fi descompus în două părți. Prima parte are distribuție gaussiană iar a doua nu poate fi modelată. Distribuția gaussiană se rezolvă la nivel de modul IRULT, prin măsurători repetitive, aplicând de exemplu filtru Kalman, iar distribuția non-gaussiană nu poate fi rezolvată ci trebuie cuantificată secvențial din pânză în pânză, iar acest lucru se poate face folosind simulatorul SiMuLoC. Aplicând diverse reguli de formare a pânzelor, evidențiate în subcapitolul 5.7 Reguli de formare a pânzelor, cuantificarea erorilor de propagare se modifică. De aceea metodologia propune un set de reguli, prin care, confiența în localizare crește.

6. APLICAREA METODOLOGIEI DE LOCALIZARE RELATIVĂ PENTRU STRATUL DE CONTROL ȘI SUPRAVEGHERE CORE-TX

În acest capitol este prezentat al treilea nivel (CENTRALIZARE) din cadrul metodologiei de localizare care cuprinde un algoritm destinat sistemelor cu procesare distribuită. Algoritmul BPF (Backtracking Particle Filter) propus, cu o tehnică originală în domeniul localizării, este un algoritm de tip „filtru de particule” sub forma de „backtracking” și care se bazează pe modelul algoritmilor probabiliști Las Vegas. Deoarece experimentele făcute în practică și expuse în capitolul 4 acoperă un caz prea izolat pentru a putea valida metodologia propusă, se execută mai multe seturi de experimente prin simulare pentru a cuprinde toate cazurile izolate ce pot apărea în practică. Prin analizarea rezultatelor experimentale și prin introducerea conceptului „bază” cu rolul de a reseta propagarea erorilor în localizare, se arată că metodologia de localizare relativă propusă este aplicabilă în contextul explorării roboților autonomi în medii necunoscute.

6.1 Introducere

Conform [16], stratul de abstractizare superior pentru modelul CORE-TX este reprezentat de entitatea BRAIN. De obicei, BRAIN se află pe un sistem cu putere mare de procesare și supraveghează toate WIT-urile. De asemenea, BRAIN este responsabil pentru interfața cu utilizatorul, acesta reprezentând punctul de intrare pentru orice informație de la utilizator. BRAIN este conectat la rețeaua WIT-urilor printr-un dispozitiv gateway/access point care de fapt este tot un WIT cu o interfață hardware adițională pentru rețea cu fir. Orice comandă elaborată de un utilizator trebuie să treacă prin BRAIN care o trimite mai departe prin sistemul CORE-TX.

Aplicarea metodologiei pentru stratul de control și supraveghere CORE-TX presupune centralizarea coordonatelor și gestionarea mobilității roboților mobili în formarea pânelor, astfel încât confiența în localizare să fie conform cerințelor. Metodologia aplicată la acest strat prin nivelul CENTRALIZARE, completează nivelul COOPERARE oferind o confiență de localizare mai ridicată. Nivelul CENTRALIZARE are la bază echipamentul cu poziție fixă: BRAIN.

La nivelul CENTRALIZARE, coordonatele roboților se obțin printr-un „filtru de particule” sub o nouă abordare. Sub noua formă, „filtrul de particule” se bazează pe metoda Las Vegas [128] și nu pe metoda Monte Carlo [129], așa cum se întâlnește în literatura de specialitate.

Algoritmii Monte Carlo sunt algoritmi aleatorii care nu oferă întotdeauna răspunsul corect. Însă, timpul de rulare a algoritmilor este determinist. Există, de asemenea, o probabilitate de adevăr că răspunsul este corect. Algoritmii Las Vegas sunt algoritmi aleatorii care oferă întotdeauna răspunsul corect. Însă, timpul de rulare nu este determinist, adică poate varia pentru aceleași date de intrare.

În acest capitol se pune accentul pe nivelul CENTRALIZARE care are la bază entitatea BRAIN cu un algoritm de tip „filtru de particule” sub forma de „backtracking” și care se bazează pe modelul Las Vegas.

6.2 Filtre de particule

Obiectivul principal al filtrării particulelor este de a urmări o variabilă de interes pe măsură ce aceasta evoluează în timp, de obicei cu funcția densitate de probabilitate PDF (Probability Density Function) non-gaussiană și potențial multi-modal [130]. Avantajul acestei abordări este că poate utiliza orice funcție de distribuție chiar și când formula matematică pentru funcția de distribuție nu poate fi modelată cu ușurință folosind o funcție analitică. Distribuția de probabilitate este aproximată folosind un set de particule. Fiecare particulă are o stare și o pondere. Însurarea tuturor particulelor oferă o aproximare a distribuției [131].

În lucrarea [114] se prezintă un exemplu simplu pentru filtre de particule. Un set de particule este o colecție de N variabile. Distribuția de probabilitate $E(x_t)$ este reprezentată de un set S_t astfel:

$$E(x_t) \approx S_t = \{x_t^{(i)}, w_t^{(i)} \mid i = 1, \dots, N\}. \quad (6-1)$$

unde, fiecare $x_t^{(i)}$ este o stare, iar $w_t^{(i)}$ sunt ponderi nenegative denumite factori de importanță, care însumați dau unitatea. Astfel $E(x_t)$ este funcția de distribuție a particulelor anterioare peste $x_t^{(i)}$ condiționată de toate datele de la senzori (măsurători z_t) disponibile la momentul t (6-2) și reprezintă incertitudinea stării la momentul t .

$$E(x_t) = p(x_t | z_1, z_2, \dots, z_t). \quad (6-2)$$

În Figura 6-1 se ilustrează utilizarea algoritmului pentru filtru de particule pentru o problemă simplă uni-dimensională. Un set de eșantioane uniform distribuite reprezintă poziția persoanei, poziție inițial necunoscută (Figura 6-1.a). Fiecare eșantion are același factor de importanță $w(x)$, după cum arată și înălțimea egală a tuturor eșantioanelor din figură. În Figura 6-1.b, senzorii detectează ușa din stânga. Filtrul de particule încorporează măsurătorile ajustând și normalizând factorul de importanță a fiecărui eșantion, așa cum este ilustrat în Figura 6-1.b. Poziționarea eșantioanelor a rămas aceeași, doar factorii de importanță atașați s-au modificat. Ponderile sunt proporționale cu probabilitatea de observare $p(z|x)$.

Când persoana se deplasează spre dreapta, se vor deplasa și eșantioanele conform pasului de predicție:

$$E^-(x_t) \leftarrow \int p(x_t | x_{t-1}) E(x_{t-1}) dx_{t-1}. \quad (6-3)$$

unde $E^-(x_t)$ reprezintă pasul de predicție, $p(x_t | x_{t-1})$ reprezintă modelul de deplasare și descrie dinamica sistemului, adică, modul în care se schimbă starea sistemului în timp, iar $E(x_{t-1})$ este pasul precedent.

Apoi, filtrul corectează estimarea folosind următoarea observație a senzorului:

$$E(x_t) \leftarrow \alpha_t p(z_t | x_t) E^-(x_t). \quad (6-4)$$

unde $E(x_t)$ este incertitudinea la momentul t , α_t este factorul de normalizare, $p(z_t | x_t)$ este modelul perceptiv al senzorului, iar $E^-(x_t)$ este incertitudinea apriorică.

Redistribuirea eșantioanelor noi se face pe baza ponderilor. Acolo unde ponderile sunt mai mari, numărul de eșantioane crește, iar acolo unde ponderea este mai mică, numărul de eșantioane scade. Numărul total de eșantioane rămâne tot timpul constant. Figura 6-1.c arată setul de eșantioane rezultat, care diferă de cel original prin acele eșantioane mai multe centrate în jurul celor 3 locații vechi. Această concentrare de eșantioane este rezultatul eșantionării proporțional cu ponderile.

În Figura 6-1.d, senzorii detectează o posibilă ușă nouă, cu probabilitatea $p(z|x)$. Balansând factorii de importanță cu proporțiile date de această probabilitate, obținem setul de eșantioane ilustrat în partea de jos (Figura 6-1.d). După următoarea

actualizare prin predicție, marea majoritate a eșantioanelor este consecventă cu locația adevărată a persoanei (Figura 6-1.e).

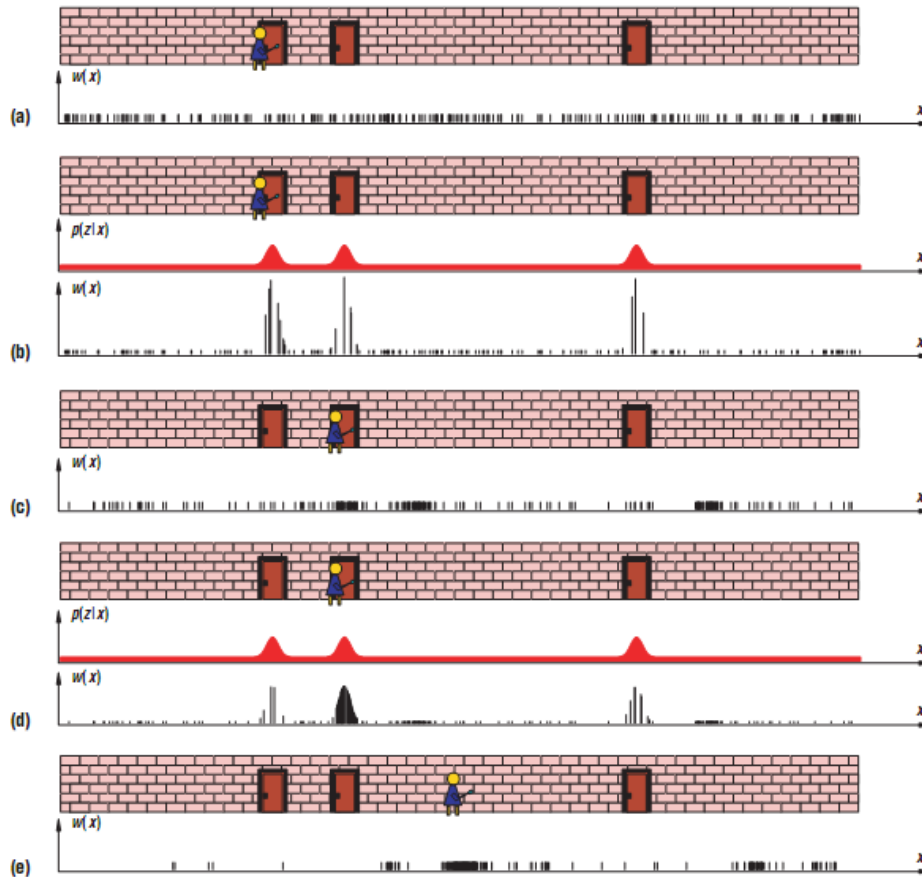


Figura 6-1 Aplicarea filtrului de particule pentru estimarea locației [114]

Avantajul cheie al filtrelor de particule este abilitatea lor de a reprezenta densități de probabilitate arbitrare. Mai mult, spre deosebire de filtrele Kalman, filtrele de particule pot converge către un anterior adevărat chiar în sisteme dinamice non-gaussiene și non-liniare. Comparate cu abordările tip grilă [132], filtrele de particule sunt foarte eficiente din cauză că ele își concentrează automat resursele (particulele) pe regiuni din spațiul stărilor cu mare probabilitate. Deoarece eficiența filtrelor de particule depinde foarte mult de numărul de eșantioane utilizate la filtrare, câteva îmbunătățiri au fost făcute ca să se facă utilizarea eșantioanelor disponibile mult mai eficace [133]. Deoarece, în cel mai rău caz complexitatea acestei metode crește exponențial cu dimensiunile spațiului stărilor, trebuie avut grijă la aplicarea filtrelor de particule la probleme de estimare multi-dimensională.

6.3 Nivelul III de localizare - CENTRALIZARE

6.3.1 Algoritmi de calcul

Algoritmii determiniști sunt cei care, date fiind niște informații de intrare vor produce, în mod previzibil, același rezultat. În general, ei folosesc fie un model analitic al calculului, fie un model iterativ care să fie convergent la rezultatul dorit.

De exemplu algoritmul „Heap Sort” [134] este un algoritm determinist cu proprietatea că rezultatul fiecărei operații este definit în mod unic și este predictibil.

Nu toți algoritmii pot fi construiți să fie determiniști. Problemele care împiedică acest lucru sunt: complexitatea ridicată și indisponibilitatea unor modele analitice. Procesele complexe, fie nu pot fi modelate pur determinist, fie solicită foarte multă putere de calcul (de ex.: determinanții Jacobieni). În unele scenarii, unde există necesitatea unui calcul rapid (care să fie făcut în timp real) și unde nu se dorește aflarea unui rezultat foarte precis, face acest tip de algoritmi nefezabili.

Algoritmii probabiliști sunt orice algoritmi care depind de variabile aleatoare în timpul execuției. În cele mai multe cazuri, algoritmii probabiliști sunt iterativi, iar variabilele aleatoare sunt folosite în luarea deciziilor dintre iterații.

Comparând algoritmii determiniști cu cei probabiliști de aceeași categorie, se poate spune că prin folosirea randomizării se ajunge la un algoritm mai simplu de descris și implementat [135]. De asemenea, la repetarea algoritmilor pentru aceleași date, algoritmul determinist trece prin exact aceiași pași în timp ce algoritmul probabilist nu [136].

Majoritatea algoritmilor probabiliști pot fi împărțiți în două mari categorii: algoritmi de tip Monte Carlo [129] și algoritmi de tip Las Vegas [128].

Algoritmii de tip Monte Carlo au întotdeauna timpul de execuție determinist dar rezultatul produs poate fi incorect, cu o anumită probabilitate [137]. Se poate estima precizia soluției, respectiv, probabilitatea ca soluția produsă să fie corectă. Un exemplu de algoritm Monte Carlo poate fi algoritmul lui Karger „min-cut”, adică rezolvarea problemei de tăietură minimă [138].

Algoritmii de tip Las Vegas produc întotdeauna un rezultat corect (respectiv, anunță un eventual eșec de a produce un rezultat corect). Caracterul lor aleator constă în variabilitatea resurselor necesare. Spre exemplu, pentru o problemă dată, poate exista o incertitudine legată de timpul necesar obținerii soluției corecte. Dar se poate face o estimare a timpului de execuție. Algoritmul este fezabil atâta timp cât estimarea de timp efectuată este finită. Un exemplu de algoritm Las Vegas poate fi „Randomized Quick Sort” cu alegere aleatoare a valorii elementului pivot [135].

În dezvoltarea unui algoritm de localizare pentru nivelul CENTRALIZARE, s-a ales modelul Las Vegas. Având în vedere propagarea erorilor în localizarea relativă, modelul Monte Carlo nu ar putea să minimizeze acest efect, chiar mai mult, ar introduce erori adiționale din cauza constrângerii de timp impuse conform definiției: „timpul de execuție este determinist, iar rezultatul produs poate fi incorect”. În schimb, modelul Las Vegas produce întotdeauna un rezultat corect în schimbul resurselor majore de calcul necesare.

Având ca și resurse de calcul un sistem pe care se află entitatea BRAIN, problema care derivă este dimensiunea resurselor sistemului. Dimensionarea resurselor depinde cel mai mult de complexitatea mediului în care se află pânzele de localizare. Prin urmare, se pune problema, de câte pânze avem nevoie astfel încât localizarea să poată fi utilizată în orice punct dorit în mediu. Soluții există la această problemă, de exemplu, construirea de baze noi atunci când resursele de calcul nu mai

ajung sau confiența de localizare scade foarte mult. Obiectivul metodologiei este de a minimiza crearea bazelor noi, deoarece acest lucru implică majorarea de costuri. De asemenea mărirea resurselor de calcul, implică majorarea de alte costuri. Un aspect foarte interesant este de găsit un punct de echilibru al costurilor, dar acesta nu este obiectivul metodologiei.

Obiectivul metodologiei este de a oferi la nivelul CENTRALIZERE un algoritm general care pune accent pe minimizarea erorilor de propagare în schimbul resurselor de calcul. Autorul tezei propune algoritmul BPF (Backtracking Particle Filter) care are la bază entitatea BRAIN și distribuirea sarcinii pe mai multe procese.

6.3.2 Algoritmul BPF (Backtracking Particle Filter)

Pentru a fi înțeles algoritmul BPF, mai întâi, autorul tezei expune în mod vizual o parte importantă din algoritm. Pentru a simplifica, în Figura 6-2 este reprezentat doar un singur pas de localizare. Nodul *necunoscut* w_3 se localizează având ca și repere două noduri *stabilite*: w_1 și w_2 . P_r reprezintă punctul cu coordonate reale, sau altfel spus, locul exact unde se află nodul w_3 . Presupunem că sistemul cunoaște coordonatele reale ale nodurilor: w_1 și w_2 . Problema care se pune este de a determina coordonatele punctului P_r . Din cauza toleranțelor în măsurarea distanțelor, se obține o poziție măsurată P_m . Poziția P_m nu coincide cu poziția P_r , dar se află la o anumită distanță maximă de poziția P_r .

Problema care se pune în continuare este de a stabili o suprafață unde ar putea să se afle P_r pornind de la punctul P_m . În Figura 6-2 se reprezintă vizual o soluție la această problemă. Suprafața căutată S_c este compusă din patru suprafețe: S_{11} , S_{22} , S_{12} și S_{21} . Suprafața S_{11} reprezintă intersecția celor patru cercuri date de distanțele d_{13m} , d_{23m} , $d_{13m}-\epsilon$ și $d_{23m}-\epsilon$ unde distanțele reprezintă razele cercurilor iar ϵ reprezintă toleranța maximă la măsurarea distanței. De asemenea, suprafața S_{22} reprezintă intersecția celor patru cercuri date de distanțele d_{13m} , d_{23m} , $d_{13m}+\epsilon$ și $d_{23m}+\epsilon$. Se observă că pentru S_{11} au fost luate toleranțele maxime inferioare iar pentru S_{22} cele superioare. În mod asemănător, suprafața S_{12} reprezintă intersecția celor patru cercuri date de distanțele d_{13m} , d_{23m} , $d_{13m}-\epsilon$ și $d_{23m}+\epsilon$ iar S_{21} reprezintă intersecția celor patru cercuri date de distanțele d_{13m} , d_{23m} , $d_{13m}+\epsilon$ și $d_{23m}-\epsilon$. În concluzie, avem suprafețele S_{ij} , unde, pentru indicii i și j se ia valoarea 1 atunci când la măsurarea distanței se consideră toleranța maximă inferioară, și se ia valoarea 2 pentru toleranța maximă superioară.

Pe baza celor patru suprafețe S_{ij} , se stabilește suprafața mare S_c . BPF nu are un mecanism prin care să indice faptul că de exemplu, în Figura 6-2, P_r se află pe suprafața S_{11} . De aceea, în cazul algoritmului BPF se consideră suprafața mare S_c pentru a genera cât mai multe poziții nominale P_n . Ideea este să se genereze uniform pe suprafață cât mai multe poziții P_n cu intenția de a atinge și poziția necunoscută P_r . Ţelul algoritmului BPF este de a determina o poziție cât mai apropiată de P_r prin filtrarea pozițiilor P_n .

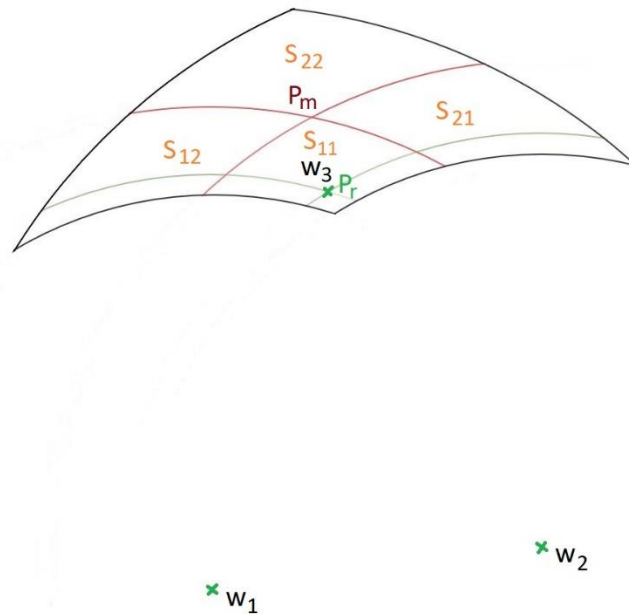


Figura 6-2 Reprezentare vizuală a unui pas al algoritmului BPF

Trebuie remarcat un aspect foarte important. Punctul P_n obținut prin filtrare devine punctul P_e (punct estimat). Doar a doua pânză are ca și reper puncte reale, celelalte pânze au ca și reper puncte estimate. De asemenea, doar a doua pânză construiește suprafața S_c pe baza punctului P_m , în rest, celelalte pânze construiesc suprafețele S_c pe baza punctelor P_e . Ideea algoritmului BPF este de a redefini punctele P_e , într-o buclă, până când se ajunge la cel puțin o soluție.

Dacă, de exemplu, o pânză este formată din doi roboți, și avem în sistem 10 pânze, atunci vom avea 18 de suprafețe S_c . Pentru pânza din home nu se creează suprafețe S_c . Cu cât crește numărul de suprafețe S_c , crește și durata de execuție a algoritmului. De aceea, algoritmul BPF este construit să fie executat pe mai multe procese. La rândul lor procesele pot fi distribuite pe mai multe procesoare/sisteme de calcul. Procesele copil generate de un proces părinte sunt concurente „cu un singur câștigător”, în sensul că, dacă un proces copil returnează o soluție, celelalte procese copil pot fi în mod automat terminate de procesul părinte. Prin astfel de abordare se câștigă enorm de mult timp în căutarea pozițiilor P_r .

În continuare se prezintă schema bloc a algoritmului BPF. Schema bloc (Figura 6-3) formată din 3 blocuri principale, prezintă modulele importate ale procesului părinte. Pentru a menține o generare uniformă a pozițiilor nominale, generarea și distribuția pozițiilor nominale se realizează de către procesul părinte. Primul bloc, blocul de distribuție, conține 3 module: modulul de setare, modulul de achiziție și modulul de generare.

În cadrul primului modul, modulul de setare, se stabilesc parametrii următori:

- numărul de roboți din axa *home*,
- coordonatele roboților din axa *home*,

- numărul de pânze,
- toleranța măsurării distanței cu ultrasunete (ϵ),
- numărul maxim de pași înapoi (back).
- numărul total de soluții,
- număr de soluții pentru un proces,

Optional, se mai pot stabili și următoarele:

- toleranța măsurării distanței cu UWB,
- increment pentru toleranță,
- număr maxim de seturi nominale pentru un increment.

În expunerea algoritmului, nu s-au luat în considerare parametrii opționali. Acești parametri se vor discuta în subcapitolele care urmează. Mai trebuie menționat faptul că în cadrul modulului de setare se execută și funcția de randomizare.

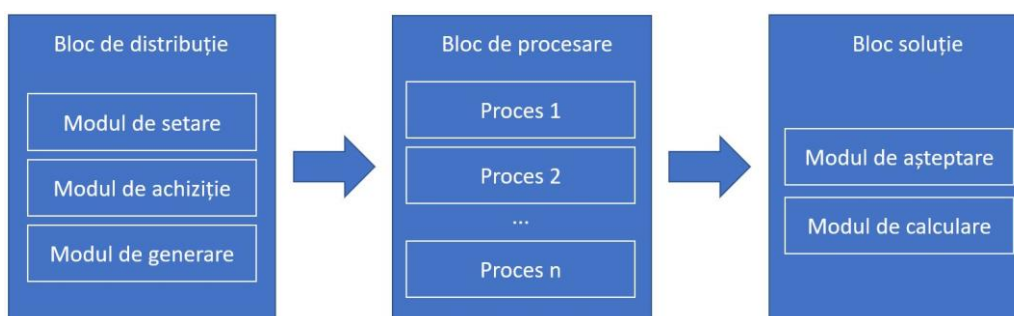


Figura 6-3 Schema bloc a algoritmului BPF

Specific pentru modulul de achiziție este achiziționarea măsurătorilor de distanță provenite de la roboții mobili, mai exact, modulul de percepție descris în subcapitolul 3.3. Trebuie menționat aici că măsurătorile vin deja filtrate cu filtrul Kalman. Filtrarea se realizează la nivel hardware (modulul de percepție).

Modulul de generare are ca și sarcină generarea valorilor nominale. Pentru fiecare suprafață S_c se generează câte un set de patru poziții nominale: câte o valoare pentru suprafețele S_{11} , S_{22} , S_{12} și S_{21} . Pentru fiecare set se generează câte un proces copil. Dacă avem n seturi, atunci avem n procese copil. Important este de menționat faptul că cele patru poziții sunt generate sub formă de distanțe nominale în cadrul procesului părinte. Procesul copil este cel care pe baza celor patru distanțe nominale, obține cele patru poziții nominale.

Blocul de procesare se referă la procesul părinte care generează n procese copil. Într-un proces copil se parcurg toate pânzele. Să zicem că avem 10 pânze iar fiecare pânză este formată de doi roboți. În total avem 18 suprafețe S_c . Pentru pânza din *home*, nu se construiesc suprafețe S_c . În cadrul unei suprafețe avem 4 poziții nominale stabilite de procesul părinte la crearea procesului copil sub forma a 4 distanțe nominale. De asemenea se ia în considerare și poziția P_m . Prin urmare, unei suprafețe S_c sunt asociate 5 poziții. O pânză are în exemplul nostru 2 suprafețe. De aici rezultă 25 de combinații posibile pentru a stabili poziția celor doi roboți în cadrul unei pânze. Problema este că suprafețele unei pânze depind de cum au fost selectate pozițiile nominale la pânza precedentă. Prin urmare, în exemplul expus, avem 25^9 variante posibile de validat, asta înseamnă aproximativ $4 \cdot 10^{12}$ variante, ceea ce

Înseamnă că un copil are foarte mult de căutat până să găsească soluția potrivită. Este posibil ca nici una din variante să nu fie bună. Foarte multe variante sunt retrase la nivel de pânză (foarte repede, fără multă procesare), atunci când se verifică dacă distanța dintre cei doi roboți din cadrul unei pânze se află la distanța corectă unul față de celălalt incluzând și toleranțele posibile. Dar foarte multe variante sunt reținute un timp foarte lung, parcurgând multe pânze și efectuând validări în serie. Aceste validări în serie duc la creșterea timpului de procesare a unui proces copil. Inițial, cele 25 de variante posibile de localizare a pânzelor, sunt adăugate într-o listă „albă”. Dacă pentru o actuală pânză, toate cele 25 de variante au fost transcrise în lista „neagră”, atunci se va trece la un pas înapoi (back), adică, se va lua în considerare pânza precedentă. Înainte să se realizeze o astfel de trecere, toate variantele din lista „neagră” (a pânzei curente) vor fi șterse, deoarece la prima revenire la pânza respectivă, suprafețele vor fi recalulate. După ce se realizează trecerea, varianta curentă va fi adăugată în lista „neagră” și se va lua următoarea variantă din lista „albă”. Fiecare pânză are propria listă „albă” și listă „neagră”. Procesul copil poate să ajungă la următoarele stări posibile:

- Nici o soluție. Acest lucru se întâmplă atunci când prin trecere la pânza precedentă (back) se ajunge la pânza cea mai apropiată de pânza din *home*, și se constată că toate variantele au fost adăugate în lista „neagră”. Cu alte cuvinte, toate variantele au fost verificate și nici una nu este bună. În acest caz procesul copil comunică procesului părinte mesajul, iar procesul se termină.
- Căutare întreruptă. Procesul se termină automat dacă numărul maxim de pași înapoi (back) este atins.
- Soluție găsită. De asemenea, putem să avem mai multe soluții găsite, dacă în modulul de setare, numărul de soluții pentru un proces a fost setat să fie mai mare decât 1.

Blocul soluție a procesului părinte conține două module: modul de așteptare și modul de calculare. Modulul așteptare reprezintă o buclă în care procesul părinte așteaptă ca un copil să returneze o soluție. Modulul de calculare reprezintă calculele pe care trebuie să le efectueze un părinte atunci când obține mai multe soluții.

În continuare se prezintă algoritmul BPF din cadrul unui proces copil:

```

01 for i ← 1 to Numar_de_panze - 1
02   do if solutii_proces = 0
03     then stop (Toate solutiile au fost gasite)
04     if pasi_inapoi = 0
05       then stop (Cautare intrerupta)
06     if i = 0
07       then stop (Fara solutii sau nu toate solutiile au fost gasite)
08   pozitie_estimata = False
09   if Panza(i).poz_nom_def = False
10     then for i1 ← 0 to 4
11       do for i2 ← 0 to 4
12         do Panza(i).com_nom_es(i1,i2,false)
13         Panza(i).actualiz_poz_nom()
14         Panza(i).com_nom_act(0,0)
15         for i1 ← 0 to 4
16           do for i2 ← 0 to 4
17             do if Panza(i).cond_indepl(i1,i2)
18               then if pozitie_estimata = False
19                 then pozitie_estimata = True
20                   Panza(i).com_nom_act(i1,i2)
21                   Panza(i).actualiz_poz_es()
22                 else Panza(i).com_nom_es(i1,i2,true)
23       else Panza(i).comb_nom_act_es()
24       if Panza(i).al_com_nom()
25         then pozitie_estimata = True
26           Panza(i).actualiz_poz_es()
27         else pozitie_estimata = False
28   if pozitie_estimata = False
29     then Panza(i).poz_nom_def = False
30       i -= 2
31       pasi_inapoi -= 1
32   else
33     Panza(i).poz_nom_def = True
34     if i = Numar_de_panze - 1
35       then Panza(i).actualiz_sol()
36         i = 0
37         solutii_proces -= 1

```

Algoritm 6-1 Implementarea unui proces copil standard

De asemenea, algoritmul este redat și în Figura 6-4 sub forma unui „Flowchart”.

Procesul copil moștenește, printre altele, două valori importante de la procesul părinte: *solutii_proces* (linia 02 și 37) și *pasi_inapoi* (linia 04 și 31). Aceste valori au fost stabilite de către procesul părinte în modulul de setare și reprezintă: număr de soluții pentru un proces respectiv numărul maxim de pași înapoi (back). De asemenea, procesul copil are o variabilă cu un rol important: *pozitie_estimata* (linia 08, 18, 19, 25, 27 și 28). Această variabilă poate avea două valori: „True” sau „False”. Prin această variabilă se stabilește dacă au fost definite punctele P_e pentru pânza curentă. Dacă și pentru ultima pânză se reușește să se definească puncte P_e , atunci procesul copil are o soluție de returnat pentru procesul părinte.

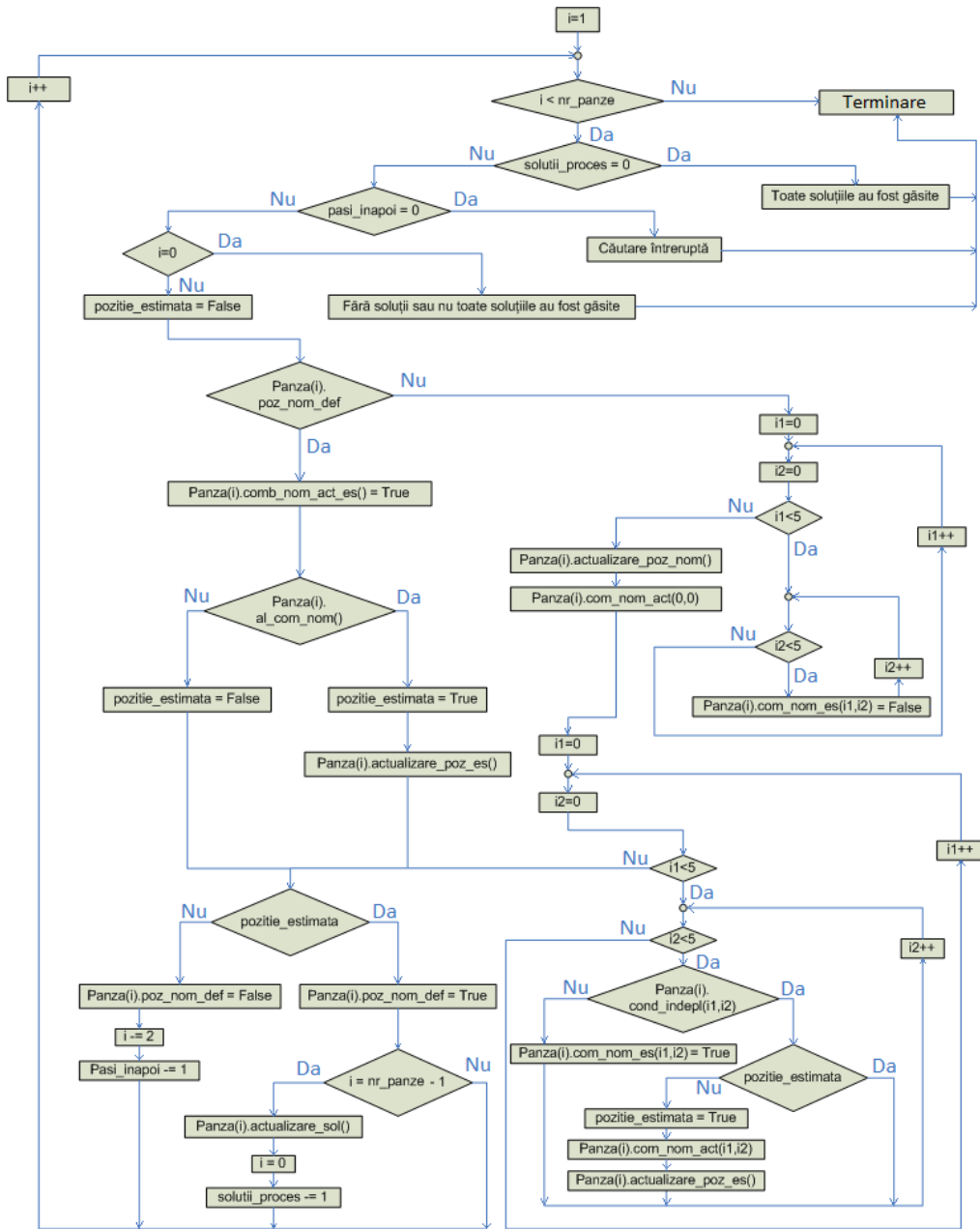


Figura 6-4 Schema logică a unui proces copil

Clasa „Panza” are un atribut cu un rol important: poz_nom_def (linia 09, 29 și 33). Acest atribut poate avea două valori: „True” sau „False”. Atributul poz_nom_def ține minte dacă pentru pânda curentă au fost deja definite suprafețele S_c.

Metoda `actualiz_poz_nom()` (linia 13) din cadrul clasei „Panza” actualizează cele 4 poziții nominale obținute de la procesul părinte sub forma a 4 distanțe nominale. Cele 4 distanțe nominale au fost generate în modulul de generare. Tot în această metodă se actualizează și poziția P_m care este moștenită de la procesul părinte sub forma a două distanțe măsurate. Atunci când se apelează metoda, actualizarea se face pentru toți roboții care formează pânza în cauză.

Metoda `com_nom_act(..., ...)` (linia 14 și 20) apelată prin parametrii (0,0), stabilește prima variantă din cele 25 variante posibile de localizare a roboților în cadrul unei pânze. Prima variantă nominală este combinația a celor două poziții P_m corespunzător celor două suprafețe. Atunci când metoda este apelată prin parametrii (i1,i2), se stabilește varianta corespunzătoare parametrilor.

Metoda `com_nom_es(..., ..., ...)` (linia 12 și 22) apelată prin parametrii (i1,i2,false), transcrie combinația dată prin i1 și i2 din lista „neagră” în lista „albă”. Al treilea parametru (false) indică faptul că transcrierea se face în lista „albă”. Pentru valoarea „true”, transcrierea se face în lista „neagră”.

Metoda `comb_nom_act_es()` (linia 23), transcrie combinația actuală (curentă) în lista „neagră”.

Metoda `actualiz_poz_es` (linia 21 și 26), actualizează pozițiile estimate a roboților din cadrul unei pânze. Pozițiile estimate devin pozițiile nominale curente (P_n), varianta curentă din cele 25 posibile.

Metoda `al_com_nom()` (linia 24), verifică în lista albă dacă mai există o altă variantă. În caz că există atunci se apelează metoda `com_nom_act(..., ...)` cu parametrii găsiți în lista albă și se returnează „True”. În caz că lista albă este goală, atunci se returnează „False”.

Metoda `actualiz_sol()` (linia 35), adună soluțiile găsite. Toate soluțiile vor fi transmise procesului părinte.

Metoda `cond_indepl(..., ...)` (linia 17), reprezintă un punct cheie în cadrul algoritmului BPF. Aici se verifică următoarele 5 distanțe: $robot_1-robot_2$ (w_1-w_2), $robot_1-referință_1$ (w_1-rw_1), w_1-rw_2 , w_2-rw_1 și w_2-rw_2 . Cele 5 distanțe se compară cu distanțele moștenite de la procesul părinte. Diferența trebuie să fie în marja de eroare ϵ . În cazul în care cel puțin o singură distanță nu intră în marja de eroare, metoda returnează valoarea „False”. În rest, metoda returnează „True”. Prima condiție este obligatorie. Celelalte 4 condiții sunt opționale deoarece atunci când se face determinarea pozițiilor nominale P_n , se ține cont de toleranța ϵ . Prin urmare, cele 4 condiții opționale se verifică doar pentru „control”. În cazul în care sistemul robotic este echipat și cu dispozitive UWB, atunci tot aici se vor valida încă 4 distanțe corespunzătoare distanțelor măsurate cu dispozitivele UWB.

Revenim la exemplul nostru cu 10 pânze, unde fiecare pânză este formată din doi roboți. Am văzut până acum că este nevoie de 18 suprafețe S_c . În continuare se pune întrebarea: De câte suprafețe S_c avem în total nevoie dacă avem tot 10 pânze unde fiecare pânză este formată din trei roboți. Răspunsul la această întrebare este tot 18 suprafețe. Pe acest concept se bazează metodologia de față, indiferent de numărul roboților din cadrul unei pânze, se vor lua în considerare doar două suprafețe S_c . De aceea în capitolul 4 a fost definită pânza ca fiind compusă din minim doi roboți. În aplicațiile simple în care avem nevoie doar de un singur robot în cadrul unei pânze, algoritmul BPF nu poate fi aplicat, probabil că se poate face o simplificare, dar aceasta nu face parte din strategia metodologiei propuse. Prin metodologia propusă se pune accent pe aplicații complexe, unde cel puțin doi roboți formează o pânză. În cazul în care avem trei sau mai mulți roboți în cadrul unei pânze, suprafețele S_c se vor construi pentru cei doi roboți care respectă cel mai bine regulile de poziționare prezentate în

subcapitolul 5.7 și de asemenea, trebuie să se stabilească în așa fel încât suprafața acestora să fie cât mai mică.

6.4 Simulatorul Simon-TX

Pentru a putea executa algoritmul BPF, și pentru al valida, autorul tezei a dezvoltat simulatorul Simon-TX. Simulatorul este dezvoltat în Microsoft Visual Studio utilizând limbajul de programare C# și baza de date Microsoft SQL pentru implementarea părții Back-End (BRAIN) și generarea de cod HTML5 prin folosirea tool-ului Construct, dezvoltat de Scirra Ltd, pentru implementarea părții Front-End. Comunicarea dintre Front-End și Back-End a fost realizată folosind protocolul WebSocket. Algoritmul BPF a fost implementat și în simulatorul SiMuLoC, doar pentru un singur proces copil, fiind același cu procesul părinte. Rezultatele experimentale expuse în acest capitol sunt cele obținute folosind simulatorul SiMuLoC deoarece simulatorul Simon-TX este încă în dezvoltare.

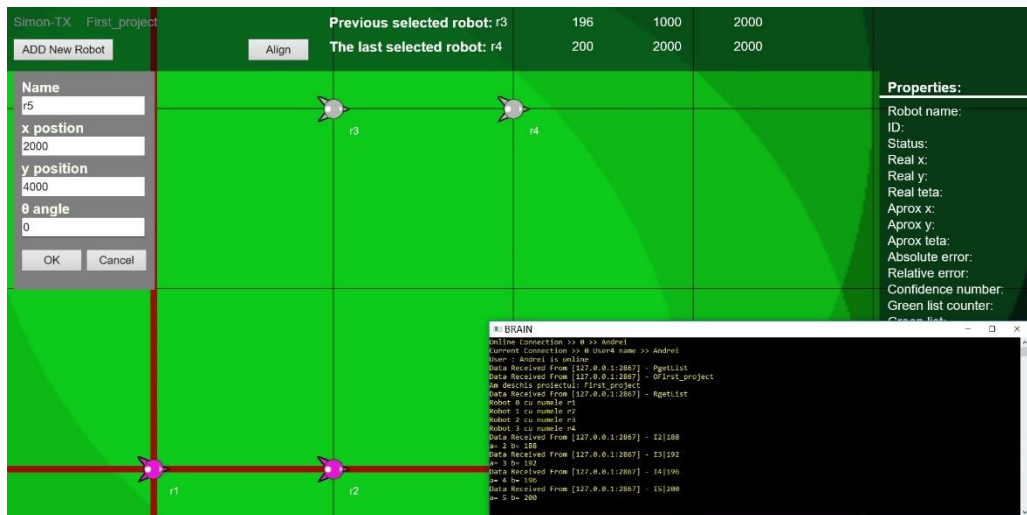


Figura 6-5 Interfața simulatorului Simon-TX

Simulatorul Simon-TX, partea Front-End (Figura 6-5), poate fi utilizat prin intermediul unui browser pe orice calculator cu o conexiune la internet. Pentru al folosi este nevoie de un cont format dintr-un nume de utilizator și o parolă.

Asemănător ca și în cazul simulatorului SiMuLoC, în Simon-TX, fiecare utilizator poate crea proiecte. Proiectele pot fi salvate, cu scopul de a reveni oricând la o configurație deja începută sau se poate continua o generare de poziții (prin ieșirea din modul pauză). Rezultatele se stochează în baza de date Microsoft SQL (Back-End). Rezultatele pot fi observate pe interfața grafică (Front-End) prin simpla selectare a robotului dorit.

Motivul pentru care s-a ales implementarea unui simulator nou este faptul că simulatorul SiMuLoC nu oferă posibilitatea de a lucra pe o suprafață mai mare de 14m x 20m. De asemenea, simulatorul SiMuLoC este limitat la o execuție cu un singur proces, nu a fost dezvoltat pentru a fi executat pe un server.

6.5 Exemplificarea metodologiei pe un studiu de caz

Pentru a putea compara metoda de localizare BPF cu metoda ILS, în continuare se ia în considerare același studiu de caz ca și în subcapitolul 5.6.

Așa cum s-a prezentat în subcapitolul 5.6, toate pânzele sunt create folosind doi roboți și, prin urmare, procedura de localizare va trebui să se bazeze pe două referințe mobile. Pentru o corectă comparație (1:1), în continuare se ia în considerare configurația 2 (CONF2) de poziționare a roboților folosită în subcapitolul 5.7.1 pentru a evidenția regula PRP. De asemenea, trebuie remarcat faptul că experimentele, în simulare, prezentate în acest subcapitol au fost realizate cu un singur proces copil. Motivul este de a evidenția importanța parametrilor de configurație de bază. Astfel au fost efectuate mai multe experimente iar experimentele diferă prin parametrii de configurare a algoritmului BPF.

Conform algoritmului BPF prezentat în subcapitolul 6.3.2, pe scurt, următoarele etape sunt parcurse:

- Pasul 1 – se face localizarea unui robot cu ajutorul a două referințe. Două locații măsurate (soluții) vor fi obținute, ca și intersecție a două cercuri. Observație: Măsurătorile distanțelor sunt efectuate repetitiv, pentru a filtra zgomotul de măsurare aleatoriu de distribuție Gauss.
- Pasul 2 – se selectează una dintre cele două soluții de la pasul 1, pe baza rezultatului obținut din metoda LOAD (informația de orientare).
- Pasul 3 – se creează o suprafață (P_c) pe baza soluției obținute de la pasul 2 și cunoscând toleranța (ϵ) modulului IRULT. Observație: Suprafața estimată trebuie stabilită ca urmare a apariției unui zgomot non-gaussian în măsurarea distanței.
- Pasul 4 – Selectarea unei soluții nominale de pe suprafața estimată. Soluția nominală reprezintă aici o soluție preliminară.
- Pasul 5 – Validarea soluției nominale selectate de la pasul anterior prin compararea distanțelor dintre roboți. Un număr total de 5 distanțe pot fi verificate aici: robot₁-robot₂ (w_1-w_2), robot₁-referință₁ (w_1-rw_1), w_1-rw_2 , w_2-rw_1 și w_2-rw_2 . Toate comparațiile sunt efectuate între distanțe măsurate (ținând cont de toleranțe) și distanțele calculate pe baza soluției nominale. Dacă o singură distanță nu se potrivește, atunci altă valoare nominală va fi selectată. Dacă nu se găsește nici o soluție pentru a potrivi toate cele 5 condiții, atunci poziția pânzei anterioare trebuie recalculată (abordare backtracking). Doar prima distanță trebuie să fie validată obligatoriu, celelalte 4 se validează opțional (se poate renunța la validarea acestora).

6.5.1 Experiment de bază

Primul experiment efectuat în simulator pentru a verifica performanța algoritmului BPF a obținut denumirea BPF-B100K. B100K semnifică un număr maxim de 100.000 de pași înapoi (back). Litera B este scrisă cu majusculă și semnifică faptul că este un experiment de bază (cu configurație simplă) având pentru ceilalți parametrii valoarea 1 (default). Experimentul BPF-B100K este efectuat executând Algoritmul 6-1.

În cazul studiului de caz curent, avem complexitate 25^{11} semnificând aproximativ $2 \cdot 10^{15}$ de variante posibile pentru un set de valori nominale. (Observație: Cea mai lungă cale din graful format de pânze conține 12 noduri-pânze. Pentru prima pânză nu se creează suprafețe S_c . De aici rezultă puterea 11.) 10^5 de pași înapoi acoperă mai mult decât 10^5 de variante, dar nu tot timpul cele aproximativ $2 \cdot 10^{15}$ variante posibile. Câteodată, pentru a acoperii toate cele $2 \cdot 10^{15}$ variante posibile, algoritmul BPF se execută pe o durată de timp foarte lungă. Dar există seturi de valori nominale pentru care durata de execuție este mult mai scurtă. Este foarte important să fie găsit un set mai „potrivit” care să ducă la o soluție într-un timp mult mai scurt. De aici vine ideea de întrerupere a procesului prin setarea parametrului „număr maxim de pași înapoi (back)” și luarea a altor valori nominale. În cazul de față, același proces copil preia alte valori nominale. De aici a venit și ideea de multi-proces care ar ajuta foarte mult în obținerea soluțiilor într-un timp mult mai scurt. În prezentarea experimentelor din acest subcapitol (chiar și pe întreaga lucrare), accentul nu s-a pus pe durata de execuție, ci pe acuratețea de localizare. De aceea prin folosirea unui singur proces copil, rezultatele în acuratețe nu diferă cu cazul în care ar fi fost folosite mult mai multe procese copil. În cazul experimentului BPF-B100K, procesul copil a obținut după fiecare întrerupere un set nou până când s-a ajuns la o soluție.

În Figura 6-6 și Tabelul 6-1 se prezintă rezultatele din simulare. Trebuie remarcat faptul că experimentul a fost repetat de 30.000, de fiecare dată generându-se alte toleranțe pentru valoarea ϵ . Ca și în cazul experimentelor din subcapitolele 5.6 și 5.7, toleranța maximă pentru valoarea ϵ a fost considerată $\pm 2\text{mm}$.

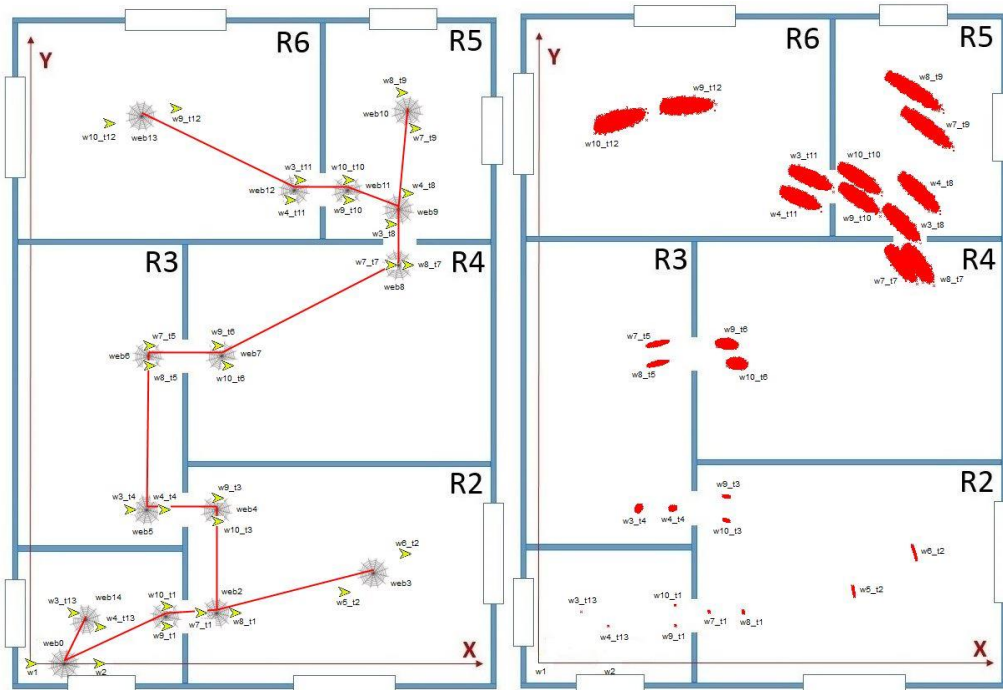


Figura 6-6 Simulare - metoda BPF-B100K (CONF2)

În Figura 6-6, în partea stângă, este prezentată configurația roboților, de asemenea se pot observa conexiunile între pânzele sistemului. Tot în Figura 6-6 dar

În partea dreaptă, sunt reprezentate pozițiile generate a roboților după efectuarea celor 30000 de generări. Fiecare poziție generată a fost marcată cu un punct. Astfel, prin generări multiple prin metoda Monte Carlo, s-au obținut zonele (suprafețele) de toleranță. Se observă că la distanță mai mare de origine, suprafața de generare a punctelor crește tot mai mult. În Tabelul 6-1 pot fi observate erorile absolute fiecărui robot în parte și de asemenea eroarea relativă.

Tabelul 6-1 Metoda BPF-B100K (CONF2)

CONF2 - BPF - 30000 - IRULT 2				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
w1	0	0	0	0
w2	2000	0	0	0
w9_t1	4012	1100	24.1365837525051	0.727578182321751
w10_t1	4003	1700	22.6646955733755	0.683209280039097
w7_t1	5000	1500	32.6163713165606	0.677639526856646
w8_t1	6000	1500	52.813275526548	1.09725152109084
w5_t2	9230	2110	171.17473670256	1.78627334181159
w6_t2	11010	3260	246.619878434644	2.57357202876323
w9_t3	5500	4900	115.518335036382	1.46909459155835
w10_t3	5500	4200	101.581420756899	1.29185307068158
w3_t4	2930	4555	140.663426781145	1.41608816698975
w4_t4	3930	4555	105.053473438294	1.0575953113143
w7_t5	3479	9400	350.397566106677	2.41845254118804
w8_t5	3479	8820	325.172062371396	2.24434549960897
w9_t6	5500	9400	342.905882532049	2.05832001699314
w10_t6	5800	8820	316.201350354852	1.8980239243193
w7_t7	10600	11800	813.537337783484	3.61345991134177
w8_t7	11100	11800	845.374678100185	3.75487069554939
w3_t8	10600	13000	882.551111150216	3.65232621182334
w4_t8	11100	13900	967.785094704189	4.00505627848948
w7_t9	11310	15820	1119.57761459364	4.13422789962462
w8_t9	10910	16890	1183.71226625347	4.37105584506455
w9_t10	9350	13700	858.215516547166	3.33135736411503
w10_t10	9350	14300	898.140913257908	3.48633680900163
w3_t11	7920	14300	836.633370227492	3.05991304010639
w4_t11	7620	13700	783.425539245886	2.86531007344236
w9_t12	4260	16410	934.159778099169	2.88879464972966
w10_t12	2300	15970	926.149336492962	2.86402316909381
w3_t13	1250	1500	5.50417505989185	0.380454247087507
w4_t13	2040	1090	5.70743594883538	0.394503849003124

6.5.2 Experiment cu metoda incrementului

Experimentul BPF-b100n10i10s3 este mai complex și a necesitat modificarea Algoritmului 6-1 (de bază). Ideea acestui experiment este de a căuta treptat soluții, prima dată luând puncte P_n cât mai aproape de P_e și treptat luând pe toată suprafața S_c .

În continuare este prezentat algoritmul modificat:

```

01 for i ← 1 to Numar_de_panze - 1
02   do if solutii_proces = 0
03     then stop (Toate solutiile au fost gasite)
04   if pasi_inapoi = 0 or i = 0
05     then i = 1
06         pasi_inapoi = 100
07         set_nom -= 1
08         if set_nom = 0
09           then set_nom = 10
10             if inc_tol < 1
11               then inc_tol += 0.1
12         for j ← 1 to Numar_de_panze - 1
13           do Panza(j).poz_nom_def = False
14   pozitie_estimata = False
15   if Panza(i).poz_nom_def = False
16     then for i1 ← 0 to 4
17       do for i2 ← 0 to 4
18         do Panza(i).com_nom_es(i1,i2,false)
19         Panza(i).actualiz_poz_nom()
20         Panza(i).com_nom_act(0,0)
21         for i1 ← 0 to 4
22           do for i2 ← 0 to 4
23             do if Panza(i).cond_indepl(i1,i2)
24               then if pozitie_estimata = False
25                 then pozitie_estimata = True
26                   Panza(i).com_nom_act(i1,i2)
27                   Panza(i).actualiz_poz_es()
28                 else Panza(i).com_nom_es(i1,i2,true)
29           else Panza(i).comb_nom_act_es()
30           if Panza(i).al_com_nom()
31             then pozitie_estimata = True
32               Panza(i).actualiz_poz_es()
33             else pozitie_estimata = False
34   if pozitie_estimata = False
35     then Panza(i).poz_nom_def = False
36         i -= 2
37         pasi_inapoi -= 1
38   else
39     Panza(i).poz_nom_def = True
40     if i = Numar_de_panze - 1
41       then Panza(i).actualiz_sol()
42         i = 0
43         solutii_proces -= 1
44         inc_tol = 0.1
45         pasi_inapoi = 100
46         set_nom = 10

```

Algoritm 6-2 Implementarea unui proces copil cu metoda incrementului

Observație: față de Algoritmul 6-1 au mai fost adăugate liniile marcate cu „bold”.

Pentru această implementare, procesul copil moștenește doi parametri opționali de la procesul părinte:

- increment pentru toleranță (inc_tol)
- număr maxim de seturi nominale pentru un increment (set_nom)

b100n10i10s1 are următoarea semnificație:

- b100 – înseamnă maxim 100 de pași înapoi (pasi_inapoi = 100),
- n10 – maxim 10 seturi nominale pentru un increment (set_nom = 10),
- i10 – toleranța se incrementează în 10 pași (inc_tol = 0.1)
- s1 – număr de soluții pentru un proces (solutii_proces = 1).

În cazul Algoritmului 6-2, metoda actualiz_poz_nom() (linia 19) din cadrul clasei „Panza” actualizează cele 4 poziții nominale în cadrul procesului copil, practic nu mai moștenește de la procesul părinte așa cum a fost în cazul algoritmului 6-1. Astfel, cele 4 distanțe nominale se generează conform incrementului (inc_tol). Cu alte cuvinte, procesul copil devine un mic generator de valori nominale valabil doar pentru procesul curent nu și alte procese. Tot în această metodă se actualizează și poziția P_m care este moștenită de la procesul părinte sub forma a două distanțe măsurate. Atunci când se apelează metoda, actualizarea se face pentru toți roboții care formează pânda în cauză.

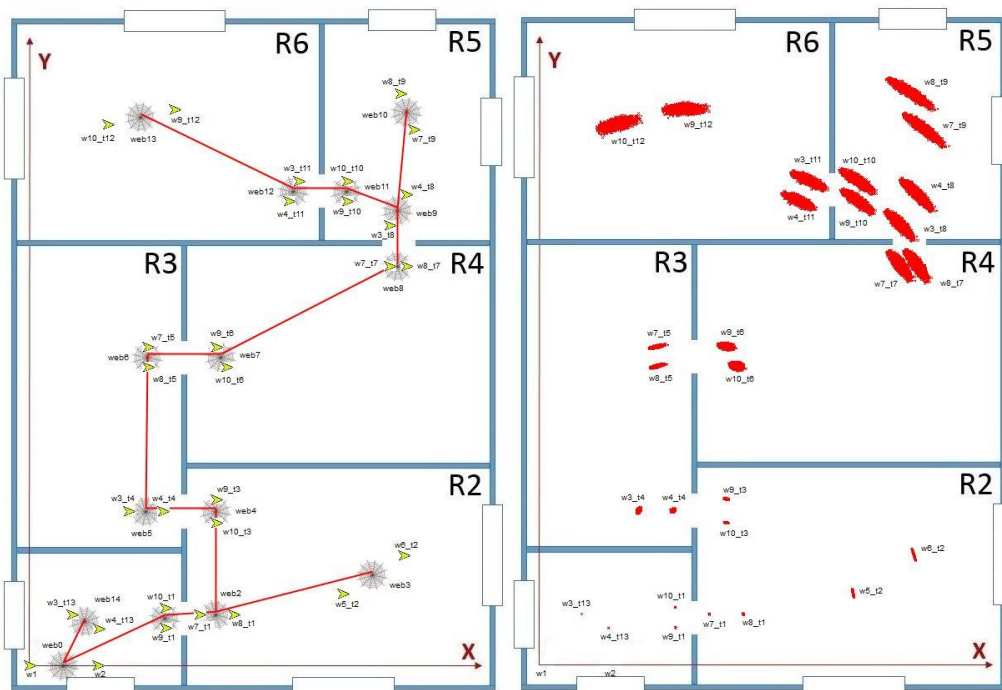


Figura 6-7 Simulare - metoda BPF-b100n10i10s1 (CONF2)

Rezultatele generate în simulator sunt expuse în Figura 6-7 respectiv Tabelul 6-2. După cum se observă față de Algoritmul 6-1, noul algoritmul (Algoritmul 6-2) aduce o îmbunătățire din punct de vedere al acurateții în localizare. De exemplu, dacă roboții w_9 și w_{10} au avut la momentul t_{12} eroarea absolută 934mm respectiv

926mm (Tabelul 6-1), prin aplicarea metodei „incrementului”, eroarea absolută a devenit 895mm respectiv 891mm (Tabelul 6-2).

Rezultatele obținute validează următoarea teorie: Printre toate cele $2 \cdot 10^{15}$ variante posibile a unui set și printre toate seturile de particule P_n , uniform generate peste toate suprafețele S_c , există o infinitate de soluții posibile. Soluțiile care au fost obținute prin generarea particulelor în apropierea punctelor P_m au o acuratețe mai bună față de celelalte soluții existente. Algoritmul 6-2 scoate la suprafață soluțiile mai bune din punct de vedere al acurateței.

Tabelul 6-2 Metoda BPF-b100n10i10s1 (CONF2)

CONF2 - BPF - 30000 - IRULT 2				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
w1	0	0	0	0
w2	2000	0	0	0
w9_t1	4012	1100	16.2175561396717	0.48886537294569
w10_t1	4003	1700	17.2139953776197	0.518902243820799
w7_t1	5000	1500	26.0822824975709	0.541886937680935
w8_t1	6000	1500	40.8516096004625	0.848735291008118
w5_t2	9230	2110	131.905307329299	1.37648194274843
w6_t2	11010	3260	186.058358387015	1.9415895827283
w9_t3	5500	4900	87.3000388912821	1.11023081260324
w10_t3	5500	4200	65.2558452137594	0.829885656168001
w3_t4	2930	4555	118.126723062362	1.18920645232217
w4_t4	3930	4555	86.5282661867895	0.871098171437954
w7_t5	3479	9400	297.715680201904	2.05484088070582
w8_t5	3479	8820	271.320781103792	1.87266264383189
w9_t6	5500	9400	291.307835189694	1.74859860627164
w10_t6	5800	8820	264.899709381167	1.59008171655954
w7_t7	10600	11800	713.077993344798	3.16725320761765
w8_t7	11100	11800	756.522641194598	3.36021975761685
w3_t8	10600	13000	788.173295261764	3.26175555089607
w4_t8	11100	13900	889.964785802423	3.68300676722244
w7_t9	11310	15820	1059.50338136145	3.91239372945213
w8_t9	10910	16890	1125.06586939992	4.15449419991912
w9_t10	9350	13700	753.254113697334	2.9239259723611
w10_t10	9350	14300	804.746126216079	3.12380384894339
w3_t11	7920	14300	730.096274179547	2.67026297228333
w4_t11	7620	13700	660.070398587585	2.41414948518868
w9_t12	4260	16410	895.699104674969	2.76985890638295
w10_t12	2300	15970	891.225274845782	2.7560240400385
w3_t13	1250	1500	4.20084980201812	0.290367063395311
w4_t13	2040	1090	4.69428832686739	0.324474042263674

6.5.3 Experiment cu metoda soluțiilor multiple

Prin executarea următorului experiment: BPF-b100kn1i1s3, se dorește să se realizeze o îmbunătățire de acuratețe în comparație cu experimentului BPF-B100K. Noutatea pe care o aduce BPF-b100kn1i1s3 este faptul că sunt căutate 3 soluții la nivel de proces copil. Soluția finală se realizează prin calcularea unui punct mediu care se află între cele 3 soluții obținute. Executarea experimentului s-a realizat cu Algoritmul 6-2. Rezultatele experimentale sunt expuse în Figura 6-8 și Tabelul 6-3.

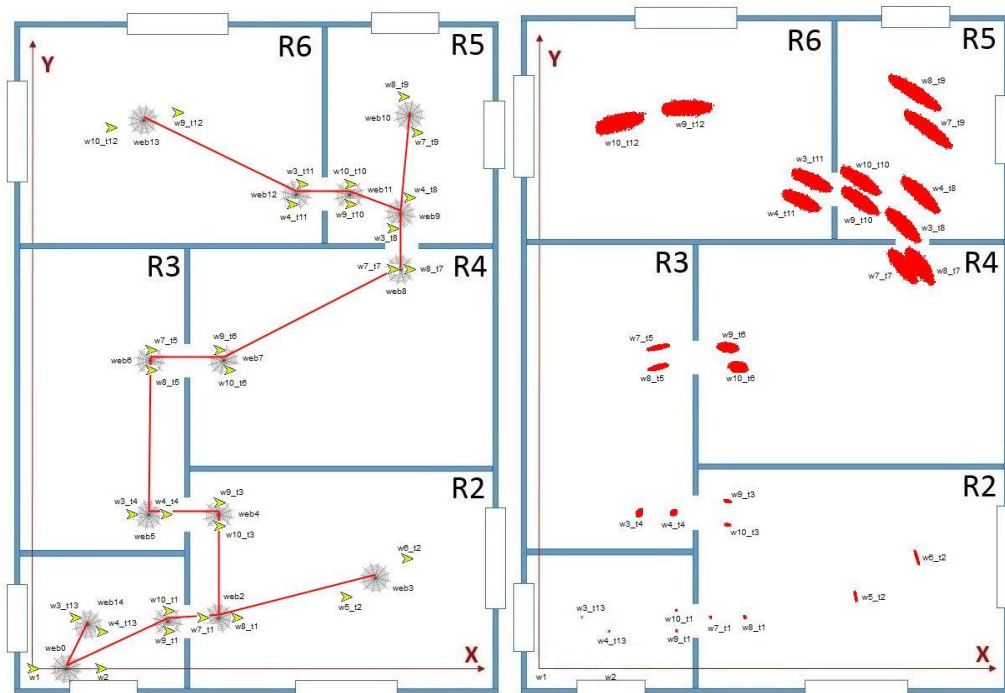


Figura 6-8 Simulare - metoda BPF-b100kn1i1s3 (CONF2)

După cum se observă față de BPF-B100K, noul experiment BPF-b100kn1i1s3 aduce o îmbunătățire din punct de vedere al acurateței de localizare. De exemplu, dacă roboții w_9 și w_{10} au avut la momentul t_{12} eroarea absolută 934mm respectiv 926mm (Tabelul 6-1), prin aplicarea metodei „soluții multiple”, eroarea absolută a devenit 833mm respectiv 842mm (Tabelul 6-3). De asemenea, dacă comparăm metoda „incrementului” cu „soluții multiple”, observăm că metoda „soluții multiple” aduce o îmbunătățire mai bună în acuratețe.

Tabelul 6-3 Metoda BPF-b100kn1i1s3 (CONF2)

CONF2 - BPF - 30000 - IRULT 2				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
w1	0	0	0	0
w2	2000	0	0	0
w9_t1	4012	1100	18.0428176667227	0.543886434655623
w10_t1	4003	1700	18.4386528717206	0.555818573101748
w7_t1	5000	1500	27.1891823330364	0.564883949626336
w8_t1	6000	1500	38.9522654875446	0.809274413109596
w5_t2	9230	2110	149.574848865531	1.56087031463212
w6_t2	11010	3260	211.398141307801	2.2060198344736
w9_t3	5500	4900	90.5936911766	1.15211755514771
w10_t3	5500	4200	73.7745329686454	0.938221342480792
w3_t4	2930	4555	127.959827027762	1.28819836862071
w4_t4	3930	4555	89.9459836284002	0.905505048462993
w7_t5	3479	9400	320.767919034076	2.21394799495663
w8_t5	3479	8820	300.445166195492	2.07367985955683
w9_t6	5500	9400	319.169929487699	1.91584305826372
w10_t6	5800	8820	294.020550868346	1.7648818993448
w7_t7	10600	11800	644.431778667946	2.86234975293911
w8_t7	11100	11800	666.903013853759	2.96215944049875
w3_t8	10600	13000	710.512859586007	2.94036765476559
w4_t8	11100	13900	779.800434001404	3.22710552296815
w7_t9	11310	15820	894.521295277204	3.30316973788873
w8_t9	10910	16890	960.315549359941	3.5461260433009
w9_t10	9350	13700	697.489598035474	2.70746340984084
w10_t10	9350	14300	733.899557561374	2.8487968912975
w3_t11	7920	14300	685.554122692487	2.50735396695333
w4_t11	7620	13700	642.932411134493	2.3514688019228
w9_t12	4260	16410	833.657404093195	2.57800122111047
w10_t12	2300	15970	842.854351846974	2.6064418520259
w3_t13	1250	1500	3.94048260834105	0.272370214901426
w4_t13	2040	1090	4.89595588952276	0.338413511825808

6.5.4 Experiment cu metoda incrementului și metoda soluțiilor multiple

Următorul experiment, BPF-b100n10i10s3, combină metoda „incrementului” cu „soluții multiple”. Executarea experimentului s-a realizat folosind Algoritmul 6-2. Rezultatele sunt expuse în Figura 6-9 respectiv Tabelul 6-4. De exemplu, dacă roboții w_9 și w_{10} au avut la momentul t_{12} eroarea absolută 934mm respectiv 926mm (Tabelul 6-1), 895mm respectiv 891mm (Tabelul 6-2), 833mm respectiv 842mm (Tabelul 6-3), prin combinarea metodelor „incrementului” și „soluții multiple”, eroarea absolută a devenit 755mm respectiv 745mm (Tabelul 6-4).

Ne întoarcem de unde am pornit: să comparăm metoda ILS cu metoda BPF. De exemplu, dacă roboții w_9 și w_{10} au avut la momentul t_{12} eroarea absolută 941mm

respectiv 918mm (pentru metoda ILS CONF2), în cazul metodei BPF, folosind aceeași configurație CONF2, putem observa că rezultatele obținute au o acuratețe mult mai bună.

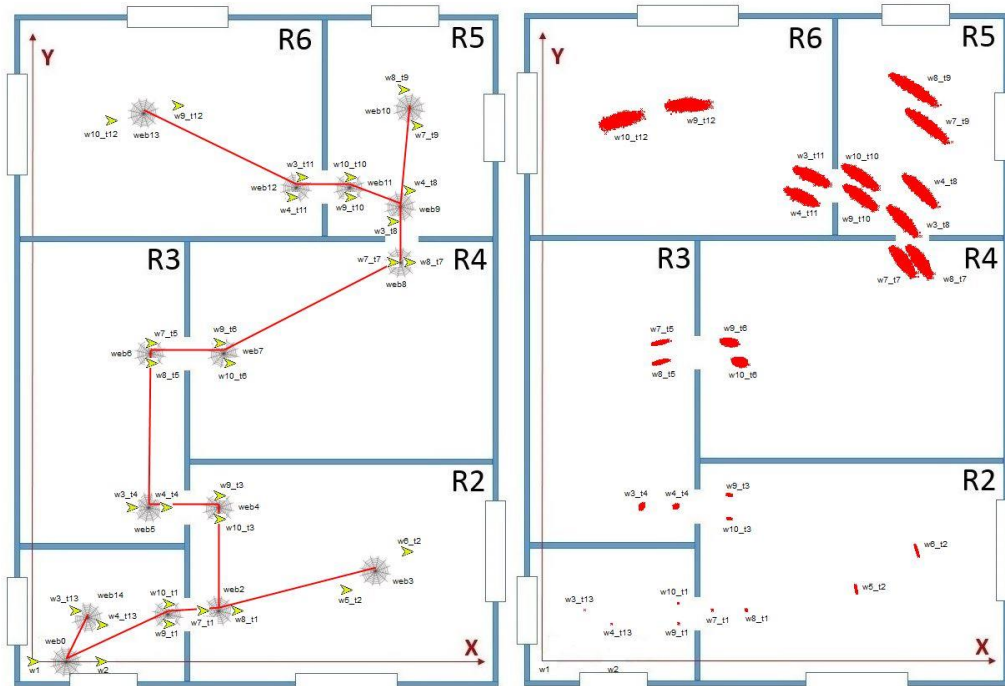


Figura 6-9 Simulare - metoda BPF-b100n10i10s3 (CONF2)

Tabelul 6-4 Metoda BPF-b100n10i10s3 (CONF2)

CONF2 - BPF - 30000 - IRULT 2				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
w1	0	0	0	0
w2	2000	0	0	0
w9_t1	4012	1100	15.3376742490319	0.462342030901351
w10_t1	4003	1700	17.1117812771094	0.515821080793775
w7_t1	5000	1500	24.4072833076221	0.507087062258074
w8_t1	6000	1500	32.3645574195658	0.672407827463746
w5_t2	9230	2110	129.040959972353	1.34659139100019
w6_t2	11010	3260	193.391423377685	2.01811289895438
w9_t3	5500	4900	89.3509637626178	1.13631327505583
w10_t3	5500	4200	68.2844804936387	0.868402067507137
w3_t4	2930	4555	113.351945539915	1.14113776734648
w4_t4	3930	4555	83.0802950322136	0.836386723950516
w7_t5	3479	9400	314.706928822867	2.17211489279989
w8_t5	3479	8820	286.893309274064	1.98014461279811
w9_t6	5500	9400	315.662805111945	1.89479126337646
w10_t6	5800	8820	289.194829458548	1.73591512017824
w7_t7	10600	11800	645.201923463613	2.86577047773664
w8_t7	11100	11800	667.825504760505	2.96625683560932
w3_t8	10600	13000	707.893063515628	2.92952595989196
w4_t8	11100	13900	775.825516050322	3.21065582748971
w7_t9	11310	15820	898.139268592083	3.31652971045651
w8_t9	10910	16890	943.391912186753	3.48363267789906
w9_t10	9350	13700	701.025531164671	2.72118893290801
w10_t10	9350	14300	733.435483520729	2.84699548309293
w3_t11	7920	14300	693.699178034233	2.53714379119357
w4_t11	7620	13700	655.753337397924	2.39836021320974
w9_t12	4260	16410	755.163984488306	2.33526825838863
w10_t12	2300	15970	745.678711431793	2.3059360106305
w3_t13	1250	1500	4.27842730769112	0.295729300459103
w4_t13	2040	1090	4.02536116547422	0.278237108159107

6.5.5 Experiment cu variația toleranței maxime la măsurarea distanței

În continuare, experimentele au fost extinse pentru a compara diferența în acuratețe dintre metoda ILS și metoda BPF luând diferite valori pentru valoarea ε . Au fost folosiți aceeași parametri ca și pentru experimentul BPF-b100n10i10s3. Pentru o comparație corectă, în continuare a fost folosită configurația CONF2 pentru poziționarea roboților. Au fost generate 30000 de particule folosind următoarele toleranțe maxime (ε) pentru măsurarea distanței: ± 1 mm, ± 2 mm, ± 3 mm, ± 4 mm și ± 5 mm. De exemplu, rezultatele metodei BPF utilizând o toleranță de măsurare a distanței de ± 2 mm au fost discutate în cazul experimentului BPF-b100n10i10s3 și sunt prezentate în Figura 6-9 respectiv Tabelul 6-4. De asemenea, rezultatele metodei

ILS utilizând o toleranță de măsurare a distanței de ± 2 mm au fost discutate în subcapitolul 5.7.1 și sunt prezentate în Figura 5-10 respectiv Tabelul 5-2. Toate rezultatele pentru cele 5 cazuri de toleranță maximă (ϵ) au fost comprimate și sunt prezentate în Figura 6-10. Metoda ILS, în acest caz, este de zece ori mai rapidă decât BPF, dar BPF are o acuratețe mai bună. Aceste rezultate au mai fost prezentate și la conferința IPIN (International Conference on Indoor Positioning and Indoor Navigation) în anul 2016 în cadrul căreia autorul tezei a luat parte [121].

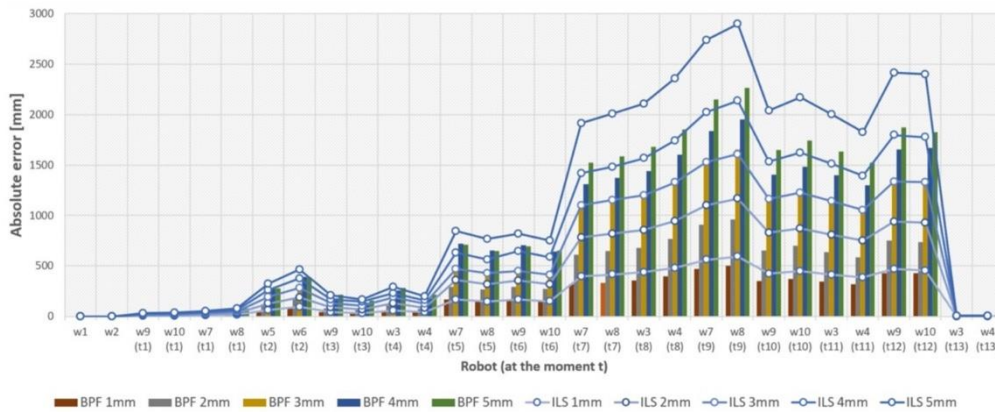


Figura 6-10 Simularea propagării erorilor utilizând metodele ILS și BPF (CONF2) [121]

6.6 Localizare relativă bazată pe măsurarea unghiului și distanței între roboți

Există mai multe sisteme de localizare relativă, în literatură, care utilizează repere mobile. Dacă comparăm metodele prin care este realizată localizarea, putem împărți aceste sisteme de localizare în trei categorii: sisteme bazate doar pe măsurarea unghiului între roboți, sisteme bazate doar pe măsurarea distanței între roboți și sisteme bazate pe măsurarea unghiului și distanței între roboți.

6.6.1 Sisteme bazate pe măsurarea unghiului între roboți

Sistemul de localizare descris în [58] și prezentat în această lucrare la subcapitolul 2.2.6 CAT IV - PhotoBeacon (sistem optic de localizare), constă într-un modul hardware compus dintr-o unitate transmițător și un receptor. Transmițătorul conține un LED de putere, care servește pentru a transmite omnidirecțional un fascicul de lumină. Receptorul conține un set de lentile poziționate în cerc și un senzor CMOS sofisticat pentru a capta lumina transmisă de la un alt robot pe plan orizontal. Folosind acest modul hardware, se poate obține unghiul dintre doi roboți. Trebuie remarcat faptul că sistemul oferă doar unghiul și nu distanța dintre emițător și receptor. Pentru

a determina locația robotului, sunt necesare cel puțin două repere (roboți mobili) de locație cunoscută.

În [7] sunt prezentate o serie de principii de poziționare cooperativă. Dintre acestea, principiul "poziționarea a trei roboți care măsoară doar unghiurile" a fost folosit în efectuarea anumitor simulări. Simulările au fost efectuate luându-se în considerare trei roboți aflați în spațiul 3D și două scenarii pentru mobilitatea lor spre obiectiv. În fiecare situație, a fost utilizat un anumit tipar de mobilitate și a fost calculat numărul de deplasări spre cea mai apropiată poziție față de obiectivul în cauză.

În general, problema acestei categorii este efectul erorilor date de unghi în obținerea distanțelor. Erorile de determinare a poziției sunt direct proporționale cu erorile unghiulare și distanțele dintre roboți. Această problemă s-a observat și în rezultatele experimentale, pentru metoda LOAD, expuse în Tabelul 4-1.

6.6.2 Sisteme bazate pe măsurarea distanței între roboți

Autorii din [87] folosesc roboți Millibots echipați cu un modul hardware care constă dintr-un traductor ultrasonic cu două funcții și anume emițător și receptor. Traductorul este poziționat pentru a transmite respectiv pentru a recepționa pe verticală, deoarece, toate undele ultrasonice care se deplasează pe orizontală sunt reflectate de un con acustic reflector. Astfel s-a obținut un efect de comunicare omnidirecțională. Scopul acestui modul hardware este de a ajuta la obținerea distanței dintre doi roboți.

Într-o altă cercetare, [139], distanța dintre roboți este estimată cu ajutorul indicatorului de nivel a semnalului recepționat (RSSI). În general, această metodă este frecvent utilizată în aplicațiile de rețele de senzori fără fir WSN (Wireless Sensor Network).

Problema acestei categorii este aceea că orientarea dintre roboți nu poate fi determinată cu ușurință doar din calculul distanțelor, sunt necesare mișcări și calcule adiționale.

6.6.3 Sisteme bazate pe măsurarea unghiului și distanței între roboți

În [140], autorii folosesc roboți Moorebots pentru a explora capacitățile unui simplu modul de localizare relativă. Fiecare modul are doisprezece diode de emisie IR (LED-uri) și patru fotodiode, poziționate la 90° una față de cealaltă. Distanța este determinată de RSSI-ul celor patru receptoare. Prin compararea valorilor RSSI, este posibilă determinarea orientării între roboți. Intervalul funcțional pentru calculul distanței este între 30 cm și 310 cm.

În altă lucrare [105], se folosește o echipă de roboți, unde fiecare robot este echipat cu un difuzor și o pereche de microfoane stereo. Fiecare robot poate determina distanțele relative și orientarea față de alți roboți din zona vizibilă, utilizând metoda TOA. Localizarea este realizată prin folosirea filtrului de particule Rao-Blackwellised RBPf (Rao-Blackwellised Particle Filter), astfel incertitudinile în determinarea unghiurilor a echipelor de roboți precum și în determinarea pozițiilor pot fi rezolvate în mod corespunzător. În continuare sunt necesare experimente suplimentare pentru a caracteriza acuratețea metodei propuse.

Această categorie de sisteme este mai apropiată de abordarea propusă în acest subcapitol. Avantajul acestei abordări este că fuziunea multiplă, cum ar fi măsurarea unghiului și măsurarea distanței, duce la creșterea confienței în localizare.

6.6.4 Algoritmul BPF pentru măsurarea unghiului și distanței între roboți

Atunci când un sistem robotic are la dispoziție dispozitive de măsurat și distanța și unghiul între roboți, atunci algoritmul BPF trebuie adaptat așa cum se va prezenta în acest subcapitol. Ceea ce este important de știut înainte, este cum se vor crea suprafețele S_c ținând cont că acestea trebuie să aibă suprafață minimă, așa cum s-a specificat și în subcapitolul 6.3.2. Există două variante pentru a construi suprafețele S_c . Suprafețele S_c pot fi construite pe baza intersecției a două distanțe, pornind de la punctul P_m/P_e , așa cum s-a prezentat în Figura 6-2. Sau, mai pot fi construite pe baza intersecției a unei distanțe și a unui unghi. În lucrarea de față, prima variantă a fost aleasă deoarece modulul IRULT oferă o acuratețe mai bună la calculul distanței față de calculul unghiului unde acuratețea este slabă.

În continuare este prezentat algoritmul modificat:

```

01 for i ← 1 to Numar_de_panze - 1
02   do if solutii_proces = 0
03     then stop (Toate solutiile au fost gasite)
04     if pasi_inapoi = 0 or i = 0
05       then i = 1
06         pasi_inapoi = 100
07         set_nom -= 1
08         if set_nom = 0
09           then set_nom = 10
10             if inc_tol < 1
11               then inc_tol += 0.1
12             for j ← 1 to Numar_de_panze - 1
13               do Panza(j).poz_nom_def = False
14 Panza(i).actualiz_beta_Măsurat()
15 pozitie_estimata = False
16 if Panza(i).poz_nom_def = False
17   then for i1 ← 0 to 4
18     do for i2 ← 0 to 4
19       do Panza(i).com_nom_es(i1,i2,false)
20 Panza(i).actualiz_poz_nom()
21 Panza(i).com_nom_act(0,0)
22   for i1 ← 0 to 4
23     do for i2 ← 0 to 4
24       do if Panza(i).cond_indepl(i1,i2)
25         then if pozitie_estimata = False
26           then pozitie_estimata = True
27             Panza(i).com_nom_act(i1,i2)
28             Panza(i).actualiz_poz_es()
29 Panza(i).actualiz_beta_es()
30           else Panza(i).com_nom_es(i1,i2,true)
31   else Panza(i).comb_nom_act_es()
32     if Panza(i).al_com_nom()
33       then pozitie_estimata = True
34         Panza(i).actualiz_poz_es()
35       else pozitie_estimata = False
36 if pozitie_estimata = False
37   then Panza(i).poz_nom_def = False
38     i -= 2
39     pasi_inapoi -= 1
40   else
41     Panza(i).poz_nom_def = True
42     if i = Numar_de_panze - 1
43       then Panza(i).actualiz_sol()
44         i = 0
45         solutii_proces -= 1
46         inc_tol = 0.1
47         pasi_inapoi = 100
48         set_nom = 10

```

Algoritm 6-3 Implementarea unui proces copil incluzând și calcularea unghiurilor

Observație: față de Algoritmul 6-2 au mai fost adăugate liniile marcate cu „bold”, linia 14 și linia 29. Trebuie menționat faptul că și în metoda `cond_indepl(..., ...)` de la linia 24 au fost adăugate condiții pentru validarea unghiurilor. Pentru a fi înțeleși toți pașii noi incluși în algoritm, autorul tezei prezintă un exemplu simplu. În Figura 5-10 se redă un exemplu de aplicație cu 6 roboți care sunt grupați în 3 pânze. Ideea acestui exercițiu este de a înțelege pas cu pas cum se determină unghiurile β în cadrul algoritmului BPF. Toți pașii sunt descriși în Tabelul 6-5.

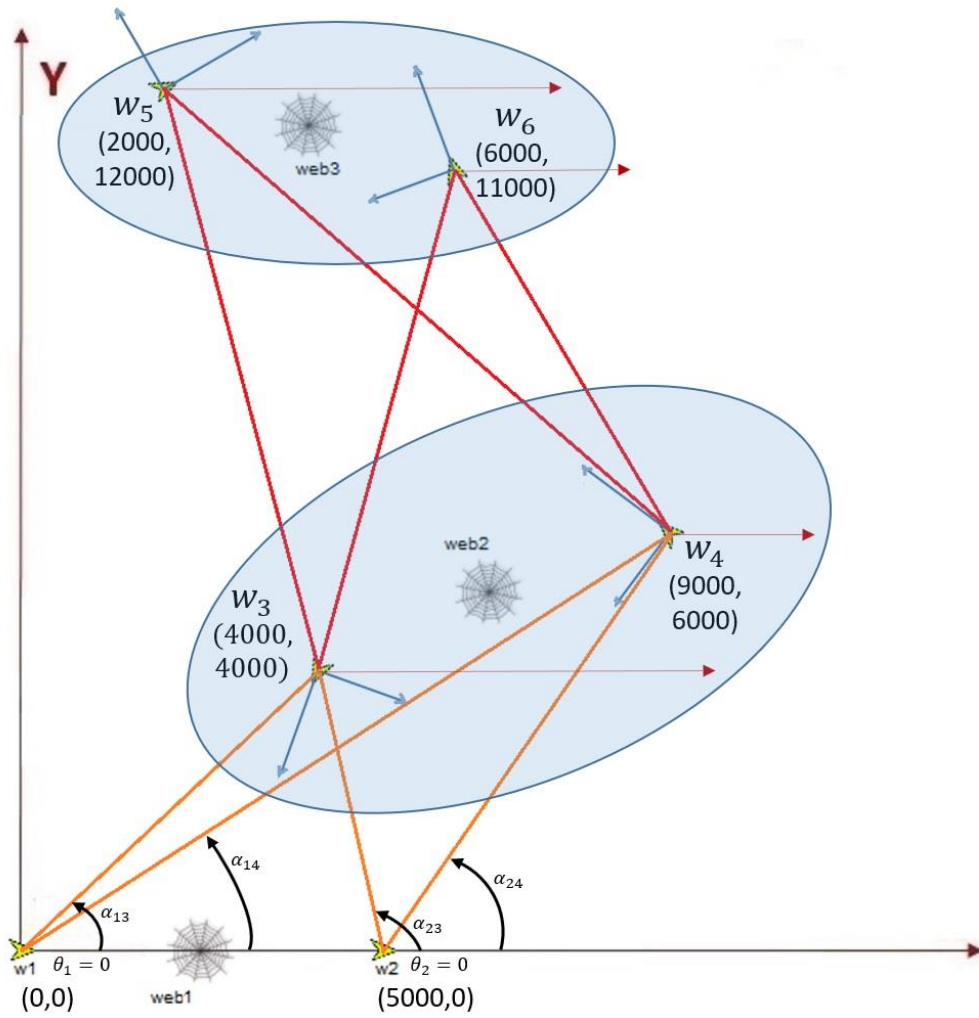


Figura 6-11 Exemplu pentru elaborarea calculului unghiurilor

Tabelul 6-5 Determinarea unghiurilor – pas cu pas

PAS	Variabilă		Descriere
A1	w ₁	Real	Cunoaștem poziția reală a robotului w ₁ Robotul se află în origine (poziția <i>home</i>) și are coordonatele x=0, y=0
A2	w ₁	Nominal	Nu este nevoie, avem direct valoarea estimată = reală
A3	w ₁	Estimat	Se deduce din inițializarea sistemului x=0, y=0
A4	θ ₁	Real	θ ₁ = 0, obs. θ este unghiul dintre axa Ox (<i>home</i>) și axa Ox (w ₁)
A5	θ ₁	Estimat	θ ₁ = 0 În cazul practic, θ _{real} nu este egal cu 0. La inițializarea sistemului, utilizatorul când poziționează roboții, nu o să poată să așeze robotul atât de precis încât unghiul θ _{real} = 0. Axa Ox a robotului w ₁ și axa Ox a robotului w ₂ ar trebui să coincidă. Deci în practică θ _{real} nu este chiar 0. La pasul C3b este folosit θ _{real} .
B1	w ₂	Real	Cunoaștem poziția reală a robotului w ₂ . Robotul are coordonatele x=5000, y=0
B2	w ₂	Nominal	Nu este nevoie, avem direct valoarea estimată
B3	w ₂	Estimat	Se deduce din inițializarea sistemului x=5000, y=0
B4	θ ₂	Real	θ ₂ = 0
B5	θ ₂	Estimat	θ ₂ = 0 (comentariile de la A5 sunt și aici valabile)
C1			Vrem să determinăm poziția robotului w ₃ în raport cu w ₁ și w ₂
C2	w ₃	Real	Acest marcaj (fundal portocaliu) semnifică: în realitate, sistemul colaborativ nu cunoaște această informație. Robotul w ₃ are coordonatele x=4000, y=4000
C3a	β ₁₃	Real	În general, β nu are legătură cu orientarea robotului β ₁₃ – putem calcula astfel: cateta opusă supra alăturată. Pentru aceasta, avem nevoie de coordonatele reale ale roboților w ₃ și w ₁ . De văzut pașii A1 și C2 Formula de calcul este: $\beta_{13} = \arctan \frac{y_1 - y_3}{x_1 - x_3} = \arctan \frac{0 - 4000}{0 - 4000} = \arctan 1 = 45^\circ$
C3b	α ₁₃	Real	α este unghiul care are legătură cu orientarea robotului $\alpha_{13} = \beta_{13} - \theta_1 = 45^\circ - 0^\circ = 45^\circ$ Obs. Dacă θ _{1 real} poate să nu fie 0, atunci și α ₁₃ poate să nu fie 45° Eroarea lui θ _{1 real} este neglijată. De văzut cum este estimat α ₁₃ la pasul C4
C4	α ₁₃	Măsurat	Presupunem că toleranța maximă pentru a determina unghiul cu ajutorul sistemului IRULT este ±2°. Această toleranță este mai mare decât toleranța cu care utilizatorul orientează robotul w ₁ respectiv w ₂ la inițializare. Atunci eroarea de orientare (reală) a robotului w ₁ respectiv w ₂ o neglijăm în simulator.

			Dacă eroarea de determinare a unghiului cu ajutorul sistemului IRULT este sub 1° , atunci ar trebui corectat θ_{1_real} de asemenea și θ_{2_real} , sau poziția lui w_2 . Acest lucru s-ar face prin alinierea roboților w_1 și w_2 . In general axa $Ox(home)$ ar trebui să fie stabilită sau de poziția robotului w_2 sau de unghiul $\alpha_{13_Măsurat}$. In cazul de față, axa $Ox(home)$ este stabilită de poziția robotului w_2 . Calculăm random de $[-2^\circ; 2^\circ]$. Să luăm valoarea 2° . Deci $\alpha_{13} = 47^\circ$
C5	β_{13}	Măsurat	$\alpha_{13_masurat} = \beta_{13_masurat} - \theta_{1_estimat}$ Deci $\beta_{13} = 47^\circ$ Dacă ar fi să aplicăm cele spuse la A5, atunci $\theta_{1_estimat}$ nu ar fi 0 aici.
C6	d_{13}	Real	De văzut pașii A1 si C2 $d_{13}=5656.854249$
C7	d_{13}	Măsurat	Presupunem că toleranța maximă pentru a calcula distanța dintre doi roboți cu sistemul IRULT este ± 2 . Calculăm random de $[-2; 2]$. Să luăm valoarea 1. Deci $d_{13} = 5657.854249$
C8	β_{23}	Real	Ca și la pasul C3, dar aici w_3 se află în cadranul II față de w_2 . Atunci considerăm următoarea formula trigonometrică: $tg(x) = a, x = arctg(a) + n \cdot \pi$ unde $n \in Z$ Atunci o să avem: $\beta_{23} = arctan \frac{y_2 - y_3}{x_2 - x_3} + n\pi = arctan \frac{0 - 4000}{5000 - 4000} + n\pi =$ $= arctan(-4) + n\pi = -75.96378369^\circ + \pi = 104.0362163^\circ$ Se poate lua în considerare formula generală (4-10)
C9	α_{23}	Real	$\alpha_{23} = \beta_{23} - \theta_1 = 104.04^\circ - 0^\circ = 104.04^\circ$
C10	α_{23}	Măsurat	Calculăm random de $[-2^\circ; 2^\circ]$. Să luăm valoarea -2° . 102.04°
C11	β_{23}	Măsurat	$\beta_{23} = 102.04^\circ$
C12	d_{23}	Real	De văzut pașii B1 si C2 $d_{23}=4123.105626$
C13	d_{23}	Măsurat	Calculăm random de $[-2; 2]$. Să luăm valoarea -1 . Deci $d_{23} = 4122.105626$
C14	w_3	Măsurat	Cunoscând distanțele măsurate de la pașii: C7 si C13, poziția o putem determina aplicând metoda de mai jos: Folosim metoda de localizare bilaterală (subcapitolul 4.6) pentru a localiza nodul necunoscut k în raport cu alte două noduri stabilite i și j . Astfel locația nodului k se obține cunoscând distanțele d_{ik} , d_{jk} și coordonatele nodurilor i și j astfel: $y_k = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

			$x_k = s + \frac{y_k(y_i - y_j)}{(x_j - x_i)}$ <p>unde,</p> $\begin{cases} a = \frac{(y_i - y_j)^2}{(x_j - x_i)^2} + 1 \\ b = \frac{2(s - x_j)(y_i - y_j)}{x_j - x_i} - 2y_j \\ c = (s - x_j)^2 + y_j^2 - d_{jk}^2 \end{cases}$ <p>Aplicând metoda de localizare bilaterală, obținem două soluții de locație pentru w_3:</p> <p>$x_{k1}=4001.955991$ $y_{k1}=-3999.457832$ si $x_{k2}=4001.955991$ $y_{k2}=3999.457832$</p> <p>Se observă intuitiv că soluția 2 este corectă. Întrebarea care se pune în continuare este cum va deosebi sistemul robotic cele două soluții.</p> <p>Răspuns: Deosebirea se va face prin comparație. Sistemul robotic mai obține o poziție <u>nominală</u> din A1, C4 și C7, adică din coordonatele lui w_1, unghiul β_{13} și distanța d_{13}. Soluția mai apropiată de poziția nominală este aleasă. Deoarece nu este considerată o problemă majoră, în simulator a fost luată poziția nominală ca fiind poziția reală (doar aici la aceasta comparație). Acest lucru nu influențează cu nimic validarea algoritmului.</p>
C15	w_3	Nominal	<p>Robotul w_3 obține coordonatele măsurate:</p> <p>$x_3=4001.955991$ $y_3=3999.457832$</p> <p>De văzut C2, coordonatele obținute prin măsurare și calcule foarte simple sunt apropiate de coordonatele reale. Trebuie ținut cont că în exemplul nostru, nu am luat cel mai nefavorabil caz, în care coordonatele obținute prin măsurare să fie la cea mai mare distanță de cele reale. Pentru aceasta putem să folosim simulatorul, și prin metoda Monte Carlo, adică simulări repetate cu distribuție uniformă, reușim să generăm aproape toate cazurile posibile.</p> <p>Robotul w_3, știe că aceste coordonate obținute nu sunt reale (șansa este foarte mică să fie reale pentru că în realitate nu avem distribuție uniformă la calculul distanțelor). Ce face robotul? Robotul ia aceste coordonate măsurate ca reper obținute din două distanțe:</p> <p>$x_3=4001.955991$ $y_3=3999.457832$ $d_{13} = 5657.854249$ $d_{23} = 4122.105626$</p>

		<p>Robotul, ia în considerare distanțele și face niște corecții, adaugă ± 2 (Toleranta IRULT) la fiecare distanță. Adică obține patru combinații:</p> <ol style="list-style-type: none"> 1) $d_{13}-2, d_{23}-2$ 2) $d_{13}-2, d_{23}+2$ 3) $d_{13}+2, d_{23}-2$ 4) $d_{13}+2, d_{23}+2$ <p>Dacă aplicăm formulele de la C14 pentru fiecare combinație, obținem 4 poziții. Dacă unim cele 4 puncte (poziții) putem spune că poziția reală a lui w_3 se află sigur în acea suprafață S_C. Știm faptul că putem selecta o poziție de pe suprafața S_C care să ne depărteze mai mult de poziția reală. Atunci, nu era mai bună poziția măsurată? Combinațiile pot fi extinse astfel:</p> <ol style="list-style-type: none"> 1) $d_{13}-2*\text{increment}, d_{23}-2*\text{increment}$ 2) $d_{13}-2*\text{increment}, d_{23}+2*\text{increment}$ 3) $d_{13}+2*\text{increment}, d_{23}-2*\text{increment}$ 4) $d_{13}+2*\text{increment}, d_{23}+2*\text{increment}$ <p>După cum se observă, față de combinațiile propuse anterior, aici apare un increment. Incrementul este o valoare între $(0, 1]$. La început incrementul ia valoarea 0.1 și se incrementează cu 0.1 până ajunge la 1. Astfel avem un efect de mărire treptată a suprafeței S_C. Concret: zicem că incrementul este 0.2 Atunci avem:</p> <ol style="list-style-type: none"> 1) $d_{13}-0.2, d_{23}-0.2$ 2) $d_{13}-0.2, d_{23}+0.2$ 3) $d_{13}+0.2, d_{23}-0.2$ 4) $d_{13}+0.2, d_{23}+0.2$ <p>Mai trebuie specificat faptul că 0.2 este valoarea maximă pentru fiecare iterație a incrementului. Cu alte cuvinte vom avea așa:</p> <ol style="list-style-type: none"> 1) $d_{13}-0.2*\text{rnd1}, d_{23}-0.2*\text{rnd2}$ 2) $d_{13}-0.2*\text{rnd1}, d_{23}+0.2*\text{rnd2}$ 3) $d_{13}+0.2*\text{rnd1}, d_{23}-0.2*\text{rnd2}$ 4) $d_{13}+0.2*\text{rnd1}, d_{23}+0.2*\text{rnd2}$ <p>Unde rnd1 respectiv rnd2 este o valoare aleatoare între $(0, 1]$</p> <p>Robotul aplică formulele de la C14 pentru fiecare combinație de mai sus. Se obțin 4 poziții. Aceste 4 poziții împreună cu poziția măsurată, reprezintă cele 5 poziții nominale ale robotului. Fiecare robot din sistem va avea câte 5 poziții nominale. Există 5×5 combinații posibile de poziții nominale pentru o pânză. Din cele 25 poziții nominale, pentru fiecare pânză, vor fi căutate poziții care să respecte anumite condiții:</p> <ol style="list-style-type: none"> 1) Se verifică dacă distanța d_{34} dintre w_3 și w_4 corespunde 2) Se verifică dacă $d_{13}, d_{14}, d_{23}, d_{24}$ corespund (este o verificare opțională)
--	--	---

			<p>3) Se verifică unghiurile (se va detalia la C17) Ce se întâmplă dacă nu se vor găsi soluții nominale care să îndeplinească condițiile propuse: Să zicem că avem 10 pânze. web1 ... web10 web2 se localizează folosind reperul web1 web10 se localizează folosind ca reper web9 Avem în total 9 pași de localizare ad-hoc. Dacă până la pasul 8 inclusiv, a fost găsită câte o soluție nominală pentru fiecare robot, iar la pasul 9 nu sunt găsite soluții nominale, atunci algoritmul se întoarce la pasul 8 ca să selecteze alte soluții nominale pentru web9. Dacă ne-am întors de 100 de ori la un pas anterior atunci generăm pentru toate pânzele alte valori nominale (se modifică rnd1 și rnd2). Dacă iar ne întoarcem de 100 de ori la un pas anterior atunci generăm alte valori nominale pentru toate pânzele. Dacă și a 10-a oara generăm alte valori nominale și nu găsim soluție pentru întregul sistem, atunci incrementăm incrementul la 0.2 și o luăm de la capăt cu următoarele combinații:</p> <ol style="list-style-type: none"> 1) $d_{13}-0.4, d_{23}-0.4$ 2) $d_{13}-0.4, d_{23}+0.4$ 3) $d_{13}+0.4, d_{23}-0.4$ 4) $d_{13}+0.4, d_{23}+0.4$ <p>Tot procesul se repetă până când incrementul ajunge la 1. Când ajunge la 1 sigur va fi găsită o soluție. Statistic vorbind, soluția va fi găsită înainte ca incrementul să ajungă la 1. Când soluția este găsită, incrementul se resetează la 0.1. Totul se repetă până când vor fi găsite, de exemplu, cele 9 soluții.</p>
C16	w ₃	Estimat	În final, soluția estimată este mijlocul celor 9 soluții de la C15.
C17	w ₃	Estimat	<p>Despre verificarea unghiurilor: Avem $\beta_{13} = 47^\circ$ și $\beta_{23} = 102.04^\circ$. Reprezintă valori măsurate. Luând în considerare faptul că toleranța IRULT este $\pm 2^\circ$ atunci vom avea extremele: $\beta_{13} - 2 = 45^\circ \Rightarrow$ extrema1 $\beta_{13} + 2 = 49^\circ \Rightarrow$ extrema2 $\beta_{23} - 2 = 100.04^\circ \Rightarrow$ extrema3 $\beta_{23} + 2 = 104.04^\circ \Rightarrow$ extrema4 Aceste condiții asigură faptul că valoarea reală este cuprinsă între cele două valori extreme. În realitate, robotul nu poate determina valoarea reală dar poate stabili un interval unde ar putea să se afle. Pe baza acestor extreme se vor determina 4 drepte prin ecuații analitice: Avem distanța $d_{13} = 5657.854249$ astfel putem determina coordonatele lui w₃ folosind cele două extreme ale lui β_{13}</p> $\begin{cases} x_3 = x_1 + d_{13} \cos \beta_{13} \\ y_3 = y_1 + d_{13} \sin \beta_{13} \end{cases}$

		<p>Obținem:</p> $X_{3_extrema1} = 4069.919746$ $Y_{3_extrema1} = 3930.275813$ $X_{3_extrema2} = 3636.798609$ $Y_{3_extrema2} = 4334.167807$ <p>De asemenea avem distanța $d_{23} = 4122.105626$ astfel putem determina coordonatele lui w_3 folosind cele două extreme ale lui β_{23}</p> <p>Obținem:</p> $X_{3_extrema3} = 4352.318426$ $Y_{3_extrema3} = 4070.904491$ $X_{3_extrema4} = 3930.341103$ $Y_{3_extrema4} = 3980.90249$ <p>Construim ecuațiile dreptelor folosind pozițiile estimate ale roboților w_1, w_2 și pozițiile extreme ale lui w_3. Vom obține 4 drepte. Țelul este să obținem 4 semiplane din fiecare dreaptă. Luăm fiecare funcție a dreptei (cele 4 pe rând) $f(x, y) = ax + by + c$ și calculăm $f(w_3)$</p> <p>Atenție, considerăm aici w_3 calculat cu distanța măsurată $d_{13} = 5657.854249$ respectiv $d_{23} = 4122.105626$ și unghiul măsurat $\beta_{13} = 47^\circ$ respectiv $\beta_{23} = 102.04^\circ$.</p> <p>Calculăm $f(x_3, y_3)$, unde $x_3 = 3858.647319$ iar $y_3 = 4137.892648$ pentru primele 2 extreme și $x_3 = 4140.151372$ iar $y_3 = 4031.428423$ pentru următoarele două extreme.</p> <p>Ne uităm la semnul valorii obținute, dacă este pozitiv, atunci avem $f(x, y) > 0$, dacă este negativ, atunci avem $f(x, y) < 0$.</p> <p>Obținem următoarele semiplane (inecuații/condiții):</p> $f_{3_extrema1}(x, y) < 0$ $f_{3_extrema2}(x, y) > 0$ $f_{3_extrema3}(x, y) > 0$ $f_{3_extrema4}(x, y) < 0$ <p>Soluțiile nominale (cele 5) discutate la C15 vor fi trecute prin aceste 4 condiții. Dacă cel puțin o condiție nu se îndeplinește, atunci soluția nominală este marcată ca și nevalidă.</p> <p>Încă nu putem să obținem valoarea estimată pentru w_3 deoarece ar mai trebui și condiția de distanță măsurată între w_3 și w_4 să fie îndeplinită. Pentru aceasta ar trebui să repetăm pașii și pentru w_4 ca și în cazul lui w_3 ca să obținem cele 4 semiplane.</p> <p>Ca să nu mai repetăm pașii și pentru w_4, o să luăm o poziție pentru w_3 și o să o declarăm poziție estimată. Să zicem: $x_3 = 3958.572322$ iar $y_3 = 4037.295638$ este poziția estimată a lui w_3.</p> <p>Această poziție îndeplinește cele 4 condiții de mai sus. Presupunem că îndeplinește și condiția de distanță măsurată între w_3 și w_4.</p>
--	--	---

			<p>Distanțele d_{13}, d_{23}, d_{14}, d_{24} sunt îndeplinite automat prin construirea algoritmului. Singura distanță pe care trebuie să o mai verificăm este d_{34}, putem să spunem distanță nominală deoarece este construită din poziții nominale. Poziția reală a lui w_4 este $x_4=9000$; $y_4=6000$. Distanța reală între w_3 și w_4 este $d_{34_real}=5385.164807$. Distanța măsurată este $d_{34_real} \pm 2$. Să zicem că $d_{34_măsurat}=5384$. Atunci condiția de distanță este următoarea: $5382 \leq d_{34_nominal} \leq 5386$ Observăm că distanța reală este inclusă în interval.</p>
C18	d_{13}	estimat	Dacă $x_3=3958.572322$ iar $y_3=4037.295638$ este poziția estimată, atunci $d_{13}=5654.206478$
C19	d_{23}	estimat	Dacă $x_3=3958.572322$ iar $y_3=4037.295638$ este poziția estimată, atunci $d_{23}=4169.451724$
C20	β_{13}	estimat	Se obține din formula (4-10) se iau în considerare pozițiile estimate: w_1 și w_3 obținem: $\beta_{13}=45.5641$
C21	β_{23}	estimat	Se obține din formula (4-10) se iau în considerare pozițiile estimate: w_2 și w_3 obținem: $\beta_{23}=104.4642$
C22	β_{31}	estimat	Se obține din formula (4-10) $\beta_{31} = \arctan \frac{y_3 - y_1}{x_3 - x_1} + n\pi$ robotul w_1 se află în cadranul III față de robotul w_3 . $\beta_{31}=225.5641$
C23	β_{32}	estimat	Se obține din formula (4-10) $\beta_{32} = \arctan \frac{y_3 - y_2}{x_3 - x_2} + n\pi$ robotul w_2 se afla în cadranul IV față de robotul w_3 . $\beta_{32}=284.4642$
C24	α_{31}	real	În exemplul dat α_{31_real} este 333° .
C25	α_{31}	măsurat	<p>α_{31} – reprezintă o eroare nouă de măsurare în sistem care are un efect asupra localizării și asupra orientării robotului. Vom vedea în continuare de ce este așa.</p> <p>Toleranța maximă pentru a determina unghiul cu ajutorul sistemului IRULT rămâne la fel definită $\pm 2^\circ$. Calculăm random de $[-2^\circ; 2^\circ]$. Să luăm valoarea 2°. Deci avem $\alpha_{31} = 335^\circ$. Zicem că robotul w_3 măsoară unghiul față de robotul w_1 și obține valoarea $\alpha_{31} = 335^\circ$.</p>
C26	θ_3	estimat	α este un unghi care nu trebuie estimat. Valoarea măsurată rămâne valabilă deoarece acest unghi este dat de modulul IRULT care este echipat cu un motor pas cu pas. Unghiul α depinde doar de rotația axului care reprezintă o măsurătoare foarte precisă. Unghiul θ ar trebui estimat. De fapt, tot timpul x, y și θ sunt necunoscute în sistem. Unghiurile α și β sunt deduse din x, y și θ .

		<p> $\alpha_{13_măsurat} = 47^\circ$ $\beta_{13_estimat} = 45.5641^\circ$ $\theta_{1_estimat} = 0^\circ$ </p> <p> Noi știm relația $\alpha_{31_real} = \beta_{31_real} - \theta_{3_real}$ Și mai știm $\alpha_{35_real} = \beta_{35_real} - \theta_{3_real}$ $\Rightarrow \beta_{31_real} - \alpha_{31_real} = \beta_{35_real} - \alpha_{35_real}$ de unde rezultă $\Rightarrow \beta_{35_real} = \beta_{31_real} - \alpha_{31_real} + \alpha_{35_real}$ </p> <p> Putem să definim următorul model: $\Rightarrow \beta_{35_măsurat} = \beta_{31_estimat} - \alpha_{31_măsurat} + \alpha_{35_măsurat}$ </p> <p> Asemănător putem să obținem: $\beta_{35_măsurat} = \beta_{32_estimat} - \alpha_{32_măsurat} + \alpha_{35_măsurat}$ </p> <p> Astfel o să avem două valori pentru $\beta_{35_măsurat}$. Putem să facem media dintre cele două și parcurgem pașii de la C17. Astfel o să reușim să estimăm poziția lui w_5. Mai departe putem să aplicăm formula din C20 ca să obținem $\beta_{35_estimat}$. Dacă știm $\beta_{35_estimat}$ putem să determinăm $\theta_{3_estimat}$ astfel: $\theta_{3_estimat} = \beta_{35_estimat} - \alpha_{35_măsurat}$ </p> <p> În simulări nu avem nevoie să calculăm $\theta_{3_estimat}$. Pentru a vedea cum se propagă erorile rămânem la modelul de sus. </p> <p> Observăm în model că toleranța unghiului se propagă de două ori. Dacă $\beta_{31_estimat}$ este estimat foarte bine, cât mai aproape de valoarea reală și dacă $\alpha_{32_măsurat}$ respectiv $\alpha_{35_măsurat}$ sunt măsurate fără eroare, atunci am putea să obținem pentru $\beta_{35_măsurat}$ chiar valoarea reală. Algoritmul tinde să facă acest lucru prin apelări tip backtracking așa cum se observă în C15. </p> <p> Pentru că nu ne interesează orientarea roboților (să calculăm), în simulator o să avem orientarea reală a roboților $\theta_{real} = 0$. Așa cum se observă mai sus, în model, propagarea erorilor din măsurarea unghiurilor este inclusă. </p>
--	--	---

Dacă reluăm problema, pe scurt, vom avea următorii pași:

Procesul tată, prin modulul de achiziție, obține valorile măsurate pentru următoarele unghiuri: α_{13} , α_{23} , α_{14} , α_{24} , α_{31} , α_{32} , α_{41} , α_{42} , α_{35} , α_{36} , α_{45} , α_{46} , α_{53} , α_{63} , α_{54} , α_{64} .

Procesul copil, atunci când se află la pânza web2, prin apelarea metodei `actualiz_beta_Măsurat()`, actualizează următoarele unghiuri:

- $\beta_{13_măsurat} = \beta_{1x_estimat} - \alpha_{1x_măsurat} + \alpha_{13_măsurat}$
- $\beta_{13_măsurat} = \beta_{1y_estimat} - \alpha_{1y_măsurat} + \alpha_{13_măsurat}$
- $\beta_{14_măsurat} = \beta_{1x_estimat} - \alpha_{1x_măsurat} + \alpha_{14_măsurat}$
- $\beta_{14_măsurat} = \beta_{1y_estimat} - \alpha_{1y_măsurat} + \alpha_{14_măsurat}$
- $\beta_{23_măsurat} = \beta_{2x_estimat} - \alpha_{2x_măsurat} + \alpha_{23_măsurat}$
- $\beta_{23_măsurat} = \beta_{2y_estimat} - \alpha_{2y_măsurat} + \alpha_{23_măsurat}$
- $\beta_{24_măsurat} = \beta_{2x_estimat} - \alpha_{2x_măsurat} + \alpha_{24_măsurat}$
- $\beta_{24_măsurat} = \beta_{2y_estimat} - \alpha_{2y_măsurat} + \alpha_{24_măsurat}$

Unde totul ce conține x și y este egal cu 0.

Procesul copil, atunci când se află la pânza web2, prin apelarea metodei `actualiz_beta_es()`, obține următoarele unghiuri:

- $\beta_{31_estimat}$
- $\beta_{32_estimat}$
- $\beta_{41_estimat}$
- $\beta_{42_estimat}$

Procesul copil, atunci când se află la pânza web3, prin apelarea metodei `actualiz_beta_Măsurat()`, actualizează următoarele unghiuri:

- $\beta_{35_măsurat} = \beta_{31_estimat} - \alpha_{31_măsurat} + \alpha_{35_măsurat}$
- $\beta_{35_măsurat} = \beta_{32_estimat} - \alpha_{32_măsurat} + \alpha_{35_măsurat}$
- $\beta_{36_măsurat} = \beta_{31_estimat} - \alpha_{31_măsurat} + \alpha_{36_măsurat}$
- $\beta_{36_măsurat} = \beta_{32_estimat} - \alpha_{32_măsurat} + \alpha_{36_măsurat}$
- $\beta_{45_măsurat} = \beta_{41_estimat} - \alpha_{41_măsurat} + \alpha_{45_măsurat}$
- $\beta_{45_măsurat} = \beta_{42_estimat} - \alpha_{42_măsurat} + \alpha_{45_măsurat}$
- $\beta_{46_măsurat} = \beta_{41_estimat} - \alpha_{41_măsurat} + \alpha_{46_măsurat}$
- $\beta_{46_măsurat} = \beta_{42_estimat} - \alpha_{42_măsurat} + \alpha_{46_măsurat}$

Procesul copil, atunci când se află la pânza web3, prin apelarea metodei `actualiz_beta_es()`, obține următoarele unghiuri:

- $\beta_{53_estimat}$
- $\beta_{54_estimat}$
- $\beta_{63_estimat}$
- $\beta_{64_estimat}$

6.6.5 Experiment cu variația toleranței maxime la măsurarea unghiului

În continuare vor fi redate două experimente reprezentative din setul de experimente efectuate cu variația toleranței maxime la măsurarea unghiului dintre două module IRULT. Experimentele expuse în acest subcapitol au următoarele caracteristici:

- au fost efectuate în simulator;
- păstrează configurația CONF2 pentru poziția roboților în cadrul pânelor;
- parametrii folosiți sunt aceiași ca și în cazul experimentului BPF-B100K;
- este folosit Algoritmul 6-3;
- numărul de generări multiple prin metoda Monte Carlo a fost: 30000;
- toleranța calculului distanțelor (ϵ) folosind modulul IRULT a fost considerată $\pm 2\text{mm}$;
- Pentru primul experiment, toleranța maximă la măsurarea unghiului este $\pm 1^\circ$ iar pentru cel de-al doilea $\pm 0^\circ 10'$.

Rezultatele primului experiment sunt redate în Figura 6-12 și în Tabelul 6-6. Dacă comparăm cu rezultatele experimentului BPF-B100K, observăm că, de exemplu, roboții w_9 și w_{10} au avut la momentul t_{12} eroarea absolută 934mm respectiv 926mm (pentru experimentul BPF-B100K), iar în actualul experiment, eroarea absolută devine 655mm respectiv 647mm. Analizând rezultatele în întregime, putem spune că Algoritmul 6-3 aduce o îmbunătățire majoră în ceea ce privește localizarea relativă.

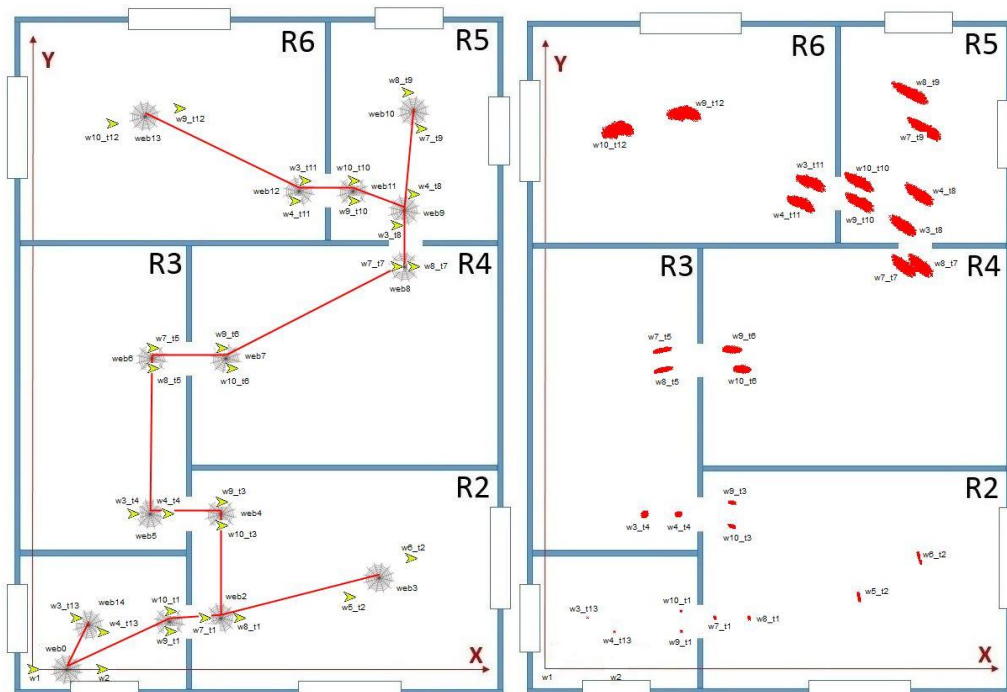


Figura 6-12 Simulare - metoda BPF-B100K-1grad (CONF2)

Tabelul 6-6 Metoda BPF-1grad (CONF2)

CONF2 - BPF - 30000 - IRULT 2				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
w1	0	0	0	0
w2	2000	0	0	0
w9_t1	4012	1100	19.8083828087033	0.597107996163456
w10_t1	4003	1700	22.2001997763926	0.669207422480036
w7_t1	5000	1500	33.0991269800603	0.687669285109834
w8_t1	6000	1500	47.7390466211625	0.991829061883616
w5_t2	9230	2110	140.481801923965	1.46598091879923
w6_t2	11010	3260	200.608727129418	2.09342820271839
w9_t3	5500	4900	93.0734733652977	1.18365397402452
w10_t3	5500	4200	82.6663501090566	1.05130226998888
w3_t4	2930	4555	110.998395722447	1.11744408859001
w4_t4	3930	4555	85.4127061593002	0.859867595085281
w7_t5	3479	9400	271.864002998292	1.87641197458718
w8_t5	3479	8820	256.570388373857	1.77085507371203
w9_t6	5500	9400	270.827395345122	1.62566312619872
w10_t6	5800	8820	250.897070985725	1.5060297583743
w7_t7	10600	11800	541.896995990135	2.40692464886968
w8_t7	11100	11800	563.517435350157	2.50295538681453
w3_t8	10600	13000	590.098279191912	2.4420471351915
w4_t8	11100	13900	650.238997943881	2.69093189746826
w7_t9	11310	15820	768.097027986783	2.83632695163727
w8_t9	10910	16890	818.25033060596	3.0215264235129
w9_t10	9350	13700	578.279767416149	2.24472352754839
w10_t10	9350	14300	607.769268407133	2.35919368614636
w3_t11	7920	14300	572.244827730008	2.09293517664537
w4_t11	7620	13700	539.704117715547	1.97392038898456
w9_t12	4260	16410	655.982058177819	2.0285581807415
w10_t12	2300	15970	647.539696744158	2.00245103171581
w3_t13	1250	1500	4.11301652290605	0.284295936712348
w4_t13	2040	1090	5.33616148078788	0.368840975517665

În cazul celui de-al doilea experiment, toleranța maximă la măsurarea unghiului este semnificativ mai mică. Rezultatele sunt redată în Figura 6-13 și în Tabelul 6-7. Prin comparație cu experimentul precedent, se observă, de exemplu, că dacă roboții w_9 și w_{10} au avut la momentul t_{12} eroarea absolută 655mm respectiv 674mm acum a eroarea absolută maximă 247mm respectiv 244mm. Analizând aceste rezultate, se poate spune pe scurt că, introducerea de măsurare a unghiului pe lângă măsurarea distanței, duce la creșterea confienței în localizare.

S-au încercat și alte experimente cu unghiuri mai mici de $0^\circ 10'$ folosind același algoritm (Algoritm 6-3), dar timpul de generare a tins spre infinit. Era de așteptat pentru că, atunci când unghiul este foarte mic trebuie definite altfel suprafețele S_c . Și anume, cea de-a doua variantă discutată în subcapitolul 6.6.4: suprafețele S_c mai pot fi construite pe baza intersecției a unei distanțe și a unui unghi. Conform celor discutate în subcapitolul 6.3.2: suprafețele S_c trebuie să se stabilească în așa fel încât suprafața acestora să fie cât mai mică. Dacă unghiul are o toleranță

foarte mică, atunci suprafața construită dintr-un unghi și o distanță este mai mică decât suprafața construită din două distanțe. Unghiurile, cu toleranță foarte mică, pot fi măsurate de dispozitive laser precum cele folosite de Kurazume în lucrarea [7].

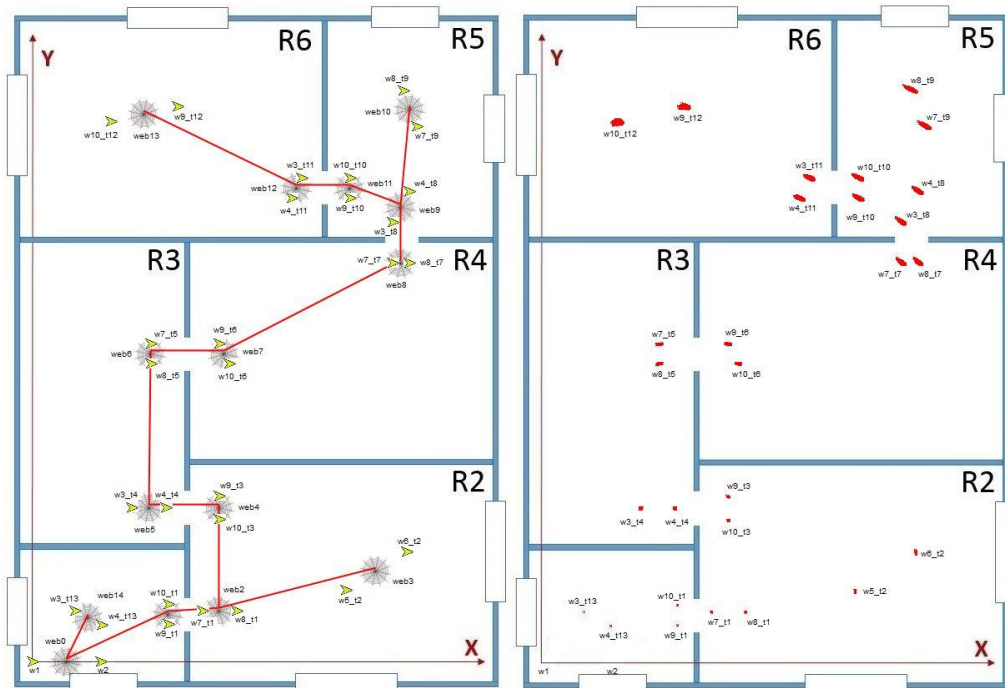


Figura 6-13 Simulare - metoda BPF-B100K-10minute (CONF2)

Tabelul 6-7 Metoda BPF-10minute (CONF2)

CONF2 - BPF - 30000 - IRULT 2				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
w1	0	0	0	0
w2	2000	0	0	0
w9_t1	4012	1100	10.6571381337924	0.321250980322835
w10_t1	4003	1700	11.1087252932571	0.334863717237564
w7_t1	5000	1500	15.2065480051218	0.315932078872354
w8_t1	6000	1500	22.5141097566741	0.467754384294889
w5_t2	9230	2110	49.9511973981618	0.521259702352855
w6_t2	11010	3260	71.7485224867036	0.748723062183341
w9_t3	5500	4900	40.6521822622824	0.516990667133094
w10_t3	5500	4200	34.9937602600389	0.445030167030382
w3_t4	2930	4555	44.0247829134563	0.443206706708746
w4_t4	3930	4555	37.0792641946347	0.373284715638302
w7_t5	3479	9400	103.048789379714	0.711245255812783
w8_t5	3479	8820	94.7481339320387	0.653953929608638
w9_t6	5500	9400	101.409166666535	0.608716642931356
w10_t6	5800	8820	94.5585131335828	0.567595046555586
w7_t7	10600	11800	205.499311274894	0.912758995331817
w8_t7	11100	11800	211.752939968791	0.940535515887565
w3_t8	10600	13000	224.378207388053	0.928560848036635
w4_t8	11100	13900	247.540228563683	1.02441394480324
w7_t9	11310	15820	295.860941906481	1.0925160922255
w8_t9	10910	16890	313.702724823963	1.15839986460092
w9_t10	9350	13700	219.329839279866	0.851378309022907
w10_t10	9350	14300	232.971098762125	0.904329938719445
w3_t11	7920	14300	217.983050788187	0.797253854992155
w4_t11	7620	13700	204.17975335696	0.74676950747588
w9_t12	4260	16410	247.239438701264	0.76456296285383
w10_t12	2300	15970	244.072671939145	0.754770056871553
w3_t13	1250	1500	5.00888825230579	0.346219513013437
w4_t13	2040	1090	4.91545297538166	0.339761170474041

6.7 Localizarea relativă prin construirea bazelor

În subcapitolul anterior am observat că fuziunea multiplă, cum ar fi măsurarea unghiului și măsurarea distanței, duce la creșterea confienței în localizare. Dar acest lucru nu elimină problema de propagare a erorilor în localizarea relativă. Fuziunea ajută la creșterea numărului maxim de pânze pentru o confiență de localizare bună sau altfel spus, ajută la extinderea lungimii maxime a traseului pe care o echipă de roboți poate să-l parcurgă în explorarea mediilor necunoscute. Problema care se pune în continuare este: Ce se poate face atunci când traseul de parcurs este mult mai lung, iar eroare de propagare a crescut semnificativ încât roboții nu mai pot continua explorarea. Metodologia descrisă în Capitolul 4 oferă o soluție: crearea de *baze*.

Rolul *bazelor* este de a reseta propagarea erorilor în localizare. Prima pânză din sistem care devine *bază* este chiar pânză cu axa *home*. Poziționarea *bazelor* se realizează printr-o abordare topologică iar poziționarea *pânzelor* se realizează printr-o abordare metrică.

În continuare autorul tezei propune un sistem prin care se arată că și bazele pot fi foarte utile în fuziunea multiplă, de data aceasta nu prin măsurarea unghiului ci prin măsurarea distanțelor adiționale față de baze. Acest lucru se poate realiza dacă fiecare robot este echipat, de exemplu, cu un modul UWB. Să zicem că pânză din *home* este formată din doi roboți. Atunci oricare robot care explorează un mediu necunoscut poate să calculeze distanțe față de pânză precedentă și în special în acest caz mai poate să calculeze două distanțe față de pânză din *home* cu rolul de *bază*.

Putem avea, de exemplu, următorul traseu de pânze: $A_0, A_1, A_2, \dots, A_i, B_0, B_1, B_2, \dots, B_j, C_0, C_1, C_2, \dots, C_k$. Unde, A_0, B_0 și C_0 sunt baze. În acest exemplu, putem spune că pânză C_9 poate să calculeze distanțe luând ca și reper pânză precedentă C_8 și de asemenea poate să calculeze distanțe luând ca și reper baza de care aparține, în cazul acesta C_0 . Pânză C_9 nu poate să calculeze distanțe față de A_0 sau B_0 din două motive:

- fizic nu este posibil pentru că semnalele UWB sunt atenuate de obstacolele din mediu;
- în cazul că nu există obstacole, nu este posibil prin definiție: poziționarea *bazelor* se realizează printr-o abordare topologică și nu metrică.

În lucrarea [141] sunt prezentate măsurători NLOS prin utilizarea unui dispozitiv radar „ultra-wideband synthetic” modificat. Măsurătorile au fost realizate pentru mai multe tipuri de materiale de construcție. În Tabelul 6-8 sunt prezentate rezultatele experimentale de atenuare la frecvența de 1,5 GHz.

Tabelul 6-8 Atenuarea prin materialele de construcție la 1,5 GHz [141]

Material	Grosime (mm)	Atenuare la 1,5 GHz (dB)
cărămidă	89	4,5
	178	6,5
	267	8,5
cărămidă cu perete din beton	90 respectiv 102	17
	90 respectiv 203	31
cărămidă cu perete din piatră	90 respectiv 194	10,5
beton (diverse categorii)	102	14 – 16,5
	203	26 – 32,5
	305	34 – 44
piatră	203	11
	406	18
	610	29
gips-carton	6	0,35
	13	0,5
	16	0,65
sticlă	6	1,2
	13	2,8
	19	3,6
lemn (uscat)	38	3,4
	76	4,4
	114	5,8
	152	7,4
lemn (umed)	38	2
	76	5,5
	114	7
	152	9
placaj (uscat)	6	0,75
	13	0,8
	19	1,1
	32	1,8
placaj (umed)	6	2,5
	13	2,5
	19	2,5
	32	3,2
beton armat	203 (0% armat)	32
	203 (1% armat) armatura de 140mm	29
	203 (2 % armat) armatura de 70mm	32,5
beton cu bare din oțel	armatura de 140mm	1,5
	armatura de 70mm	11

După cum se observă în Tabelul 6-8, atenuarea diferă în funcție de materialul din care este construit obstacolul. De aceea în practică, construirea de baze noi se realizează în funcție de două situații:

- Atunci când semnalul de la bază este atenuat complet;
- Atunci când confiența în localizare a scăzut foarte mult. Se ține cont și de faptul că roboții la nevoie ar putea să se întoarcă la baza precedentă.

În continuare se prezintă trei experimente care au ca scop să arate fuziunea multiplă în avantajul folosirii bazelor. Experimentele au următoarele caracteristici:

- au fost efectuate în simulator;
- au fost folosiți un număr total de 6 roboți și 28 de pânze;
- a fost folosită o configurație nouă (CONF7) pentru poziția roboților în cadrul pânzelor cu un traseu mai lung de explorare a mediului;
- parametrii folosiți sunt aceiași ca și în cazul experimentului BPF-B100K;
- este folosit Algoritmul 6-1 cu observația că în metoda $\text{cond_indepl}(\dots, \dots)$ au fost adăugate validările corespunzătoare măsurătorilor distanței față de bază;
- numărul de generări multiple prin metoda Monte Carlo a fost: 30000;
- toleranța calculului distanțelor (ϵ) folosind modulul IRULT a fost considerată $\pm 2\text{mm}$;
- toleranța calculului distanțelor folosind modulul UWB a fost considerată $\pm 150\text{mm}$;

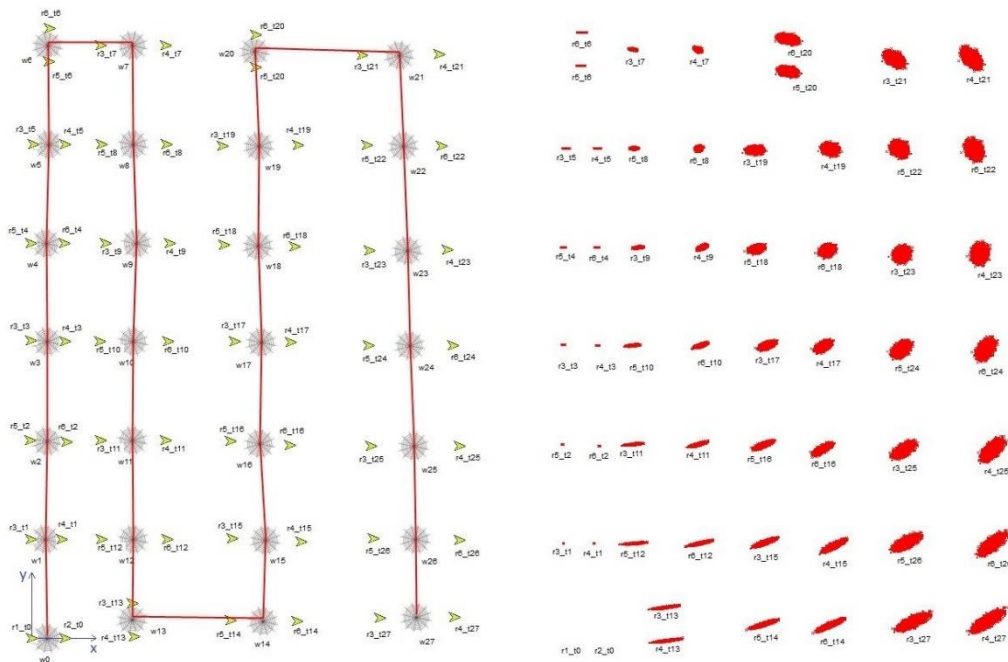


Figura 6-14 Simulare – fără UWB

Trebuie menționat faptul că în experimente a fost folosită doar o singură bază (w_0), considerând că pentru următoarele baze de fiecare dată propagarea erorilor se resetează și acolo de unde începe o bază nouă se reia totul de la început ca și când ar porni roboții din nou din poziția *home*.

În cazul primului experiment expus în Figura 6-14, nu avem module UWB, prin urmare, localizarea se face clasic asemănător cu experimentul BPF-B100K.

Rezultatele primului experiment sunt expuse în Tabelul 6-9 și de asemenea pot fi observate și în Figura 6-14. Se poate observa cum eroarea de localizare se propagă din pânză în pânză. Ultima pânză (w_{27}) este construită de roboții r_3 și r_4 la momentul t_{27} . Eroare absolută maximă de localizare la acest moment pentru r_3 și r_4 este 816mm respectiv 848mm.

Tabelul 6-9 Metoda BPF (fără UWB)

CONF7 - BPF - 30000 - IRULT 2				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
r1_t0	0	0	0	0
r2_t0	1000	0	0	0
r3_t1	-25	3003	16.0604317796361	0.534709722616746
r4_t1	907	3003	15.3510121519001	0.511090583507016
r5_t2	-50	6006	35.4720755670179	0.592909392991126
r6_t2	1055	5957	35.3540629938742	0.59093683395731
r3_t3	-25	9034	67.9031066067198	0.752577610676101
r4_t3	1006	9009	66.4108770976242	0.736039066643943
r5_t4	-25	11988	100.560227771853	0.83875227324328
r6_t4	981	11988	100.236127798042	0.836049021711188
r3_t5	49	14991	142.322364030907	0.949279492865654
r4_t5	1006	14991	142.625966707831	0.951304499948307
r5_t6	515	17502	178.430225398312	0.990801662337606
r6_t6	539	18512	198.291016972339	1.10108625825155
r3_t7	2085	17995	189.904816601146	0.92419194747493
r4_t7	4048	17995	202.659399789556	0.986263480399155
r5_t8	2110	14991	168.255459237664	0.7143906335266
r6_t8	4073	14991	169.847753964327	0.721151308298735
r3_t9	2233	11988	208.243203891024	0.784126824155961
r4_t9	4171	11988	210.596607871907	0.792988420380977
r5_t10	2061	9009	278.301671956478	0.942169535301302
r6_t10	4122	9009	292.538523534286	0.990367333542753
r3_t11	2085	6006	373.259204803193	1.14702555638856
r4_t11	4048	6006	384.302048644312	1.18096021610485
r5_t12	2110	3003	468.338072450819	1.317602451045
r6_t12	4097	3003	477.727636549233	1.34401865207169
r3_t13	3042	1058	530.802417074686	1.39705077397572
r4_t13	3091	49	563.187335703785	1.4822865871155
r5_t14	6012	541	564.302552781047	1.34530474100582
r6_t14	8024	516	594.120598041127	1.41639135484128
r3_t15	6061	3027	485.609559343648	1.09378745283586
r4_t15	8146	2929	507.793269947329	1.14375406458964
r5_t16	6012	5982	434.710789197238	0.918165328968796

r6_t16	7803	5858	453.12634966753	0.957061371021434
r3_t17	6134	9009	385.966999581129	0.765451307917869
r4_t17	7827	8985	402.197709733215	0.797640117655058
r5_t18	5815	11939	392.099893425467	0.73505957033641
r6_t18	7950	11890	389.490359688924	0.730167544657943
r3_t19	5766	14942	402.571438215006	0.713998159844504
r4_t19	8048	14967	400.928356519891	0.711084000529108
r5_t20	6772	17330	431.649661525338	0.728474989812012
r6_t20	6772	18315	463.641146641061	0.782465526284098
r3_t21	9987	17699	511.766362968467	0.804211643399711
r4_t21	12318	17724	580.428983601668	0.912111034557936
r5_t22	10134	14967	447.849208186447	0.6745225435682
r6_t22	12392	14942	516.543376441843	0.777985415117236
r3_t23	10208	11767	465.916887182862	0.669681983130348
r4_t23	12563	11791	517.175287470707	0.743357842712103
r5_t24	10183	8886	547.416343765098	0.755403049115928
r6_t24	12735	8886	595.775690401603	0.822136164264105
r3_t25	10257	5834	644.6418340517	0.853584065382998
r4_t25	12931	5834	688.416446190668	0.911546967286674
r5_t26	10330	3027	733.365940717038	0.935970758650013
r6_t26	12931	2978	776.841036025214	0.991456588682138
r3_t27	10527	615	816.015639865654	1.01081304640067
r4_t27	12784	640	848.120607107532	1.05058203875441

Următorul experiment este realizat doar cu modulele UWB fără să se ia în considerare modulele IRULT. Poziția antenelor UWB de referință este dată în mod normal de poziția roboților din *home*. Dar aici, fiind vorba de un simulator, au fost alese pentru cele două antene UWB de referință, coordonatele (0,0) și (2000). Rezultatele experimentului sunt prezentate în Figura 6-15 și Tabelul 6-10. Rezultatele sunt expuse doar pentru roboții r3 și r4 la momentele t21 și t27. Se poate observa că erorile de localizare sunt enorm de mari.

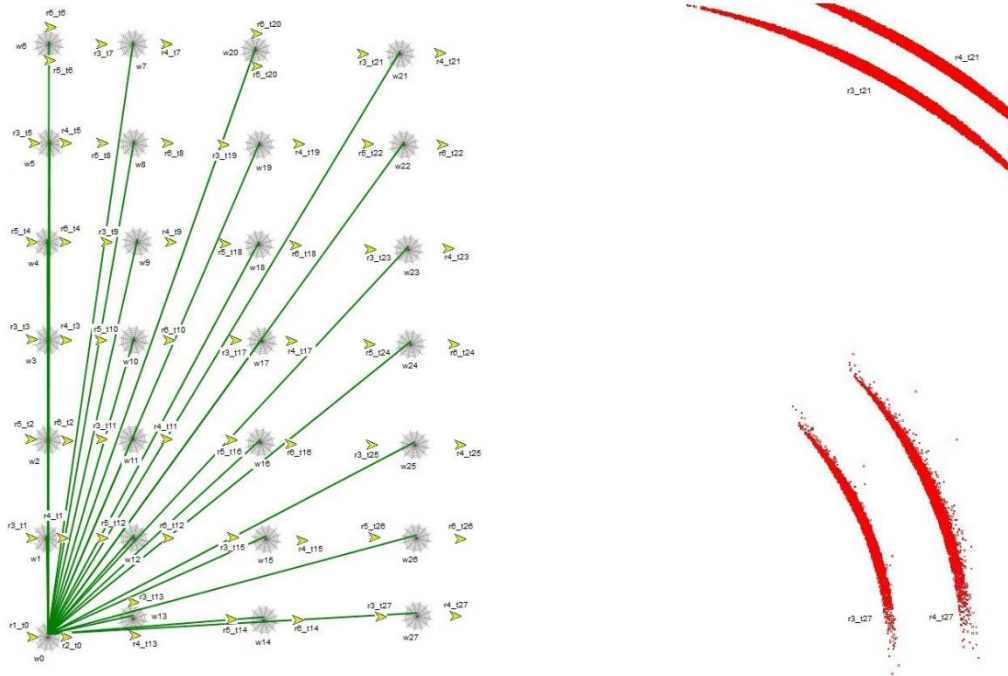


Figura 6-15 Simulare – doar UWB

Tabelul 6-10 Metoda BPF (doar UWB)

CONF7 - ILS - 30000 - UWB 150				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
r1_t0	0	0	0	0
r2_t0	1000	0	0	0
r3_t21	9987	17699	8407.72924747769	40.6796142679254
r4_t21	12318	17724	8022.02159873008	38.813422111946
r3_t27	10527	615	7517.31257155012	67.2802472125677
r4_t27	12784	640	9115.02960715342	81.5798783783574

Pentru cel de-al treilea experiment, au fost folosite ambele surse de măsurat distanța: UWB și IRULT. Cu toate că toleranța maximă de măsurare a distanțelor cu modulele UWB este mult mai mare decât cu IRULT, prin fuziune, modulele UWB aduc o îmbunătățire majoră în localizare. Dacă comparăm cu rezultatele de la primul experiment, unde pentru r3 și r4 la momentul t27 am avut 816mm respectiv 848mm, acum pentru același moment avem 173mm respectiv 219mm. Rezultate sunt expuse în Figura 6-16 și Tabelul 6-11.

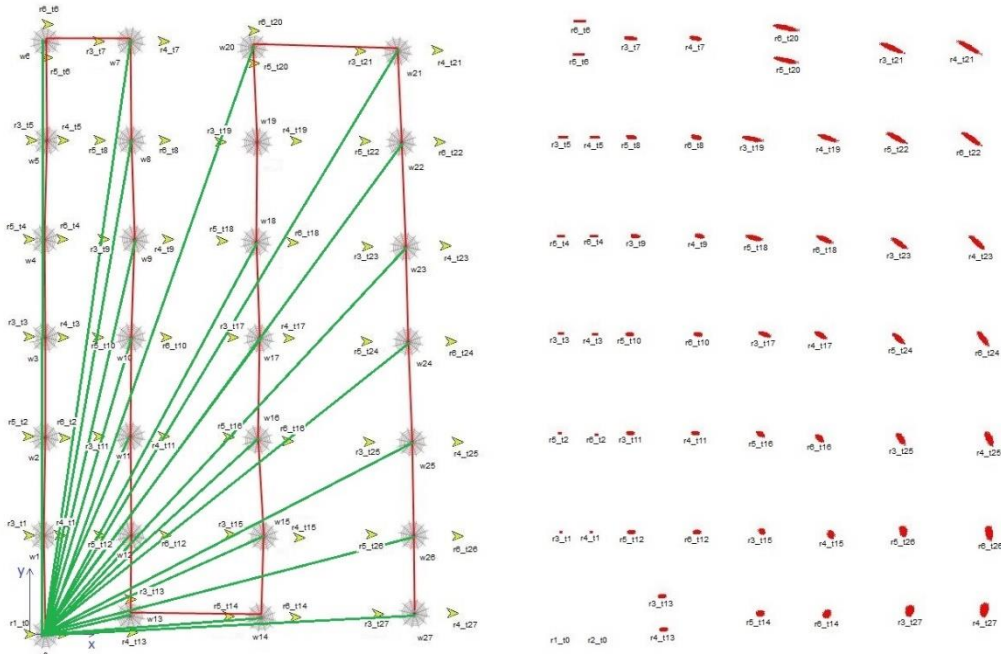


Figura 6-16 Simulare – UWB și IRULT

Tabelul 6-11 Metoda BPF (combinat)

CONF7 - BPF - 30000 - IRULT 2 - UWB 150				
Robot_moment	x [mm]	y [mm]	Eroarea absolută [mm]	Eroarea relativă [%]
r1_t0	0	0	0	0
r2_t0	1000	0	0	0
r3_t1	-25	3003	15.5544653053142	0.517864273076524
r4_t1	907	3003	15.5814333802445	0.518762137596195
r5_t2	-50	6006	37.6776922259183	0.629775880601332
r6_t2	1055	5957	37.096156088851	0.620055608175222
r3_t3	-25	9034	69.5325070342683	0.770636435108115
r4_t3	1006	9009	70.0070883543799	0.775896272158424
r5_t4	-25	11988	101.397699635104	0.845737454608155
r6_t4	981	11988	100.136298890181	0.835216369227603
r3_t5	49	14991	132.603880964854	0.884457939772544
r4_t5	1006	14991	131.730197123835	0.87863053408564
r5_t6	515	17502	159.213461281855	0.884093273728938
r6_t6	539	18512	170.279514269955	0.945541740050438
r3_t7	2085	17995	165.233623144142	0.804126965787563
r4_t7	4048	17995	167.705991305628	0.816159007875457
r5_t8	2110	14991	139.561584658495	0.592560201802003
r6_t8	4073	14991	143.156357854215	0.607823137770382
r3_t9	2233	11988	121.648988968689	0.458061697090099
r4_t9	4171	11988	122.656338045316	0.461854807345382

r5_t10	2061	9009	96.9671334006647	0.328274991570625
r6_t10	4122	9009	96.9844123291341	0.328333488093093
r3_t11	2085	6006	100.803191978717	0.309768214359496
r4_t11	4048	6006	100.200127440979	0.307914997002669
r5_t12	2110	3003	102.556289105793	0.288527510028694
r6_t12	4097	3003	102.157524934935	0.287405643838822
r3_t13	3042	1058	106.074461299277	0.279183748020425
r4_t13	3091	49	105.068491514738	0.276536075702292
r5_t14	6012	541	107.648127911465	0.256634559113692
r6_t14	8024	516	119.036907209867	0.283785559421774
r3_t15	6061	3027	99.1397810841903	0.223302541188328
r4_t15	8146	2929	118.485144220483	0.266876056292992
r5_t16	6012	5982	131.626835061493	0.27801287503937
r6_t16	7803	5858	145.690506267898	0.307717164927327
r3_t17	6134	9009	183.813867592314	0.364540402455843
r4_t17	7827	8985	200.409371496804	0.397452672631878
r5_t18	5815	11939	250.242483722039	0.46912313838617
r6_t18	7950	11890	263.797492444523	0.494534364082791
r3_t19	5766	14942	317.697627589706	0.563466505452545
r4_t19	8048	14967	331.233324191447	0.587473331442756
r5_t20	6772	17330	373.501003182127	0.63034021276976
r6_t20	6772	18315	398.932845071762	0.673260345490321
r3_t21	9987	17699	407.445481301025	0.640277329311448
r4_t21	12318	17724	431.670534205113	0.678345617923535
r5_t22	10134	14967	353.454402709981	0.532350975268577
r6_t22	12392	14942	378.124329110303	0.569507279669718
r3_t23	10208	11767	292.993913272463	0.421132502991708
r4_t23	12563	11791	324.14826465735	0.465911965579383
r5_t24	10183	8886	243.885721704891	0.33654825967485
r6_t24	12735	8886	283.727851917136	0.391528106346215
r3_t25	10257	5834	204.868855903637	0.271271241882458
r4_t25	12931	5834	252.770258923845	0.33469851601785
r5_t26	10330	3027	178.890641281596	0.228312224415568
r6_t26	12931	2978	229.402491545905	0.292778832677299
r3_t27	10527	615	173.201061404589	0.214547226750489
r4_t27	12784	640	219.731719586945	0.27218557833414

Această strategie de a crea baze noi aduce indirect și o îmbunătățire majoră în localizare prin fuziunea datelor obținute de la două surse diferite: setul de distanțe UWB (informație pentru o pânză curentă dată de o bază) și setul de distanțe IRULT (informație pentru o pânză curentă dată de o pânză precedentă).

6.8 Concluzii

De obicei în literatura de specialitate găsim sisteme de localizare pentru roboți mobili care folosesc următoarele abordări sau derivă din acestea: SLAM, Kalman Filter, ILS sau filtre de particule bazate pe metoda Monte Carlo. Fiecare abordare are o problemă specifică aparte pentru a fi aplicată pentru aplicații de explorare în medii necunoscute. De exemplu, prin abordarea SLAM, nu se poate reconstitui locația în mediu dacă mediul este dinamic. Kalman Filter are nevoie de un model pentru senzori, iar comportamentul senzorilor în practică este greu de modelat. ILS pierde din acuratețe la calcularea valorii reziduale, iar erorile se acumulează ad-hoc. Filtrele de particule prin abordările existente în literatura de specialitate necesită o hartă, în schimb atunci când se face o explorare a unui mediu necunoscut harta nu există.

În acest capitol s-a discutat un concept nou în domeniul localizării. Ideea a venit după analiza abordărilor existente. Era nevoie de o metodă care să păstreze istoricul măsurărilor și estimărilor în așa fel încât prin estimări și validări repetitive să fie găsită soluția potrivită. Apelând la metoda BPF prezentată în acest capitol, nu se pierde așa de mult din acuratețe așa cum ILS pierde din acuratețe la calcularea valorii reziduale. În schimb, metoda BPF necesită foarte multe resurse de calcul. Acum zece ani, sau mai mult, era mai greu de imaginat un astfel de concept din lipsa resurselor de calcul sau costurilor imense nesuportabile. Astăzi, când noțiunea „cloud computing” este pusă în practică, se deschid drumuri spre idei noi. Spre exemplu, entitatea BRAIN poate fi văzută ca și o soluție software care se execută în „cloud”. Algoritmul BPF, prezentat în acest capitol, a fost construit pentru nivelul CENTRALIZARE care are la bază entitatea BRAIN, entitate care poate fi reprezentată ca și un ansamblu de servicii de calcul în „cloud”.

Algoritmul BPF este un algoritm de tip „filtru de particule” dar este foarte diferit de abordarea filtrelor de particule regăsite în literatura de specialitate prin simplul fapt că nu necesită o hartă. Mai mult, filtrele de particule din literatura de specialitate se bazează pe modelul Monte Carlo, în schimb algoritmul BPF se bazează pe modelul Las Vegas.

S-au realizat mai multe experimente în simulator folosind algoritmul BPF. Din rezultatele obținute se observă că metoda BPF are acuratețe mai bună comparativ cu metoda ILS. S-a constatat că fuziunea de date este relevantă pentru a crește confiența în localizare. O fuziune de date pentru metoda ILS nu este exclusă, doar că este mai grea în implementare. De asemenea a fost propusă soluția de localizare prin construire a *bazelor* pentru problema aplicațiilor cu roboți mobili care explorează medii necunoscute indiferent că traseul este scurt sau foarte lung.

7. CONCLUZII ȘI PERSPECTIVE

„În sfârșit, în urmă cu două zile, am reușit – nu datorită eforturilor mele grele, ci datorită harului lui Dumnezeu. Asemeni unei bruște sclipiri de fulger, enigma a fost rezolvată. Nu aş putea spune care a fost firul călăuzitor care lega ceea ce cunoşteam înainte cu ceea ce a făcut reuşita mea posibilă”.

Karl Friedrich Gauss

7.1 Concluzii

În lucrarea de față se abordează problematica localizării în contextul explorării roboților autonomi în medii necunoscute. Principalul scop al activității de cercetare este de a propune o metodologie de localizare relativă, bazată pe modele matematice, algoritmi, metode și tehnologii de localizare, care poate fi utilă în aplicații de explorare cu roboți mobili autonomi în medii necunoscute.

Obiectivele propuse în cadrul acestui program de cercetare iau în vedere problemele existente ale domeniului localizării și au ca țintă atingerea scopului principal. O discuție detaliată a modului în care au fost tratate și atinse aceste obiective este prezentată în continuare.

(O1). Analiza și structurarea soluțiilor existente, evidențierea celor mai importante tehnici și tehnologii existente.

Soluțiile de localizare existente, pot avea caracteristici cu apartenență la una sau mai multe categorii din cele enumerate mai jos:

- CAT I – sisteme de localizare globală
- CAT II – sisteme de localizare locală cu repere fixe - active
- CAT III – sisteme de localizare locală cu repere fixe - pasive
- CAT IV – sisteme de localizare locală cu repere mobile
- CAT V – sisteme de localizare locală fără repere

Astfel, se poate spune că orice sistem de localizare poate să fie conceput pe baza unei sau mai multor categorii dintre cele enumerate mai sus. Metodologia propusă are caracteristici apropiate categoriei CAT IV, dar în funcție de nivelul aplicat, poate să conțină și alte caracteristici care sunt specifice celorlalte categorii. De exemplu, nivelul I de localizare (PREDICȚIE) este conceput pe baza categoriei CAT V. În schimb, nivelurile II și III (COOPERARE respectiv CENTRALIZARE) sunt concepute pe baza categoriei CAT IV. Opțional, toate nivelurile pot avea și caracteristicile categoriei CAT II atunci când, de exemplu, *bazele* sunt folosite pe post de balize UWB. Prin urmare, se poate spune că metodologia este concepută pentru un sistem de *localizare hibridă* care combină categoriile CAT II, CAT IV și CAT V.

Prin analiza caracteristicilor din cadrul soluțiilor de localizare existente, autorul tezei a definit cele 5 categorii amintite mai sus. Categoriile au fost exemplificate în subcapitolul 2.2. În subcapitolul 2.3 au fost prezentate tehnicile: TOF, TDOA, AOA, RTT, PDOA, Corelația semnalului, și RSSI. Iar în subcapitolul 2.4 au fost prezentate tehnologiile WiFi, CSS, UWB, Zigbee, Ultrasunete și IR.

(O2). Identificarea factorilor de influență și a problemelor existente în localizarea roboților mobili.

Factorii de influență în localizare, semnificativi pentru aplicații de explorare cu roboți mobili autonomi în medii necunoscute sunt împărțiți în două categorii: principali și secundari.

Factori principali:

- independența de repere fixe, aplicabilitate fără a necesita sisteme de orientare și localizare cu reper fix, aplicabilitate pentru spații închise (indoor localization), sistemul să poată fi mutat în alt mediu fără a fi nevoie de o preinstalare;
- dinamica mediului, aplicabilitate acolo unde nu există repere pasive sau sunt foarte greu de recunoscut;
- acuratețe, calculul distanței între roboții autonomi care elimină timpii imprecizabili proveniți din comunicarea datelor prin unde electromagnetice, fără a necesita sincronizarea bazelor de timp ale roboților, eliminarea problemei interferențelor distructive, eliminarea problemei erorii de măsurare cauzate de atenuarea undelor mecanice prin unghiul de incidență prea mare la recepționare.

Factori secundari:

- viteza de execuție;
- scalabilitate;
- conectivitate;
- eficiență energetică;
- cost;
- robustețe;
- standardizare.

Problemele existente în funcție de factorii de influență au fost discutate în subcapitolul 2.4. Una dintre problemele discutate se referă la robustețe. Fiecare robot, care face parte dintr-un sistem cu mai mulți roboți mobili și este conceput după modelul WIT [16], poate să comunice cu ceilalți roboți, astfel sistemul construit devine un sistem colaborativ. Scopul unui astfel de sistem este de a rezolva problemele pe care sistemul cu un singur robot mobil nu le poate atinge. Unul dintre avantajele unui sistem cu mai mulți roboți este probabilitatea mai mare de a satisface cerințele aplicației. În cazul sistemului cu un singur robot mobil, acesta fiind echipat cu mai mulți senzori este mult mai costisitor, iar pe lângă aceasta, în cazul unei defecțiuni a robotului, sistemul devine nefuncțional. În cazul sistemului cu mai mulți roboți mobili, simpli și de cost redus, cu toate că un robot se poate defecta, sistemul fiind colaborativ, poate să atingă țelul dorit.

(O3). Dezvoltarea unor metode de calcul de distanță între doi roboți folosind tehnologii bazate pe semnale mecanice.

Calculul distanței între doi roboți mobili se realizează printr-un proces colaborativ. Această informație este utilă în cazul localizării roboților. Cunoscând distanțele între roboți, am putea să determinăm locația acestora. Un robot mobil care respectă modelul WIT, ar putea să comunice cu un alt robot și să calculeze distanța față de acesta. Modulul care implementează funcționalitatea de calcul a distanței este modulul de percepție care la rândul lui este echipat cu modulul IRULT. În subcapitolele 3.5 și

3.6 s-a discutat despre două tehnici de măsurare a distanței între doi roboți. S-a arătat cum MTDOA și metodele CTOF îndeplinesc cerințele specificate (factorii de influență în localizare), de exemplu: cost redus, acuratețe, fără a fi nevoie de o preinstalare.

(O4). Propunerea unui sistem hardware pentru roboți mobili care să ofere facilități de localizare având la bază metodele de calcul a distanței.

În capitolul 3 autorul tezei a expus sistemul hardware pentru poziționare relativă, IRULT, cu un set comun de cerințe pentru sistemul robotic țintă CORE-TX. Analizând modulele destinate localizării în literatura de specialitate și observând problemele acestora, s-a ajuns la o propunere de modul nou destinat mediilor colaborative robotizate, care are ca scop principal calculul distanței între doi roboți mobili.

(O5). Elaborarea unei metodologii de localizare aplicabilă pe o gamă largă de sisteme luând ca exemplu sistemul CORE-TX.

Metodologia prezentată în capitolul 4 a fost dezvoltată în paralel cu dezvoltarea modului IRULT. Noțiunile au fost stabilite pe măsură ce s-au obținut rezultatele experimentale și s-a observat că definirea unei metodologii este absolut necesară, mai ales pentru că în prezent, nu există niciun standard care să guverneze cercetarea localizării în mediul interior al clădirilor [81]. Pentru dezvoltare s-a luat ca și referință sistemul CORE-TX, dar s-a ținut cont de generalizare și aplicabilitate pe o gamă largă de sisteme.

Anumite aspecte care țin de metodologie și nu sunt prezentate în capitolul 4, apar în continuare în capitolele 5 și 6, cum ar fi de exemplu modelul matematic de localizare pentru nivelul COOPERARE și algoritmul BPF pentru nivelul CENTRALIZARE. Aceste informații au fost introduse în capitole separate pentru că necesită o evaluare separată prin simulări.

(O6). Concepția și definirea unui model matematic de localizare bazat pe cooperare și localizare multilaterală și validarea acestuia.

În subcapitolul 5.4.2 s-a propus un model matematic, generalizat, oferind posibilitatea de a localiza o pânză în raport cu alta, indiferent de numărul de roboți care alcătuiesc o pânză. Localizarea prin COOPERARE se face secvențial din pânză în pânză, astfel o pânză care trebuie localizată este dependentă doar de pânza de referință. Acest lucru face posibilă implementarea modelului matematic la nivel de WIT. Avantajul acestei idei este că sistemul de localizare la nivelul COOPERARE poate fi funcțional și atunci când, din diverse motive, lipsește comunicarea cu entitatea BRAIN.

S-a ales ca validarea modelului să fie făcută prin simulări de tip Monte Carlo, deoarece, în practică, erorile de localizare cu propagare secvențială sunt foarte greu de reprodus. Atunci când distribuția zgomotului în cazul măsurării distanței nu poate fi modelată, singurul lucru ce se poate face este să fie găsit intervalul maxim și minim al distribuției. Cu un astfel de zgomot, găsirea celor mai defavorabile cazuri în practică este imposibilă. De aceea, aplicând metoda Monte Carlo în simulări, se poate evalua modelul rapid și corect. În simulările efectuate în simulatorul SiMuLoC se ține cont de acest aspect. S-a considerat că zgomotul aleator în cazul măsurării distanțelor poate fi descompus în două părți. Prima parte are distribuție gaussiană iar a doua nu poate fi modelată. Distribuția gaussiană se rezolvă la nivel de modul IRULT, prin măsurători repetitive, aplicând de exemplu filtrul Kalman, iar distribuția non-gaussiană nu poate

fi rezolvată ci trebuie cuantificată secvențial din pânză în pânză, iar acest lucru se poate face folosind simulatorul SiMuLoC.

(O7). Definirea unor reguli, pe baza cărora să se păstreze la un nivel superior confiența de localizare a roboților mobili.

Regulile de formare a pânzelor: PRP, DST, SRT și PNR au fost expuse în subcapitolul 5.7. Aplicând aceste reguli, cuantificarea erorilor de propagare se modifică. De aceea metodologia definește aceste reguli, prin care, confiența în localizare crește. Evaluarea acestor reguli a fost realizată, de asemenea, în subcapitolul 5.7. Aceste reguli sunt valabile și pentru nivelul CENTRALIZARE.

(O8). Exemplificarea metodologiei pe un studiu de caz concret.

În subcapitolul 5.7 a fost descris un scenariu pentru aplicațiile de supraveghere interioară, unde s-au utilizat roboți distribuiți folosind metodologia de localizare. A fost prezentată o configurație simplă de poziționare a pânzelor în cadrul unui birou cu 6 camere. Sistemului de supraveghere în interior a fost evaluat aplicând în primă fază metoda ILS (nivelul COOPERARE) iar apoi în a doua fază a fost aplicată metoda BPF (nivelul CENTRALIZARE). Evaluarea a fost realizată în capitole separate. Pentru metoda ILS rezultatele sunt prezentate în capitolul 5 iar pentru metoda BPF rezultatele sunt prezentate în capitolul 6. Examinând rezultatele acestor simulări, s-a observat că acuratețea de localizare a metodei BPF este așa cum era de așteptat mai bună decât cea a metodei ILS. În plus, acuratețea de localizare a ambelor metode este suficient de fiabilă pentru a fi utilizată pentru aplicații de supraveghere interioară, atunci când toleranța pentru măsurarea distanței nu depășește toleranța maximă propusă (toleranța țintă) de măsurare a distanței prin intermediul modulelor IRULT ($\pm 2\text{mm}$). Această toleranță poate fi depășită atunci când, de exemplu, se face o fuziune dintre măsurătorile distanțelor și a unghiurilor dintre roboți sau atunci când sistemul dispune și de module UWB.

(O9). Elaborarea unui algoritm de localizare de confiență superioară, destinat sistemelor cu procesare distribuită.

Aplicarea metodologiei pentru stratul de control și supraveghere CORE-TX presupune centralizarea coordonatelor și gestionarea mobilității roboților mobili în formarea pânzelor, astfel încât confiența în localizare să fie conform cerințelor. Metodologia aplicată la acest strat prin nivelul CENTRALIZARE, completează nivelul COOPERARE oferind o confiență de localizare mai ridicată. Nivelul CENTRALIZARE are la bază echipamentul cu poziție fixă: BRAIN.

La nivelul CENTRALIZARE, coordonatele roboților se obțin print-un „filtru de particule” sub o nouă abordare. Algoritmul BPF este un algoritm de tip „filtru de particule” dar este foarte diferit de abordarea filtrelor de particule regăsite în literatura de specialitate prin simplul fapt că nu necesită o hartă. Mai mult, filtrele de particule din literatura de specialitate se bazează pe modelul Monte Carlo, în schimb algoritmul BPF se bazează pe modelul Las Vegas.

Obiectivul metodologiei este de a oferi la nivelul CENTRALIZARE un algoritm general care pune accent pe minimizarea erorilor de propagare în schimbul resurselor de calcul. Autorul tezei a propus algoritmul BPF care are la bază entitatea BRAIN și distribuirea sarcinii pe mai multe procese. În capitolul 6 au fost expuse mai multe experimente în simulator folosind algoritmul BPF. Din rezultatele obținute se observă

că metoda BPF are acuratețe mai bună comparativ cu metoda ILS. Algoritmul oferă o integrare ușoară a oricărei informații ce ține de localizare, cum ar fi de exemplu: măsurarea unghiului sau obținerea distanțelor prin intermediul modulelor UWB. Din experimentele efectuate s-a constatat că fuziunea de informații multiple este relevantă pentru a crește confiența în localizare.

7.2 Sinteza contribuțiilor

În cele ce urmează va fi prezentat un scurt rezumat al contribuțiilor cuprinse în această lucrare de doctorat, structurate la nivel de capitole:

Capitolul 1:

- Identificarea celor mai semnificative exemple de aplicații în domeniul roboților autonomi pentru care se poate aplica metodologia propusă din lucrarea de față: roboți pentru pază, roboți salvatori, roboți subacvatici, roboți pentru operațiuni miniere, roboți de curățenie și roboți pentru monitorizarea fermelor.
- Identificarea factorilor de influență în localizarea roboților mobili, semnificativi pentru aplicații cu roboți mobili autonomi în medii necunoscute. Factorii de influență sunt: independența de repere fixe, dinamica mediului, viteza de execuție, acuratețea, scalabilitatea, conectivitatea, eficiența energetică, costul, robustețea și standardizarea. (O2)

Capitolul 2:

- Clasificarea sistemelor de localizare pe 5 categorii: localizare globală, localizare locală cu repere fixe active, localizare locală cu repere fixe pasive, localizare locală cu repere mobile și localizare locală fără repere. (O1)
- Exemplificarea celor 5 categorii de localizare.
- Analiza problemelor în localizare în funcție de factorii de influență identificați și propunerea soluțiilor. (O2)

Capitolul 3:

- Identificarea celor mai semnificative module în domeniul roboților autonomi care folosesc tehnologia semnalelor ultrasonice și identificarea problemelor acestora.
- Propunerea și implementarea unui modul nou (IRULT) care se bazează pe semnale ultrasonice pentru a măsura distanța în raport cu alte module IRULT și care are un set comun de cerințe pentru sistemul robotizat țintă: CORE-TX (O4). Modulul IRULT a fost patentat în [B2]. După ce cererea de brevet a fost depusă, această activitate a fost publicată în [C4].
- Dezvoltarea unui algoritm de aliniere pentru două turele care sunt echipate cu câte un modul IRULT. Această activitate a fost publicată în [C3] și [F1].

- Dezvoltarea metodelor MTDOA și CTOF pentru măsurarea distanței dintre două module IRULT (O3). Această activitate a fost publicată în [A1].

Capitolul 4:

- Propunerea unei metodologii de localizare compusă din trei niveluri: PREDICȚIE, COOPERARE și CENTRALIZARE. (O5)
- Definirea unor concepte de bază în cadrul metodologiei (O5). Parțial, conceptele au fost publicate în [C2].
- Elaborarea metodei de orientarea și localizare LOAD.
- Elaborarea metodei de localizare bilaterală.

Capitolul 5:

- Elaborarea modelului matematic de localizare bazat pe cooperare și localizare multilaterală. (O6)
- Dezvoltarea simulatorului SiMuLoC pentru validarea modelului matematic.
- Definirea unor reguli de formare a pânzelor. (O7)
- Exemplificarea metodologiei pe un studiu de caz concret (O8). Această activitate a fost publicată în [C5].

Capitolul 6:

- Elaborarea algoritmului de localizare BPF, de confiență superioară, destinat sistemelor cu procesare distribuită. (O9)
- Dezvoltarea simulatorului Simon-TX pentru testarea algoritmului BPF și validarea metodologiei.
- Extinderea algoritmului BPF pentru localizare relativă bazată pe măsurarea unghiului și distanței între roboți.
- Extinderea algoritmului pentru localizarea relativă prin construirea bazelor.

7.3 Perspective de cercetare și dezvoltare

Problemele abordate în activitatea de cercetare doctorală, conceptele, modelele și algoritmii permit continuarea preocupărilor și deschid noi perspective în domeniul de mare interes al explorării roboților autonomi în medii necunoscute.

Cercetarea de față se dorește a fi continuată urmând două direcții principale:

- 1) Dezvoltarea simulatorului Simon-TX și evaluarea performanței algoritmului BPF pentru procese multiple în simulator.
- 2) Extinderea algoritmului BPF pentru minimizarea suprafețelor S_C . Prin această actualizare poate fi redus timpul de execuție a algoritmului.

Cele două direcții principale sunt detaliate în continuare:

- 1) Experimentele expuse în lucrarea de față au fost realizate cu un singur proces pentru a evalua acuratețea rezultatelor de localizare obținute. Nu s-a ținut cont de timpul de execuție. Se dorește ca experimentele să fie extinse pentru a evalua performanța algoritmului BPF pentru procese

multiple. În acest fel se poate face și o evaluare a timpului de execuție în funcție de numărul de procese folosite.

- 2) Algoritmul BPF ar putea fi extins cu scopul de a minimiza suprafețele S_C . Dacă avem suprafețele S_C iar soluția corectă se află pe suprafețe mai mici S_m unde $S_m \subset S_C$, atunci timpul de găsim a punctelor P_n pe suprafețele S_m crește proporțional cu raportul dintre S_C și S_m . Pentru a reduce acest timp, trebuie ca raportul dintre S_C și S_m să fie redus. Minimizarea suprafeței S_C a fost încercată prin reducere la una dintre suprafețele: S_{11} , S_{22} , S_{12} , S_{21} . Acest lucru a fost realizat cu statistica punctelor P_n trecute în lista neagră. S-a realizat o statistică pentru fiecare suprafață S_{ij} . Care suprafață avea cele mai puține puncte P_n adăugate în lista neagră a devenit suprafață înlocuitoare pentru suprafața S_C . Această tehnică nu a funcționat, dar este propusă pentru a fi analizată și îmbunătățită în viitor. De exemplu, suprafața S_C ar putea fi împărțită în 9 suprafețe pentru realizarea statisticii în loc de 4.

Există și alte idei care sunt foarte interesante și pot fi luate în considerare în viitor. În cercetarea doctorală nu s-a pus accent pe faptul că roboții ar trebui să se întoarcă la bazele precedente. Sistemul BRAIN ar trebui să aibă și un mecanism prin care stabilește momentul când trebuie să fie creată o bază nouă astfel încât să permită bazei create posibilitatea de reîntoarcere în poziția home.

În simulatorul Simon-TX ar putea fi implementată o funcție care calculează câte variante au fost retrase din cele posibile. În cazul studiului de caz prezentat avem complexitate 25^{11} semnificând aproximativ $2 \cdot 10^{15}$ de variante posibile pentru un set. 10^5 de pași înapoi acoperă mai mult decât 10^5 de variante. Această nouă funcție ar putea să returneze o valoare în procent care exprimă raportul dintre variantele retrase și cele posibile. Această informație poate să deschidă idei noi pentru optimizarea timpului de execuție a algoritmului BPF.

Metoda BPF implementată în simulatorul Simon-TX ar putea să genereze rezultate de localizare ca și referință pentru dezvoltarea altor sisteme de localizare unde modelarea este greu de realizat. Astfel, BRAIN poate oferi un serviciu de simulare în care utilizatorul ar putea să introducă tot felul de parametri și resurse. Prin analiza rezultatelor, utilizatorul ar putea să vadă dacă are nevoie de mai multe resurse sau sunt suficiente pentru a rezolva problema de localizare. După ce clarifică aceste aspecte ar putea să pornească la drum cu proiectarea și implementarea sistemului propriu. Un astfel de serviciu oferit de platforma CORE-TX, ar ajuta foarte mult utilizatorul să se informeze de la început dacă un sistem pe care vrea să-l construiască va avea erori peste limita care îi impune cerința. De exemplu ar putea să-și dea seama dacă mai are nevoie și de alte repere în plus față de cele propuse sau ar putea să-și dea seama de toleranța maximă pe care trebuie să o aibă dispozitivul de măsurat distanța încât să poată să-și îndeplinească cerința impusă.

O altă idee ce poate fi implementată în viitor cu ajutorul simulatorului Simon-TX este să se determine numărul minim de roboți necesari pentru a explora un labirint. Se presupune că mediul este dinamic și obstacolele labirintului greu de identificat. Soluțiile deja existente în literatura de specialitate bazate pe categoria CAT III de localizare, unde se poate folosi doar un singur robot pentru explorare, nu sunt aplicabile în acest caz. Pentru un astfel de studiu de caz, aplicarea metodologiei prezentate în această lucrare de doctorat deschide drumul spre o altă cercetare interesantă.

REFERINȚE BIBLIOGRAFICE

- [1] „CCCTI - Centrul de Cercetare în Calculatoare și Tehnologia Informației”. [Online]. Valabil la: <http://research.cs.upt.ro/>.
- [2] „Digital Signal Processing Laboratories in Timisoara”. [Online]. Valabil la: <http://dsplabs.cs.upt.ro/>.
- [3] K. Ohno, S. Kawatsuma, T. Okada, E. Takeuchi, K. Higaschi, și S. Tadokoro, „Robotic control vehicle for measuring radiation in Fukushima Daiichi Nuclear Power Plant”, prezentat la Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on, 2011.
- [4] F. Baccelli și J. Bolot, „Modeling the economic value of the location data of mobile users”, prezentat la INFOCOM, 2011 Proceedings IEEE, 2011, pp. 1467-1475.
- [5] D. Reyes, G. Millan, R. Osorio-Corparan, și G. Lefranc, „Mobile Robot Navigation Assisted by GPS”, *IEEE Lat. Am. Trans.*, vol. 13, nr. 6, pp. 1915-1920, iun. 2015.
- [6] X. Wang, S. Vozar, și E. Olson, „FLAG: Feature-based Localization between Air and Ground”, în *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3178-3184.
- [7] R. Kurazume, S. Nagata, și S. Hirose, „Cooperative positioning with multiple robots”, prezentat la Robotics and Automation, 1994 IEEE International Conference on, 1994.
- [8] S. Witwicki *et al.*, „Autonomous Surveillance Robots: A Decision-Making Framework for Networked Multiagent Systems”, *IEEE Robot. Autom. Mag.*, vol. 24, nr. 3, pp. 52-64, sep. 2017.
- [9] N. Michael *et al.*, „Collaborative mapping of an earthquake-damaged building via ground and aerial robots”, *J. Field Robot.*, vol. 29, nr. 5, pp. 832-841, sep. 2012.
- [10] S. Nawaz, M. Hussain, S. Watson, N. Trigoni, și P. N. Green, „An Underwater Robotic Network for Monitoring Nuclear Waste Storage Pools”, în *Sensor Systems and Software: First International ICST Conference, S-CUBE 2009, Pisa, Italy, September 7-9, 2009, Revised Selected Papers*, S. Hailes, S. Sicari, și G. Roussos, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 236-255.
- [11] D. Ferguson *et al.*, „An Autonomous Robotic System for Mapping Abandoned Mines”, în *Advances in Neural Information Processing Systems (NIPS 03)*, 2003.
- [12] J. H. Lee, J. S. Choi, B. H. Lee, și K. W. Lee, „Complete coverage path planning for cleaning task using multiple robots”, în *2009 IEEE International Conference on Systems, Man and Cybernetics*, 2009, pp. 3618-3622.
- [13] Joon Seop Oh, Yoon Ho Choi, Jin Bae Park, și Y. F. Zheng, „Navigation of cleaning robots using triangular-cell map for complete coverage”, în *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, 2003, vol. 2, pp. 2006-2011 vol.2.
- [14] R. Kurazume și S. Hirose, „Development of a Cleaning Robot System with Cooperative Positioning System”, *Auton. Robots*, vol. 9, nr. 3, pp. 237-246, 2000.

- [15] D. A. Shaikh, G. Ghorale Akshay, A. Chaudhari Prashant, și L. Kale Parmeshwar, „Intelligent Autonomous Farming Robot with Plant Disease Detection using Image Processing”, *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 5, nr. 4, pp. 1012-1016, apr. 2016.
- [16] R. D. Cioargă, M. V. Micea, B. Ciubotaru, D. Chiciudean, și D. Stănescu, „CORE-TX: Collective Robotic Environment - the Timisoara Experiment”, prezentat la Proc. 3rd Romanian-Hungarian Joint Symp. Applied Computational Intellig., SACI 2006, 2006.
- [17] M. D. Venkata, M. M. M. Pai, R. M. Pai, și J. Mouzna, „Traffic monitoring and routing in VANETs — A cluster based approach”, prezentat la ITS Telecommunications (ITST), 2011 11th International Conference on, 2011.
- [18] I. X. Gao, Z. FeiFei, și M. Ito, „Underwater acoustic positioning system based on propagation loss and sensor network”, prezentat la OCEANS, 2012 - Yeosu, 2012.
- [19] R. Piza, J. Salt, A. Cuenca, și V. Casanova, „Control network architecture for an autonomous underwater vehicle divisamos project”, prezentat la OCEANS, 2011 IEEE - Spain, 2011.
- [20] A. Parness, M. Frost, N. Thatte, și J. P. King, „Gravity-independent mobility and drilling on natural rock using microspines”, prezentat la Robotics and Automation (ICRA), 2012 IEEE International Conference on, 2012.
- [21] K. Soowoong, S. Jae-Young, și Y. Seungjoon, „Vision-based cleaning area control for cleaning robots”, *Consum. Electron. IEEE Trans. On*, vol. 58, pp. 685-690, mai 2012.
- [22] M. Rampinelli, T. F. Bastos, R. F. Vassallo, și D. Pizarro, „Implementation of an intelligent space for localizing and controlling a robotic wheelchair”, prezentat la Biosignals and Biorobotics Conference (BRC), 2012 ISSNIP, 2012.
- [23] R. A. Gonzales, F. A. Gaona, și R. R. Peralta, „An autonomous robot based on a Wheelchair”, prezentat la Electrical Communications and Computers (CONIELECOMP), 2012 22nd International Conference on, 2012.
- [24] T. Matsumoto, S. Satake, T. Kanda, M. Imai, și N. Hagita, „Do you remember that shop? — Computational model of spatial memory for shopping companion robots”, prezentat la Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on, 2012.
- [25] H. Hlavacs și K. A. Hummel, „Cooperative Positioning when Using Local Position Information: Theoretical Framework and Error Analysis”, *Mob. Comput. IEEE Trans. On*, vol. 12, pp. 2091-2104, oct. 2013.
- [26] A. Chilian, H. Hirschmuller, și M. Gerner, „Multisensor data fusion for robust pose estimation of a six-legged walking robot”, prezentat la Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, 2011.
- [27] C. Guanghui, N. Matsuhira, J. Hirokawa, H. Ogawa, și I. Hagiwara, „Mobile robot global localization using particle filters”, prezentat la Control, Automation and Systems, 2008. ICCAS 2008. International Conference on, 2008.
- [28] A. A. B. Ariffin, N. H. A. Aziz, și K. A. Othman, „Implementation of GPS for location tracking”, prezentat la Control and System Graduate Research Colloquium (ICSGRC), 2011 IEEE, 2011.

- [29] T. Venkateswara Rao, M. V. Venkataramana Rao, și P. V. Narasaiah, „Mobile Robot Self Planning and navigation based on Artificial landmark Localization method and Binocular Stereo Vision”, *Int. J. Electron. Signals Syst. IJESS*, vol. 1, pp. 42-45, 2012.
- [30] M. Brkic, M. Lukic, J. Bajic, B. Dakic, și M. Vukadinovic, „Hardware realization of autonomous robot localization system”, prezentat la MIPRO, 2012 Proceedings of the 35th International Convention, 2012.
- [31] J. Urena, D. Ruiz, J. C. Garcia, J. J. Garcia, A. Hernandez, și M. C. Perez, „LPS self-calibration method using a mobile robot”, prezentat la Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE, Binjiang, China, 2011.
- [32] G. P. Settembre, P. Scerri, A. Farinelli, K. Sycara, și D. Nardi, „A Decentralized Approach to Cooperative Situation Assessment in Multi-Robot Systems”, prezentat la International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2008.
- [33] Z. Zhiwei *et al.*, „High-precision localization using visual landmarks fused with range data”, prezentat la Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011.
- [34] D. Yingjie, D. Runwei, și L. Hong, „A Probabilistic Method of Bearing-only Localization by Using Omnidirectional Vision Signal Processing”, prezentat la Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on, 2012.
- [35] Marius Hofmeister, Maria Liebsch, și Andreas Zell, „Visual Self-Localization for Small Mobile Robots with Weighted Gradient Orientation Histograms”, prezentat la 40th International Symposium on Robotics (ISR), Barcelona, Spain, 2009.
- [36] H.-S. Ahn și W. Yu, „Indoor Localization Techniques Based on Wireless Sensor Networks”, în *Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions*, InTech, 2009, pp. 277-302.
- [37] X. Kuai, K. Yang, S. Fu, R. Zheng, și G. Yang, „Simultaneous Localization and Mapping (SLAM) for Indoor Autonomous Mobile Robot Navigation in Wireless Sensor Networks”, prezentat la Networking, Sensing and Control (ICNSC), 2010 International Conference on, 2010.
- [38] A. Chatterjee, A. Rakshit, și N. N. Singh, „Simultaneous Localization and Mapping (SLAM) in Mobile Robots”, în *Vision Based Autonomous Robot Navigation*, Springer Berlin Heidelberg, 2013, pp. 167-206.
- [39] J. A. Fernandez-Madrigal și J. L. Blanco Claraco, *Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods*. IGI Global, 2012.
- [40] H. Durrant-Whyte și T. Bailey, „Simultaneous Localization and Mapping: Part I”, *Robot. Autom. Mag. IEEE*, vol. 13, pp. 99-110, 2006.
- [41] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, și R. J. O’Dea, „Relative location estimation in wireless sensor networks”, *IEEE Trans. Signal Process.*, vol. 51, nr. 8, pp. 2137-2148, aug. 2003.
- [42] S. Y. Cho, J. Y. Kim, și M. Enkhtur, „P2P Ranging-Based Cooperative Localization Method for a Cluster of Mobile Nodes Containing IR-UWB PHY”, *ETRI J.*, vol. 35, nr. 6, pp. 1084-1093, dec. 2013.

- [43] „Official U.S. government information about the Global Positioning System (GPS) and related topics”. [Online]. Valabil la: <https://www.gps.gov/>.
- [44] „Information and Analysis Center for Positioning, Navigation and Timig”. [Online]. Valabil la: <https://www.glonass-iac.ru/en/>.
- [45] E. J. Nnenna și O. H. Onyekachi, „Mobile Positioning Techniques in GSM Cellular Networks: A Comparative Performance Analysis”, *Int. J. Comput. Technol. Electron. Eng. IJCTEE*, vol. 2, nr. 6, pp. 21-29, dec. 2012.
- [46] M. V. Micea, *Telecomunicații Digitale Moderne: Suport de curs*, Ediția a 3-A. Politehnica University of Timisoara, 2008.
- [47] E. Trevisani și A. Vitaletti, „Cell-ID location technique, limits and benefits: an experimental study”, în *Sixth IEEE Workshop on Mobile Computing Systems and Applications*, 2004, pp. 51-60.
- [48] „The Cricket Indoor Location System”. [Online]. Valabil la: <http://cricket.csail.mit.edu/>.
- [49] N. B. Priyantha, „The Cricket Indoor Location System”, Ph.D. Thesis, Dept. of Electrical Eng. and Comput. Sci., Massachusetts Institute of Technology, USA, 2005.
- [50] S. Fazli și L. Kleeman, „A real time advanced sonar ring with simultaneous firing”, prezentat la 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), 2004, vol. 2, pp. 1872-1877 vol.2.
- [51] K. h. Hwang, D. e. Kim, D. h. Lee, și T. y. Kuc, „A Simple Ultrasonic GPS System for Indoor Mobile Robot System using Kalman Filtering”, prezentat la 2006 SICE-ICASE International Joint Conference, 2006, pp. 2915-2918.
- [52] S. Hirata, M. K. Kurosawa, și T. Katagiri, „Real-time ultrasonic distance measurements for autonomous mobile robots using cross correlation by single-bit signal processing”, prezentat la 2009 IEEE International Conference on Robotics and Automation, 2009, pp. 3601-3606.
- [53] K. Nakahira, T. Kodama, S. Morita, și S. Okuma, „Distance measurement by an ultrasonic system based on a digital polarity correlator”, *IEEE Trans. Instrum. Meas.*, vol. 50, pp. 1748-1752, 2001.
- [54] M. S. Reynolds, „A phase measurement radio positioning system for indoor use”, Master, Massachusetts Institute of Technology, Massachusetts, 1999.
- [55] HAGISONIC CO.,LTD, „Localization system StarGazer for Intelligent Robots”. [Online]. Valabil la: www.hagisonic.com.
- [56] Z. Parisek, Z. Ruzsa, și G. Gordos, „Mathematical algorithms of an indoor ultrasonic localisation system”, *Infocommunications J.*, vol. 64, pp. 30-36, 2009.
- [57] I. Ulrich și I. Nourbakhsh, „Appearance-based place recognition for topological localization”, în *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 2000, vol. 2, pp. 1023-1029 vol.2.
- [58] S. Bergbreiter, A. Mehta, și K. S. J. Pister, „PhotoBeacon: design of an optical system for localization and communication in multi-robot systems”, prezentat la RoboComm '07 Proceedings of the 1st international conference on Robot communication and coordination, 2007.

- [59] P. M. Maxim *et al.*, „Trilateration localization for multi-robot teams”, prezentat la in Proceedings of the Sixth International Conference on Informatics in Control, Automation and Robotics, Special Session on MultiAgent Robotic Systems (ICINCO'08), Funchal, Madeira - Portugal, 2008.
- [60] F. Rivard, J. Bisson, F. Michaud, și D. Letourneau, „Ultrasonic relative positioning for multi-robot systems”, prezentat la Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, 2008.
- [61] R. K. Ghosh, V. Kataria, și V. Mishra, „Indoor navigation system using optical mouse sensor and smartphone”, în *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2014, pp. 426-430.
- [62] O. Maye, J. Schaeffner, și M. Maaser, „An Optical Indoor Positioning System for the Mass Market”, în *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication (WPNC'06)*, Hannover, Germany, 2006, pp. 111-115.
- [63] S. Ko, J. Choi, și B. Kim, „Indoor Mobile Localization System and Stabilization of Localization Performance using Pre-filtering”, *Int. J. Control Autom. Syst.*, vol. 6, nr. 2, pp. 204-213, apr. 2008.
- [64] G. Welch și G. Bishop, „An Introduction to the Kalman Filter”, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.
- [65] M. Akhlaq și T. R. Sheltami, „RTSP: An Accurate and Energy-Efficient Protocol for Clock Synchronization in WSNs”, *Instrum. Meas. IEEE Trans. On*, vol. 62, pp. 578-589, mar. 2013.
- [66] F. Gustafsson și F. Gunnarsson, „Positioning using time-difference of arrival measurements”, în *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, 2003, vol. 6, pp. VI-553-6 vol.6.
- [67] J. Bachrach și C. Taylor, „Localization in Sensor Networks”, în *Handbook of Sensor Networks*, John Wiley & Sons, Inc., 2005, pp. 277-310.
- [68] P. V. Nikitin, R. Martinez, S. Ramamurthy, H. Leland, G. Spiess, și K. V. S. Rao, „Phase based spatial identification of UHF RFID tags”, în *2010 IEEE International Conference on RFID (IEEE RFID 2010)*, 2010, pp. 102-109.
- [69] M. V. Micea, *Prelucrarea numerică a semnalelor în domeniul timp*. Timișoara: Editura Politehnica, 2009.
- [70] A. Bry, M. Cavolowsky, M. Donahoe, A. Roldan, și B. Bingham, „The P.O.E. System and The Crickets”, Olin College of Engineering, 2006.
- [71] A. Suzuki, K. Kumakura, Y. Choi, și T. Iyota, „Accuracy of distance measurements using signal tracking of spread-spectrum ultrasonic waves with CDMA”, în *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2014, pp. 575-581.
- [72] S. Fuicu, M. Marcu, B. Stratulat, și A. Gîrban, „Effectiveness and Accuracy of Wireless Positioning System”, *WSEAS Trans. Comput.*, vol. 8, nr. 1, pp. 1471-1483, ian. 2009.
- [73] J. Blumenthal, R. Grossmann, F. Golasowski, și D. Timmermann, „Weighted Centroid Localization in Zigbee-based Sensor Networks”, în *2007 IEEE International Symposium on Intelligent Signal Processing*, 2007, pp. 1-6.

- [74] P. Pivato, L. Fontana, L. Palopoli, și D. Petri, „Experimental assessment of a RSS-based localization algorithm in indoor environment”, prezentat la Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE, 2010.
- [75] P. Torteeka, X. Chundi, și Y. Dongkai, „Hybrid technique for indoor positioning system based on Wi-Fi received signal strength indication”, în *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2014, pp. 48-57.
- [76] A. R. Jiménez și F. Seco, „Comparing Decawave and Bespoon UWB location systems: Indoor/outdoor performance analysis”, în *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1-8.
- [77] D. D. Wentzloff, „Pulse-Based Ultra-ideband Transmitters for Digital Communication”, Teză de doctorat, Massachusetts Institute of Technology, Massachusetts, 2007.
- [78] Álvaro Gutiérrez, Elio Tuci, și Alexandre Campo, „Evolution of Neuro-Controllers for Robots' Alignment using Local Communication”, *Int. J. Adv. Robot. Syst.*, vol. 6, nr. 1, p. 6, ian. 2009.
- [79] A. E. Turgut, H. Çelikkanat, F. Gökçe, și E. Şahin, „Self-organized flocking in mobile robot swarms”, *Swarm Intell.*, vol. 2, nr. 2, pp. 97-120, dec. 2008.
- [80] H. Celikkanat, „Optimization of self-organized flocking of a robot swarm via evolutionary strategies”, în *2008 23rd International Symposium on Computer and Information Sciences*, 2008, pp. 1-4.
- [81] F. Zafari, A. Gkelias, și K. K. Leung, „A Survey of Indoor Localization Systems and Technologies”, *CoRR*, vol. abs/1709.01015, 2017.
- [82] C. Medina, J. C. Segura, și A. De la Torre, „A Synchronous TDMA Ultrasonic TOF Measurement System for Low-Power Wireless Sensor Networks”, *Instrum. Meas. IEEE Trans. On*, vol. 62, pp. 599-611, mar. 2013.
- [83] S. Latif, R. Tariq, W. Haq, și U. Hashmi, „INDOOR POSITIONING SYSTEM using ultrasonics”, prezentat la Applied Sciences and Technology (IBCAST), 2012 9th International Bhurban Conference on, 2012.
- [84] L. E. Navarro-Serment, R. Grabowski, C. J. J. Paredis, și P. K. Khosla, „Millibots”, *Robot. Autom. Mag. IEEE*, vol. 9, pp. 31-40, dec. 2002.
- [85] H. B. Brown, J. M. Vande Weche, C. A. Bererton, și P. K. Khosla, „Millibot trains for enhanced mobility”, *Mechatron. IEEEASME Trans. On*, vol. 7, pp. 452-461, dec. 2002.
- [86] F. Franceschini, D. Maisano, L. Mastrogiacomo, și B. Pralio, „Ultrasound Transducers for Large-Scale Metrology: A Performance Analysis for Their Use by the MScMS”, *Instrum. Meas. IEEE Trans. On*, vol. 59, pp. 110-121, ian. 2010.
- [87] R. Tinos, L. E. Navarro-Serment, și C. J. J. Peredis, „Fault tolerant localization for teams of distributed robots”, prezentat la Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on, 2001.
- [88] S. Shoval și J. Borenstein, „Measurement of Angular Position of a Mobile Robot Using Ultrasonic Sensors”, prezentat la ANS Conference on Robotics and Remote Systems, 1999.
- [89] S. Shoval și J. Borenstein, „Measuring The Relative Position And Orientation Between Two Mobile Robots With Binaural Sonar”, prezentat la ANS 9th International Topical Meeting on Robotics and Remote Systems, 2001.

- [90] F. Figueroa și E. Barbieri, „An ultrasonic ranging system for structural vibration measurements”, *Instrum. Meas. IEEE Trans. On*, vol. 40, pp. 764-769, aug. 1991.
- [91] F. Montes-Gonzalez, D. Flandes-Eusebio, și L. Pellegrin-Zazueta, „Action Selection and Obstacle Avoidance using Ultrasonic and Infrared Sensors”, în *Frontiers in Evolutionary Robotics*, InTech, 2008.
- [92] F. Mondada, E. Franzi, și A. Guignard, „The Development of Khepera”, prezentat la Experiments with the Mini-Robot Khepera, Proceedings of the First International Khepera Workshop, 1999.
- [93] F. Montes-Gonzalez, J. Velasquez-Matus, F. Aldana-Franco, și R. Palacios, „The Development of an Ultrasonic Turret Extension for the Khepera Robot to Avoid Legged Object”, în *Introduction to Modern Robotics*, iConcept Press, 2013.
- [94] G. Oberholzer, P. Sommer, și R. Wattenhofer, „SpiderBat: Augmenting wireless sensor networks with distance and angle information”, prezentat la Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on, 2011.
- [95] B. Kusy, P. Dutta, P. Levis, M. Maroti, A. Ledeczi, și D. Culler, „Elapsed Time on Arrival: A simple and versatile primitive for canonical time synchronization services”, *Int J Ad Hoc Ubiquitous Comput*, vol. 1, pp. 239-251, 2006.
- [96] A. Stancovici, M. V. Micea, V. Crețu, și V. Groza, „Relative positioning system using Inter-Robot Ultrasonic Localization Turret”, în *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, 2014, pp. 1427-1430.
- [97] M. V. Micea, A. Stancovici, și V. I. Crețu, „Sistem și metodă pentru orientare și localizare relativă a unor subsisteme autonome”, RO 129802 A0, 30-sep-2014.
- [98] M. V. Micea, A. Stancovici, și I. Sinziana, „Distance Measurement for Indoor Robotic Collectives”, în *Mobile Robots / Book 2*, InTech, 2011.
- [99] M. Parrilla, J. J. Anaya, și C. Fritsch, „Digital signal processing techniques for high accuracy ultrasonic range measurements”, *Instrum. Meas. IEEE Trans. On*, vol. 40, pp. 759-763, aug. 1991.
- [100] D. Marioli, C. Narduzzi, C. Offelli, D. Petri, E. Sardini, și A. Taroni, „Digital time-of-flight measurement for ultrasonic sensors”, *Instrum. Meas. IEEE Trans. On*, vol. 41, pp. 93-97, feb. 1992.
- [101] A. Stancovici, S. Indreica, M. V. Micea, V. Crețu, și V. Groza, „Relative localization methodology for autonomous robots in collaborative environments”, în *2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2013, pp. 1730-1733.
- [102] S. Indreica, A. Stancovici, M. V. Micea, V. Crețu, și V. Groza, „Simulator based study of robot alignment and localization”, în *2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, 2013, pp. 107-111.
- [103] D. Swanson și R. Stejskal, „Back-EMF method detects stepper-motor stall”, STMicroelectronics, ian. 2012.
- [104] M. V. Micea, A. Stancovici, D. Chiciudean, și C. Filote, „Indoor Inter-Robot Distance Measurement in Collaborative Systems”, *Adv. Electr. Comput. Eng.*, vol. 10, pp. 21-26, aug. 2010.

- [105] L. Yuanqing, P. Vernaza, H. Jihun, și D. D. Lee, „Cooperative relative robot localization with audible acoustic sensing”, prezentat la Intelligent Robots and Systems, IROS 2005, IEEE/RSJ International Conference on, 2005.
- [106] R. Kurazume, S. Hirose, S. Nagata, și N. Sashida, „Study on cooperative positioning system (basic principle and measurement experiment)”, în *Proceedings of IEEE International Conference on Robotics and Automation*, 1996, vol. 2, pp. 1421-1426 vol.2.
- [107] I. Rekleitis, G. Dudek, și E. Miliot, „Multi-robot collaboration for robust exploration”, în *Annals of Mathematics and Artificial Intelligence*, Springer International Publishing, 2001, pp. 7-40.
- [108] S. Y. Chen, „Kalman Filter for Robot Vision: A Survey”, *IEEE Trans. Ind. Electron.*, vol. 59, pp. 4409-4420, 2012.
- [109] P. Goel, S. I. Roumeliotis, și G. S. Sukhatme, „Robust localization using relative and absolute position estimates”, prezentat la Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289), 1999, vol. 2, pp. 1134-1140 vol.2.
- [110] J. Yao, D. Di, G. Jiang, S. Gao, și H. Yan, „Real-Time Acceleration Harmonics Estimation for an Electro-Hydraulic Servo Shaking Table Using Kalman Filter With a Linear Model”, *IEEE Trans. Control Syst. Technol.*, vol. 22, pp. 794-800, 2014.
- [111] S. Boyd, „Lecture 8 - The Kalman filter”, Stanford University, 2008.
- [112] S. O. Haykin, *Adaptive Filter Theory (4th Edition)*. Prentice Hall, 2001.
- [113] H. Cho și S. W. Kim, „Mobile Robot Localization Using Biased Chirp-Spread-Spectrum Ranging”, *IEEE Trans. Ind. Electron.*, vol. 57, pp. 2826-2835, 2010.
- [114] D. Fox, J. Hightower, L. Liao, D. Schulz, și G. Borriello, „Bayesian Filtering for Location Estimation”, *IEEE Pervasive Comput.*, vol. 2, pp. 10-19, 2003.
- [115] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, și A. A. F. Loureiro, „Chapter 12 - Localization systems for wireless sensor networks”, în *Algorithms and protocols for wireless sensor networks*, Hoboken, New Jersey: John Wiley & Sons, Inc., 2009, pp. 307-340.
- [116] G. H. Golub și C. F. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins University Press, 1996.
- [117] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. New York: John Wiley & Sons, 1991.
- [118] I. Biton, M. Koifman, și I. Y. Bar-Itzhack, „Improved Direct Solution of the Global Positioning System Equation”, *J. Guid. Control Dyn.*, vol. 21, pp. 45-49, 1998.
- [119] S. Y. Cho și Y. W. Choi, „Access point-less wireless location method based on peer-to-peer ranging of impulse radio ultra-wideband”, *IET Radar Sonar Navig.*, vol. 4, pp. 733-743, 2010.
- [120] O. Lipovan, *Analiză matematică - Calculul diferențial*. Timișoara: Editura Politehnica, 2004.

- [121] A. Stancovici, M. V. Micea, și V. Cretu, „Cooperative positioning system for indoor surveillance applications”, în *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1-7.
- [122] D. Medeiros, „DWSIM - Process Simulation, Modeling and Optimization Source Code - Matrix Operations - cMatLib.vb”. [Online]. Valabil la: <https://github.com/DanWBR/dwsim5/blob/master/DWSIM.Math/cMatLib.vb>. [Data accesării: 03-iun-2018].
- [123] T. Theodoridis și H. Huosheng, „Toward Intelligent Security Robots: A Survey”, *IEEE Trans. Syst. Man Cybern. Part C Cybern. TSMC11*, vol. 42, pp. 1219–1230, 2012.
- [124] B. Gaspers, J. Welle, și D. Schulz, „Opening doors with a mobile manipulator without force-torque feedback”, prezentat la Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on, 2013.
- [125] R. P. Rocha, D. Portugal, M. Couceiro, F. Araujo, P. Menezes, și J. Lobo, „The CHOPIN project: Cooperation between human and robotic teams in catastrophic incidents”, prezentat la Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on, 2013.
- [126] R. Salman, I. Willms, T. Sakamoto, T. Sato, și A. Yarovoy, „Environmental imaging with a mobile UWB security robot for indoor localisation and positioning applications”, prezentat la Microwave Conference (EuMC), 2013 European, 2013.
- [127] A. Amanatiadis *et al.*, „The AVERT project: Autonomous Vehicle Emergency Recovery Tool”, prezentat la Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on, 2013.
- [128] L. Babai, „Monte Carlo algorithms in graph isomorphism testing”, Universite de Montreal, Canada, Technical Report D.M.S. No. 79-10, 1979.
- [129] D. Fox, W. Burgard, H. Kruppa, și S. Thrun, „Collaborative Multi-Robot Localization”, prezentat la Proc. of the German Conference on Artificial Intelligence (KI), Germany, Germany, 1999.
- [130] I. Rekleitis, „A Particle Filter Tutorial for Mobile Robot Localization”, McGill University, Montreal, Quebec, Canada, Technical Report TR-CIM-04-02, 2004.
- [131] S. Banerjee, „Improving Accuracy in Ultra-Wideband Indoor Position Tracking through Noise Modeling and Augmentation”, (2012). All Dissertations. 1066. https://tigerprints.clemson.edu/all_dissertations/1066.
- [132] F. Dellaert, D. Fox, W. Burgard, și S. Thrun, „Monte Carlo localization for mobile robots”, în *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, 1999, vol. 2, pp. 1322-1328 vol.2.
- [133] D. Fox, W. Burgard, și S. Thrun, „Markov Localization for Mobile Robots in Dynamic Environments”, *J Artif Int Res*, vol. 11, nr. 1, pp. 391–427, iul. 1999.
- [134] V. I. Cretu, *Structuri de date și algoritmi. Structuri de date fundamentale*, vol. 1. Timisoara: Orizonturi Universitare, 2000.
- [135] R. Motwani și P. Raghavan, *Randomized Algorithms*, First Edition. Cambridge University Press, 1995.
- [136] H. Edelsbrunner, *Design and Analysis of Algorithms*. Duke University, North Carolina, USA, 2008.
- [137] M. T. Goodrich și R. Tamassia, *Algorithm Design and Applications*. Willey, 2014.

- [138] D. R. Karger, „Randomization in Graph Optimization Problems: A Survey”, *Optima*, pp. 1-11, iun. 1998.
- [139] Z. Yunlong, X. Songdong, Z. Jianchao, și D. Jing, „Target Position Estimation Aided Swarm Robotic Search under Conditions of Relative Localization Mechanism”, prezentat la Computing, Measurement, Control and Sensor Network (CMCSN), 2012 International Conference on, 2012.
- [140] J. Pugh și A. Martinoli, „Relative Localization and Communication Module for Small-Scale Multi-Robot Systems”, prezentat la Robotics and Automation, ICRA 2006, Proceedings 2006 IEEE International Conference on, 2006.
- [141] W. C. Stone, „Electromagnetic Signal Attenuation in Construction Materials”, National Institute of Standards and Technology, Gaithersburg, Maryland, USA, NIST Construction Automation Program 3, oct. 1997.

PUBLICAȚII

A. Lucrări științifice publicate în reviste indexate ISI

1. M. V. Micea, **A. Stancovici**, D. Chiciudean, C. Filote, "Indoor Inter-Robot Distance Measurement in Collaborative Systems" *Advances in Electrical and Computer Engineering*, vol. 10, issue 3, Suceava, Romania, pp. 21-26, Aug. 2010.

B. Patente indexate ISI

1. M. G. Marcu, **A. Stancovici**, V. Stângaciu, C. Stângaciu, A. Topîrceanu, B. Popescu, D. Volcinschi, S. O. Fuicu, „System for measuring and analyzing the energy consumption in electric devices by using consumption signatures”, Patent pending for invention, Patent Number RO127698-A0, 30 Jul 2012.

2. M. V. Micea, **A. Stancovici**, V. Cretu, „System and Method for Orientation and Relative Localization of Autonomous Subsystems”, Patent pending for invention, Patent Number RO129802-A0, 30 Sept 2014.

C. Lucrări științifice publicate în volumele unor manifestări științifice (Proceedings) indexate ISI

1. B. Popescu, **A. Stancovici**, V. Stangaciu, C. Certejan, M. Marcu, „Intelligent Wireless Distributed Network for Power Consumption Monitoring and Analysis”, 2009 IEEE Toronto International Conference – Science and Technology for Humanity, TIC-STH 2009, Toronto, Canada, pp.730-735, Sep. 2009.

2. M. Marcu, **A. Stancovici**, „System classification based on power signatures”, Innovative Smart Grid Technologies (ISGT Europe), 2011 2nd IEEE PES International Conference and Exhibition on, Manchester, Dec. 2011.

3. **A. Stancovici**, S. Indreica, M. V. Micea, V. Cretu, V. Groza, "Relative Localization Methodology for Autonomous Robots in Collaborative Environments", International Instrumentation and Measurement Technology Conference (I2MTC), 2013 IEEE, pp. 1730 – 1733, Minneapolis, USA, May 2013.

4. S. Indreica, **A. Stancovici**, M. V. Micea, V. Cretu, V. Groza, „Simulator Based Study of Robot Alignment and Localization”, 2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE), Washington, USA, October 2013.

5. **A. Stancovici**, M. V. Micea, V. Cretu, V. Groza, "Relative Positioning system using Inter-Robot Ultrasonic Localization Turret", International Instrumentation and Measurement Technology Conference (I2MTC), 2014 IEEE, pp. 1427–1430, Montevideo, Uruguay, May 2014.

6. **A. Stancovici**, M. V. Micea, V. Cretu, "Cooperative Positioning System for Indoor Surveillance Applications", International Conference on Indoor Positioning and Indoor Navigation, IPIN 2016, Alcalá de Henares, Madrid, Spain, October 2016.

D. Lucrări științifice publicate în volumele unor manifestări științifice (Proceedings) indexate BDI

1. D. Tudor, **A. Stancovici**, B. Popescu, G. Reisinger, V. Cretu, "Zombie: a Home Automation Prototype for Retrofitted Environments", International Conference on Pervasive Technologies Related to Assistive Environments (PETRA), Corfu, Greece, 2009.
2. **A. Stancovici**, O.-C. Szanto, V.-C. Ardelean, R.-G. Barbatei, M. Marcu, „IQube: a cube for learning”, International Games Innovation Conference, IGIC 2011, California, USA, 2011.

E. Lucrări științifice publicate în volumele unor manifestări științifice

1. B. Popescu, **A. Stancovici**, O. Szanto, S. Fuicu, M. Marcu, "Intelligent measuring and control system for honeybees' life support", Proceedings of the 8th International Conference on Technical Informatics, CONTI 2008, Timisoara, Romania, pp.45-50, Jun. 2008.
2. M. Marcu, B. Popescu, **A. Stancovici**, V. Stangaciu, C. Certejan, "Power Characterization of Electric, Electronic and Computing Devices", Scientific Bulletin of the Electrical Engineering Faculty, 10 (1), pp. 99-106, 2009.

F. Capitle de cărți publicate în străinătate

1. M. V. Micea, **A. Stancovici**, S. Indreica, "Distance Measurement for Indoor Robotic Collectives", in Mobile Robots - Control Architectures, Bio-Interfacing, Navigation, Multi Robot Motion Planning and Operator Training, section 16, Janusz Bedkowski (Eds.), InTech, Rijeka, Croatia, pp. 353-372, Dec. 2011, ISBN 978-953-307-842-7.