Universitat de Girona

LOCAL MODE PREDICTIVE CONTROL FOR NAVIGATION OF A WHEELED MOBILE ROBOT USING MONOCULAR INFORMATION

Lluís PACHECO VALLS

ISBN: 978-84-692-9877-0
Dipòsit legal: GI-1492-2009
http://www.tesisenxarxa.net/TDX-1222109-123408/

ADVERTIMENT. La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del servei TDX (www.tesisenxarxa.net) ha estat autoritzada pels titulars dels drets de propietat intel·lectual únicament per a usos privats emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA. La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del servicio TDR (www.tesisenred.net) ha sido autorizada por los titulares de los derechos de propiedad intelectual únicamente para usos privados enmarcados en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio TDR. No se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING. On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the TDX (www.tesisenxarxa.net) service has been authorized by the titular of the intellectual property rights only for private uses placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading and availability from a site foreign to the TDX service. Introducing its content in a window or frame foreign to the TDX service is not authorized (framing). This rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author.

UdG

UNIVERSITAT DE GIRONA
Institute of Informatics and Applications

Doctoral Thesis

Local Model Predictive Control for Navigation of a Wheeled Mobile Robot Using Monocular Information

Lluís Pacheco Valls

Advisors:

Dr. Ningsu Luo
Dr. Xevi Cufí

Girona, June 2009

PhD Program in Information Technology

Local Model Predictive Control for Navigation of a Wheeled Mobile Robot Using Monocular Information

List of Figures vii

List of Tables xv

List of Acronyms and Abbreviations xvii

1 Introduction 1

2 WMR Navigation Using Computer Vision

2.1 Introduction 5

2.2 Indoor navigation 5

2.2.1 Navigation with maps 5

2.2.2 Navigation based on map creation 7

2.2.3 Navigation without maps …..……………………………... 8

2.3 Outdoor navigation 9

2.3.1 Structured outdoor navigation 9

2.3.2 Unstructured outdoor navigation 9

2.3.3 Light and shadow in outdoor navigation ...…...………… 10

2.4 The proposed basic system 11

2.4.2 WMR sensorial system suggestions …………………… 11

2.4.1 Proposed navigation methods …………………………… 12

3 3D Machine Vision Techniques

3.1 Introduction 15

3.1.1 Stereo vision 15
3.1.2 Shape from X 15
3.1.2.1 Shape from intensity 15
3.1.2.2 Shape from motion 16
3.1.2.3 Shape from focus and defocus 16
3.1.3 Monocular 3D scene analysis 16
3.2 Optical flow methods 17
3.2.1 Field movement analysis and optical Flow 17
3.2.2 Computing optical flow using BCM: the aperture problem 18
3.2.3 Smoothness constraints 19
3.2.4 A short state of the art review 19
3.3 DFF methods 20
3.3.1 The optical transfer function (OTF) as a focus measure . 20
3.3.2 Focus measure operators 21
3.3.3 Statistical methods in the focus measure quality 22
3.3.4 3D scene perception 23
3.4 3D recovery by DFD methods 24
3.4.1 Depth information using two images 24
3.4.2 The camera optics 25
3.5 First monocular perception proposal 26

4 Model Based Predictive Control Strategies

4.1 Introduction 27
4.2 The system identification 27
4.2.1 Linear difference equation-ARX models 28
4.2.1.1 Linear least square method 28

4.2.2 Experiment design 28
4.2.3 Software for system identification 29
4.3 Model predictive control 30
4.3.1 The model predictive control basic ideas 30
4.3.1.1 Advantatges and disadvantatges of MPC 31
4.3.1.2 The MPC algorithms 31
4.3.1.3 The control horizon 32
4.3.1.4 The input and output trajectories 33
4.3.2 Some aspects of the MPC academic research 34
4.3.3 The MPC for WMR control 35
4.4 Model predictive control proposal 36

5 WMR Platform Description

5.1 Introduction 37
5.2 Mechanical description 38
5.3 Electrical description 39
5.4 System architecture 41
5.4.1 The proposed system architecture ………..……………….. 41
5.4.2 The proposed software support tools ……………….….. 42

6 The Developed Monocular Machine Vision System

6.1 Introduction ..……………………………………………..…..... 43
6.2 3D reconstruction by DFF methods …………………….............. 43
6.2.1 Calibration of the optics parameters ….….……………... 43
6.2.2 Focus measure and 3D reconstruction 46
6.3 Constrained monocular obstacle perception with just one frame ……………………..….. 48

- **6.3.1 Monocular perception algorithms 48**
- **6.3.2 Experimental results 49**
- **6.3.3 Conclusions 54**

6.4 Improving the obstacle detection and optical Flow estimation 54

- **6.4.1 The local optical flow methodology 54**
- **6.4.2 The computer vision proposed algorithms 56**

6.5 The local planar monocular occupancy grid framework 57

- **6.5.1 The bayesian occupancy grid proposal 58**

6.6 Conclusions and future work 60

7 WMR Experimental Model and Local MPC trajectory-tracking

7.1 Introduction 61

7.2 The kinematic system 63

7.3 The WMR dynamic models 64

- **7.3.1 The system identification 64**
- **7.3.2 The reduced system models 66**

7.4 The dynamic MPC techniques for local trajectory-tracking 68

- **7.4.1 The LMPC formulation 68**
- **7.4.2 The LMPC algorithms and simulation results 69**
- **7.4.3 Trajectory-tracking simulation results 73**
  - **7.4.3.1 Trajectory-tracking results using the trajectory distance factor 74**
  - **7.4.3.2 Trajectory-tracking results using the orientacion distance Factor 81**

7.5 Trajectory-tracking experimental results: tuning the cost function parameters by factorial design 88

7.5.1 The local field of perception and the attraction potential fields 88

7.5.2 The trajectory-tracking experimental approach by using LMPC methods 89

7.5.3 Velocity estimation trough Kalman filter 91

7.5.4 Experimental tuning of APD and TDD factors 92

7.5.5 Experimental time improvements by using APD and TDD trajectory-flexible factors 99

7.5.6 Experimental tuning of APD and OD factors 102

7.5.7 Experimental time improvements by using APD and OD trajectory-flexible factors 109

7.6 Conclusions 113

8 Monocular Local Navigation Results

8.1 Introduction 115

8.2 The local navigation framework 115

8.2.1 The perception horizon 116

8.2.2 The local attraction potential fields 118

8.3 The system of navigation 119

8.3.1 The field of view and the local grid size 119

8.3.2 The navigation algorithms 121

8.4 Experimental results 125

8.4.1 The time-integration of the frames within the local occupancy grid 125

8.4.2 Navigation experimental results 128

8.4.2.1 First navigation experiment: static obstacle avoidance 129

8.4.2.2 Second navigation experiment: straight and left turning corridor 133

8.4.2.3 Third navigation experiment: straight, left and right turning, and straight corridor 138

8.5 Conclusions …………………………...……………………………….. 146

9 Conclusions

9.1 Introduction ..…………………………...…………………. 149
9.2 Contributions ……………………………....……………... 149
9.3 Publications ………………………………….....………….. 151
9.4 Future Work ………………………….....………………. 153
Bibliography 155

List of Figures

Figure 2.1: The proposed navigation scheme.

Figure 3.1: Two point projection relationship.

Figure 3.2: Optical flow and gradient directions.

Figure 3.3: PSF Hankel transform.

Figure 3.4: The high frequency components can produce lateral lobules.

Figure 3.5: The shape from focus concept.

Figure 3.6: NCP are used to interpolate a surface.

Figure 3.7: The Pentland camera model.

Figura 3.8: The Subbarao camera models.

Figure 3.9: The Hiura multifocus camera implementation.

Figure 4.1: MATLAB Graphical interface for System Identification.

Figure 4.2: The MPC strategy.

Figure 4.3: Multiple move option.

Figure 4.4: Single move option.

Figure 4.5: Future input parameterization.

Figure 4.6: Set-point option.

Figure 4.7: Zone option strategy.

Figure 4.8: Reference trajectory option.

Figure 4.9: Funnel strategy.

Figure 5.1: The picture shows the PRIM I that is mainly used as teaching and research activities platform. The picture on the right shows the platform PRIM II that was designed with the aim of to be used in social events as multimedia information platform.

Figure 5.2: The mobile platform PRIM I at 18th IAVE in Barcelona.

Figure 5.3: The basic hardware structure of the robots.

Figure 5.4: The system architecture.

Figure 5.5: Software tools and system architecture.

Figure 6.1: Optical model of the camera.

Figure 6.2: Field of view of the camera.

Figure 6.3: Transfer function Zoom-Focal length.

Figure 6.4: Shape from Focus method.

Figure 6.5: Field of view and 3D perception of the work area.

Figure 6.6: Field of view and 3D perception using active vision.

Figure 6.7: The robot PRIM, and the on-robot monocular camera configuration.

Figure 6.8: Typical indoor scenario.

Figure 6.9: Local perception grid coordinates relative to the robot position, at the 96x72 and 9x7 size level. At the 9x7 size level it is possible to distinguish 7 different rows (Row 1 being the closest to the robot).

Figure 6.10: Fragments of high resolution floor images (768x576 pixels) under different light conditions corresponding to 200, 800, 1400, and 2000 lx, respectively.

Figure 6.11: Typical floor scenario acquired at 400 lx, represented in a 96x72 image size, averaged 9x7 modified Laplacian and standard deviation values.

Figure 6.12: Typical floor scenario, with an object without texture acquired at 400 lx, represented in a 96x72 image size, averaged 9x7 modified Laplacian and standard deviation values.

Figure 6.13: Obstacle acquired at 180 lx, represented in a 96x72 image size, averaged 9x7 modified Laplacian and standard deviation values.

Figure 6.14: False obstacle caused by a flat shape acquired at 1400 lx, represented in a 96x72 image size, averaged 9x7 modified Laplacian and standard deviation values.

Figure 6.15: False obstacle caused by light reflection acquired at 200 lx, represented in a 96x72 image size, averaged 9x7 modified Laplacian and standard deviation values.

Figure 6.16: Camera system producing an image that is a perspective projection of the world.

Figure 6.17: Magnification changes of the floor model by considering the optical axis ray as the unit of magnification.

Figure 6.18: a) Image with obstacles, 96x72; b) Modified Laplacian measures; c) 9x7 Modified Laplacian mean values; d)) 9x7 standard deviation mean.

Figure 6.19: In 6.19.a$_1$, 6.19.b$_1$, 6.19.c$_1$ and 6.19.d$_1$ are shown 4 images corresponding to floor marks acquired from different positions that have been taken in advancing sense. The binary results are shown in 6.19.a$_2$, 6.19.b$_2$, 6.19.c$_2$ and 6.19.d$_2$. The probability of being floor is depicted in sequences 6.19.a$_3$, 6.19.b$_3$, 6.19.c$_3$ and 6.19.d$_3$, and 6.19.a$_4$, 6.19.b$_4$, 6.19.c$_4$ and 6.19.d$_4$.

Figure 6.20: It is shown the occupancy results when an obstacle is met. Figure. 6.20.a1, b1, c1, and d1 shows respectively the first image, the binary results, and the occupancy grid 2D and 3D probabilty results. Figure 6.20.a2, b2, c2, and d2, shows the corresponding results obtained when the second acquired image is overlapped into the local occupancy grid framework.

Figure 7.1: Positioning of robot as function of the angular movement of each wheel.

Figure 7.2: Structure of MIMO system.

Figure 7.3: Left speed Output for a left PRBS input signal.

Figure 7.4: Coupled effects at the left wheel for medium speeds.

Figure 7.5: Different gain for slow speeds for the same consigns.

Figure 7.6: Experimental and model data for medium speeds.

Figure 7.7: Subinterval search.

Figure 7.8: Trajectory tracking and command speed profiles during 2 m straight line tracking simulation, by computing the cost function for all the input combinations or by using the gradient descent method.

Figure 7.9: Two interval search grids.

Figure 7.10: Predicted coordinates from speed zero, n=5 and m=3; where n and m are the prediction and control horizons.

Figure 7.11: LMPC simulated results following a 45° trajectory.

Figure 7.12: Predicted coordinates from speed zero, n=10 and m=5.

Figure 7.13: Trajectory tracking in red (n=10 and m=5), and in blue (n=5 and m=3). The larger prediction horizon shows a closer final point achievement and a worse trajectory tracking.

Figure 7.14: Wheel speeds during straight line tracking of 2 meters. The red and blue dots respectively show the right and left speeds (with n=10 and m=5). The magenta and green dot lines depict the right and left speeds (with n=5 and m=3).

Figure 7.15 Trajectory tracking simulated results for a two meters straight line by using optimal search method. The parameters used in the cost function are the final point distance and trajectory distance.

Figure 7.16: Trajectory-tracking simulated results for a two meters straight line by using gradient interval descent search method. The parameters used in the cost function are the final point distance and trajectory distance.

Figure 7.17: Trajectory-tracking simulated results by using optimal search method. The trajectory is composed by the "*xy*" coordinate, (0, 0), (0, 100) and (-50, 200), given in

cm.The parameters used in the cost function are the final point distance and trajectory distance.

Figure 7.18: Trajectory-tracking simulated results by using gradient descent method. The trajectory is composed by the “*xy*” coordinates, (0, 0), (0, 100) and (-50, 200), given in cm.The parameters used in the cost function are the final point distance and trajectory distance.

Figure 7.19: Trajectory-tracking simulated results by using optimal search method. The trajectory is composed by the “*xy*” coordinates, (0, 0), (25, 50), (25, 100), (0, 150), and (0, 200), given in cm.The parameters used in the cost function are the final point distance and trajectory distance.

Figure 7.20: Trajectory-tracking simulated results by using gradient descent method. The trajectory is composed by the “*xy*” coordinates, (0, 0), (25, 50), (25, 100), (0, 150), and (0, 200), given in cm. The parameters used in the cost function are the final point distance and trajectory distance.

Figure 7.21: Trajectory-tracking simulated results by using optimal search method. The trajectory is composed by the “*xy*” coordinates, (0, 0) and (0, 200), given in cm. The parameters used in the cost function are the final point distance and orientation deviation factors.

Figure 7.22: Trajectory-tracking simulated results by using gradiend descent search method. The trajectory is composed by the “*xy*” coordinates, (0, 0) and (0, 200), given in cm. The parameters used in the cost function are the final point distance and orientation deviation factors.

Figure 7.23: Trajectory-tracking simulated results by using optimal search method. The trajectory is composed by the “*xy*” coordinates, (0, 0), (0, 100), and (-50, 200), given in cm. The parameters used in the cost function are the final point distance and orientation deviation factors.

Figure 7.24: Trajectory-tracking simulated results by using gradiend descent search method. The trajectory is composed by the “*xy*” coordinates, (0, 0), (0, 100), and (-50, 200), given in cm. The parameters used in the cost function are the final point distance and orientation deviation factors.

Figure 7.25: Trajectory-tracking simulated results by using optimal search method. The trajectory is composed by the “*xy*” coordinates, (0, 0), (25, 50), (25, 100), (0, 150), and (0, 200), given in cm. The parameters used in the cost function are the final point distance and orientation deviation factors.

Figure 7.26: Trajectory-tracking simulated results by using the gradiend descent search method. The trajectory is composed by the “*xy*” coordinates, (0, 0), (25, 50), (25, 100), (0, 150), and (0, 200), given in cm. The parameters used in the cost function are the final point distance and orientation deviation factors.

Figure 7.27: Floor local perception grid coordinates relative to the robot position, at the 96x72 size level.

Figure 7.28: Trajectory tracking tested from point to point by using the available local map coordinates provided by the monocular perception system.

Figure 7.29: Experimental results of the Kalman filter.

Figure 7.30: The different factor combinations and the influence directions, in which the performances should be analysed.

Figure 7.31: (a) Trajectory-tracking experimental results by using flexible or fixed cost function. (b) WMR orientation experimental results by using flexible or fixed cost function. (c) Left wheel speed results by using flexible or fixed cost function. (d) Right wheel speed results by using flexible or fixed cost function.

Figure 7.32: (a) Trajectory-tracking experimental results by using flexible or fixed cost function. (b) WMR orientation experimental results by using flexible or fixed cost function. (c) Left wheel speed results by using flexible or fixed cost function. (d) Right wheel speed resultsby using flexible or fixed cost function.

Figure 7.33: (a) Trajectory-tracking experimental results by using flexible or fixed cost function. (b) WMR orientation experimental results by using flexible or fixed cost function. (c) Left wheel speed results by using flexible or fixed cost function. (d) Right wheel speed results by using flexible or fixed cost function.

Figure 7.34: (a) Trajectory-tracking experimental results by using flexible or fixed cost function. (b) WMR orientation experimental results by using flexible or fixed cost function. (c) Left wheel speed results by using flexible or fixed cost function. (d) Right wheel speed results by using flexible or fixed cost function.

Figure 7.35: (a) Trajectory-tracking experimental results by using flexible or fixed cost function. (b) WMR orientation experimental results by using flexible or fixed cost function. (c) Left wheel speed results by using flexible or fixed cost function. (d) Right wheel speed results by using flexible or fixed cost function.

Figure 8.1: Fixed camera configuration and pose (angles α=37º, β=48º and φ=32º,), H=109cm.

Figure 8.2: Example of local visual perception using 96x72 or 9x7 grids.

Figure 8.3: Unsafe trajectories, planned within local grid, arise when is not considered the heuristic WP concept.

Figure 8.4: The trajectories should include, safety reactive distances, WP, and necessary free of obstacles zone for the next acquired frame.

Figure 8.5: Two different desired points *Ad* and *Bd*; *Ald* and *Bld* are computed by using Euclidean and the angular distances.

Figure 8.6: WP concept is considered as a necessary distance in order to plan safety trajectories within the local occupancy grid.

Figure 8.7: WMR camera configuration set-up and floor field of view.

Figure 8.8: Local occupancy grid dimensions and the monocular visual perception obtained.

Figure 8.9: Compressed image, 96x72, of a typical indoor scenario.

Figure 8.10: Occupancy grid corresponding to the image of the Figure 8.9. The rectangular free zone, depicted in blue, is the necessary free dead zone of the first acquired frame.

Figure 8.11: The algorithms used for searching the meaningful obstacle vertexes.

Figure 8.12: Meaningful vertexes considered when right or left obstacles are met.

Figure 8.13: The basic navigation algorithms.

Figure 8.14: It is shown the reactive turning and down vertex necessary constraints when a new obstacle avoidance strategy should be faced.

Figure 8.15: The frames acquired at positions (x, y, θ): (a) (0, 0, 90º); (c) (0, 21cm, 90.5º); (f) (0, 41cm, 90º); (i) (0, 62cm, 89.5º). The occupancy grid predictions and the fusions with the acquired frames are also depicted.

Figure 8.16: The frames acquired at positions (x, y, θ): (a) (0, 0, 90º); (c) (0, 46cm, 89º); (f) (1cm, 92cm, 88º); (i) (2cm, 137cm, 88º). The occupancy grid predictions and the fusions with the acquired frames are also depicted.

Figure 8.17: The frames acquired at positions (x, y, θ): (a) (0, 0, 90º); (c) (0, 71cm, 88º). The prediction of (b) referred to the coordinates of (c) is shown in (d). The fusion is shown in (e).

Figure 8.18: Simplified map scenario where the robot trajectory toward the goal is depicted with blue dots. The obstacles are drawn in black.

Figures 19.a and 19.b: It is presented the real scenario that has been drawn in Figure 18. It is shown the obstacles placed on the floor that the WMR should avoid.

Figure 20: In (a), (b), (c), (d), and (e) are depicted the monocular frames acquired while the WMR is avoiding the first obstacle placed on the direction towards the objective. (f), (g), (h), (i) and (j) show the occupancy grid obtained by integrating the acquired frames.

Figure 21: It is showed the set of new frames acquired after the first obstacle avoidance strategy. (f), (g), (h), (i) and (j) show the occupancy grid built by time-integration of the new acquired frames during the approaching navigation towards the goal.

Figures 8.22: (a) and (b) are the last monocular acquired frames that are free of obstacles. (c) and (d) are the corresponding occupancy grids that integrate the free of obstacle frames.

Figure 8.23: Simplified map scenario where the robot trajectory toward the goal is depicted with blue dots. The obstacles are drawn in black.

Figure 8.24: (a) picture shows the straight corridor navigation where left wall and right obstacles appear. (b) Picture shows the left corner corridor that the WMR should overcome in order to attain the desired coordinates.

Figure 8.25: (a) and (b) are the two first acquired frames depicting the left corridor wall that doesn't allow the left turning of the robot towards the desired coordinates. Figure 8.25.c depicts the occupancy grid built by the integration of the first two frames where the left corridor wall appears as an obstacle that should be avoided.

Figures 8.26.a, 8.26.b, and 8.26.c: Third, fourth and fifth acquired frames during the WMR navigation. Figures 8.26.d, 8.26.e, and 8.26.f show the corresponding occupancy grids. It is noted that the occupancy grid is aware of the left corridor wall even when the new acquired frame depicts a free of obstacle field of view.

Figures 8.27.a and 8.27.b: Front placed obstacles appear in the images acquired in the sixth and seven frames. Figures 8.27.c and 8.27.d depict the occupancy grids that were built by integrating sixth and seventh frames. They still were having memory of the left wall.

Figure 8.28.a and 8.28.b depict the eighth and ninth frames. Figure 8.28.c shows the occupancy grid obtained after the integration of frame ninth; it has memory of the past obstacles.

Figure 8.29: Simplified map scenario where the robot trajectory toward the goal is depicted with blue dots. The obstacles are drawn in black.

Figure 8.30.a, b, c, and d: The scenario where the third experimented has been developed. It is the larger trajectory tested. Indoor corridor navigation includes straight, left turning, right turning, and straight trajectories.

Figures 8.31.a, 8.31.b, and 8.31.c: Left obstacles appear in the first acquired frames. The navigation is constrained by these obstacles. Figure 8.31.d shows the integration of the first three frames. The left and right obstacles are constraining the WMR movements towards the objective.

Figure 8.32.a: Fourth frame depicts the obstacles placed on the right side of the corridor. Figure 8.32.b shows the corresponding occupancy grid where appears the right side obstacle and the left side obstacles due to the integration of the previous acquired frames.

Figure 8.33.a: The left side corridor wall appears when the WMR turns towards left. Figure 8.33.b shows the occupancy grid corresponding to the fifth frame integration. Short term memory of the right obstacles is attained.

Figure 8.34.a and 8.34.b: The left wall and a free of obstacles scenario are shown in these frames. Figure 8.34.c depicts the occupancy grid obtained by integrating the sixth and seventh frames. The obstacle shape of the left side wall is oversized due to the 3D perspective.

Figure 8.35.a and 8.35.b: Eight and ninth frames depict the presence of front obstacles. Figure 8.35.c shows the occupancy grid obtained after the integration of the ninth frame.

Figures 8.36.a and 8.36.b: Monocular perception of eleventh and twelfth frames. In Figures 8.36.c and 8.36.d are time integrated within the occupancy grid framework. The left side wall and front obstacles appear as obstacles.

Figures 8.37.a and 8.37.b: Acquired frames when right turning navigation is done. Figure 8.37.c shows the time integration of the fourteenth frame.

Figure 8.38.a and 8.38.b: Monocular acquired frames during right turning movement. Figures 8.38.c and 8.38.d depict the occupancy grid obtained after the integration of fifteenth and sixteenth frames. It is important to be aware that the right corner appears as an obstacle due to the short-term memory.

Figure 8.39.a: Seventeenth acquired frame. Figure 8.39.b depicts the occupancy grid obtained with the integration of the seventeenth frame with the previous acquired frames. It is noted that the marks placed on the floor appear as obstacles that can be removed by using OF methods.

Figures 8.40.a, 8.40.b, and 8.40.c: The eighteenth, nineteenth, and twentieth monocular acquired frames depict marks placed on the floor. Figures 8.40.d, 8.40.e, and 8.40.f show the occupancy grid that arise with the integration of the eighteenth, nineteenth and twentieth frames.

List of Tables

Table 2.1: The selected sensors and their main/ leading features

Table 5.1: The basic description related to the robot mechanics

Table 6.1: Modified Laplacian and standard deviation for each 9x7 image row
Table 6.2: Floor scenario; modified Laplacian and standard deviation for each 9x7 image window.
Table 6.3: Textureless object and floor scenario; FM and standard deviation for each 9x7 image window.

Table 7.1: The second order WMR model
Table 7.2: The reduced WMR model
Table 7.3: LMPC processing times
Table 7.4: Point to point trajectory tracking statistics
Table 7.5A: Performance for straight line trajectory
Table 7.5B: Performance for wide left turning
Table 7.5C: Performance for less left turning
Table 7.5D: Performance for wide right turning
Table 7.5E: Performance for less right turning
Table 7.6: Main and lateral effects
Table 7.7: Factor selection as function of the trajectories
Table 7.8: Results obtained by using fixed or flexible cost function
Table 7.9: Results obtained by using fixed or flexible cost function
Table 7.10A: Performance for straight line trajectory
Table 7.10B: Performance for wide left turning
Table 7.10C: Performance for less left turning
Table 7.10D: Performance for wide right turning
Table 7.10E: Performance for less right turning
Table 7.11: Main and lateral effects
Table 7.12: Results obtained by using fixed or flexible cost function
Table 7.13: Results obtained by using fixed or flexible cost function
Table 7.14: Results obtained by using fixed or flexible cost function
Table 7.15: Results obtained by using fixed or flexible cost function

Table 8.1: First experiment coordinates from where the monocular frames are acquired

Table 8.2: Second experiment coordinates from where the monocular frames are acquired

Table 8.3: Third experiment coordinates from where the monocular frames are acquired

List of Acronyms

ABEL: advanced Boolean equations language

APD: approaching point distance

ARMAX : autoregressive moving average with extra inputs

ARMS: absolute root mean square error

ARX: autoregression with extra inputs

AUM: autofocus uncertain measure

BCA: brightness constancy assumption

BCM: brightness constancy model

CAD: computer assisted design

CLF: control Lyapunov function

CS: complex spectrogram

DC: direct current

DFD: depth from defocus

DFF: depth from focus

EKF: extended Kalman filter

FCCU: Fluid Catalytic Cracking Unit

FIR: finite impulse response

FIS: focused image surface

FM: focus measure

FMF: focus measure filter

GPS: general positioning system

LAN: local area network

LPMOG: local planar monocular occupancy grid

LMPC: local model predictive control

LP: Linear programming

LPF: low pass filter

LSQ: least squares quadratic

LUT: look up table

MCL: Monte Carlo localization

MIMO: multiple inputs multiple outputs

MIL: Matrox Imaging Libraries

MKT: microcontroller software tools

MMIP: mobile multimedia information point

MOSFET: Metal oxide semiconductor field effect transistor

MPC: model predictive control

MPHC: model predictive heuristic control

NCP: nine control points

OD: orientation deviation

OF: optical flow

OFM: optical flow methods

OFT: optical flow techniques

OPUOF: obstacle perception using one frame

OTF: optical transfer function

PAL: phase alternation line

PC: personal computer

PCB: printed circuits boards

PID: proportional integral derivative

PLD: programmable logic device

PRIM: Platform Robot Information Multimedia

PRBS: pseudo random binary signals

PSF: point spread function

PVC: Poly-Vinyl Chloride

PWM: pulse width modulation

QP: quadratic program

RGB: red green blue

SFF: shape from focus

SFM: shape from motion

SISO: single input single output

SLAM: simultaneous localization and mapping

TDD: trajectory deviation distance

WMR: wheeled mobile robot

WP: wide-path

Chapter 1

Introduction

1 Introduction

Despite the enormous progress in robotics over the last half century, this field is still in its infancy. As compared with the human behaviour, the ability of a robot is limited in its movement, understanding complex sensorial inputs or performing higher level reasoning. The key requirements for enabling further progress to be made are the integration of several different fields of knowledge, such as computing, communications and control sciences, with the aim of enabling robots to use higher level reasoning and decision tools with a strong theory base [Murray et al., 03].

The research work presented integrates control science and robot vision knowledge in a computer science environment. It is aimed at differential driven wheeled mobile robots (WMR) with a free rotating wheel. The biological systems – i.e. walking humanoid systems – seem to be more advanced and are being successfully used as reference model sources by robotic researchers. They can overcome small obstacles, as well as up-down stairs [Denk and Schmidt, 03] and its locomotion advantages with respect to WMRs are clear. However, WMRs have some advantages in the aspects of speed, price, or power consumption.

Among many objectives in the mobile robot navigation, it is very important to achieve the feasible and accurate trajectory following. Environmental understanding and obstacle avoidance are research topics that allow the improvement of robot skills. The perception of the navigation environment is based on sensor systems that provide distance measurements in the vicinity of the robot. This essential task can be accomplished by different range systems such as ultrasonic sensors, laser rangefinders and vision-based systems. Among them, the computer vision-based system is one of the most attractive and useful sensing methods. It presents some interesting aspects such like its falling down price and capacity of providing richer information than other traditional ranging devices. Moreover, the continuously increasing capability of personal computer has augmented the field of real time applications of the vision-based systems.

The navigation control signals should include the policy of obstacle avoidance, as well as the final desired coordinate achievement. The scientific community developed several studies in this field [Rimon and Koditschek, 92]. Some approaches in mobile robots proposed the use of potential fields that satisfy stability in a Lyapunov sense, and the use of MPC (model predictive control) on a short prediction horizon [Ögren and Leonard, 05]. Recently, real time mobile robot MPC implementations have been developed using global vision sensing [Gupta et al., 05].

The relevant work developed consists in integrating local control and WMR navigation by using monocular perception. Therefore LMPC (local model predictive control) is presented as a fast and accurate methodology to track changing trajectories under dynamic environments. Moreover, the navigation strategies are oriented to goal achievement by using the desired coordinates as an artificial potential field that attracts the WMR towards the goal position while local obstacles coordinates are obtained by using monocular perception techniques.

This dissertation is organised as follows. First, it is presented the useful methodologies relevant to the research work, especially on the mobile robot navigation based on computer vision, 3D machine vision techniques and model-based control strategies. Then, it is presented the research developed and results obtained on the model-based control by using the monocular perception techniques to the WMR platforms. Finally, conclusions are drawn and further research works are planned.

The details of the dissertation content are given as follows:

Chapter 2 explains the different navigation strategies based on computer vision methods. These methods are classified according to the field of application as well as the computer vision methods and objective policies used.

Chapter 3 explains 3D machine vision system methods within a state of the art context. The present work focuses on monocular perception with optical flow, DFF (depth from focus) and DFD (depth from defocus) methods being the most suitable ones. The objective of the present research is the development of monocular methodologies for detecting obstacles.

Chapter 4 introduces the model based control strategies. Firstly the basic identification methods are commented. Once the methodology to obtain the dynamic model is attained, the classical PID speed control can be efficiently applied. The basic trajectories following methods are presented, as well as their applicability in WMR platforms. Model predictive control (MPC) is another interesting and suitable strategy. Finally, the applicability of such techniques to the WMR's is contemplated.

In Chapter 5 it is presented the available lab platforms used in this work. The robot description concerning with the basic mechanical and electrical system is explained. Furthermore, the system architecture is also commented. The philosophy of such platforms as open systems is also declared. Hence, pluridisciplinar teaching and research activities are important issues that can be tackled by the use of these platforms.

Chapter 6 presents the main research developed concerning the local monocular perception system. The DFF experimental results by using DFF methods and their application in WMR platforms, when static position is assumed, are presented. The use of such techniques, under homogeneous floor radiance, has been experimented in typical indoor scenarios. The use of cooperative optical flow and DFF methods is analysed in order to improve 3D understanding. However, due to the camera pose used in this thesis, the time-integration of the different acquired frames is developed by using the occupancy grid framework, the odometer system information, and the floor model.

Chapter 7 depicts the experimental local trajectory-tracking results obtained by using local MPC techniques based on the experimental WMR dynamic model knowledge. In this sense, the algorithm, simulation and experimental results are presented. The trajectories tested are obtained by considering the local narrow perception field of view obtained by the camera system. The study and tuning of the local MPC cost function parameters are considered as important issues of this chapter.

In the Chapter 8 the monocular local reactive navigation results are presented. In the present case, the use of monocular vision system as a main source of information provides a grid where global objectives can be accomplished by using the local information as data source that can allow a feasible real time trajectory planning in a reduced field of view. Artificial potential fields are also suggested as a way to attract the WMR to the desired configuration. Moreover, the occupancy grid size increases the environment description and a local memory concerning the passed obstacles can be achieved. The local navigation strategy algorithms are explained.

Finally, Chapter 9 presents the conclusions as well as the work to be developed.

Chapter 2

WMR Navigation Using Computer Vision

2.1 Introduction

This section presents a classification and description of relevant computer vision methods used in mobile robot navigation [DeSouza and Kak, 02]. Mobile robot navigation using computer vision in a WMR work environment is classified as either indoor or outdoor. Techniques based on map environment description are used in indoor navigation. Thus, according to the environmental map used, the following navigation classification is obtained:

- Navigation with maps
- Navigation based on map creation
- Navigation without maps

Outdoor navigation can be classified by considering the environment as either structured or unstructured.

2.2 Indoor navigation

A need for more environmental knowledge has arisen since early research work. Indoor navigation can be depicted by CAD models. Image sequences are used to determine the necessary environmental perception. Previous environmental knowledge can be used as a source for indoor navigation classification:

- **Navigation with maps.** Geometric models of the environment created by the users are used.
- **Navigation based on map creation.** WMR sensors are used to build the environment map that can be used to plan the robot's movements.
- **Navigation without maps.** These systems do not use any environment representation but merely recognize objects that can be tracked.

2.2.1 Navigation with maps

Navigation based on maps includes geometric environmental information. The first research was developed using 2D environment models consisting of a free or occupied grid [Moravec and Elfes, 85]. The virtual potential field idea improved performance by using objective attraction and obstacle repulsion forces [Borenstein and Koren, 89]. WMR navigation strategies should be planned so that obstacle collisions are avoided during navigation towards the objective. The meaningful navigation idea consists of providing feasible and expected visual landmarks, thereby allowing the robot's map position to be calculated using information about recognized landmarks. Localization of the WMR using machine vision system computation is done in 4 steps:

- Sensor information acquisition, in this context meaning digital image acquisition.
- Landmark detection: scenario data analysis from where relevant information is searched for (smooth and edge filters; grey level, region based segmentations; colour or texture based segmentations; distances; movement analysis; etc).
- Landmark detection is correlated with the corresponding map position.
- The robot position estimation is done using the previous data.

WMR localization can be considered as absolute or incremental. Incremental navigation methods consider the initial robot coordinates as known; machine vision system information is used as a way to improve the robot's positioning. Otherwise, absolute navigation methods would not know the initial robot coordinates.

Absolute localization

The navigation system should build a correspondence between the WMR's perception information and the database. Estimations based on sensor uncertainty probability allow better robot localization. Thus, the database correspondences arise in multiple perspectives when just one observation is used. The ambiguities can be solved by using statistical methods. In this connection, Markov analysis assumes that the past is independent of a future that arises from a present state. Markov's statement is known as the static world assumption: the present robot position is the only state that can have more influence than a single sensor measure [Thrun, 00].

Monte Carlo localization (MCL) localizes the robot through multiple sensor measures [Dellaert et al., 99]. The perception measures are weighted according to their importance. MCL localizes the robot by building a new database as a function of robot action and existing perception information. The Kalman filter and Gaussian distribution measures can perform good statistical analysis [Cox, 94]. Valuable research analyzing interval uncertainty or position estimation based on triangulation has also been reported [Atiya and Hager, 93].

Incremental localization

There are three different incremental localization techniques:

- Localization based on geometrical representation
- Localization based on topological space representation
- Localization based on landmark detection

1) Localization based on geometrical representation

In many practical situations, the initial robot position is known. The navigation strategy should measure positional uncertainties when the WMR movement commands are provided and sensor data can be used to estimate new absolute positions [Kosaka and Kak, 92]. This incremental localization system has the following three parts:

- WMR positional uncertainty (x, y, ϕ) is represented by a Gaussian distribution.
- Orientation errors during translation motions, and translation errors during rotation motions, should be considered. Thus, discrepancies between commands and movements should be analyzed.
- A Kalman filter design is proposed using these main steps:

- A constrained landmark search within image data, considering camera calibration. The line search can be simplified by using the Hough transform space.
- The uses of linear constrained equations to derive the Kalman filter equations.
- Robot uncertainty should be considered by the image space.
- A Kalman filter is used to obtain covariance and mean position values.

Once the image landmarks are detected, the WMR's position is computed by linear transforms using the camera and the robot's coordinate system knowledge. The image and robot position uncertainties can be used to compute the robot position probability based on statistical parameters.

The Kalman filter reduces the noise of WMR position measures. Thus, the covariance and mean position measures are actualized each time a movement command is given. Thereafter, the influences of navigation uncertainties can be analyzed in more detail [Miura and Shirai, 93].

2) Localization based on space topological representation

The topological representation of space in incremental navigation methods was proposed as a neuronal system applied to NEURO-NAV where the environment is represented in a useful simplified graph used by a vision system [Kuipers and Byun, 91]. Using graphs, the corridors, unions or no-way-out ends can be represented by squares, circles or diamonds; each node can have attributes to show, for instance, the number of doors belonging to a corridor. The NEURO-NAV system was composed of simultaneous landmark detection and navigation systems. The commands produced by the neural network are fuzzy. The complete set of inferred fuzzy rules can be found in [Pan et al., 95].

3) Localization based on landmark

Landmark detections can provide a robot's position. Landmark classification can be done by making a division between natural and artificial.

Circular patterns composed of encoded colour marks were proposed in [Kakuba and Arenas, 87]. Path landmarks consisting of magnetic lines were proposed in [Tsumura, 86]. Natural landmark detection is based on the environment map, where the robot's vision system should detect the landmarks. The robot's position is known through the positions of relative landmarks, including doors and windows [Hashima et al., 97].

2.2.2 Navigation based on map creation

The robot's own perception, acquired during navigation, is used to create an environmental representation. The space has been represented as a grid of free or occupied positions. The WMR's performance depends on sensor system precision and the accuracy of the map built by the robot.

The topological representation of space arising from grid information has been developed in [Yamauchi and Beer, 96]. These representations include metric and topological data to improve the WMR's navigation once the map has been built. Node classification research, including odometer and sonar sensor data fusion, has been developed. Hence, grid knowledge and topological information can be obtained using neural networks and Bayesian integration [Thrun, 98].

Other studies use sensor fusion and robot training, where featured scenarios are used as landmarks [Zheng et al., 91]. The robot is able to achieve proper localization using a database. The use of omnidirectional vision systems using conic mirrors has also been proposed [Yagi et al., 94].

2.2.3 Navigation without maps

This section presents the systems that *a priori* have no environmental description or map building data. The locomotion commands are generated using featured environmental information. These elements may be walls, doors, tables, etc. Therefore, the objects' absolute or relative positions are not necessary, but navigation is referenced to these elements. The most significant navigation strategies are:

- Optical flow based navigation
- Object recognition strategies with database
- Object recognition strategies without database

Research developing navigation without maps can be found in [Nakamura and Asada, 95]. Navigation strategies are based on behaviour; e.g. a WMR with a sonar sensor ring can demonstrate obstacle avoidance behaviour. Object tracking is another example of behaviour without maps [Huber and Kortenkamp, 95].

Navigation using optical flow

The first work was developed by Santos-Victor and was based on the lateral visual perception of insects [Santos-Victor et al., 93]. The bee's perception system uses mainly movement features instead of distance information. Similarly, multiple perspective view and optical flow perception lead to better collision avoidance performance. The implementation of these methods in robots has led to some kinds of improved behaviour, for instance, equidistant wall corridor navigation. Two optical flow constraints are the need for scene texture and the fact that some robot movements are not allowed. The analysis of other kinds of behaviour, such as object tracking, has been studied with pan and tilt camera movements [Bernardino and Santos-Victor, 98]. Another study consists of wall tracking behaviour where distances are obtained from optical flow analysis [Dev et al., 97].

Navigation based on database object recognition

This method consists of creating a database with images featured from around the world that can be associated with WMR commands and control.

The use of a neural network is proposed in [Gaussier et al., 97], in order to achieve the necessary action using perception data.

A panoramic scenario description can be achieved by combining different field of view images. The significant images contain relevant attributes which are used by a neural network in order to determine the necessary associations that will allow the final position to be reached.

The representation of the path by a set of different images should provide enough information to plan the necessary control commands as a function of the different acquired images corresponding to the selected trajectory [Jones et al., 97].

Navigation based on object recognition without a database

Navigation without maps can be based on a featured objects search. In [Kim and Nevatia, 98] a symbolic navigation strategy is proposed, based on the use of robot commands as a data source in order to achieve proper navigation. The command information should include the landmarks necessary to allow a feasible path and object recognition such as "*go to the door*", "*go forward to next desk*", etc. The machine vision method proposed consists of a trinocular system that is used for landmark detection as well as to plan movements towards the object through a 2D grid.

2.3 Outdoor navigation

Outdoor navigation methods normally include landmark and obstacle detection, map creation and actualizations as well as position estimation. These methods can be classified as functions of the environment:

- Structured outdoor navigation
- Unstructured outdoor navigation

2.3.1 Structured outdoor navigation

The first outdoor vision systems basically used obstacle avoidance policies. Roads are reported as an interesting research area in relation to outdoor structured environments. Road models include information such as width or vanishing points. Shadow and light variations are problems that should be considered.

An important outdoor navigation research project is the Navlab robot from Carnegie-Mellon University [Thorpe et al., 87]. The first work with Navlab used colour machine vision systems to detect and avoid obstacle collisions. Image processing was based on space-resolution methods, where high resolution images were used to detect the textures and low resolution to classify the colour. A proposed road detector algorithm consisted of pixel classification, detection of the probable road position and colour actualization. Pixel classification is done by performing a probabilistic analysis of colour and texture. The Hough Transform can be used to compute the vanishing road point using the longer lines. Road changes are detected by estimating the road limits as well as the mean and variance changes from frame to frame. An experiment was carried out on the system with a network able to determine driver reactions to road variations [Pomerleau, 91].

The outdoor navigation project EUREKA "*Prometheus*" consists of creating an automatic pilot which can be used as a traffic warning when dangerous situations are detected [Graefe, 93]. Constrained region analysis is known as adapted perception. Here, explored image size and computing time are reduced, and robot speed can be increased [Kelly and Stenz, 97].

2.3.2 Unstructured outdoor navigation

Outdoor navigation environments, where there are no useful visual navigation features, are known as unstructured. For instance, navigation on the planet Mars can be shown as a typical outdoor unstructured environment. Two basic navigation strategies are used:

- Closer initial point navigation strategies, without a defined objective
- Navigation strategies where an object position should be reached

WMR localization is mainly based on environmental camera perception, where the use of distant points, such as mountains or hills as marks, allow triangulation positioning. Map construction and robot positioning using stereo vision techniques has been reported in [Krotkov and Herbert, 95]. The proposed system includes sensor fusion with the available data coming from encoders, compasses and inclinometers. The Pathfinder project can be considered one of the most important [Matthies et al., 95]. Pathfinder landed on Mars in July of 1997 and the mission was developed using two main systems:

- The static module, known as the "*lander*", consisting of a tetrahedral measuring 1.5 m by 1 m and weighing 245 kg was a multi-spectral stereo vision system with pan and tilt camera movement controlled by a computer.
- The rover, 65 cm long, 48 cm wide and 30 cm high, was equipped with a controller based on an Intel 8085 processor and 500kb of memory.

The rover's navigation consists of four basic functions:

- Object designation
- Path selection
- Rover localization
- Risk detection

Object designation is carried out by human operators using the image information provided by the lander. The 3D coordinates corresponding to the robot path are determined by pixel selection. Once 3D coordinates are selected, the goal is attained by following these rules: when no danger is present, the rover goes ahead or turns towards the object; otherwise, when danger is detected left, the rover turns to the right until no risk is detected, and vice versa. The localization missions are updated daily by the lander. The vehicle can advance at 15 cm/s and, on average, safety stops are produced each 6 cm. Obstacle detection is done using two cameras and five laser diodes that provide a structured pattern up to 30 cm in front of the vehicle. Other Pathfinder research is addressed at improving map creation by using maximal likelihood algorithms [Olson and Matties, 98].

Hence, last Mars exploration news arises from the WMR Spirit, which landed in Mars in January of 2004. It has been doing a continuous surface exploration for more than four years and revelling hints of water history on Mars (1).

2.3.3 Light and shadow in outdoor navigation

Change in light is a common outdoor problem, meaning that image contrast and texture differences depend on the time of the day, the season of the year, weather conditions, etc.

Colour constancy can be an interesting environmental feature that allows more system robustness [Thorpe et al., 88]. Subsequent work used colour features to compensate for sunny conditions or colour space changes that improve the results [Mori et al., 97].

(1) http://www.space.com/missionlaunches/spirit_rolling_040115.html

2.4 Proposed basic system

WMR Navigation based in computer vision is presented in this work. In this context, monocular perception has been employed. The motivation arises from the fact of using the PRIM WMR platform available in our lab. Moreover, it is pointed that the research group VICOROB, where the author belongs, have not previous experiences with monocular perception systems. Therefore, the work presented complements the knowledge of the research group based on machine vision systems that use multiple cameras. The proposed sensor system as well as its uses and possibilities are presented in this section. Navigation strategies and the author philosophy underlying this work are also commented on.

2.4.1 WMR sensorial system suggestions

The work developed in this thesis is done by using monocular perception. However, this section presents the sensorial system of the WMR from a general point of view. Therefore, at this point, the proposed system should be complete and satisfactory for indoor WMR navigation. Sensor selection criteria are their low cost and the possibility of developing a wide range of navigation strategy applications. The selected WMR sensors are:

- An odometer system
- A sonar sensor ring or laser range finders
- A machine vision system

The odometer system is widely used in outdoor and indoor navigation methods based on computer vision perception. It allows the WMR's relative position to be known and referenced to its initial one. It can be used as an incremental localization system, providing incremental robot positions from frame to frame. The WMR coordinates are given by (x, y, θ), the robot position and orientation. The odometer system algorithms, using an encoder, have interesting features as well as drawbacks:

- Fast and easy algorithms, with a processing time of milliseconds. The low level controller design needs include a sample and actuation time of less order than the constant of time of the control system, which in WMR is usually in tenths of a second.
- Odometer systems based on encoders have dead-reckoning problems. They can be caused by certain WMR movements that produce wheel slip. These problems are also related to the robot's speed, trajectory and dynamics.

The sonar sensors or laser range finders allow vertical obstacle detection and easy heuristic and fuzzy navigation strategies. Good results are shown when tasks such as obstacle contour, wall tracking or equidistant obstacle navigation are commanded. The strategies proposed are similar to those used in the rover Pathfinder [Matthies et al., 97]. A sensor fusion strategy is proposed as future work.

The machine vision system should provide the main 3D environment perception. The following aspects are considered:

- Odometer dead-reckoning problem can be minimized by using structured natural landmarks, such as doors, walls, windows, corners, columns, etc. Thus, the WMR machine vision system should detect obstacle coordinates in order to find a trajectory free of obstacles and detect the landmark positions that can reduce dead-reckoning problems.

- Map building is also possible using sensor fusion and mosaic techniques. Hence, simultaneous localization and mapping (SLAM) is a well known WMR research topic. However, it is beyond the scope of this research.
- The use of active vision methods can be used in texture less environments as a way to solve these constraints to obtain some obstacle information.
- The camera should be able to make pan-and-tilt movements as well as have a flexible configuration.

The different sensor constraints and successes are depicted in Table 2.1.

Table 2.1: The selected sensors and their main/ leading features

Sensor	Failures	Successes
Sonar Sensor or laser range finder	Down steep slopes and precipices Small objects **Computing time**	Vertical Obstacles Textureless obstacles Glass doors or walls
Machine Vision System	**Textureless obstacles** Glass doors and walls **Computing time**	Down steep slopes and precipices Small objects Accurate measurements
Odometer System	Dead-reckoning	Computing time Accurate measurements

2.4.2 Proposed navigation methods

This section introduces the proposed navigation methods where flexibility criteria are considered. A comparative analysis with the methods explained in the previous sections is also presented.

The indoor navigation proposal of this work is based on previously explained strategies:

- Objective oriented navigation proposal consist in different local attraction fields which can be understood as a simplified map while collision avoidance is reached by perception data analysis [Borenstein and Koren, 89], [Moravec and Elfes, 85].
- Map localization is based on natural landmark detection. Hence, robot localization is considered as incremental considering that the WMR coordinates are know. However, landmark positions are considered as absolute coordinates that can set to zero dead reckoning errors [Hashima et al., 97].

The authors believe that the performance of a wide range of robot tasks can be achieved using the robot control commands and simplified map as a source of information to be obtained and controlled. This navigation strategy is similar to some simple agents tested on real animals such as cows [Butler, 06]. Therefore, the proposed methodology will use local artificial potential attraction field coordinates as a way to attract WMR towards a local desired goal. In this context, the use of monocular perception and local control methods, which allow dynamic navigation towards the desired local goal while obstacle avoidance is performed, are presented and tested in this work for static obstacles. Thus, the methodology presented can improve the WMR robot navigation allowing flexible navigation in dynamics environments and avoiding complete map descriptions. Therefore, local landmarks coordinate used as artificial attraction fields can be enough to achieve feasible and accurate navigation. In this way, navigation from landmark to landmark can be accomplished by the methodology presented. Moreover the use of a simplified map based on natural landmark information can improve WMR localization setting to zero the dead reckoning errors.

Map representation should allow navigation based on a sufficient number of natural landmarks to reach the final object. Hence, when the robot meets a corner with a cross corridor and should turn left, the corner is considered as a natural landmark to be reached and turning left can be considered as further attraction field that must be activated once right positioning is reached. Another meaningful example is to plan that the robot will leave a room and turn left along a corridor until the fourth door on the right is attained. In this case the left hand corridor environment description is not necessary. The summarized features of the proposed strategy are:

- Greater simplicity than topological representation
- Indoor navigation based on natural landmarks that are used as map representation
- Similarity to the human behaviour shown when navigation objectives are accomplished with a minimal set of information

Localization is achieved by using natural landmarks; it deals with structured indoor navigation scenarios where the typical environment can be used as natural landmarks [Hashima et al., 97].

Outdoor navigation is constrained by the local perception of the proposed sensorial system that normally will be less than 2 meters. Hence, only low speeds can be reached and as a result applications may be limited. Therefore, a complementary sensor system should be added as future work. Moreover, the use of other sensor systems, with a large range, can be used to deal with dynamic obstacles. Fig. 2.1 shows the scheme of the proposed navigation strategy where the local research developed in this thesis is drawn as green blocs and the necessary complementary blocs for achieving global navigation are depicted in red.

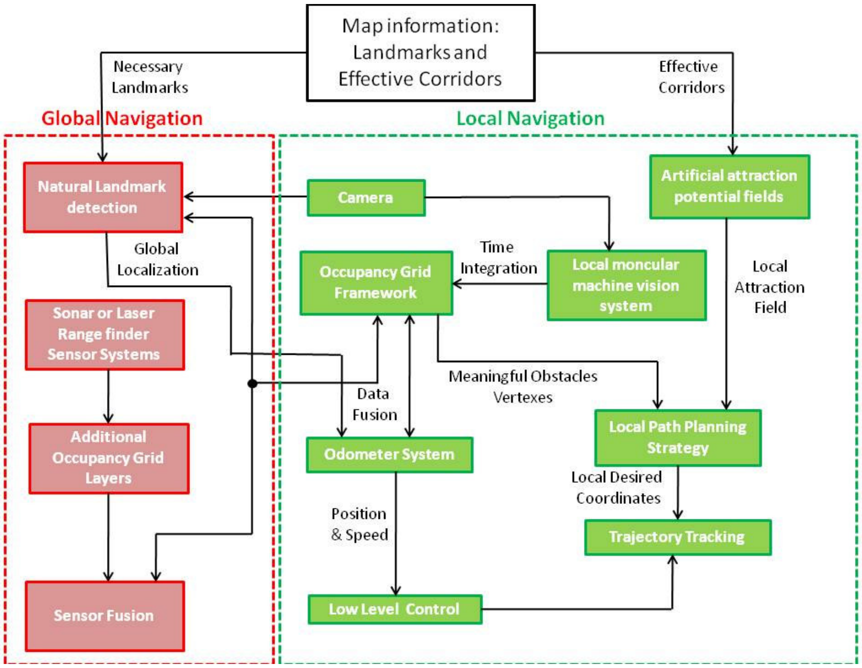


Figure 2.1: The proposed navigation scheme.

Chapter 3

3D Machine Vision Techniques

3.1 Introduction

Scene images formed by an optic system, usually with lenses, contain three-dimensional information arising from photometric and geometric data. The problem of 3D understanding, when the data source is provided by an image sequence, is known as inverse Optics. The geometrical information comes from the object distances and shapes. The photometric information consists of object colour and radiance. The methods used can be classified as active or passive. Active methods use external energy sources, such as lasers or sonar, to obtain the scene information. Passive methods use the normal environmental light. 3D scene data can be obtained by using several image analysis techniques:

- Stereo Vision
- Shape from X

Once the different methods are explained, a comparative performance analysis is done. However, this research work is focused on monocular vision-based methods. The motivation to select monocular perception techniques is based on the fact that the group VICOROB has a lack of results by using these techniques. Therefore, the research presented is aimed to complement this aspect. Thus, once 3D machine vision techniques are introduced, the research is focused into analyse in depth the monocular methodologies.

3.1.1 Stereo vision

Stereo vision techniques seem to provide the easiest clues for obtaining 3D information [Dhond et al., 89]. The 2D image perception corresponding to two simultaneous images, acquired from different 3D camera positions, is used to obtain the correspondence from scenario and image points. Solving the correspondence problem allows 3D scene coordinates to be computed. Some problems, such as occlusions, should be considered when point correspondences are computed.

3.1.2 Shape from X

Several methods can be used to obtain the object shapes and 3D coordinates. X represents the various options:

- Shape from intensity
- Shape from motion
- Shape from focus and defocus

3.1.2.1 Shape from Intensity

Shape information is obtained by a sequence of intensity images where each frame has a different light source. This technique is divided into three categories: shape from shading, shape from photometric stereo and shape from photometric sampling.

- **Shape from shading**. The objective of shape from shading methods is to reconstruct the 3D shape of an object from its 2D intensity image. It is assumed a proper reflectance map, which models the relationships between the intensity and surface shape, is given. This constraint can be expressed as geodesic curves that allow object shape recovery [Horn and Brooks, 89]. This methodology allows an easy one-image process but low resolution and reflections are drawbacks to be considered.

- **Shape from photometric stereo**. This method tries to solve the shape from shading problems by computing the reflectance map as a function of the object shape but also considering the light source position [Woodham, 80]. The image provides different shape constraints which are used to compute the object shape by minimizing a cost function.
- **Shape from photometric sampling**. Different light sources and an image sequence corresponding to the different light sources are used [Nayar et al., 90]. The use of additional light sources minimises the photometric stereo problems caused by inadequate light positions.

3.1.2.2 Shape from Motion

3D coordinates are computed using motion information. Two techniques differ in terms of fixed or unfixed camera poses. When the scene remains fixed, and just the camera is moved, the problem is similar to stereo vision. Thus, different images of the same object are acquired from different positions. However, unlike stereo vision systems, only one camera is used. These methods are used in "*eye-in-hand*" manipulators. Knowledge of the 3D coordinates using optical flow, known as visual servoing [Hager and Hutchinson, 94], has been applied to robot navigation. Optical flow analysis can provide 3D object understanding as well as motion information.

3.1.2.3 Shape from focus and defocus

The focus position is used as a clue to compute the object shape and distance. Objects placed at different distances are focused at different distances. Two methods perform 3D scene recovery using camera setup information. Depth from Focus (DFF) methods search for the best focused image within an image sequence. Depth from Defocus (DFD) methods compare the relative defocus blur to obtain depth information using only two or three images of the same scene acquired from different focus positions. The first reported research can be found in [Born, 65].

3.1.3 Monocular 3D scene analysis

This subsection analyses the monocular possibilities related to the methods explained previously. The monocular constraint of this research avoids stereo vision studies. More extensive state-of-the-art research is developed considering the monocular 3D techniques previously introduced.

A comparison between the techniques presented previously reveals advantages and disadvantages. Passive methods have been selected so current perception systems should be robust enough to allow navigation with normal illumination sources. Thus, in textureless environments where active methods have been highly successful, the use of sensor fusion is proposed, and shape from intensity methods are not considered. Furthermore, shape from shading methods are based on previous shape knowledge. The two remaining monocular methods, DFF and SFM (Shape from Motion), are discussed in the next sections. Hence, several differences are considered:

- DFF methods use an image sequence of the same scene acquired with different camera setups.
- SFM methods analyse the optical flow of an image sequence to obtain 3D structures.

A priori, DFF method constraints referring to the need for an image sequence of the same scene seem very restrictive. In the next sections optical flow and the state of the art of DFF are presented and their applicability is analysed.

3.2 Optical flow methods

The motion of the image brightness pattern observed by a camera, when it or some scene object is moving, is called optical flow. Techniques to measure it use a time varying sequence of images as a source of information. Unlike DFF methods, camera calibration is needed [Elsayed, 03]. The reconstruction of 3D information from 2D information could also be considered as an ill-posed problem [Aloimonos and Duric, 92]. Hence, it has been suggested, due to the inherent differences between a 2D motion field and intensity variations, that normally only qualitative information can be extracted [Verri and Poggio, 87]. Despite this, the interesting features of optical flow become evident in a wide range of applications: Science and military (aerial images), autonomous robot navigation (obstacle avoidance and visual odometry), safety systems, sports (movement analysis), multimedia (videoconference tracking), 3D reconstruction by multiple views.

3.2.1 Field movement analysis and optical flow

When objects or cameras move in a fixed environment, the image change can be used to obtain the relative movement or the object shape. Field movement refers to each image pixel having an associated velocity vector. Hence, each instant of time, each image point P_i has its corresponding scene point P_o. Fig. 3.1 shows that both points are related by the image perspective projection [Horn, 98].

$$\vec{v}_o = \frac{d\vec{r}_o}{dt} \qquad \text{and} \qquad \vec{v}_i = \frac{d\vec{r}_i}{dt} \qquad (3\text{-}1)$$

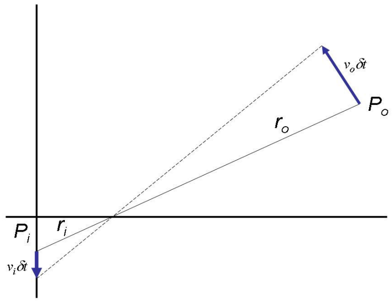


Figure 3.1: Two point projection relationship.

where v_o and v_i represent the velocity of the object point and image point respectively, dr_o and dr_i the object and image point displacement, and dt the differential of time considered. These vectors can be assigned to all the image points and are known as field movement. The optical flow is the apparent movement of light patterns. It ideally corresponds to the field movement, but several exceptions exist:

- An opaque sphere rotation cannot provide image movement.
- Lighting changes can provide inexistent movements.

Optical flow is known as image flow when:

- The surfaces have many marks or texture and homogeneous lighting.
- There are constrained scene dynamic movements.
- The brightness constancy assumption (BCA) is made.

Normally, neighbouring points have similar velocities. However, object boundaries are exceptions. The use of BCA is known as the brightness constancy model (BCM) [Negahdaripour, 02].

3.2.2 Computing optical flow using BCM: the aperture problem

Let *i(x,y)* be the irradiance of an image point *(x,y)* at instant of time *t*. If *v(x,y)* and *u(x,y)* are *x* and *y* optical flow speed components, then the irradiance time changes, using BCM, are:

$$i(x + u\delta t, y + v\delta t, t + \delta t) = i(x, y, t) \qquad (3\text{-}2)$$

The BCM constraint will set the algorithm performance. Designating i_x and i_y as the intensity gradients, the BCA constraint can be expressed as:

$$(i_x, i_y).(u, v) = -i_t \qquad (3\text{-}3)$$

Then, as shown in Fig. 3.2, the brightness optical flow direction is given by $(i_x, i_y)^T$ and the intensity gradient component is:

$$\frac{i_t}{\sqrt{i_x^2 + i_y^2}} \qquad (3\text{-}4)$$

The optical flow components that have right angles relative to the vector $(i_x, i_y)^T$ cannot be determined. This is known as the aperture problem.

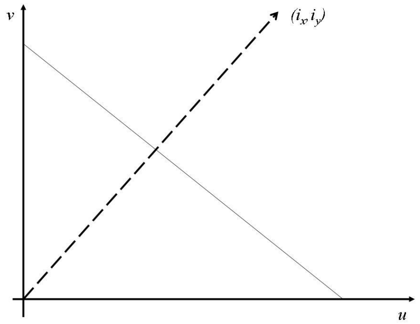


Figure 3.2: Optical flow and gradient directions.

Research into point displacements is done by looking in the next frame point neighbourhood. However, the uniqueness correspondence probability is low because:

- The neighbours have different brightnesses, due the BCM violation.
- More than one neighbour has the same brightness.
- More constraints are necessary.

3.2.3 Smoothness constraints

Because optical flow is continuous and rigid bodies are assumed, the neighbourhood pixels have similar movements. Thus, field movement discontinuities arise from depth boundary discontinuities. The smoothness constraint can be expressed using the Taylor approximation series:

$$i(x+u\delta t, y+v\delta t, t+\delta t) = i(x, y, t) \qquad (3\text{-}5)$$

$$i(x, y, t) + \delta x\frac{\partial i}{\partial x} + \delta y\frac{\partial i}{\partial y} + \delta t\frac{\partial i}{\partial t} = i(x, y, t) \qquad (3\text{-}6)$$

The image flow is computed on each pixel using a neighbourhood cluster. The error, e_0, should be minimised with a cost function based on the displacement for a set of points belonging to the same object.

$$e_0 = \iint (i_x u + i_y v + i_t)^2 \, dxdy \qquad (3\text{-}7)$$

The optimal solution is obtained when the displacement gradient is set to zero. An example of the optical flow algorithm is:

- Each image pixel should have a determined neighbourhood where displacements are computed.
- The image can be divided by using the windows determined in the previous step.
- The mean neighbourhood velocity should be similar to the velocity of each pixel.
- Each cluster should belong to a unique background.
- When errors are big the cluster selected may belong to more than one object.

The new value of (u, v) at a point is set equal to the average of the surrounding values, minus an adjustment, which in velocity space is in the direction of the brightness gradient.

3.2.4 A short state of the art review

The scientific community has developed a large number of quantitative methods to test the performance of different methods. Special attention has been paid to the differential techniques also known as variational methods. The use of a global smoothness term to constrain the velocity of the field of view was reported in [Horn and Schunck, 81]. Other differential methods assumed constant model image flow regions and consequently field flow is measured locally using weighted least squares methods [Lucas and Kanade, 81]. Frequency-based methods in the Fourier domain have been proposed. The output frequency of velocity-tuned filters, such as Gabor filters, can also provide optimal flow estimation [Heeger, 88]. The velocity phase behaviour of band pass filter outputs is another suitable methodology [Fleet and Jepson, 90] . Therefore, the OFT allows motion and structure recovery when the aperture problem is not an issue [Horn, 98]. The interests of some scientists have dealt with the performance of the different methods in terms of: convergence results, number of iterations, filter design, confidence measures, or number of frames needed [Barrow et al., 94]. The different quantitative approaches are generally based on two classical, feasible assumptions, BCM and optical flow smoothness. Thus, image motion discontinuities are due to the depth and motion discontinuity boundaries. The following concluding aspects are considered:

- The optical flow should be computed in homogeneous regions and smoothness constraints should be accomplished.
- Optical flow changes can be used as image segmentation clues.

Hence, there are places where image flow changes suddenly are useful as image segmentation clues but can cause problems such as optical flow estimation clusters.

The assumptions, in terms of accuracy or constraints, will set the method performance. Thus, it can be assumed that the camera is moving through a fixed environment, or there is more than one moving object with independent motion. The drawbacks presented, when noise or large displacements between frames exist, can be solved using space resolution image compression that reduce the noise, and improve the search process when long distances between frames are produced [Adelson et al., 84]. An algebraic multigrid method for anisotropic variational approaches, using a unidirectional multi-resolution search, is reported in [Ghosal and Vaneck, 96]. The use of combined local and global methods for optical flow computations has depicted dense flow fields with high noise robustness and low average angular errors [Bruhn et al., 02]. Real time variational optical flow computation using nondyadic grid hierarchies with bidirectional cycles allow global and local image flow estimation with no restriction about the image size or traversed levels [Bruhn et al., 05]. Image flow computation can solve important objectives of robotics, such as motion detection and scene structure analysis. Robot skill improvements can include interesting features such as: mosaic based positioning, visual servoing using special cameras with omnidirectional perception, structure from motion or visual odometry, among others [Unnikrishnan and Kelly, 02]. Visual odometry and structure scene analysis have been successfully reported and the experiences show advantages in slipping surfaces when compared with traditional odometer systems based on encoders. The scene structure analysis is used as an obstacle avoidance detection method that includes precipice detection [Campbell et al., 04]. In the work of Campbell, Gaussian and Sobel filters are used within the Canny edge detector algorithm to detect edge magnitude and orientation and provide a feasible boundary binarization. Structure and motion detection is done over detected edges.

3.3 DFF Methods

DFF techniques use an image sequence of the same scene acquired from different focus positions. The focus position differences consist of small increments. Thus, each image pixel has a focus position which is unique, has maximum energy and is also known as focus measure (FM). The thin lenses formula is used to compute the distances between the focused objects and the image plane [Jarvis, 83]. It has an exact value for optical systems without aberrations and points are closer to the domain of paraxial optics.

3.3.1 The optical transfer function (OTF) as a focus measure

The Fourier optics, for linear shift-invariant systems, has the following relationship:

$$I_i(\omega,\nu) = H(\omega,\nu)F(\omega,\nu) \qquad (3\text{-}8)$$

Thus, the image *I* can be expressed by an image convolution of the focused image *F* by the point spread function *(PSF)* represented by *H*. The Hankel transform, applied to the PSF for rotationally symmetric functions, is:

$$H(\rho) = 2\frac{J_1(2\pi R\rho)}{(2\pi R\rho)} \qquad (3\text{-}9)$$

$$I_i(\omega,\nu) = 2\frac{J_1(2\pi R\rho(\omega,\nu))}{(2\pi R\rho(\omega,\nu))}F(\omega,\nu) \qquad (3\text{-}10)$$

The PSF Hankel transform is shown in Fig. 3.3. The principal lobule area grows when the defocus is decreased. Thus, the main lobule volume is considered as a focus measure [Subbarao et al., 92].

$$FM_0 = \iint \left| I_i(\omega, \nu) \right| d\omega d\nu \qquad (3-11)$$

The larger focus measure value is obtained for perfectly focused images.

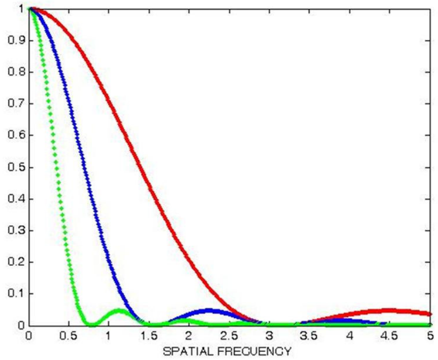


Figure 3.3: PSF Hankel transform.

3.3.2 Focus measure operators

There is a set of operators that measure the image energy and are consequently used as focus measures [Krotkov, 86]:

I. The gray level energy

$$FM_1{}' = \frac{1}{N^2} \sum_x \sum_y i_i^2(x,y) - \mu_i^2 \qquad \text{where} \qquad \mu_i = \frac{1}{N^2} \sum_x \sum_y i_i(x,y) \qquad (3-12)$$

II. The image gradient energy

$$FM_2 = \sum_x \sum_y \left(i_x^2 + i_y^2 \right) \qquad (3-13)$$

where

$$i_x(x, y) = i_i(x+1, y) - i_i(x, y)$$
$$i_y(x, y) = i_i(x, y+1) - i_i(x, y)$$

III. The Laplacian energy

$$FM_3 = \sum_x \sum_y \left(i_{xx} + i_{yy} \right)^2 \qquad (3-14)$$

where

$$\begin{aligned} i_{xx} + i_{yy} = & -i_i(x-1, y-1) - 4i_i(x-1, y) - i_i(x-1, y+1) - 4i_i(x, y-1) + 20 i_i(x, y) \\ & - 4i_i(x, y+1) - i_i(x+1, y-1) - 4i_i(x+1, y) - i_i(x+1, y+1) \end{aligned}$$

IV. The Laplacian magnitude energy

$$\nabla_M^2 i_i(x,y) = \left| \frac{\partial^2 i_i}{\partial x^2} \right| + \left| \frac{\partial^2 i_i}{\partial y^2} \right| \qquad (3-15)$$

$$FM_4 = \sum_{x=i+N/2}^{i+N/2} \sum_{y=j-N/2}^{j+N/2} ML(x, y) \quad \text{when} \quad \mathrm{ML(x, y)} \geq T_1$$

$$ML(x,y) = \left|2i(x,y) - i(x - step, y) - i(x + step)\right| + \left|2i(x,y) - i(x, y - step) - i(x, y + step)\right|$$

In [Tyan, 97] the experimental results obtained show the following conclusions:

- The ARMS (absolute root mean square) error decreases by increasing the signal to noise relationship.
- Small standard deviation does not mean a minimal auto focus error.
- The best focus measure can be affected by the object.

The use of different textured objects and their focus measures computed with different operators has been analysed. Results have shown that both square of the gradient magnitude and square of the Laplacian can provide the better performance as function of the object texture. However, the gray level variance is not appropriate for some textured objects.

3.3.3 Statistical methods in the focus measure quality

Textureless images can provide ambiguous results due to the noise influence. Statistical methods applied to the different operators can provide the minimal thresholds for obtaining feasible results. Useful statistics such as the variance of focus measures are used as quality parameters. Thus, mean value, standard deviation and variance are proposed. The mean value provided by other measures is used as an expected future result. The standard deviation depicts deviations from the mean value and variance is the square of the standard deviation. Experimental focus measure thresholds consisting of mean value plus three times the standard deviation are proposed for 15x15 windows [Surya, 94]. These quality parameters can be computed for the different operators.

The noise is affected by the focus measure filter (FMF) convolution. Thus, the FM has two components, signal and noise, and on average noise effects an image sequence similarly to all the measures. Pre-processing and post-processing the signal improve the results. The following three normalizations are commonly used.

- Magnification normalization is usually ignored due to the small value.
- The image mean, as i.e. five times the same image, reduces the effects of noise.
- Gray level normalization increases robustness against light variations.

The energy measures are based on the main lobule volume, so the focus values should increase in monotone way. However, images with high frequency lateral lobules can produce local maximum. Fig. 3.4 shows high frequency lateral lobules. Low pass filters (LPF) are used to reduce the lateral lobules and improve the search for the best focus position. Ideal LPF are very expensive computationally, thus lateral lobules are reduced with respect to the main lobule. The Gaussian filters are the most widely used.

The LPF-obtained signal has many advantages:

- Noise reduction
- Aliasing attenuation
- Unimodal and monotonic focus measures

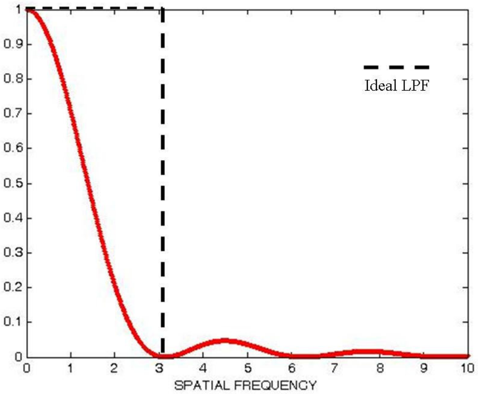


Figure 3.4: The high frequency components can produce lateral lobules.

The Gaussian filters have band pass features. Thus, focus measures preserve monotonic and unimodal features. The noise and lateral lobule high frequencies are reduced. The post-processing methods consisting of mean filters, such as 5x5, can reduce erroneous distances due to the lack of texture by considering wider patches.

3.3.4 3D Scene perception

3D reconstruction using the focus position is known as Shape from Focus (SFF) [Nayar, 92]. The focused image plane is moved through the object and the focus positions relative to the different object regions are searched for, as shown in Fig. 3.5. Interpolation methods are used to reduce the number of acquired images. The local measures allow system adaptation to possible object texture changes.

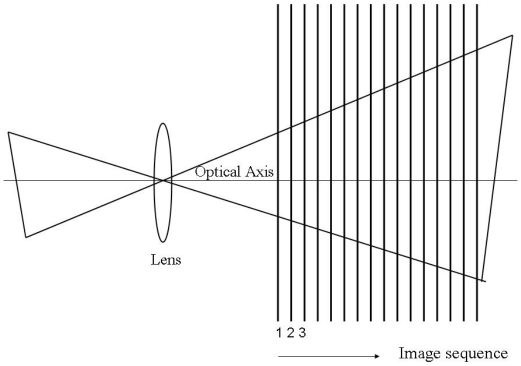


Figure 3.5: The shape from focus concept.

The focused image surface (FIS) is based on the fact that all the focus measures are computed on pixels belonging to the focused image surface of an image sequence [Subbarao and Choi, 95]. Using planar approximations for each surface image window, a second more accurate search using sub-windows can be done. In [Choi and Yun, 99] an FIS search for complex surfaces is proposed and the use of planar approaches in complex surfaces becomes unfeasible. Thus, small window sizes are needed and a consequent lack of information is produced. The use of 9 control points (NCP) for each window is proposed and depicted in Fig. 3.6. The objective is to find a curve, using the Lagrange equations, that fits into the 9 points. The 9 points can control the surface form, and using the curve surface the FIS window can be obtained.

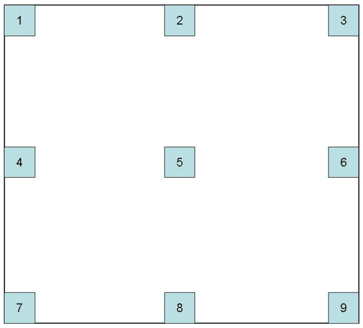


Figure 3.6: NCP are used to interpolate a surface.

3.4 3D recovery by DFD methods

DFF methods obtain scene depth information using the relationship between unfocused images, normally two, of the same scene but with different camera setups. It is an ill-posed problem; the geometrical image features and the number of images used are considered in the different models developed by the scientific community. The optical transfer function (OTF) of the defocused images depicts frequency image dependence, and scene texture is then a meaningful parameter to infer depth information. The lack of texture makes the use of passive techniques to obtain the distance unfeasible. The majority of developed methods belong to the frequency or space domain. In this section reported research considering mathematical models as well as the optic systems developed are introduced.

3.4.1 Depth information using two images

The first work used one focused and another unfocused image, and their relationship was used to obtain the distance [Pentland, 85]. Other works extend the research to the case of two unfocused images [Subbarao, 87]. The unfocused image g(x,y) is the result of the focused image f(x,y) convolved by the defocus operator.

$$g(x,y) = f(x,y) * h(x,y) \qquad (3\text{-}16)$$

The Gaussian operator provides good models for planar surfaces. A space domain transformation using a third degree image function as a model was proposed by Subbarao and developed by Surya [Surya, 94; Subbarao, 91].When frequency space is

$$P(\omega,\nu) = GG^{*} \qquad (3-17)$$

considered, the spectral power between the two images is used to obtain the scene depth.

Bidimensional Fourier transform noise can be minimised using the unidimensional Fourier transform [Subbarao, 94]. Frequency space research using a set of frequency Gabor filters, that measure the magnitude as a focus measure, is presented in [Xiong and Shafer, 93; Xiong and Shafer, 94]. The convolution ratio between the defocus operator of the images, represented by $h_3(x,y)$, is used to obtain the distance. Ratio normalization in frequency space has been proposed [Watanabe and Nayar, 98]. Frequency space inverse filtering regularization improves the system robustness [Ens and Lawrence, 91]. The $H_3(\omega,\nu)$ ratio through a quadratic model is proposed using least squares methods [Ens and Lawrence, 93]. The complex spectrogram (CS) and the Wigner distribution provide useful frequency measures [Saleh and Subotic, 85]. The distance computation, using complex the spectrograms and the Wigner distribution in two defocused images, is presented in [Rajagolapan and Chaudhuri, 97]. Finally, other multiple mathematical models are tested by the researchers. Thus, the blur parameter estimation and scene distance can be modelled using Markov fields [Rajagolapan and Chaudhuri, 99]. Another model formulation, where radiance is considered as belonging to the Hilbert space, is reported in [Favaro and Soatto, 02].

3.4.2 The camera optics

Early research was done by Pentland: "*Surprisingly, the idea of using focal gradients to infer depth appears to have never been investigated*" [Pentland, 87]. Pentland proposed comparing two images of the scene, acquired at the same time. One image is formed by a pinhole camera and the other by a normal camera. The simultaneous analysis of two images creates some problems:

- The two images should contain the same scene.
- Magnification problems should be considered.

These problems can be solved by using the camera model shown in Fig. 3.7. The lens system uses a half-silvered mirror to duplicate the original image. Thus, the two images are considered as identical except in global intensity and brightness normalization is necessary.

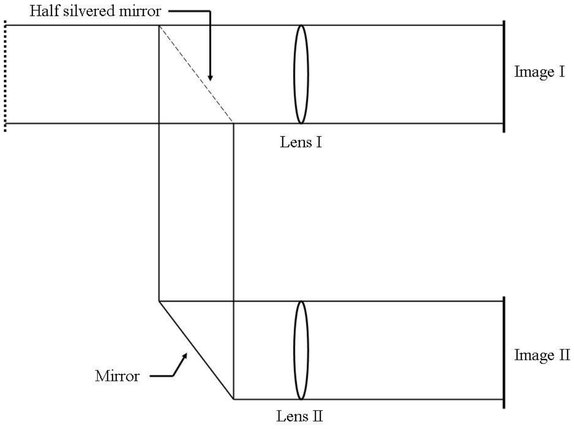
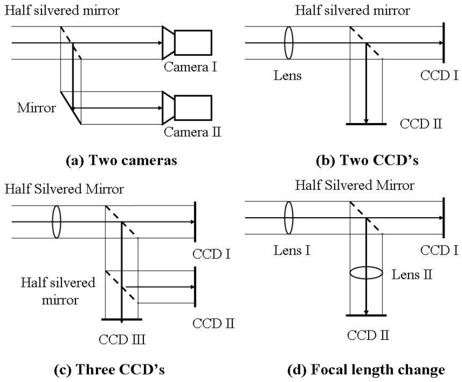


Figure 3.7: The Pentland camera model.

Other camera models, depicted in Fig. 3.8, have been proposed in [Subbarao, 93]. A multifocus camera that acquires three images with three different focus positions was implemented in [Hiura et al., 98]. The system, shown in Fig. 3.9, uses the RGB camera outputs to transmit 3 images. The magnification problem due to focus changes can be solved using camera calibration or telecentric lenses [Wilson and Shafer, 94; Nayar et al., 96]. The chromatic aberrations are due to the different colour wavelengths and are an intrinsic camera lens feature. The chromatic aberration used to infer the autofocus direction as well as the defocus blur in each colour band was proposed in [Garcia et al., 00].



(a) Two cameras **(b) Two CCD's** **(c) Three CCD's** **(d) Focal length change**

Figura 3.8: The Subbarao camera models.

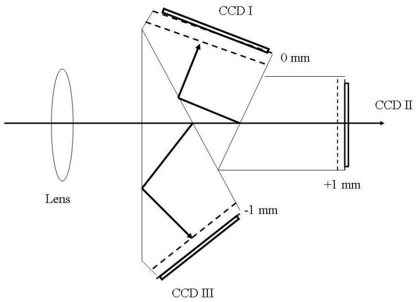


Figure 3.9: The Hiura multifocus camera implementation.

3.5 First Monocular Perception Proposal

Once the monocular perception state of the art, concerning DFF and OF methods, has been presented, their applicability to WMR should be considered. Despite the efforts, a real time DFD range sensor has only been implemented over small scene areas using active illumination to constrain the frequency spectrum [Nayar et al., 96]. Even nowadays, DFD used in robotics could be computationally expensive. The lack of commercial camera models as shown in the previous sub-section is another negative aspect of promoting their use in robotics. The need for images, of the same field of view, acquired with different camera setups, can be considered as the main drawback of DFF methods. However, when DFF methods are compared with stereo vision systems, several DFF features stand out. They include reliability, better robustness, less resolution and sensitivity, and correspondence problem avoidance [Schechner and Kiryati, 98]. Their use in WMR has been reported in [Nourbakhsh et al., 97]. In the work of Nourbakhsh, three different focused images, with almost the same scene and acquired with three different cameras, were used. Less than two depth bits are obtained and robust obstacle avoidance is achieved. However, monocular uses have not been proposed and exploring monocular DFF use and constraints is one of the current research objectives.

The use of optical flow (OF) in mobile robotics has been exploited in multiple applications. Structure from motion (SFM) is a well known research topic of computer vision science. The real time 3D structure of the environment can be attained using nonlinear filters in a monocular image sequence [Campbell et al., 04]. The aim of the present work is to explore qualitative OF methods when monocular DFF approaches do not provide enough information to achieve proper WMR navigation.

When the computational cost is considered, the photometric optics (DFF methods) make less effort than geometric optical methods (OF and stereo vision methods). The use of sensor fusion information is also analysed. Thus, knowledge of the robot position or speed can reduce the OF search time when static obstacles are assumed. Additionally, the use of sensor fusion in textureless or special environments is proposed. Space resolution representations are considered; hence local and global solutions as well as bidirectional search were proposed as interesting research topics [Bruhn et al., 02; Bruhn et al., 05]. Other interesting aspects such as the use of chromatic aberrations are not considered in the present research. The need for a wide field of view to plan the WMR movements increases the DOF (depth of field). Thus, camera configuration is not appropriate for these computations.

Chapter 4

Model Based Predictive Control Strategies

4.1 Introduction

Engineering and Science advances are highly related with automatic control advances. The control science is based on the model of the system knowledge. Usually, linear control assumes lineal models and lineal feedback control laws [Ogata, 93]. Nowadays, the majority of control laws are computed by digital system and hence discrete system control should be considered [Ogata, 96]. Therefore, the classic continuous lineal control theory should be considered as an important source of information due to the relevant research developed. In this sense, it becomes an important engineering objective to obtain accurate dynamic models suitable for controlling the system. There are two main methods for obtaining the system models:

- Mathematical modelling. Physical, chemical, or economical laws are used in order to describe the system dynamics.
- System identification. The model is obtained by a set of experiments that allow computing the parameters of the model.

System identification is usually used due to the complexity of the mathematical modelling in many cases. However, some *a priori* knowledge on the physical laws is useful for planning the experiment design as well as choosing the identification methodology [Luo, 01].

In this chapter, it is presented firstly the basic system identification methodology. Once it is attained the problem is focused on the WMR control by using lineal control laws with an interval of validity [Vehí et al., 01]. Furthermore, the use of MPC (Model Predictive Control) in order to control the platform is the main objective that focuses on the control research developed in this work.

4.2 The system identification

Based upon the *a priori* system knowledge, the selection of the model structure to be used in the parameter identification should take into account the following aspects:

- Sample time
- Number of measures during the experiment
- Delay
- Open or closed loop identification
- Data filtering

Usually, the identification methods are classified into parametric and non parametric ones [Lju, 89]. In the present research, it is focused on parametric methods, in which the objective is to find the model parameters by using a set of experimental tests that included relevant aspects as: transient analysis, frequency response, spectral analysis, correlation analysis [Norton, 86].

4.2.1 Linear difference equation-ARX models

Denoting the system's inputs and output at time t by *u(t)* and *y(t)*, respectively, a basic relationship between the input and output is a linear difference equation:

$$y(t)+a_1y(t-1)+\ldots+a_ny(t-n)=b_1u(t-1)+\ldots+b_mu(t-m) \qquad (4\text{-}1)$$

It is noted that equation (4-1) is expressed in discrete time, since the data are collected by sampling. Equation (4-1) can be used to determine the next output value:

$$y(t)=-a_1y(t-1)-\ldots-a_ny(t-n)+b_1u(t-1)+\ldots+b_mu(t-m) \qquad (4\text{-}2)$$

More compact notation is attained by using vectors representation.

$$\theta=[a_1,\ldots,a_n,b_1,\ldots,b_m]^T \qquad (4-3)$$

$$\varphi(t)=[-y(t-1),\ldots,-y(t-n),u(t-1),\ldots,u(t-n)]^T \qquad (4-4)$$

Hence, the problem consists in determining the parametric vector θ. Once the model is determined outputs can be estimated by the model:

$$\hat{y}(t)=\varphi(t-1)\theta \qquad (4\text{-}5)$$

4.2.1.1 Linear least square method

The parameter estimation is done by using the least square method. Thus, the inputs and outputs of the system are recorded over a time interval $1\leq t\leq N$. Hence, the following data are given:

$$Y=[y(1),y(2),\ldots,y(N)]^T \quad \Phi=[\varphi(1),\varphi(2),\ldots,\varphi(N)]^T \qquad (4\text{-}6)$$

Number of measures in practical applications should be bigger than the number of parameters to recognise. The difference between the measured and estimated (or calculated) values is given by:

$$\varepsilon(i)=y(i)-\hat{y}(i)=y(i)-\varphi(i)\theta \qquad (4\text{-}7)$$

Denote E as the error vector, then the estimation is done by using (4-5) in order to fit the calculated values as accurate as possible to the measured outputs by the least square method.

$$\min V(\theta,t)=\frac{1}{N}\sum_{i=1}^{N}(y(t)-\hat{y}(t))^2=\frac{1}{N}\sum_{i=1}^{N}(y(t)-\varphi(t-1)\theta)^2=\frac{1}{N}\|E\|^2 \qquad (4-8)$$

Then, the vector $\varphi(t)$ can be found by using the identification toolbox of MATLAB.

Model structures as (4-5) are linear in θ that are known in statistics as linear regression, and the vector $\varphi(t)$ is called the regression vector. The model structure (4.1) has the standard name ARX-model (Autoregression with extra inputs), in which the regression vector $\varphi(t)$ contains the past values of the input and output variables.

4.2.2 Experiment design

The objective of an experiment design is to make a collected data set as informative as possible with respect to the models using the data [Van Overschee and Moor, 96]. Thus, some basic points should be considered:

- The input signal should expose all the relevant properties of the system.

- The input should be rich in frequency that contains the different number of frequencies greater than the order of the model to be built. A good choice is to let the input to be random and consequently produce all the frequencies. The use of PRBS (Pseudo Random Binary Signals) inputs is usually a good choice; hence the whole frequency spectrum is excited.
- The input signal switching should allow the output more or less to settle.
- A typical good sample frequency is 10 times of the system bandwidth that corresponds to 5-7 samples along the rise time of a step response.

The model selection and validation are important aspects that should be considered. Thus, the quality of the model is validated on several models in order to verify how well it can reproduce the behaviour of a set of new data; i.e., comparing the estimated output generated by the model with the real output obtained by the experiments.

The residual analysis from identification process consisting of prediction errors that the model could not explain is another important aspect. The model has not squeezed out all relevant information about the system from the data. It is good practice to always check the residual for these dependencies.

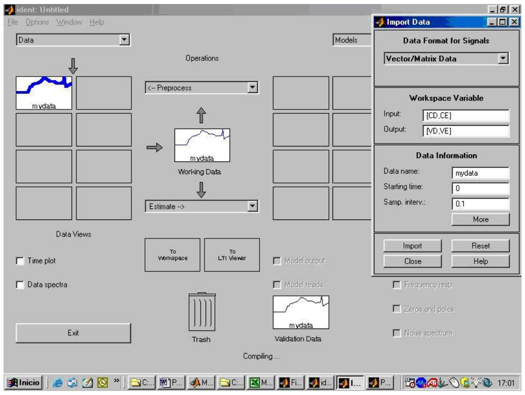
4.2.3 Software for system identification

In practice, system identification is characterised by heavy numerical calculations that determine the best model in each given class of models. This is mixed with several user choices, trying different model structures, filtering data and so on. In practical applications it is needed a good software support. There are many different commercial packages for identification available, such as Mathwork's System Identification Toolbox, and Matrix's System Identification. They offer the following routines:

- Handling of data, plotting, etc. Filtering data, removal drift, choice of data segments, etc.
- Non parametric identification methods. Estimation of covariance, Fourier transforms correlation and spectral analysis, etc.
- Parametric estimation methods. Calculation of parametric estimates in different models structures.
- Presentation of models. Simulation of models estimation and plotting of poles and zeros, computation of frequency functions, and plotting Bode diagrams, etc.
- Model validation. Computation and analysis of residual. Comparison between different model properties, etc.

The existing program packages differ mainly in various user interfaces and by different options regarding the choice of model structure. For example, MATLAB's Identification Toolbox covers all the lineal models structures discussed here, including arbitrarily parameterized linear models in continuous time.

Regarding the user interface, there is a clear trend to make them graphically oriented. This avoids syntax problems and relies more on "click and move", and at the same time avoid tedious menu-labyrinths. Fig. 4-1 shows the graphical interface corresponding to the MATLAB Identification Toolbox.



Data Format for Signals: Vector/Matrix Data

Input: [CD,CE]

Output: [VD,VE]

Data name: mydata

Starting time: 0

Samp. interv.: 0.1

Compiling ...

Figure 4.1: MATLAB Graphical interface for System Identification.

4.3 Model predictive control

The model predictive control, MPC, has many interesting aspects for its application to mobile robot control. It is the most effective advanced control technique, as compared to the standard PID control, that has made a significant impact to the industrial process control [Maciejowski, 02].

The start was related with heuristic methods relayed with industrial applications. In [Richalet et al., 76] describe applications to a Fluid Catalytic Cracking Unit (FCCU) main fractionator column, a power plant steam generator and a Poly-Vinyl Chloride (PVC) plant. All of these examples are constrained multivariable processes. The main fractionator example involved controlling key tray temperatures to stabilise the composition of heavy and light product streams. The controller adjusted product flow-rates to compensate for inlet temperature disturbances and to maintain the level of a key internal tray. The power plant steam generator problem involved controlling the temperature and pressure of steam delivered to the turbine. This application is interesting because the process response time varied inversely with load on the system. This nonlinearity was overcome by executing the controller with a variable sample time. Benefits for the main fractionator application were reported as \$150,000/yr, due to increasing the flow-rate of the light product stream. Combined energy savings from two columns in the PVC plant were reported as \$220,000/yr.

In the next subsections the basic ideas of MPC are introduced. The academic point of view is also briefly commented; considering aspects as nonlinear optimization, stability and robustness. Finally, the MPC applicability for WMR is also presented.

4.3.1 The model predictive control basic ideas

MPC usually contains the following three ideas [Camacho and Bordons, 02]:

- The model of the process is used to predict the future outputs along a horizon time.
- An index of performance is optimized by a control sequence computation.
- It is used a receding horizon idea, so at each instant of time the horizon is moved towards the future. It involves the application of the first control signal of the sequence computed at each step.

The receding horizon strategy is illustrated at Fig. 4.2.

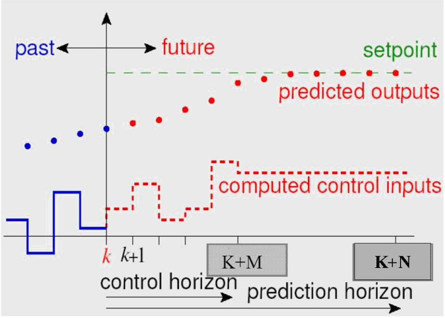


Figure 4.2: The MPC strategy.

The predicted future outputs $\hat{y}(k+n|n), n=1\ldots N$ for the prediction horizon N are computed at each instant of time t using the process model. The future outputs depend upon the known values up to instant t (past inputs and outputs), including the current output and the future control signals and outputs.

The sequence of future control signals is computed and optimised by a performance criterion. Usually it is minimised the error between a reference trajectory and the predicted outputs. Normally the control effort is included in the performance criterion.

Only the current control signal is transmitted to the process. At the next sampling instant the output is measured and it is repeated all the sequence.

4.3.1.1 Advantatges and disadvantatges of MPC

The main advantatges are:

- The concepts are intuitive and attractive to industry.
- Can be used to control a wide range of processes.
- Can deal with SISO and MIMO systems.
- The system constraints can be properly considered in the optimization process.
- It is an open technology which allows future extensions.

The unique important disadvantage is the need of an appropriate model of the plant.

4.3.1.2 The MPC algorithms

Once the model is obtained, the concept is based on minimising a cost function, related to the objectives, through the selection of the optimal inputs. In this case, the cost function can be expressed as follows:

$$J(n,m)=\min_{\left\{U(k+i|k)\right\}_{j=0}^{j=m-1}}\left\{\begin{array}{l}\left[X(k+n|k)-X_d\right]^T P\left[X(k+n|k)-X_d\right]\\+\sum_{i=1}^{n-1}\left[X(k+i|k)-X_d\right]^T Q\left[X(k+i|k)-X_d\right]\\+\sum_{j=0}^{m-1}U^T(k+i|k)RU(k+i|k)\end{array}\right\} \tag{4-9}$$

The first term of (4-23) is referred to the desired coordinates achievement, the second to the trajectory that should be followed, and the last one to the input signals. The parameters P, Q and R are weighting parameters that express the importance of each term. The system constrains are also considered[Wan, 07]:

$$\left\{\begin{array}{ll}\left|U(k+i|k)\right|\le G_1 & \alpha\in[0,1)\\\left\|\left[x_{k+i},y_{k+i}\right]-\left[x_o,y_o\right]\right\|\ge G_2 & \\\left\|\left[x_{k+n},y_{k+n}\right]-\left[x_d,y_d\right]\right\|\le\alpha\left\|\left[x_{k+n},y_{k+n}\right]-\left[x_d,y_d\right]\right\| & \end{array}\right\} \tag{4 - 10}$$

The limitation of the input signal is taken into account in the first constraint. The second constraint is related to the obstacle points where the robot should avoid the collision. The last one is just a convergence criterion.

The MPC algorithms run in the following steps:

- To read the actual position
- To minimise the cost function and to obtain a series of optimal input signals
- To choose the first obtained input signal as the command signal.
- To go back to the step 1 in the next sampling period

When the model is nonlinear, the minimization of the cost function is a nonlinear problem, and then following equation should be verified:

$$f(\alpha x+\beta y)\leq \alpha f(x)+\beta f(y) \qquad (4-11)$$

It is a convex optimization problem [Boyd and Vanderberghe, 04].

4.3.1.3 The control horizon

The Figure 4.6 shows the control horizon, where the input can be changed during *M* samples of time. Normally the prediction horizon, *N*, is bigger than the control horizon. Thus, considering the input parameterization there are three choices [Qin and Badgwell, 03]:

- Multiple move option, the sequence of future inputs moves to be spread over a finite control horizon. Fig. 4.3 shows this concept where a separate input adjustment is made for each time point of the control horizon. The system performance improves as M increases, at expense of additional computation.
- The algorithms compute a single future input change, as it is shown in Fig. 4.4 This greatly simplifies the algorithm computation, but involves a closed loop performance sacrifice.
- Future input profile parameterization using a set of polynomial basis functions. It allows a relative complex input profile specification over a large control horizon. Fig. 4.5 represents this strategy.

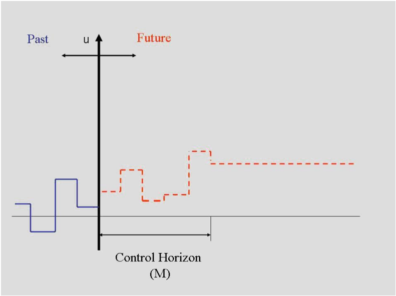


Figure 4.3: Multiple move option.

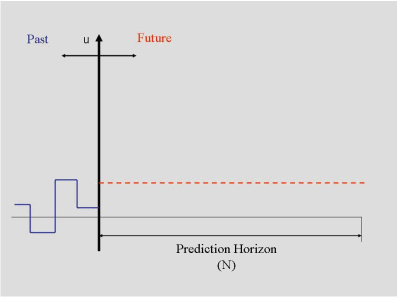


Figure 4.4: Single move option.

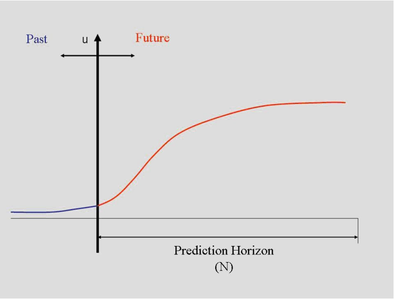


Figure 4.5: Future input parameterization.

4.3.1.4 The input and output trajectories

MPC controllers use four basic options to specify the behaviour, as function of penalised deviations from the objective function:

- The fixed set point option, shown in Fig. 4-6, has penalised deviations in both sides of the objective function. In the practice this type of specification is very aggressive and may lead to very large input adjustments.
- The zone control option consists in designing the controllers in order to keep the system between upper and lower boundaries. Fig. 4.7 shows this strategy. One way to implement zone control is to define upper and lower soft constraints.
- The reference trajectory consist in considering a first or second order trajectory from the current value to the set point, with the speed response determined by one or more time constants. The future deviations from reference trajectory are penalised. Fig. 4.8 depicts this option.
- The funnel option allows separating specification of upper and lower trajectories with different dynamics and set points. In the controller objective, only deviations above the upper trajectory and deviations below the lower trajectory are penalised. This provides additional freedom during the transient time that the controller can utilise in other tasks. Fig. 4-9 shows this strategy.

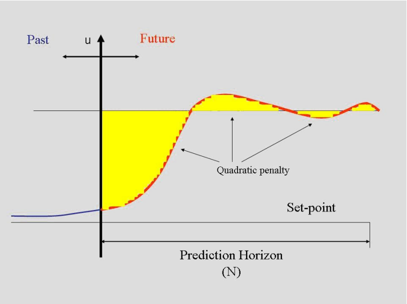


Figure 4.6: Set-point option.

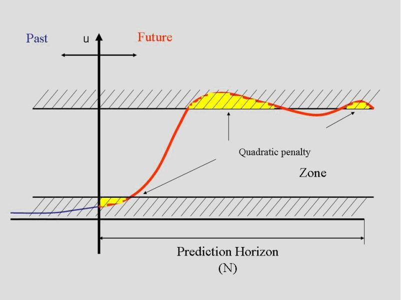


Figure 4.7: Zone option strategy.

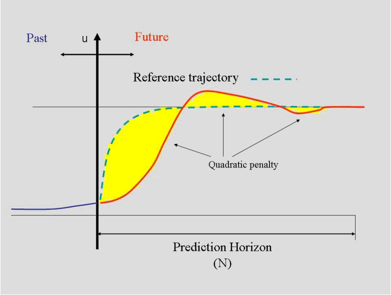


Figure 4.8: Reference trajectory option.

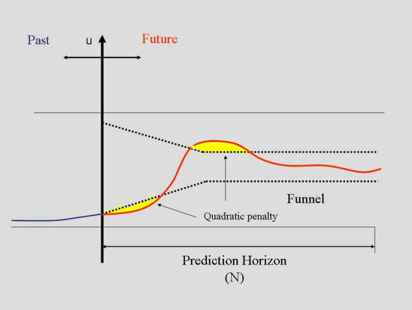


Figure 4.9: Funnel strategy.

4.3.2 Some Aspects of the MPC Academic Research

In this subsection the subject is focused from a closer academic point of view [Camacho and Bordons, 02]. Hence, some aspects should be pointed:

- QP (quadratic program) and LP (linear programming) algorithms are important research topics.
- The model knowledge is an important issue. Thus, Kalman filter and Extended Kalman Filter, (EKF), improve the results achieved. Adaptation control techniques depict an interesting research area.
- Interval analysis obtaining different linear grid with different control strategies.
- Hybrid systems are considered. Hence, the system is considered as a set with different process phases, models, or objectives. Thus, different states with different behaviours are considered; as i.e. start, stop, or the steady states can have different dynamics.

The horizon of prediction is an interesting aspect to consider. Hence, analysis for nonlinear constrained system that include a variable time horizon has reported robustness and less computational effort. The use of an inequality in the terminal constraints was also proposed [Michalska and Mayne, 93]. The idea of moving horizon estimation was also developed in [Zimmer, 94].

The optimization problem for nonlinear system can be considered as a convex problem. It can be tacked using interval analysis [Limon et al., 05]. Convex problem can be solved using a set membership estimation for piecewise affine discrete time systems. The use of piecewise affine can approximate the nonlinear problem by a set of different zones. Hence, the set can be represented by zonotopes. The volume of zonotopes is minimised each sample instant solving a convex optimisation problem [Alamo et al., 03]. When the problem is not convex it is difficult to solve. Hence, different methods have been developed for dealing with such special cases. The stability is an important aspect to be considered. Thus, an infinite horizon controller is developed to allow input and state constraints in a receding horizon [Rawling and Muske, 93]. The feasibility of the constraints guarantees closed loop stability for all the choices of the tuning parameters in the control law. The infinite horizon needs may increase the on-line computation demand. In order to overcome this difficulty, the idea of a quasi-infinite horizon was suggested [Chen and Allgöwer, 97]. The quasi-infinite approach consists in using a finite horizon cost function that includes a terminal penalty term. In [Rao et al., 03] is proposed a general theory for constrained moving horizon estimation. The asymptotic and bounded stability conditions are established by using state estimate optimization with a moving horizon approximation.

The robustness is an important issue that should be analysed. Hence, it can reduce the required tuning and test time of industrial applications. Recent robust research works can be found in [Rakovic et al., 03].

4.3.3 The MPC for WMR control

Real time mobile robot MPC implementations have been developed using global vision sensing [Gupta et al., 05]. In [Küne et al., 05], it was studied the MPC based optimal control useful for the case when nonlinear mobile robots are used under several constraints, as well as the real time implementation possibilities when short prediction horizons are used. In general, the global trajectory planning becomes unfeasible since the sensorial system of some robots is just local. By using a MPC, the idea of the receding horizon can deal with the local sensor information.
The MPC is based on minimizing a cost function, related to the objectives, through the selection of the optimal inputs. In this case, the cost function can be expressed as follows:

$$J(n,m) = \min_{\left\{U(k+i|k)_{j=0}^{j=m-1}\right\}} \left\{ \begin{array}{l} [X(k+n|k) - X_d]^T P [X(k+n|k) - X_d] \\ + \sum_{i=1}^{n-1} [X(k+i|k) - X_d]^T Q [X(k+i|k) - X_d] \\ + \sum_{j=0}^{m-1} U^T(k+i|k) R U(k+i|k) \end{array} \right\} \tag{4-12}$$

The first term of (4-26) is referred to the desired coordinates achievement, *$X_d=(x_d,y_d,\theta_d)$,* the second to the trajectory that can be followed, and the last one to the input signals. The parameters *P*, *Q* and *R* are weighting parameters that express the importance of each term. The system constrains are also considered:

$$\left\{ \begin{array}{ll} |U(k+i|k)| \le G_1 & \alpha \in [0,1) \\ \|[x_{k+i}, y_{k+i}] - [x_o, y_o]\| \ge G_2 & \\ \|[x_{k+n}, y_{k+n}] - [x_d, y_d]\| \le \alpha \|[x_{k+n}, y_{k+n}] - [x_d, y_d]\| & \end{array} \right\} \tag{4 - 13}$$

The limitation of the input signal is taken into account in the first constraint. The second constraint is related to the obstacle points where the robot should avoid the collision. The last one is just a convergence criterion.

The cost function can be related to three interesting aspects of the robots as final coordinate achievement, trajectory following, and speed value. Thus by using appropriates parameters *P*, *Q* and *R* the objectives can be achieved.

The input constraints as the use of a single move option present advantages such like the reduction in the computation time and the smooth behaviour of the robot during the prediction horizon. Thus, the set of available input can be reduced to one value, in order to reduce the optimal signal value search

4.4 Model predictive control proposal

In this work the control strategy selected is based on MPC techniques where the control horizon will be constrained to the perception field of view acquired by a monocular camera. Therefore, the local desired coordinates to be reached are given as an attraction potential field that attracts the WMR towards the goal. The perception system produces data of the obstacles to be avoided during WMR navigation. Dynamic scenarios are assumed consequently the path planning is computed at each perception step and flexible trajectories are tracked. In this context a LMPC (local model predictive control) strategy is proposed. The LMPC proposal consists of the following features:

- The trajectory to be tracked is constrained to a straight line that goes from the robot coordinates from where the last monocular frame is acquired to the free occupancy grid cell that approaches the WMR to the goal, while collision avoidance is performed.
- The horizon of prediction is set to less than one second. Navigation under dynamic environments is mainly reactive. Consequently is not necessary to plan longer trajectories.
- The control horizon is constrained to selecting one single increment value [Maciejowski, 02].

The LMPC proposal can be considered as reactive short term WMR navigation in spite of other proposals [Gupta et al., 05], [Küne et al., 05]. Therefore, it is suitable for dynamic environments with very low computational cost.

Chapter 5

WMR Platform Description

5.1 Introduction

This chapter presents the available WMR platforms that are used in order to test the research developed in this work. They are differential driven WMR with a free rotating wheel. The platform objectives are the creation of a low cost open mobile robot platform PRIM (Platform Robot Information Multimedia). The philosophy of the robot includes important goals which are the use of this platform as a full open educational tool, as well as an available research platform. A third objective, as result of the research developed is their use as a mobile multimedia information point (MMIP), in commercial applications [Gluer and Schmidt, 00]. Fig. 5.1 depicts respectively the PRIM I and PRIM II available platforms.

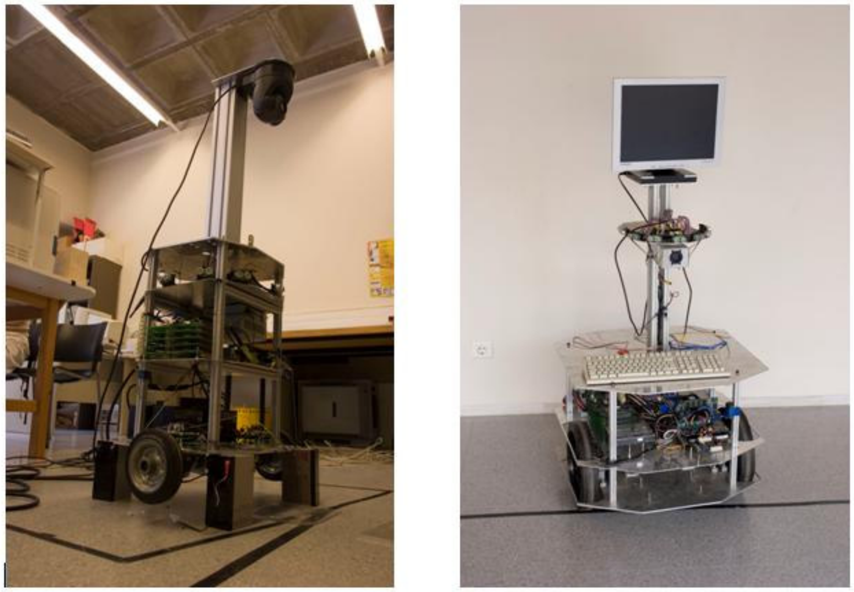


Figure 5.1: The picture on the left shows the PRIM I that is mainly used as teaching and research activities platform. The picture on the right shows the platform PRIM II that was designed with the aim of to be used in social events as multimedia information platform.

The educational goal should be considered under the roof of our university reality, where study and understanding in several academic fields such as electronics, mechanics, control, and computer science, can be reinforced by the use of these kinds of platforms. The uses of such open platforms rather than other kinds of commercially available mobile robot, which act as closed platforms is clear; students working with open platforms have the full platform access. Hence learning is attained in electronics, control and modelling, sensor fusion, and computer science among other areas. However this renewed path should not only be attained at first degree level, but it should also include research activities coming from the mutual interaction and integration between subjects, and the high level control strategies. The use of low cost open platforms allows the students to develop understanding in a multidisciplinary context.

It is obvious that the design and implementation of such systems, is also of interest from a commercial point of view. Thus, this idea is not only restricted to the university community. The creation of an open ethic product is another objective, and has been supported by Catalan association "*Ateneu Informatic*". In the context of this institution the idea of building a high performing low cost autonomous multimedia information tool began. The idea consisted in fusing robotic technology, telecommunications, and computer science into just one technological product. The product must be ethical; it is an MMIP, but it is also a declaration of principles, before the market incorporation. Their ethical and esthetical characteristics should be defined, involving public administration, universities, and non governmental associations. In this context of an ethical product, the robot was presented at the 2004 Universal Culture Forum of Barcelona at the 18th IAVE World Volunteer Conference from the 17th to 21st of August of 2004, (see Fig. 5.2) [Izquierdo et al., 04]. To summarise, it was concluded that the ethical and esthetical characteristics should cover all the needs that the third sector could possibly demand.

Figure 5.2: The mobile platform PRIM I at 18th IAVE in Barcelona.

The particular motivation to work with these platforms arose from the following characteristics: platform flexibility, low cost, open system, and multiple applications.

The robots PRIM are designed for indoor navigation, and previously gained knowledge from our experience in robotics was taken into consideration during its construction [Batlle et al., 95]; [Salvi et al., 96]. In the following section the basic mechanical, electronic, and architecture characteristics are presented.

5.2 Mechanical Description

The robot structures are made from aluminium. Fig. 5.3 shows the different levels where different parts are placed. The both robots share some basic characteristics. Thus, at the first level there are two differential driven wheels, controlled by two dc motors, and a third omni-directional wheel that gives the third contact point with the floor. On the second level there is the PC computer, and on the third level the specific hardware, the sonar sensors are placed. The forth level could be used, according to the flexibility of the system, to place the machine vision system and/or the multimedia set up depending on the platform application.

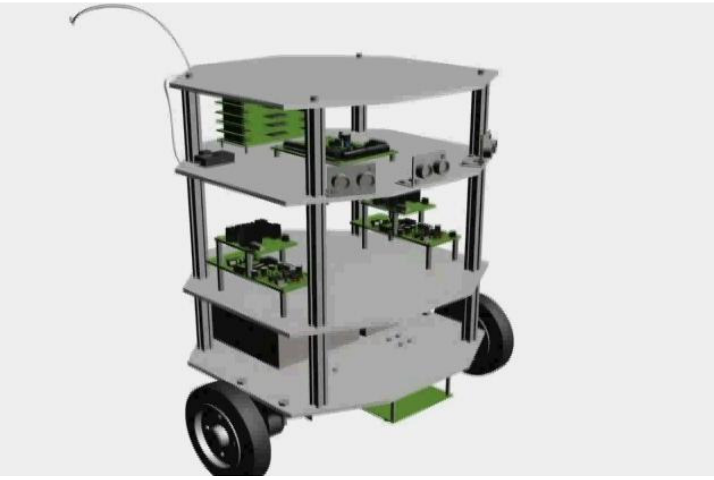


Figure 5.3: The basic hardware structure of the robots.

Table 5.1 summarises the basic mechanical description of PRIM I and PRIM II.

Table 5.1: The basic description related to the robot mechanics.

Features	PRIM I	PRIM II
Wide	**580 mm**	660 mm
Large	**400 mm**	625 mm
Height	**1200 mm**	1500 mm
Distance between wheels	**560 mm**	540 mm
Diameter of the wheels	**160 mm**	300 mm
Weight	**20 Kg**	45 kg
Maximal speed	**0.48 m/s**	0,9 m/s
Motor max. cont. torque	**131 mNm**	131 mNm
Gear reduction	**86:1**	86:1
Total robot force	**141N**	141N

The platform PRIM II, in essence, has the same hardware of PRIM I. The main differences are the speed and weight. The bigger speed is due to the larger diameter of the wheels, and the difference of weight is caused by a bigger inner space that allows a major degree of autonomy.

5.3 Electrical Description

The two platforms have the same basic electrical hardware; hence the description depicted in this section embraces both platforms.

The system can be powered by 12V dc batteries or by an external power source through a 220V ac, and a switch selects both modes of operation. The 12V provided by the batteries or the external source of 220V are transformed by a power PC source of 12V into the wide range of dc needed by the system. The battery system is actually composed of a set of 4 batteries of 12V with 7Ah each that provide between 3 and 4 hours of autonomy. The space remaining on the platform PRIM II means that the system could be expected to have up to 8 hours of autonomy.

The robot is equipped with two DC motors that act over the two independent traction wheels.

The robot has the following sensorial system:

- Two encoders connected to the rotation axis of each DC motor.
- An array of sonar composed by 8 ultrasounds sensors.
- A machine vision system composed by a monocular camera.

The meaningful hardware consists of the following electronic boards:

- The dc motor power drivers based on a MOSFET bridge that controls the energy supplied to the actuators.
- A set of PCB (printed circuits boards) based on PLD (programmable logic devices) act as interface between the PC system, the encoders, and the dc motors. The interface between the PLD boards and the PC is carried out by the parallel port.
- A µc processor board controls the sonar sensors. The communication between this board and the PC is made through a serial port. This board is also in charge of a radio control module that enables the tele-operation of the robot.
- The PC is the core of the basic system, and it is where the high level decisions are taken.

The PLD boards generate 23 khz PWM (pulse width modulation) signals for each motor and the consequent timing protection during the command changes. This protection system provides a delay during the power connection, and at the change of the rotation motor sense. A hardware ramp is also implemented in order to give a better transition between command changes. The speed is commanded by a byte and thus it can generate from 0 to 127 advancing or reversed speed commands by the use of the parallel port as interface between the PC and the PLD's boards. The PLD boards also measure the pulses provided by the encoders, during an adjustable period of time, giving to the PC the speed of each wheel at every 25ms. An absolute counter provides the counts in order to measure the robot position by the odometer system The absolute position of each encoder is also measured by two absolute counters used in order to measure the position and orientation of the robot by the odometer system. The shaft encoders provide 500 counts/rev since encoders are placed at the motor axes; it means that the encoders provide 43000 counts for each turn of the wheel.

The system has two modes of operation; tele-operated, and autonomous mode. In autonomous mode when the system starts running, the board disables the PLD boards until it has received from the serial port a control word. This prevents the noise reception going into the PLD boards, through the parallel port. Moreover, the µc has the control of the sonar sensors, so for each sensor a distance measure is obtained. The ultrasound sensor range is comprised between 3cm and 6m. The data provided by these boards is gathered through the serial port in the central computer. The rate of communication with these boards is 9600 b/s. The worst measuring time for each sonar sensor is 36ms in order to achieve robust and feasible control timing. The data transmission sends only the two actualised sonar distances. Hence, the total distance information reaches in each 400ms.

The flexibility of the system allows different hardware configurations as a function of the desired application and consequently the ability to run different programs on the µc or PLD boards. However, the platform is actually being tested under the configuration explained previously. The open platform philosophy is reinforced by the use of the similar µc and PLD boards that are used as teaching tools at our school.

5.4 System Architecture

This section introduces the WMR system architecture. It is considered as an open system that is used as an educational tool. The proposed educational open framework has its advantages in many aspects. First, the use of a structure similar to that employed by students at the laboratories can enable their easy understanding and prototyping of new low level hardware. Also, the design flexibility allows the development of different applications in the context of an open platform. Furthermore, the reinforcement of the teaching activities can be achieved through the knowledge integration of different subjects [Pacheco et al., 2008A]. Furthermore, special motivation of this work arises from the fact of using such system as a research tool exploring other capabilities.

In this context, it is presented the proposed system architecture as well as the proposed software support tools.

5.4.1 The proposed system architecture

The main decision system arises from the PC that controls the hardware shown at this point. The data gathering and control by digital computer is set to 100ms. The platform acts as an open system that allows the connection of other PCs though a LAN. These possibilities cover two complementary aspects of the system that are the multimedia point of information and the machine vision system, as an advanced sensor system; Fig. 5.4 shows the system architecture.

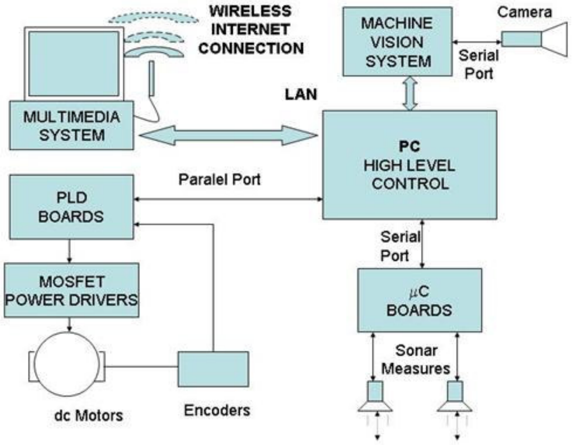


Figure 5.4: The system architecture.

The multimedia system is composed of a PC with a tactile screen that allows the interaction with people. The idea of this application as a multimedia information point means that this computer should be configured with the software information that the users applications demand. A wireless Internet connection allows the communication with the whole world, and a set of multiple possibilities.

The machine vision system is composed of the following components: A remote camera with motorised focus, iris, and zoom control by a serial port, two steep motors that control the pan and tilt position of the camera, and specific hardware boards running on a PC, exclusively used by the machine vision system. The system is connected to the main control system through a LAN.

5.4.2 The proposed software support tools

The software of the high level control PC is implemented on C language and runs under the LINUX operative system. The software implemented has access through the parallel port to the PLD boards and through the serial port to the microcontroller boards.

The PLD boards allow real time access to the encoder data as well as to the power drivers. The software of the PLD boards has been implemented by using the Synario development tools, which allow to design by using basic hardware description languages as schematics and ABEL (advanced Boolean equations language). The software can be downloaded to the PLD boards by using the parallel port.

The microcontroller boards have access to the sonar system and switch on the PLD boards once the PC control system has started with the right configuration. The software of the microcontroller boards was implemented in C language. The ICC-8051 compiler (Intel C compiler) was used. Download was developed by using the serial port and the MKT (microcontroller software tools) of Maxim.

The LMPC source code has been implemented in C language on the high level PC system. It consists of straight line trajectory tracking. Moreover, the monocular machine vision system runs on an external PC by using MIL (Matrox Imaging Libraries) utilities under Windows operative system. The source code for the machine vision system has been implemented by using VBASIC language. The monocular machine vision system and the control system have communication through an external LAN. Figure 5.5 depicts the system architecture and software support tools used in this work.

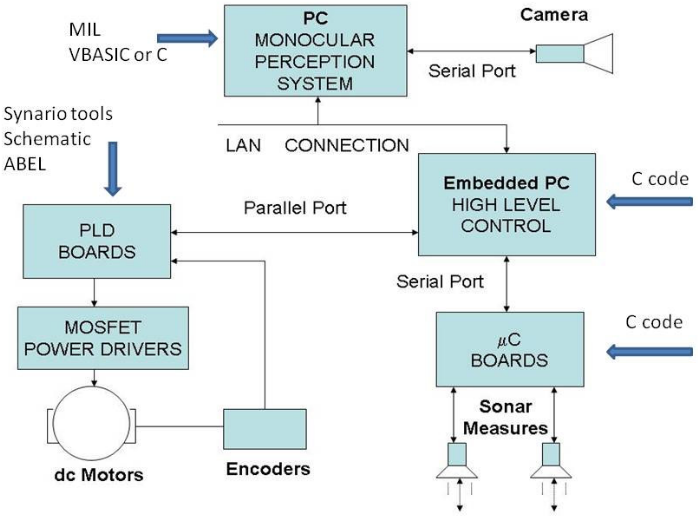


Figure 5.5: Software tools and system architecture.

Chapter 6

The Developed Monocular Machine Vision System

6.1 Introduction

This chapter introduces the computer vision work developed until now. Arising from the DFF techniques interest; research work was addressed testing 3D reconstruction performance as well as their use in WMR's when static positions are attained. The second work explored the main constraint that consists of the needs of having an image sequence to obtain 3D information. Results show that to search within an image sequence can be avoided when homogeneous floor radiance is attained. Thus, the knowledge of these values can be considered as a method to separate the work space from the obstacles. However, experimental results show that false obstacles are detected. Therefore, qualitative optical flow analysis is presented as an available methodology for solving the false obstacle problem.

Qualitative obstacle structure analysis is introduced through the time integration of image sequences that can afford information for WMR navigation. Moreover, 3D obstacle structure can be inferred in future works. The occupancy grid framework is used as a way to relate different frames that share a common scene part perception. By using the floor and camera models the prediction of the future field of perception can be done, by using the previous and present robot coordinates provided by the odometer system. The matches between predictions and frames are proposed as clues for revealing the floor structure when the aperture problem is not presented. In this sense, scene obstacles are expected to be met when discrepancies occur.

6.2 3D reconstruction by DFF methods

The DFF & DFD methods relate the camera system with the scene characteristics through the paraxial optics [Horn, 68]. Unfortunately commercial cameras do not have this information available, thus, a previous camera calibration should be done before the use of these techniques. However commercial cameras are controlled through motor positioning and without any relation between the motors position and the optical set-up. The transference function between the motor positions and the optical characteristics can be obtained by a set of experiments. Furthermore, for each range of distance an ideal set up configuration is obtained. Once the calibration is done the system performance is experimented using DFF techniques. In this work it is used a commercial camera that allows modifying the focus and the zoom. The skills of the results are analysed in the context of the indoor navigation of a mobile robot that enables the robot for a partial 3D understanding of the main goal scene [Pacheco, et al., 04].

6.2.1 Calibration of the optics parameters

In this subsection it is presented the set of calibration experiments designed in order to compute the depth of the scene using DFF techniques. These techniques are based on the thin lens formula, thus the whole lenses system of the camera has its own equivalent thin lenses model. The model of thin lenses relates the focal length, *f,* with the distance between the lenses and an object point*, u,* and the distance between the lenses and the convergence point of the light rays, *v*.

$$\frac{1}{f}=\frac{1}{u}+\frac{1}{v} \qquad (6\text{-}1)$$

When the point is focused $D = u + s$, (s is the image detector distance, and D the distance between the scene object and the image detector, see Fig. 6.1), the following equation is obtained:

$$D=\frac{s^2}{s-f} \qquad (6\text{-}2)$$

The equation (6-2) can be used for obtaining the distance when the optical set up of the camera is known. From the equation (6-2) it is obtained the equation that expresses the s distance as a function of the focal length, *f,* and the distance *D*.

$$s=\frac{D}{2}-\frac{1}{2}\sqrt{D^2-4Df} \qquad (6\text{-}3)$$

Fig. 6.1 shows the optical paraxial model of the camera, and the most important parameters. Unfortunately this optical information in the majority of commercial cameras is not available. The commercial camera used in this work is EVI-311 from Sony, it has only available the focal lengths for the minimal and maximal zoom. The parameters used, in this calibration work, are the zoom and the focus motor position.

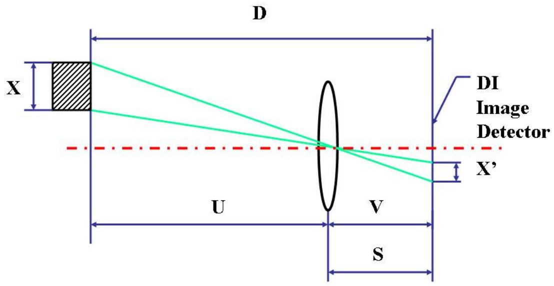


Figure 6.1: Optical model of the camera.

From equation (6-2) it can be obtained an expression of the distance measure error, e_D, as function of the focus value, *f*, the distance error position, e_s, and the image detector distance *s*.

$$e_D=\left(\frac{(s+e_s)^2}{(s+e_s)-f}\right)-\left(\frac{s^2}{s-f}\right) \qquad (6\text{-}4)$$

When the focus value is fixed and the position error of the image detector is a constant, equation (6-4) shows that the error in the distance measure increases with the distance of the object. Thus, it is concluded that it is convenient to work with the maximal focal length value position possible. The work area of the camera is known as field of view [Horn, 98]. This parameter can be expressed with two angles, as shown in Fig. 6.2. Thus, the maximal values of the *x*, *y* coordinates can be expressed as:

$$\left|x_{\max}\right| = (z - s)\tan\left(\frac{\beta}{2}\right) \qquad (6\text{-}5)$$

Considering that for normal cameras approximately $f = s$, and using f instead of s, (Fig. 6.2), the following equations are obtained:

$$\alpha = 2a\tan\left(\frac{Y_d}{2f}\right) \qquad (6\text{-}7)$$

$$\left|y_{\max}\right| = (z - s)\tan\left(\frac{\alpha}{2}\right) \qquad (6\text{-}6)$$

where, X_d and Y_d represent the size of the image detector. Using the focal length values of the minimal and maximal zoom positions in equations (6-7) and (6-8) the dimensions of the image sensor can be found.

$$\beta = 2a\tan\left(\frac{X_d}{2f}\right) \qquad (6\text{-}8)$$

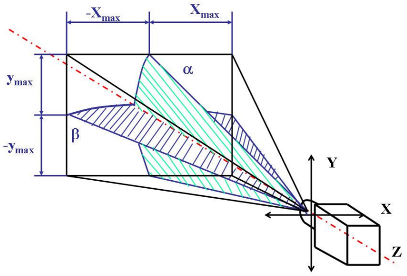


Figure 6.2: Field of view of the camera.

Once these values are known we can obtain the field of view as function of the focal length. The focal length can be obtained measuring the experimental angles, α and β, and using the equations (6-7) and (6-8) for a set of different zoom values. From these values using LSQ method a third order polynomial can be obtained. It relates the zoom position of the motors Z, with the focal length.

$$f = 4.38x10^{-8}Z^3 - 1.955x10^{-5}Z^2 + 0.0144Z + 5.613 \qquad (6\text{-}9)$$

Fig. 6.3 shows the relation between the zoom position and the focal length. Once the focal length is got, the optimal zoom position associated with the estimated distance of the object can be obtained.

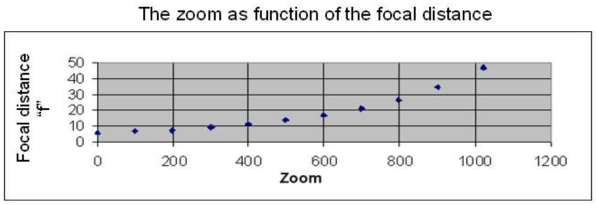


Figure 6.3: Transfer function Zoom-Focal length.

The second step consists of the calibration of the focus motor position; using targets placed at different known distances, for a set of zoom positions. The motor focus position that gives the image with more energy, for the selected zoom position, is considered as the distance of the target. These experiments are tested with three different targets in order to find the more robust distance associated at each focus position. Thus, it is determined a LUT with the distance associated for each focus measure. Sets of statistical parameters related to focus measure are also obtained. Another interesting calibration parameter is the minimal distance differentiable [Nayar and Nakagawa, 94]. It is useful in order to know the minimal step value for obtaining the best-focused image. The results show that the best zoom gives us the better resolution. Thus, for instance, the minimal step for a 550-zoom position is 30 motor steps, and for a 1023 zoom position it is only 5 motor steps.

6.2.2 Focus measure and 3D reconstruction

The DFF methods can be useful for understanding the final scene in this type of applications related to path planning for mobile robots. The 3D reconstruction is done by the search of the maximal value of the focus operator, in a sequence of images of the same scene, as it is shown in Fig. 6.4.

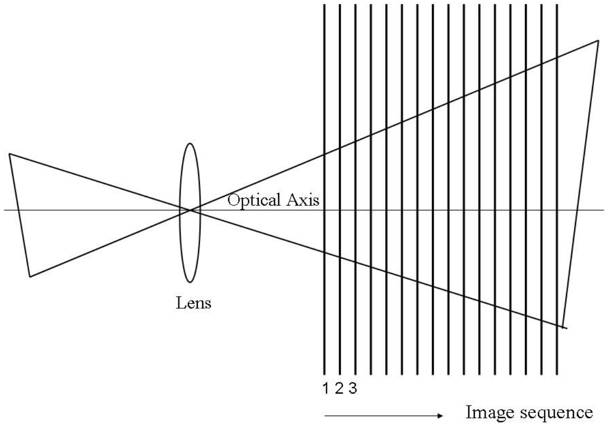


Figure 6.4: Shape from Focus method.

In this work, each image has a different focus set up. These methods are known as Shape from Focus [Nayar, 92]. The focus operator is calculated for the different decomposition windows of the image. Thus, for each window a better focus position is found, so it means that the knowledge of the distance is obtained. The focus operator measures the energy of the image, thus it has the maximal value at the best-focused image. A set of focus operators such as the energy of the grey level variance, the gradient, or the Laplacian of the grey level image has been proposed in the literature [Subbarao et al., 92]. The focus measure is obtained through the convolution of the operators across the selected window. The number of images used is among 10 to 15. For each image a set of windows are selected. However the best focused window is searched. For each window a better-focused position should be found, and then a full 3D map of the scene is obtained.

The position of the focus is found by interpolation near the three best positions acquired in order to avoid an exhaustive research of the best-focused image [Subbarao and Tyan, 96]. The size of the windows is set to 20x20 pixels. The images are normalised by the mean grey value in order to get robustness from the external lighting conditions. A band pass filter is also necessary in order to have a monotonically behavior of the focus operator, as it was seen in Chapter 3 section 3.3.4. The erroneous measures are removed setting a threshold of focus quality and filtering with a 5x5 mean filter. The time of the process has been decreased by the use of the Gaussian space resolution or down sampling techniques [Gonzalez and Woods, 02]. Figs. 6.5 and 6.6 show the 3D perception of the field of view. The distance of the objects is represented in mm by a set of different colors. The lack of texture in some areas, see Fig. 6.5, arises in a lack of information for performing the 3D reconstruction of the scene. It can be solved by the use of active vision systems that increase the system performance, as shown in Fig. 6.6. However, active techniques were not used in the local navigation research presented in this thesis.

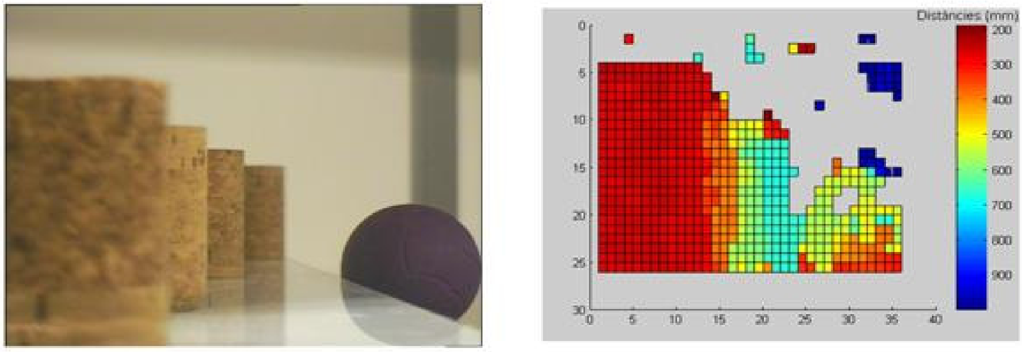


Figure 6.5: Field of view and 3D perception of the work area.

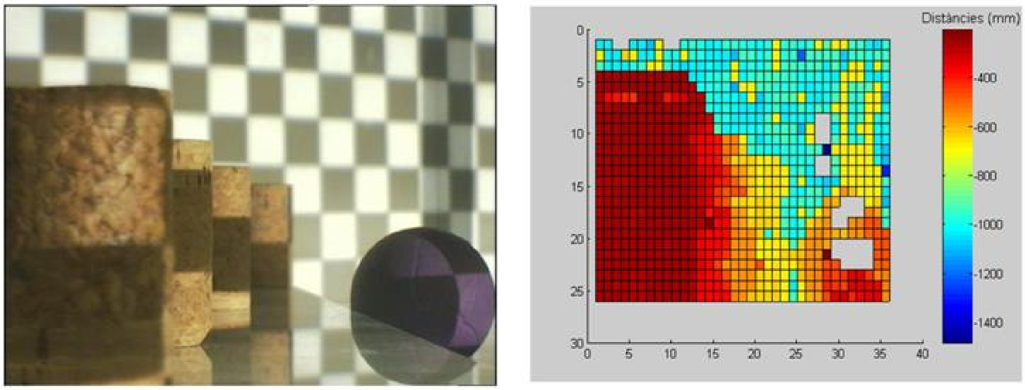


Figure 6.6: Field of view and 3D perception using active vision.

The 3D perception is achieved using DFF techniques. The results are obtained with a reduced field of view and for objects with distances of less than 2 meters. However, their use as a suitable methodology for WMR perception is not appropriate by using the techniques depicted in this section. The main reason is that a wide field of view is necessary for obtaining a large scenario description. Therefore, a wide field of view corresponds to a zoom position where a large focus value is obtained, see Fig. 6.3.

Thus, when large focus values are used there is a lack of resolution and accuracy for objects with distances of more than two meters.

The next section presents studies on the use of DFF techniques when a wide field of view is used. Furthermore, a monocular system and a single frame are proposed.

6.3 Constrained monocular obstacle perception with just one frame

This section presents a monocular perception system tested on WMR. Its significant contribution is the use of a single image to obtain depth information (one bit) when robots detect obstacles [Pacheco, et al., 07]. The use of one single image arises from the fact of monocular perception. The field of view is constrained to the WMR neighbourhood, and wide perception is pursued. 3D reconstruction by searching the best focused window within a sequence of acquired frames, with different focus values, is not appropriate. Therefore, different scenario description will be done due to the WMR movement. Moreover, there is a lack of resolution that arises from the present camera configuration and possible obstacle distances. In this context, the research explored in this section studies the use of one single image, under the present system set up, for obstacle detection. Thus, the obstacles are detected by one single bit (true or false) when only one frame is acquired. Therefore, OPUOF (obstacle perception using one frame) is presented as an important research in this Chapter. However, some constraints referred to the environment should be assumed. Flat and homogeneous floor radiance is assumed. Results emerge from using a set of multi-resolution focus measurement thresholds to avoid obstacle collision. The algorithm's simplicity and the robustness achieved can be considered as the key points of this work. On-robot experimental results are reported and a broad range of indoor applications is possible. However, false obstacle detection occurs when the constraints fail.

6.3.1 Monocular perception algorithms

The implemented algorithms of the machine vision system are based on important assumptions that are generally obtained in normal indoor scenarios, but also in many outdoor scenarios. These constraints are flat and homogenous energy radiance of the floor surface and experimental knowledge of the focus measurement threshold values. Two important aspects, image window size and camera pose, should be considered. The size of windows should be big enough to get energy information. For example, in the work of Surya, images of 150x150 pixels were used, and the focus measures were computed in 15x15 pixel regions in the images [Surya, 94]. The camera pose will set the scenario perspective and consequently the floor position coordinates that should be used in the WMR navigation strategy. Fig. 6.7 shows the camera configuration considered in this work. α, β and φ are angles of the vertical and horizontal field of view and the tilt camera pose respectively.

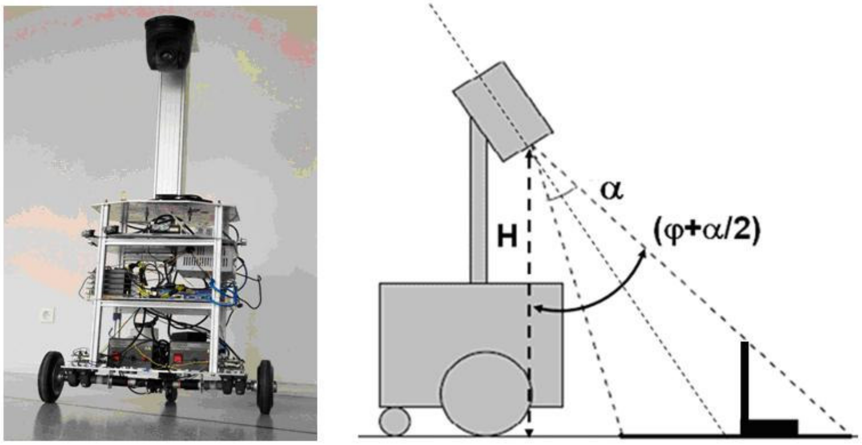


Figure 6.7: The robot PRIM, and the on-robot monocular camera configuration.

The vertical coordinate of the camera is represented by *H*. The relative robot coordinates corresponding to each pixel can be computed using trigonometric relationships and the corresponding knowledge of the camera configuration [Horn, 98]. Using trigonometric relationships, the flat floor scene coordinates can be computed as follows:

$$Y_j = H\tan(\varphi - \alpha/2 + \Delta\alpha) \qquad (6\text{-}10)$$

$$\Delta\alpha = K_j \frac{\alpha}{R} \qquad (0 \le \mathrm{K_j} \le R)$$

$$X_{i,j} = \pm \frac{H}{\cos(\varphi - \alpha/2 + \Delta\alpha)} \tan(\Delta\beta) \qquad (6\text{-}11)$$

$$\Delta\beta = K_i \frac{\beta}{C} \qquad (0 \le \mathrm{K_i} \le C/2)$$

K_i and K_j are parameters used for covering the image pixel discrete space. Thus, *R* and *C* represent the image resolution through the total number of rows and columns. It should be noted that for each row position corresponding to scene coordinates Y_j, there are *C* column coordinates $X_{i,j}$. The above equations provide the available local map coordinates when no obstacle is detected.

The multigrid representation using low-pass filtering processes can improve the surface radiance homogeneity. Scale space representations can reduce the search space, increasing the computation performance [Wang et al., 05]. Therefore, a Gaussian filter is applied to the frames acquired in PAL format, at 768x576 pixels. Three decreasing resolution levels have been used with picture sizes of 384x288, 192x144 and 96x72 [Gonzalez and Woods, 02]. The average image brightness is also computed. In order to achieve better robustness against changes in lightness, brightness normalization is performed by dividing each image's pixel brightness by the mean [Surya, 94].

6.3.2 Experimental results

In this subsection the results from the computation of the homogeneous floor radiance thresholds to detect obstacles in the robot trajectory are presented. These values are validated using previously acquired test images. Once thresholds are calculated, the perception algorithms are tested during robot navigation experiences. False obstacle detection analysis can be carried out using obstacle image test files acquired during navigation. Another interesting parameter that has been analysed is the processing time for each image. The experimental results report robust and effective obstacle detection in typical indoor environments, such as the one shown in Fig. 6.8. The visual position grid where paths should be planned to avoid obstacle collisions is shown in Fig. 6.9.

Figure 6.8: Typical indoor scenario.

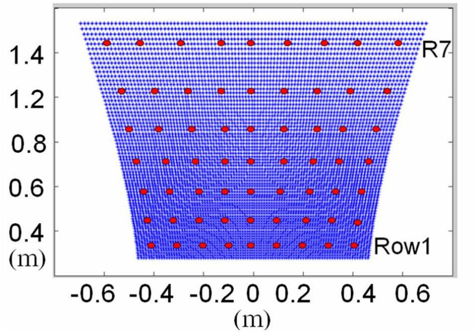


Figure 6.9: Local perception grid coordinates relative to the robot position, at the 96x72 and 9x7 size level. At the 9x7 size level it is possible to distinguish 7 different rows (Row 1 being the closest to the robot).

It is in the neighborhood of those points where the information about radiance is obtained. It should be noted that, using the odometer system, the robot position and orientation are available at a rate of 100ms. Hence, the robot navigation can be tested under different control algorithms, such as heuristic rules, path following under discontinuous control laws, or predictive control techniques which are depicted in Chapter 7 [Pacheco and Luo, 07A]. Fig. 6.10 depicts high resolution, 130x130 pixel windows, corresponding to different floor images used to compute focus measurement thresholds, where the floor texture is clearly visible. This robot environment can be considered as a good indoor benchmark scenario because great variations in light are produced. Light illumination can change from 2000 lx, when light from the sun is clearly present through the windows, to less than 200 lx in the darker corridor zones.

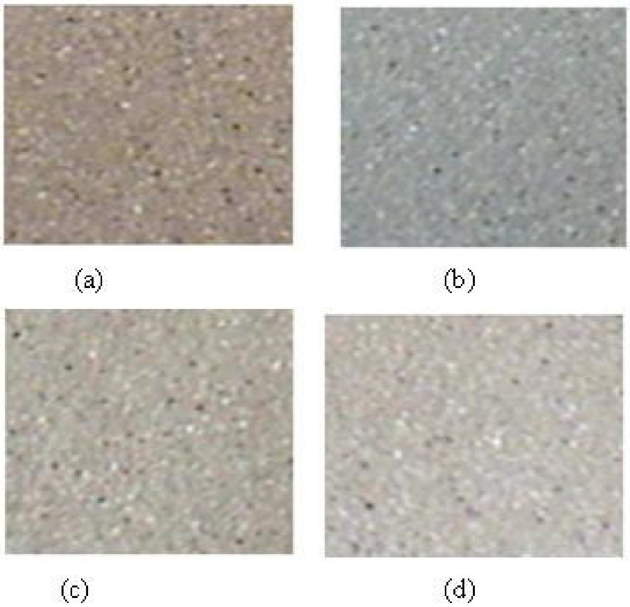


Figure 6.10: Fragments of high resolution floor images (768x576 pixels) under different light conditions corresponding to 200, 800, 1400, and 2000 lx, respectively.

The energy floor radiance measures obtained for each 9x7 visual perception row are shown in Table 6.1. They were computed in a wide range of different light conditions. For each light condition, five samples were used, and the averaged mean results and standard deviation are depicted. The image perspective emerges from a set of multi-resolution thresholds as a function of the camera distances. Note that the table shows larger values at closer 9x7 row positions (row 1 is closest to the robot).

Table 6.1: Modified Laplacian and standard deviation for each 9x7 image row.

Row		200 lx	400 lx	800 lx	1400 lx	1800 lx	2000 lx
1	Mod. Lapl.	$4.5x10^{-2}$	$3.6x10^{-2}$	$3.5x10^{-2}$	$4x10^{-2}$	$3.6x10^{-2}$	$3.8x10^{-2}$
	Std. dev.	$3.1x10^{-3}$	$2.5x10^{-3}$	$2.4x10^{-3}$	$2.7x10^{-3}$	$2.4x10^{-3}$	$2.7x10^{-3}$
2	Mod. Lapl.	$4.6x10^{-2}$	$3.5x10^{-2}$	$3.5x10^{-2}$	$3.9x10^{-2}$	$3.3x10^{-2}$	$3.7x10^{-2}$
	Std. dev	$3.1x10^{-3}$	$2.3x10^{-3}$	$2.4x10^{-3}$	$2.6x10^{-3}$	$2.2x10^{-3}$	$2.6x10^{-3}$
3	Mod. Lapl.	$4.6x10^{-2}$	$3.4x10^{-2}$	$3.4x10^{-2}$	$3.8x10^{-2}$	$3.4x10^{-2}$	$3.5x10^{-2}$
	Std. dev	$3x10^{-3}$	$2.4x10^{-3}$	$2.3x10^{-3}$	$2.6x10^{-3}$	$2.3x10^{-3}$	$2.3x10^{-3}$
4	Mod. Lapl.	$4x10^{-2}$	$3.2x10^{-2}$	$3.2x10^{-2}$	$3.6x10^{-2}$	$3.2x10^{-2}$	$3.3x10^{-2}$
	Std. dev	$2.8x10^{-3}$	$2.2x10^{-3}$	$2.2x10^{-3}$	$2.5x10^{-3}$	$2x10^{-3}$	$2.1x10^{-3}$
5	Mod. Lapl.	$3.6x10^{-2}$	$3x10^{-2}$	$3x10^{-2}$	$3.4x10^{-2}$	$2.8x10^{-2}$	$3.1x10^{-2}$
	Std. dev	$2.5x10^{-3}$	$2x10^{-3}$	$2x10^{-3}$	$2.3x10^{-3}$	$1.9x10^{-3}$	$2.1x10^{-3}$
6	Mod. Lapl.	$3.4x10^{-2}$	$3x10^{-2}$	$2.9x10^{-2}$	$3x10^{-2}$	$2.6x10^{-2}$	$2.9x10^{-2}$
	Std. dev	$2.4x10^{-3}$	$2x10^{-3}$	$1.9x10^{-3}$	$2x10^{-3}$	$1.7x10^{-3}$	$1.9x10^{-3}$
7	Mod. Lapl.	$3.3x10^{-2}$	$2.5x10^{-2}$	$2.6x10^{-2}$	$2.9x10^{-2}$	$2.3x10^{-2}$	$2.6x10^{-2}$
	Std. dev	$2.2x10^{-3}$	$1.7x10^{-3}$	$1.8x10^{-3}$	$1.9x10^{-3}$	$1.6x10^{-3}$	$1.8x10^{-3}$

The setup reported in Table 6.1 results from scenarios without obstacles. However, greater measures of focus filters are expected when obstacles appear in the scene.

The results of Table 6.1 were obtained from a vertical coordinate of the camera of 107cm. When, the camera vertical coordinate is increased the focus measures decrease. Figure 6.11 shows the floor scenario acquired when the camera coordinate is 170cm high from the floor. Figure 6.11 also shows the corresponding focus measures and standard deviations.

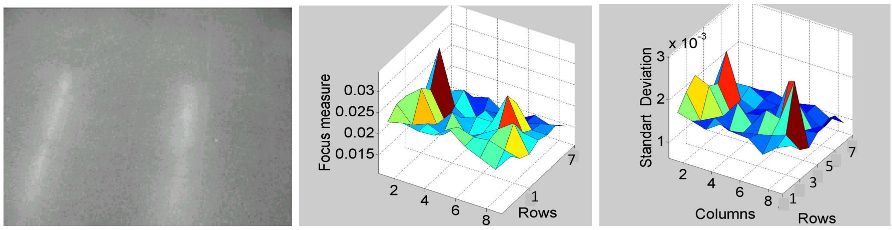


Figure 6.11: Typical floor scenario acquired at 400 lx, represented in a 96x72 image size, averaged 9x7 modified Laplacian and standard deviation values.

Table 6.2 reports the focus measures obtained for each 9x7 image window.

Table 6.2: Floor scenario, modified Laplacian and standard deviation for each 9x7 image window.

R	Col.	9	8	7	6	5	4	3	2	1
1	FM.	$2.2x10^{-2}$	$1.8x10^{-2}$	$1.9x10^{-2}$	$2.1x10^{-2}$	$2.4x10^{-2}$	$2.1x10^{-2}$	$2.1x10^{-2}$	$2.2x10^{-2}$	$2.1x10^{-2}$
	σ	$1.6x10^{-3}$	$1.4x10^{-3}$	$1.2x10^{-3}$	$1.4x10^{-3}$	$1.5x10^{-3}$	$1.6x10^{-3}$	$1.3x10^{-3}$	$1.4x10^{-3}$	$1.5x10^{-3}$
2	FM.	$2.0x10^{-2}$	$2.0x10^{-2}$	$2.2x10^{-2}$	$2.1x10^{-2}$	$2.2x10^{-2}$	$1.9x10^{-2}$	$2.1x10^{-2}$	$1.9x10^{-2}$	$2.3x10^{-2}$
	σ	$1.3x10^{-3}$	$1.6x10^{-3}$	$1.5x10^{-3}$	$1.4x10^{-3}$	$1.3x10^{-3}$	$1.2x10^{-3}$	$1.5x10^{-3}$	$1.3x10^{-3}$	$1.8x10^{-3}$
3	FM.	$1.9x10^{-2}$	$2.5x10^{-2}$	$2.2x10^{-2}$	$2x10^{-2}$	$1.8x10^{-2}$	$2.3x10^{-2}$	$1.9x10^{-2}$	$2.6x10^{-2}$	$2.3x10^{-2}$
	σ	$1.2x10^{-3}$	$2.7x10^{-3}$	$1.7x10^{-3}$	$1.8x10^{-3}$	$1.3x10^{-3}$	$1.7x10^{-3}$	$1.3x10^{-3}$	$1.9x10^{-3}$	$2.1x10^{-3}$
4	FM.	$2.0x10^{-2}$	$2.1x10^{-2}$	$2.6x10^{-2}$	$2.1x10^{-2}$	$2.1x10^{-2}$	$2.1x10^{-2}$	$1.9x10^{-2}$	$1.7x10^{-2}$	$2.4x10^{-2}$
	σ	$1.4x10^{-3}$	$1.3x10^{-3}$	$2.4x10^{-3}$	$1.4x10^{-3}$	$1.4x10^{-3}$	$1.3x10^{-3}$	$1.1x10^{-3}$	$1.8x10^{-3}$	$1.4x10^{-3}$
5	FM.	$1.8x10^{-2}$	$1.9x10^{-2}$	$1.9x10^{-2}$	$2.4x10^{-2}$	$1.8x10^{-2}$	$2x10^{-2}$	$1.6x10^{-2}$	$1.8x10^{-2}$	$3.1x10^{-2}$
	σ	$1.4x10^{-3}$	$1.2x10^{-3}$	$1.7x10^{-3}$	$1.1x10^{-3}$	$1.3x10^{-3}$	$1.2x10^{-3}$	$1.4x10^{-3}$	$2.4x10^{-3}$	$1.4x10^{-3}$
6	FM.	$1.9x10^{-2}$	$2x10^{-2}$	$1.9x10^{-2}$	$1.8x10^{-2}$	$1.5x10^{-2}$	$1.8x10^{-2}$	$1.4x10^{-2}$	$2.1x10^{-2}$	$1.9x10^{-2}$
	σ	$1.4x10^{-3}$	$1x10^{-3}$	$1.3x10^{-3}$	$1.1x10^{-3}$	$1.3x10^{-3}$	$1.3x10^{-3}$	$1.5x10^{-3}$	$1.5x10^{-3}$	$1.2x10^{-3}$
7	FM.	$1.7x10^{-2}$	$1.6x10^{-2}$	$1.7x10^{-2}$	$1.5x10^{-2}$	$1.6x10^{-2}$	$1.8x10^{-2}$	$1.7x10^{-2}$	$1.7x10^{-2}$	$1.6x10^{-2}$
	σ	$1.1x10^{-3}$	$1.1x10^{-3}$	$1.3x10^{-3}$	$1.2x10^{-3}$	$1.2x10^{-3}$	$1.3x10^{-3}$	$1.2x10^{-3}$	$1x10^{-3}$	$1.1x10^{-3}$

Both tables confirm the fact that FM decrease as function of the camera distance. Therefore, closer rows have larger FM values. Moreover light reflections, which arise from the lighting system, can cause larger values. For texture-less floors, the methodology presented is not suitable for analysing such cases an object without texture was placed on the floor. Figure 6.12 shows the floor scenario where an object without texture was included; the system camera coordinate was set to 170cm high from the floor.

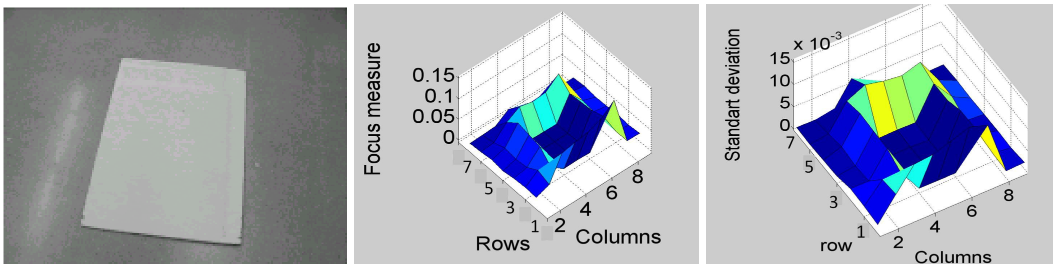


Figure 6.12: Typical floor scenario, with an object without texture acquired at 400 lx, represented in a 96x72 image size, averaged 9x7 modified Laplacian and standard deviation values.

Table 6.3 reports the focus measures and standard deviations obtained for each 9x7 image window corresponding to Figure 6.12.

Table 6.3: Textureless object and floor scenario; FM and standard deviation for each 9x7 image window.

R	Col.	9	8	7	6	5	4	3	2	1
1	FM.	$2x10^{-2}$	$2.2x10^{-2}$	$14x10^{-2}$	$11x10^{-2}$	$10x10^{-2}$	$10x10^{-2}$	$9x10^{-2}$	$3x10^{-2}$	$2.2x10^{-2}$
	σ	$1.6x10^{-3}$	$1.4x10^{-3}$	$1.3x10^{-2}$	$1.5x10^{-2}$	$1.3x10^{-2}$	$1.2x10^{-2}$	$1.2x10^{-2}$	$6.7x10^{-3}$	$1.7x10^{-3}$
2	FM.	$1.7x10^{-2}$	$2.2x10^{-2}$	$7.9x10^{-2}$	$0.9x10^{-2}$	$1.1x10^{-2}$	$0.9x10^{-2}$	$3.4x10^{-2}$	$4.6x10^{-2}$	$2.4x10^{-2}$
	σ	$1.2x10^{-3}$	$1.7x10^{-3}$	$9.3x10^{-3}$	$8.3x10^{-4}$	$8x10^{-4}$	$8.4x10^{-4}$	$6.2x10^{-3}$	$6.3x10^{-3}$	$2.1x10^{-3}$
3	FM.	$1.9x10^{-2}$	$2.6x10^{-2}$	$7x10^{-2}$	$1.2x10^{-2}$	$1x10^{-2}$	$1.1x10^{-2}$	$6x10^{-2}$	$3x10^{-2}$	$2.2x10^{-2}$
	σ	$1.2x10^{-3}$	$2.4x10^{-3}$	$9.3x10^{-3}$	$8.7x10^{-4}$	$8.6x10^{-4}$	$8.4x10^{-4}$	$7.8x10^{-3}$	$2.3x10^{-3}$	$2.3x10^{-3}$
4	FM.	$1.7x10^{-2}$	$2.3x10^{-2}$	$6.7x10^{-2}$	$1.1x10^{-2}$	$1x10^{-2}$	$1.1x10^{-2}$	$6.7x10^{-2}$	$2.7x10^{-2}$	$1.8x10^{-2}$
	σ	$1.2x10^{-3}$	$1.6x10^{-3}$	$7.9x10^{-3}$	$7.4x10^{-4}$	$8.6x10^{-4}$	$8x10^{-4}$	$8x10^{-3}$	$2x10^{-3}$	$1.4x10^{-3}$
5	FM.	$1.5x10^{-2}$	$2.7x10^{-2}$	$6.4x10^{-2}$	$1x10^{-2}$	$1x10^{-2}$	$1.1x10^{-2}$	$7.1x10^{-2}$	$3.1x10^{-2}$	$2.1x10^{-2}$
	σ	$1.2x10^{-3}$	$3.1x10^{-3}$	$7.9x10^{-4}$	$9.2x10^{-4}$	$9.8x10^{-4}$	$9.2x10^{-4}$	$8.3x10^{-3}$	$2.5x10^{-3}$	$1.2x10^{-3}$
6	FM.	$2x10^{-2}$	$2.6x10^{-2}$	$8.8x10^{-2}$	$6.3x10^{-2}$	$6x10^{-2}$	$6.8x10^{-2}$	$4.9x10^{-2}$	$2x10^{-2}$	$1.7x10^{-2}$
	σ	$1.3x10^{-3}$	$3.2x10^{-3}$	$9x10^{-3}$	$7.7x10^{-3}$	$8.2x10^{-3}$	$9.1x10^{-3}$	$7.3x10^{-3}$	$1.7x10^{-3}$	$1.1x10^{-3}$
7	FM.	$1.6x10^{-2}$	$1.6x10^{-2}$	$1.6x10^{-2}$	$1.6x10^{-2}$	$2.2x10^{-2}$	$3.9x10^{-2}$	$1.7x10^{-2}$	$1.5x10^{-2}$	$1.5x10^{-2}$
	σ	$1.2x10^{-3}$	$1.1x10^{-3}$	$1.5x10^{-3}$	$1.2x10^{-3}$	$2.7x10^{-3}$	$6.9x10^{-3}$	$1.3x10^{-3}$	$1x10^{-3}$	$1.1x10^{-3}$

Table 6.3 show that low FM measures were obtained for objects with a lack of texture when it is compared with the FM measures obtained in Table 6.2. In this way, standard deviation of lack of texture objects also have low values as compared with the floor values obtained in Table 6.2. These results show that OPUOF method developed maintains a degree of floor energy radiance even after decreasing the space resolution level at the 96x72 image size.

Experimentally good results can be achieved using different row thresholds consisting of the maximum modified Laplacian value plus 3 times the maximum standard deviation. The different row thresholds correspond to different scene distances and different radiance values as a distance function. This result can be considered as an important improvement that arises from the present research when it is compared with standard segmentation. Using the acquired images of the obstacles, a more in-depth analysis can be performed during the navigation tests. Fig. 6.13 presents the modified Laplacian energy and standard deviation values, when typical indoor obstacles are present. The modified Laplacian has been used with good results, however other DFF filters can be tested, see section 3.3.3.

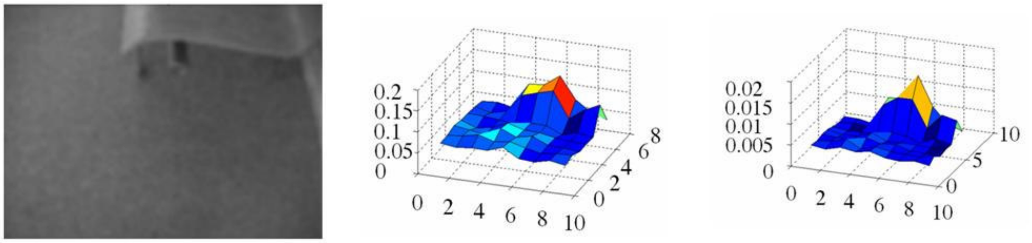


Figure 6.13: Obstacle acquired at 180 lx, represented in a 96x72 image size, averaged 9x7 modified Laplacian and standard deviation values.

However, it should be pointed out that false obstacles are detected by the methodology presented. Figures 6.14 and 6.15 show that flat shape lines placed on the floor or light reflections, respectively, can be considered as obstacles (false positives). These false obstacles emerge when constraints fail, which can be considered the main drawback of the method. The OPUOF methodology may be improved by time-integrating the acquired frames. For example, OFM can allow for scene structure analysis and, consequently, avoid false obstacle detection.

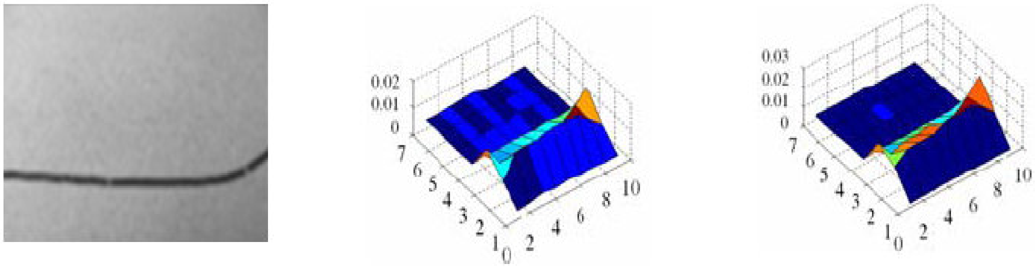


Figure 6.14: False obstacle caused by a flat shape acquired at 1400 lx, represented in a 96x72 image size, averaged 9x7 modified Laplacian and standard deviation values.

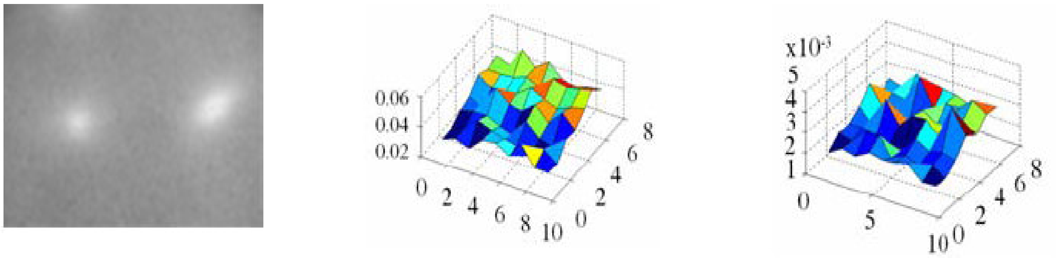


Figure 6.15: False obstacle caused by light reflection acquired at 200 lx, represented in a 96x72 image size, averaged 9x7 modified Laplacian and standard deviation values.

The processing speed of the machine vision system can be increased. Using a 2.7 GHz desktop, the total frame processing time to compute the three levels of the Gaussian pyramid compression is close to 2s. The use of down-sampling techniques in the first two compression levels, instead of the Gaussian space resolution method, allows a reduction of computational effort, and the system processes between 2 and 3 frames each second.

6.3.3 Conclusions

The OPUOF work presented on this subsection has shown that feasible obstacle avoidance can be achieved using a set of previously computed multi-resolution thresholds, based on DFF filters. The methodology can be easily transferable to WMRs. Algorithm robustness simplicity and fast computation are other interesting features of this research. The method constraints are provided by a large range of indoor and outdoor environments. The experimental results report robust and effective obstacle detection in typical indoor environments. False obstacle detection, when constraints fail, can be considered as the weak point of this work. In order to solve this limitation, OPUOF results are time-integrated by using odometer sensor information as it is depicted in the next section.

6.4 Improving the obstacle detection and optical flow techniques

The good results attained under homogeneous constraints and false failures problems focus the research presented in this section. Thus, as i.e., optical flow methods can be applied only in the scene areas where the possible obstacles are detected. OPUOF can detect relative obstacle positions, speeding up OFM matches between frames, and OFM can use a frame sequence of the same obstacles in order to obtain qualitative structure information. Robot position and orientation, established using the odometer system, are interesting data that can help to detect the floor obstacle contact coordinates from frame to frame. Analysis of the discrepancies can provide qualitative structure knowledge. The actual research is aimed to solving false obstacle detection by obtaining the obstacle structure.

6.4.1 The local optical flow methodology

The image brightness pattern motion observed by a camera, when the camera or some scene object is moving, is called optical flow. The techniques use a time varying sequence of images as a source of information. Reconstruction of 3D information through 2D information could also be considered as an ill-posed problem. Therefore, the OFT allows the motion and structure recovery when the aperture problem is not presented [Horn, 98]. The different quantitative approaches are generally based on two classical feasible assumptions: BCM (brightness constancy model), and optical flow smoothness. Image motion discontinuities are due to the depth and motion discontinuity boundaries. Therefore, suggestions to compute the algorithms over small neighborhoods, or region based matched methods has turn on. Combining local and global optic flow differential methods have been proposed as a way to share benefits from the complementary advantages and short-comings [Bruhn et al., 02].

In the present research the camera field of view depicts only the vicinity of the WMR. Hence, perspective projection, as it is shown in Fig. 6.16, should be assumed. Introducing the coordinate system, where z coordinates are aligned to the optical camera, and xy-plane is parallel to image plane, the image *P'* of the point *P* of some scene object is given by the following expressions:

$$-z = r\cos\alpha \Rightarrow r = -\frac{z}{\cos\alpha} = -z\sec\alpha \qquad (6-12)$$

$$z' = r'\cos\alpha \Rightarrow r' = \frac{z'}{\cos\alpha} = z'\sec\alpha \qquad (6-13)$$

$$\frac{x'}{z'} = \frac{x}{z} \qquad \text{and} \qquad \frac{y'}{z'} = \frac{y}{z} \qquad (6-14)$$

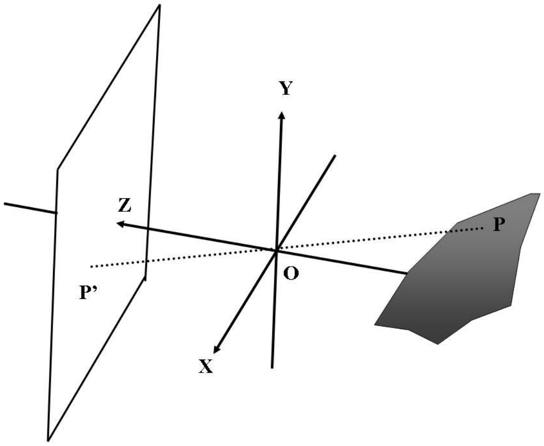


Figure 6.16: Camera system producing an image that is a perspective projection of the world.

Where *z'* is the distance between image plane and the camera lenses, and *x'* and *y'* the image coordinates. The object point coordinates, referred to the optic center *O*, are given by *P= (X, Y, Z)*, being *r* the distance between *P* and *O* and α the angle.

The ratio of the distance between two points measured in the image plane and the corresponding points measured in the scene is called magnification *m*.

$$m = \frac{\sqrt{(\Delta' x)^2 + (\Delta' y)^2}}{\sqrt{(\Delta x)^2 + (\Delta y)^2}} = \frac{z'}{z} \qquad (6-15)$$

For reduced field of views when the optical rays are parallel to the optical axis the magnification *m* is constant for all the image points. However, due to the field of view and camera pose assumed in this research, magnification changes are expected even just considering a flat floor scenario. Hence, the perspective image formation model arises in magnification changes. Fig. 6.17 shows the magnification changes of the floor model by considering the optical axis ray as the unit of magnification. Therefore, these changes in the magnification made more complicate to look for image patches with similar motion in order to detect obstacle depth boundaries.

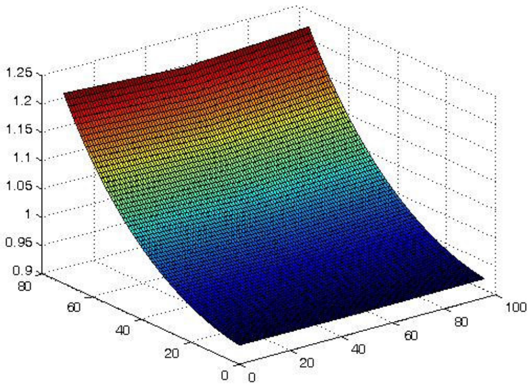


Figure 6.17: Magnification changes of the floor model by considering the optical axis ray as the unit of magnification.

However by using the floor model and the odometer system data, binary floor boundary positions can be predicted from frame to frame; then predicted discontinuities can arise due to the fact that obstacle shapes don't belong to the floor plane.

6.4.2 The computer vision proposed algorithms

The machine vision system algorithms implemented are based on binary results obtained by the OPUOF algorithms explained at subsection 6.3.1. The camera position will set the field of view of the floor, i.e., in the experimental results reported, the camera is 109 cm high from floor, the tilt angle is set to 58º, and horizontal, and vertical field of view are respectively 48º, and 37º. Thus, a LUT (look up table) with the calibrated real floor coordinates, corresponding to the image resolution used sizes, is computed.

Image space resolution has been used; hence, it can speed up the computations reducing the search space. Fig. 6.18 shows the modified Laplacian energy and standard deviation values, using 9x7 and 96x72 space-resolutions, when typical indoor obstacles are presented. It is shown that 9x7 space resolutions can detect radiance discontinuities; however they have a large lack of resolution manifested trough soft slopes. It is necessary to use fine space-resolution, as for instance 96x72, where sharper edges are attained. Thus, 9x7 resolutions are used to detect the local patches where obstacle segmentation is computed within 96x72 space resolution.

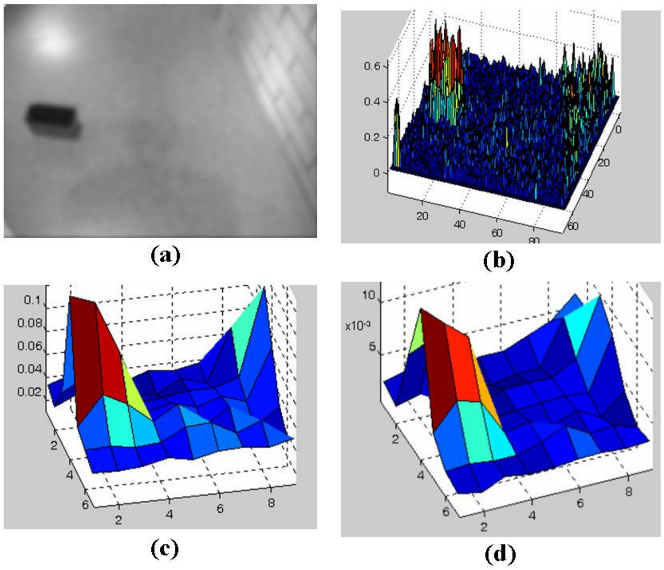


Figure 6.18: a) Image with obstacles, 96x72; b) Modified Laplacian measures; c) 9x7 Modified Laplacian mean values; d)) 9x7 standard deviation mean.

The binary images are obtained in the 96x72 space resolution level. The binary image is composed by a set of binary regions, which correspond to the different obstacles found. Binary region labeling is done, and blob analysis is performed. Small particles are removed, and close blobs are considered as part of the same obstacle when the distance between blobs is less than a heuristic parameter, as i.e. three pixels. Once significant regions are obtained, these regions are considered as obstacles and bigger size coordinates are computed. The down, top, left, and right vertexes coordinates are obtained for the different obstacle candidates. For each obstacle region a set of parameters that allow further work, not presented in this research, are suggested:

- The shape of each obstacle can be encoded by using a simple run length coding algorithm.
- Edges and size can also be locally computed.
- Other parameters as texture and colour can also be considered.

Thus, it is proposed as future work to use the relative robot coordinate increments provided by the odometer system, and the qualitative structure estimations to compare the predicted positions with the binary results obtained. Thus, consecutive OPUOF obstacle matches can be compared as a qualitative methodology where discrepancies can be analised for inferring obstacle structure or floor radiance discontinuities.

Time integration of the different acquired frames is a research topic introduced in the next subsection. Thus, the robot coherent interaction with the world can be faced up by using the occupancy grid framework that provides a robust and unified approach to a variety of problems in spatial robot perception and navigation [Elfes, 89].

6.5 The local planar monocular occupancy grid framework

In this section it is presented the LPMOG (local planar monocular occupancy grid) methodology used in this research. The occupancy field can be depicted by a probability density function that relates sensor measures to the real cell state, describing the occupancy probability for the different grid positions. The tessellated probabilistic representation has been widely adopted by scientist community in navigation or mapping issues. Outdoor applications can have available GPS information that provide feasible robot localization, thus developed occupancy grid applications deal with the integration of on robot information in order to improve the environment knowledge and the robot navigation while indoor applications research has been mainly concentrated in SLAM issues [Thrun, 02]. However both applications share benefits that arise from occupancy grid framework as multi-resolution and multidimensional representations or sensor modeling. Hence, their use allows sensor fusion or multiple layer representations to segment dynamic objects [Coue et al., 06].

The perception system used in this work consists of monocular and odometer system data. The use of these systems in SLAM is reported in [Cumani et al., 04]. However, in this case the camera pose and the narrow field of view provide just the vicinity of the robot where only floor is expected to be found, in order to plan goal approaching trajectories. The use of floor models has been proposed as an interesting issue in order to plan WMR trajectories [Kim and Kim, 04]. In this work, it is also utilised the floor model; hence obstacle binary results provided in subsection 6.3 are time integrated within the LPMOG framework by considering such model. Moreover, obstacle structure can be inferred as future work by considering optical flow magnification changes discrepancies from the floor model.

The Bayesian estimation formulation has been proposed by researchers for static occupancy, not only for the use of the last sensor observation measure. Thus, the last occupancy estimate should also be considered. Using 2D grids for static indoor mapping is proposed in [Thrun, 02]. Other works propose multidimensional grids in which multi target tracking algorithms with obstacle state space are used with Bayesian filtering techniques [Coue et al., 06]. Integrating perception and planning is an interesting topic. For example, planning cycles that consider a time horizon in which partial trajectories are planned until the robot state is close to the goal are reported in [Benenson et al., 06].

The next subsection introduces the occupancy grid formulation used as a tessellated probabilistic representation.

6.5.1 The Bayesian occupancy grid proposal

The occupancy grid is considered as a discrete stochastic process defined over a set of continuous spatial coordinates (x, y). Hence, the space is divided into a finite number of cells representing a 2D position, $1\le j \le R$ $1\le i \le C$. The R and C parameters are the number of rows and columns of the grid respectively. The cell column coordinates are designated by X_i and the rows by Y_j. It is assumed that local occupancy grid data is provided by the on-robot perception system. The occupancy probability is divided into only two ranges: free and occupied. The grid can be updated by using the sensor models and the current information. Hence, given a sensor measurement m, the occupancy probability $P(O,)$ for the different cells, $P(C_{ij})$, can be computed by applying Bayes rule:

$$P(O|C_{ij}) = \frac{P(C_{ij}|O)P(O)}{P(C_{ij}|O)P(O) + P(C_{ij}|/\mathrm{O})P(/O)} \qquad (6\text{-}16)$$

Hence, the probability that a cell is occupied $P(O/C_{ij})P(O)$ is given by the cell occupancy sensor measurement statistics $P(C_{ij}/O)P(/O)$ by considering also the probability that the cell will be free $P(C_{ij}//O)$. Thus, free cells have binary results equal to zero; these not occupied cells belong to coordinates for image pixels within floor radiance thresholds. Other available coordinates are provided trough time integration of the acquired frames when predicted radiance energy occupancy cells, by using the floor model, have coincidences with the acquired frames. The unknown probability value is set to 0.5. Therefore, by using (6-16) expression with predicted occupied cells and acquired frames, the grid positions belonging to the floor will provide larger occupancy values. Obstacles positions give intermediate occupancy probabilities due to the discrepancies between the predicted and the acquired image values that arise due to the 3D obstacle shape. Fig. 6.19 and Fig. 6.20 show some results when marks belonging to the floor or obstacles are met. Hence, in Fig. 6.19 overlapping between marks belonging to the floor are observed, and consequently the probability of being floor is increased. Otherwise, in Fig. 6-20 when real obstacles are met probability has middle values. The expected results are that overlapping zones between predicted radiance discontinuities and new acquired frames will increase the probability values when the floor model is accomplished. However, when discrepancies arise because the 3D obstacle perspective mismatches with the floor model, then decreasing probabilities are obtained.

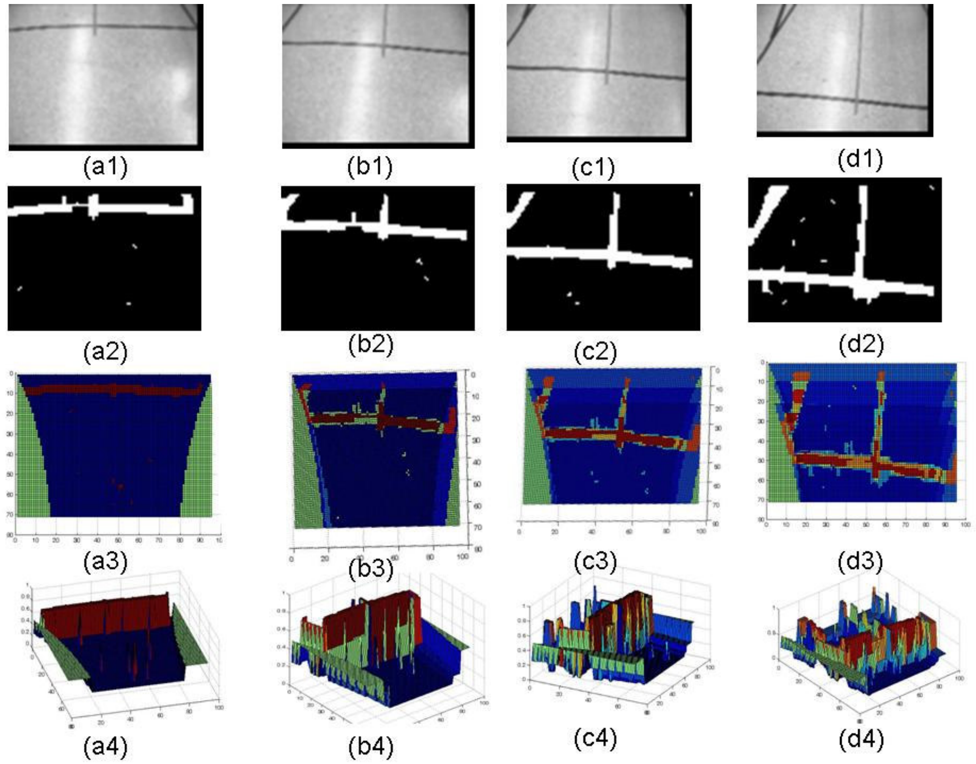


Figure 6.19: In 6.19.a$_1$, 6.19.b$_1$, 6.19.c$_1$ and 6.19.d$_1$ are shown 4 images corresponding to floor marks acquired from different positions that have been taken in advancing sense. The binary results are shown in 6.19.a$_2$, 6.19.b$_2$, 6.19.c$_2$ and 6.19.d$_2$. The probability of being floor is depicted in sequences 6.19.a$_3$, 6.19.b$_3$, 6.19.c$_3$ and 6.19.d$_3$, and 6.19.a$_4$, 6.19.b$_4$, 6.19.c$_4$ and 6.19.d$_4$.

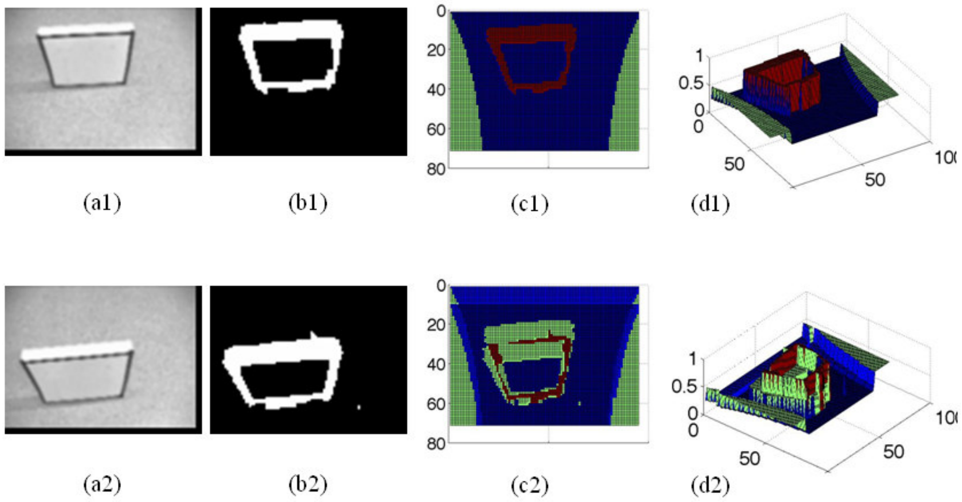


Figure 6.20: It is shown the occupancy results when an obstacle is met. Figures 6.20.a1, b1, c1, and d1 shows respectively the first image, the binary results, and the occupancy grid 2D and 3D probabilty results. Figures 6.20.a2, b2, c2, and d2, shows the corresponding results obtained when the second acquired image is overlapped into the local occupancy grid framework.

The results of Fig. 6-19 and Fig. 6-20 illustrate the LPMOG probability values for sequences of images. However, in Fig. 6-19 an expected perfect coincidence, drawn in red colour, for a mark placed on the floor is not completely accomplished; and in Fig. 6-20 when a real obstacle is met some coincidences are obtained. These results should focus the future research. Thus, the results obtained in this section show that the matches produced during time integration are not good enough. The errors are mainly due to the following aspects:

- The lack of an accurate camera calibration
- Floor model deviations
- Odometer system errors, mainly produced during WMR turning actions.

Thus, the parameters suggested in subsection 6.4.2 can afford more clues to classify the binary regions as obstacles or floor discontinuities.

6.6 Conclusions and Future Work

OPUOF is presented as a suitable monocular machine vision system for WMR perception when flat floor and homogeneous floor radiance are expected. Moreover, the LPMOG framework is introduced as a probabilistic framework where sequences of images can be time-integrated. Time-integration can be used as a suitable system to analyse qualitative OF when radiance discontinuities are produced. The results achieved show that probabilistic results are not enough for obtaining the obstacle structure. Thus, future research should include other features as blob shape, size, and edge evolutions as additional clues for inferring obstacle structure within the methodology presented. The use of vertical models could also be suggested.

Moreover, Chapter 8 will use the LPMOG perception system presented in this section. Therefore, local WMR navigation will be tested by using the perception techniques introduced in this Chapter. In this way, the LPMOG framework will be used as a way to integrate sequences of images. Thus, the reduced field of view of the frames acquired can be increased by using occupancy grids with a bigger size than the one obtained by each image. As a result, better environment knowledge is attained and consequently better robot navigation is accomplished.

Chapter 7

WMR Experimental Model and Local Model Predictive Control Trajectory Tracking

7.1 Introduction

This chapter extends previously developed work for mobile robots to trajectory tracking with dynamic model predictive control techniques. The use of dynamic models and experimental cost function factor adjustments are important issues of the above work. In order to do this, a set of dynamic models obtained from experimental robot system identification are used for predicting the horizon of available coordinates. The use of contractive constraints guarantees the convergence of the robot coordinates towards the desired ones. Local trajectory planning is another relevant idea of this work. Hence, when dynamic environments or obstacle avoidance policies are considered, the navigation path planning should be constrained to the robot neighbourhood. The testing and analysis of the experimental results of trajectory tracking are reported. In this context, the different parameter weight performances of the cost function are studied. In this way, the factor tuning is tested by using different kinds of trajectories.

Moreover, other important aspects, such as local on-robot perception and navigation issues are considered as important constraints in order to accomplish the different missions. Studies concerning the relationships between the perception and control systems have been developed. Efficient and safe navigation is a requirement usually in a trade-off relationship. Hence, when the WMR speed is increased the efficiency is also increased. When the sensing observations decrease, the uncertainties in localization and environment recognition increase. The use of the fastest safe speed for WMR movements in unknown environments has been reported in [Miura et al., 06]. The control strategies, applied to WMR, should fulfil safety rules as well as achieve the final desired configuration. The scientific community has developed several studies in this field. The dynamic window approach is based on the dynamic constraints of WMR and uses the available robot speeds for planning the avoidance of obstacle collisions, safe stop and goal achievements [Fox et al., 97]. In the work of Rimon the methodologies for the exact motion planning and control are presented, based on the artificial potential fields where the complete information on free space and the goal is encoded [Rimon and Koditschek, 92]. However, under dynamic environments the global solution becomes unfeasible for the majority of applications where the scenario should be considered as partially unknown due to the lack of global sensors or the existence of dynamic obstacles. Hence, some approaches on mobile robots propose the use of potential fields, which satisfy the stability in a Lyapunov sense, in a short prediction horizon that allows obstacle avoidance policies as well as final goal approaches to be dealt with [Ögren and Leonard, 05]. Thus, convergence to goal and reactive behavior can be achieved by

casting the two objectives in a MPC and CLF (control Lyapunov function) framework [Primbs et al., 99].
From the control science point of view, the accuracy and performance of WMR trajectory tracking are subject to nonholonomic constrains and consequently it is usually difficult to achieve a stabilised tracking of trajectory points using linear feedback laws [Brockett, 83]. In the research results presented by Hindman and Hauser [Hindman and Hauser, 92], with nonholonomic systems similar to that presented in this work, it is demonstrated through Lyapunov stability theory that asymptotic stability exists in the control system with respect to the desired trajectory. In the case considered, where trajectory-tracking is performed by using different linear models, the Euclidean distance between the robot and the desired trajectory can be used as a potential function. Such functions are CLF, and consequently asymptotic stability with respect to the desired trajectory can be achieved. Path following stabilisation of nonholonomic systems can be reached using time-varying, discontinuous or hybrid feedback laws. The use of piecewise analytic feedback laws in discontinuous stabilising approaches are reported in [Sørdalen and Canudas de Wit, 93] where its application is mainly addressed to WMR. The time-varying feedback laws allow the use of closed-loop algorithms that give finite time convergence [M'Closkey and Murray, 97]. Other approaches with adaptive capabilities of including robot-model uncertainties, or fuzzy inference that compensates environment perturbations such as variable friction, are proposed in [Pourboughrat and Karlsson, 02; Raimondi and Melluso, 05]. Input-output linear dynamic models have been presented as a suitable solution to following reference trajectory dynamics [Sarkar et al., 94]. Trajectory tracking using discontinuous control laws can be implemented by using heuristic concepts related to experimental dynamic knowledge of the system. Therefore, when the error of path distance of the robot is greater than a heuristic threshold, the control law can minimise the distance to the path; otherwise the angular error deviation should be minimised [Pacheco and Luo, 06]. Knowledge of different models can provide information about the dynamics of the robot, and consequently about the reactive parameters, as well as the safe stop distances. In the cases studied in this work, three different reactive distances corresponding to three different dynamic models are proposed. The allowed trajectory-distances set the speed that can be reached.

Trajectory tracking by using predictive control techniques has also been proposed [Ollero and Amidi, 91]. The majority of the research developed by using MPC techniques and its application to WMR is based on the fact that the reference trajectory is known beforehand [Klancar and Skrjanc, 07]. The use of mobile robot kinematics to predict future system outputs were proposed in most of the different research developed [Ollero and Amidi, 91; Klancar and Skrjanc, 07]. The use of multilayer neural network has been employed to model the nonlinear kinematical behaviour of a mobile robot [Gu and Hu, 02]. The use of kinematics should include velocity and acceleration constraints to prevent WMR of unfeasible trajectory-tracking objectives [Klancar and Skrjanc, 07]. MPC techniques based on a linear time-varying description of the system dynamics has been also proposed [Küne et al., 05]. The work of Kühne shows the possibility of using linear time-varying descriptions of the system, as well as real-time implementations, when short prediction horizons are used. The MPC applicability for agricultural vehicle guidance has been depicted in [Lenain et al., 05], where kinematics models and different on-field fixed trajectories were tested. However, due to the unknown environment uncertainties, short prediction horizons have been proposed by other authors [Pacheco and Luo, 07D]. The use of dynamic input-output models is also proposed as a way to include the dynamic constraints within the system model for controller design. This chapter extends the use of dynamic models when local trajectory tracking is performed [Pacheco and Luo, 07D]. The techniques for obtaining the dynamic models of the WMR as well as the cost function performance for tracking the local trajectories are presented. Furthermore, experimental studies are done as a way of inferring the weight of the different parameters used in the cost function. Factor tuning is achieved by considering aspects, such as the time taken, or trajectory deviation, within a set of different local trajectories.

The Chapter is organised as follows. Section 2 presents the robot kinematical system used for performing the trajectory tracking. The experimental methodology used to obtain the different dynamic models is depicted in Section 3. Section 4 introduces the MPC formulation for trajectory tracking as well as the preliminary simulations tested to conduct the experimental results. In Section 5 the navigation strategy is introduced by considering a local environment description provided by a monocular perception system introduced on Chapter 6. Moreover, in this section different factorial experimental results used to adjust the weighting parameters of the cost functions are also presented. The conclusion and description of future work are given in Section 6.

7.2 The Kinematic System

The WMR is a rigid body and consequently non-deforming wheels are considered. It is assumed that the vehicle moves without slipping on a plane, so there is pure rolling contact between the wheels and the ground. Denoting *(x, y, θ)* as the coordinates of position and orientation, and *u = [v, w]* as the velocity vector; where *v* and *w* respectively denote the tangential and angular velocities. Then, the kinematic model of the WMR is given by:

$$\begin{aligned} dx &= v\cos\theta \\ dy &= v\sin\theta \\ d\theta &= \omega \end{aligned} \qquad (7\text{ - }1)$$

By using the discrete time representation (with *T* being the sampling period and *k* the time instant) and Euler's approximation, the following discrete time model is obtained for the robot dynamics:

$$\begin{aligned} x(k+1) &= x(k)+v(k)\cos\theta(k)T \\ y(k+1) &= y(k)+v(k)\sin\theta(k)T \\ \theta(k+1) &= \theta(k)+\omega(k) \end{aligned} \qquad (7\text{ - }2)$$

The WMR platform uses incremental encoders in order to obtain the coordinates of position and orientation. Figure 7.1 describes the positioning of robot as a function of the radius of left and right wheels *(R_e, R_d)*, and the angular incremental positioning (θ_e,θ_d), with *E* being the distance between two wheels and *dS* the incremental displacement of the robot. The position and angular incremental displacements are expressed as:

$$dS = \frac{R_d d\theta_d + R_e d\theta_e}{2} \qquad d\theta = \frac{R_d d\theta_d - R_e d\theta_e}{E} \qquad (7\text{-}3)$$

The coordinates *(x, y, θ)* can be expressed:

$$\begin{aligned} x(k+1) &= x(k) + dS\cos(\theta(k) + d\theta) \\ y(k+1) &= y(k) + dS\sin(\theta(k) + d\theta) \\ \theta(k+1) &= \theta(k) + d\theta \end{aligned} \qquad (7-4)$$

Thus, the incremental position of the robot can be obtained by the odometer system through the available encoder information from (7-3) and (7-4).

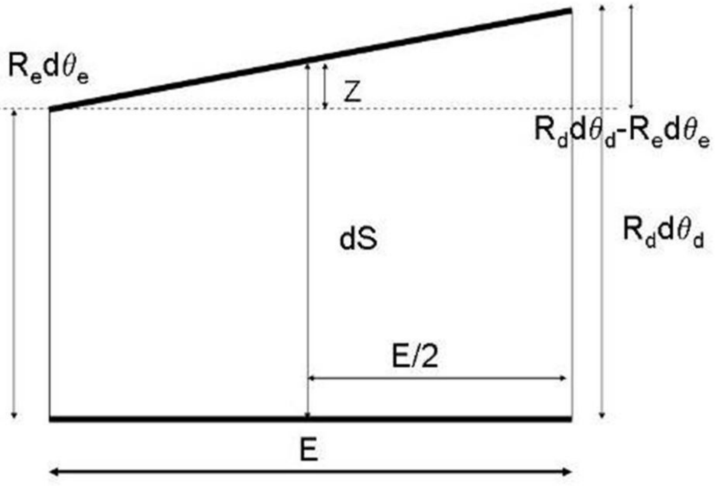


Figure 7.1: Positioning of robot as function of the angular movement of each wheel.

7.3 The WMR experimental dynamic models

This Section deals with the problem of modeling the dynamics of the WMR system. The aim of this work is to obtain a set of dynamic models, for high, medium, and slow velocities, in order to predict the range of possible coordinates when MPC strategies are used. Thus, dynamics constraints such as range of speeds or accelerations are included within the dynamic models. The methodology developed is explained in the next two subsections

7.3.1 The system identification

The model is obtained through the approach of a set of lineal transfer functions that include the nonlinearities of the whole system. The parametric identification process is based on black box models [Lju, 89; Norton, 86]. The nonholonomic system dealt with in this work is considered initially to be a MIMO (multiple input multiple output) system, as shown in Figure 7.2, due to the dynamic influence between two DC motors. This MIMO system is composed of a set of SISO (single input single output) subsystems with coupled connection.

The parameter estimation is done by using a PRBS (Pseudo Random Binary Signal) as an excitation input signal. It guarantees the correct excitation of all dynamic sensible modes of the system along the whole spectral range and thus results in an accurate precision of parameter estimation. The experiments to be realised consist in exciting the two DC motors in different (low, medium and high) ranges of speed.

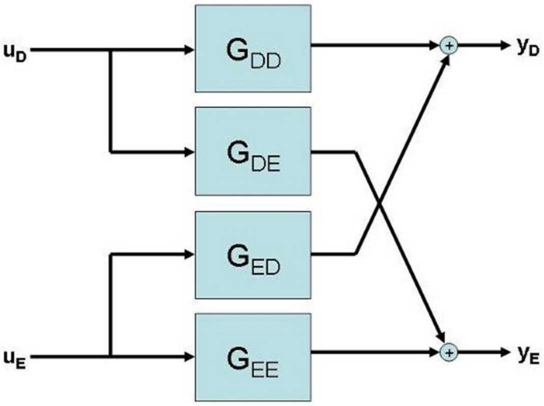


Figure 7.2: Structure of MIMO system.

The ARX (auto-regressive with external input) structure has been used to identify the parameters of the system. The problem consists in finding a model that minimises the error between the real and estimated data. By expressing the ARX equation as a lineal regression, the estimated output can be written as:

$$\hat{y} = \lambda\varphi \tag{7-5}$$

with $\hat{y}$ being the estimated output vector, λ the vector of estimated parameters, and φ the vector of measured input variables. By using the coupled system structure, the transfer function of the robot can be expressed as follows:

$$\begin{pmatrix} Y_D \\ Y_E \end{pmatrix} = \begin{pmatrix} G_{DD} & G_{ED} \\ G_{DE} & G_{EE} \end{pmatrix} \begin{pmatrix} U_D \\ U_E \end{pmatrix} \tag{7-6}$$

where Y_D, and Y_E represent the speeds of right and left wheels, and U_D and U_E the corresponding speed commands, respectively. In order to have information about the coupled system, the matrix of transfer function should be identified. Figure 7.3 shows the speed responses of the system corresponding to the PBRS input signals.

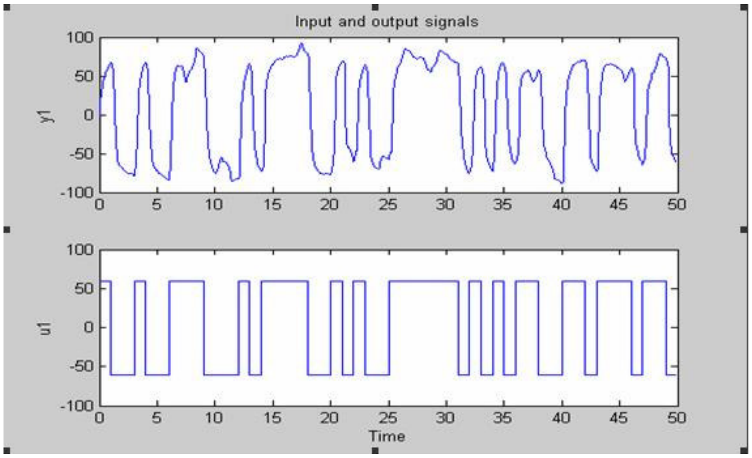


Figure 7.3: Left speed Output for a left PRBS input signal.

The filtered data, which represent the average value of five different experiments with the same input signal, are used for identification. The system is identified by using the identification toolbox "ident" of Matlab for the second order models. Table 7.1 shows the continuous transfer functions obtained for the three different lineal speed models.

Table 7.1: The second order WMR model

Linear Transfer Function	High velocities	Medium velocities	Low velocities
G_{DD}	$\frac{0.20s^2-3.15s+9.42}{s^2+6.55s+9.88}$	$\frac{0.20s^2+3.10s+8.44}{s^2+6.17s+9.14}$	$\frac{0.16s^2+2.26s+5.42}{s^2+5.21s+6.57}$
G_{ED}	$\frac{-0.04s^2-0.60s-0.32}{s^2+6.55s+9.88}$	$\frac{-0.02s^2-0.31s-0.03}{s^2+6.17s+9.14}$	$\frac{-0.02s^2-0.20s+0.41}{s^2+5.21s+6.57}$
G_{DE}	$\frac{-0.01s^2-0.08s-0.36}{s^2+6.55s+9.88}$	$\frac{0.01s^2+0.13s+0.20}{s^2+6.17s+9.14}$	$\frac{-0.01s^2-0.08s-0.17}{s^2+5.21s+6.57}$
G_{EE}	$\frac{0.31s^2+4.47s+8.97}{s^2+6.55s+9.88}$	$\frac{0.29s^2+4.11s+8.40}{s^2+6.17s+9.14}$	$\frac{0.25s^2+3.50s+6.31}{s^2+5.21s+6.57}$

7.3.2 The reduced system models

The coupling effects should be studied as a way of obtaining a reduced-order dynamic model. It can be seen from Table 7.1 that the dynamics of two DC motors are different and the steady gains of coupling terms are relatively small (less than 20% of the gains of main diagonal terms). Thus, it is reasonable to neglect the coupling dynamics so as to obtain a simplified model. In order to verify the above facts from real results, a set of experiments have been done by sending a zero speed command to one motor and different non-zero speed commands to the other motor. In Figure 7-4, a response obtained on the left wheel is shown, when a medium speed command is sent to the right wheel. The experimental result confirms that the coupled dynamics can be neglected.

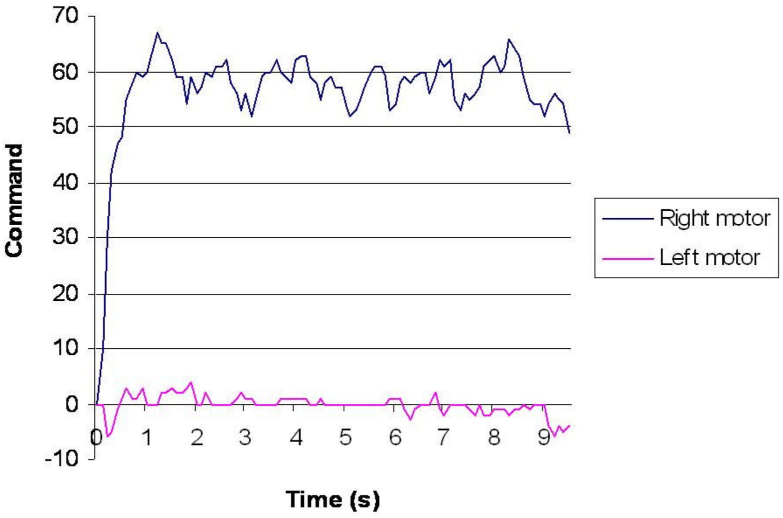


Figure 7.4: Coupled effects at the left wheel for medium speeds.

The existence of different gains in steady state is also verified experimentally. As shown in Figure 7.5, the gain of the left DC motor is greater than that of the right motor in the range of low speed. Finally, the order reduction of the system model is carried out through the analysis of pole positions by using the root locus method.

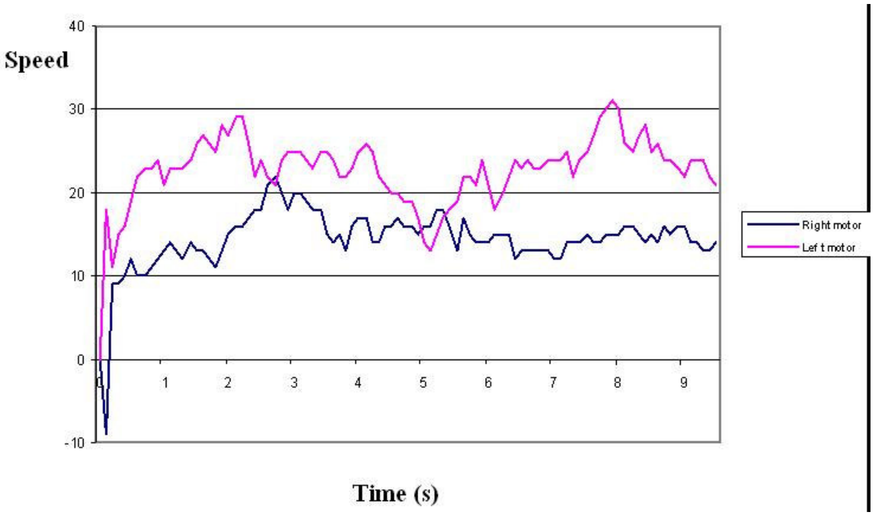


Figure 7.5: Different gain for slow speeds for the same consigns.

It reveals the existence of a dominant pole and consequently the model order can be reduced from second order to first order. Table 7.2 shows the first order transfer functions obtained.

Table 7.2: The reduced WMR model

Linear Transfer Function	High velocities	Medium velocities	Low velocities
G_{DD}	$\frac{0.95}{0.42s+1}$	$\frac{0.92}{0.41s+1}$	$\frac{0.82}{0.46s+1}$
G_{EE}	$\frac{0.91}{0.24s+1}$	$\frac{0.92}{0.27s+1}$	$\frac{0.96}{0.33s+1}$

Afterwards, as shown in Figure 7.6, the system models are validated through the experimental data by using the PBRS input signal.

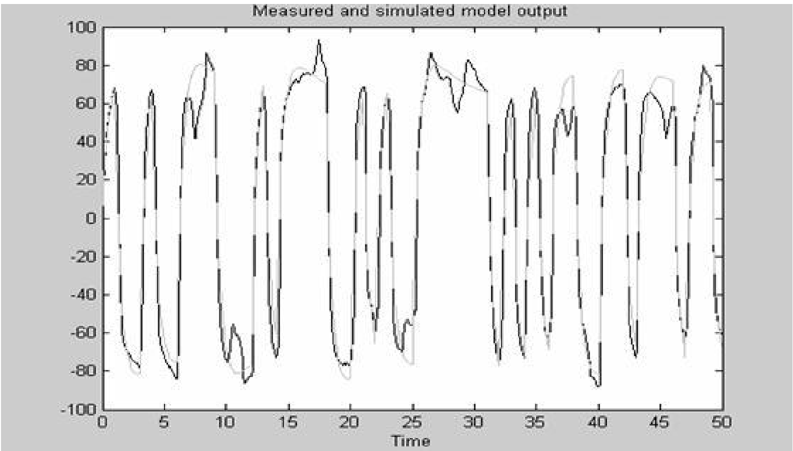


Figure 7.6: Experimental and model data for medium speeds.

7.4 The dynamic MPC techniques for Local Trajectory-Tracking

The minimisation of path tracking error is considered as a challenging subject in mobile robotics. MPC has many interesting features for its application to mobile robot control. It is the most effective advanced control technique, as compared with standard PID control, which has made a significant impact on industrial process control [Maciejowski, 02]. In this section the LMPC (local model predictive control) techniques based on the dynamics models obtained in the previous section are presented. The use of dynamics models avoids the velocity and acceleration constraints used in other MPC research based on kinematic models. Moreover, contractive constraints are proposed as a way of guaranteeing convergence towards the desired coordinates. In general, real-time implementation is possible when a short prediction horizon is used. By using LMPC, the idea of a receding horizon can deal with the local sensor information. The LMPC and contractive constraint formulations as well as the algorithms and simulations implemented are introduced in the next subsections.

7.4.1 The LMPC formulation

The main objective of highly precise motion tracking consists in minimising the error between the robot and the desired path. In general, global trajectory planning becomes unfeasible since the sensorial system of some robots is just local [Noborio and Schmidt, 96]. By using MPC, the idea of the receding horizon can deal with the local sensor information. In this way, LMPC is proposed so as to use the available local perception data in the navigation strategies. More specifically, LMPC is based on minimising a cost function related to the objectives for generating the optimal inputs. Define the cost function as follows:

$$J(n,m) = \min_{\left\{U(k+i|k)_{i=0}^{i=m-1}\right\}} \left\{ \begin{array}{l} [X(k+n|k) - X_{ld}]^T P [X(k+n|k) - X_{ld}] \\ + \sum_{i=1}^{n-1} \left[X(k+i|k) - \overline{X_{ld}X_{l0}}\right]^T Q \left[X(k+i|k) - \overline{X_{ld}X_{l0}}\right] \\ + \sum_{i=0}^{m-1} U^T(k+i|k) R U(k+i|k) \end{array} \right\} \tag{7-7}$$

The first term of (7-7) refers to the attainment of the desired local coordinates, $X_{ld}=(x_d, y_d, \theta_d)$. The second one is related to the distance between the predicted robot positions and the trajectory segment $\overline{X_{ld}X_{l0}}$ given by a straight line between the initial robot coordinates $X_{l0}=(x_{l0}, y_{l0}, \theta_{l0})$, from where the local perception is acquired, and the desired local position $X_{ld}=(x_{ld}, y_{ld}, \theta_{ld})$ related to the local perception. $X(k+n|k)$ represents the terminal value of the predicted output after the horizon of prediction n and $X(k+i|k)$ $(i=1,\dots n-1)$ represents the predicted output values within the prediction horizon. The third term is related to the input signals denoted as U. The parameters P, Q and R are weighting parameters that express the importance of each term, m the control horizon. The system constraints are also considered:

$$\left\{ \begin{array}{l} G_0 < |U(k+1|k)| \le G_1 \\ \left|[X_{k+n}] - [X_{ld}]\right| \le \alpha \left|[X_k] - [X_{ld}]\right| \text{ or } \left|[\theta_{k+n}] - [\theta_{pk}]\right| \le \alpha \left|[\theta_k] - [\theta_{pk}]\right| \quad \alpha \in (0,1] \end{array} \right\} \tag{7-8}$$

where X_{k+n} denotes the predicted coordinates and X_k the actual coordinates, θ_{k+n} denotes the predicted orientation from the actual robot coordinates and θ_k the actual orientation. The parameter θ_{pk} is the desired orientation from the actual WMR position towards the desired coordinates. The limitation of the input signal is taken into account in the first constraint, where G_0 and G_1 respectively denote the dead zone and saturation of the DC motors. The latter is a contractive constraint [Wan, 07]. In this case, coordinates or orientation convergence should be accomplished. Contractive constraints arise in a convergence towards the desired trajectory, until updating a new trajectory command**.** Therefore, path planning consists of a set of points, obtained within the available field of perception, and tracked by the LMPC strategy.

7.4.2 The LMPC algorithms and simulation results

By using the basic ideas introduced in the previous subsection, the LMPC algorithms have the following steps:

1. Read the actual position
2. Minimise the cost function and to obtain a series of optimal input signals
3. Choose the first obtained input signal as the command signal.
4. Go back to the step 1 in the next sampling period.

The minimisation of the cost function is a nonlinear problem in which the following equation should be verified:

$$f(\alpha x+\beta y)\leq \alpha f(x)+\beta f(y) \qquad (7-9)$$

It is a convex optimisation problem caused by the trigonometric functions used in (7-4). The use of interior point methods can solve the above problem [Boyd and Vanderberghe, 04]. Among many algorithms that can solve the optimisation, descent methods are used, such as the gradient descent method. The gradient descent algorithm has been implemented in this work. In order to obtain the optimal solution, some constraints over the inputs are taken into account:

- The signal increment is kept fixed within the prediction horizon.
- The input signals remain constant during the remaining interval of time.

The above considerations will result in the reduction of the computation time and the smooth behaviour of the robot during the prediction horizon. Thus, the set of available input is reduced to one value [Maciejowski, 02]. In order to reduce the optimal signal value search, the possible input sets are considered as a bi-dimensional array, as shown in Figure 7.7. Then, the array is decomposed into four zones, and the search is just located to analyse the centre points of each zone. It is considered to be the region that offers better optimisation, where the algorithm is repeated for each sub-zone, until no sub-interval can be found.

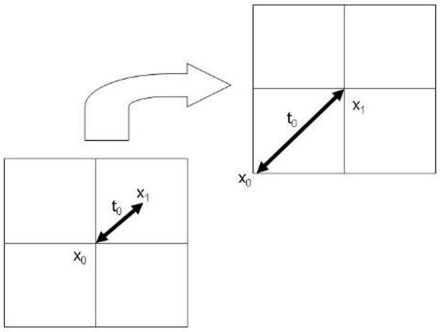


Figure 7.7: Subinterval search.

The results were obtained by testing all possible inputs and the subinterval search algorithm, which were compared by simulating a 2m straight line tracking, as shown in Figure 7.8. The results show the discrepancy between the two methods.

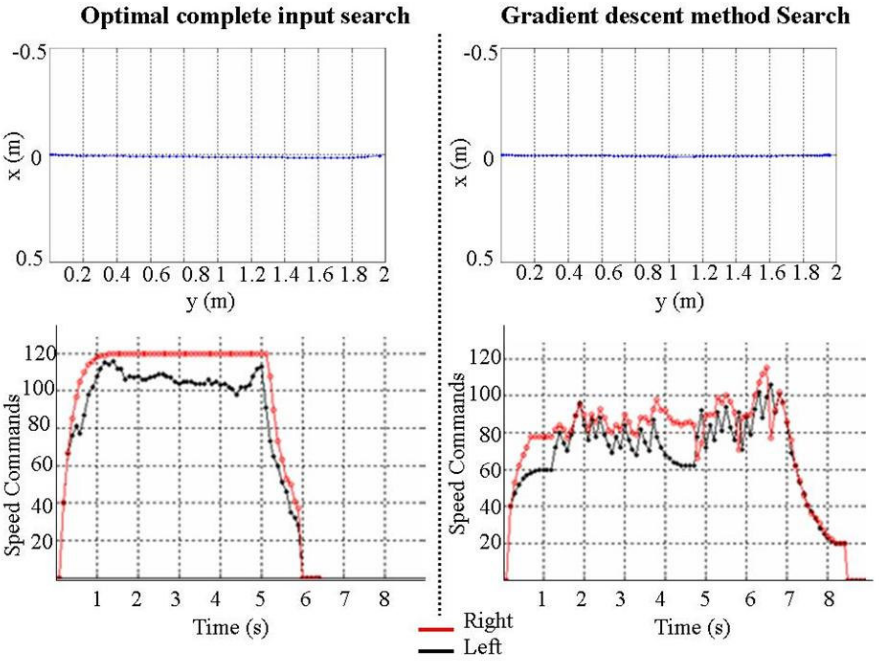


Figure 7.8: Trajectory tracking and command speed profiles during 2 m straight line tracking simulation, by computing the cost function for all the input combinations or by using the gradient descent method.

The subinterval gradient descent method does not usually give the optimal solution. However, when acceleration is produced both results are similar. Figure 7.9.a illustrates this case by showing the results of cost function obtained by all possible inputs, in which a unique minimum is found. However, when the acceleration process is finished and other criteria as final point and desired trajectory distances are considered, only suboptimal solutions are obtained. As shown in Figure 7.9.b, local minimum can be obtained instead of being globally optimal. The gradient descent results can be considered as nearly optimal.

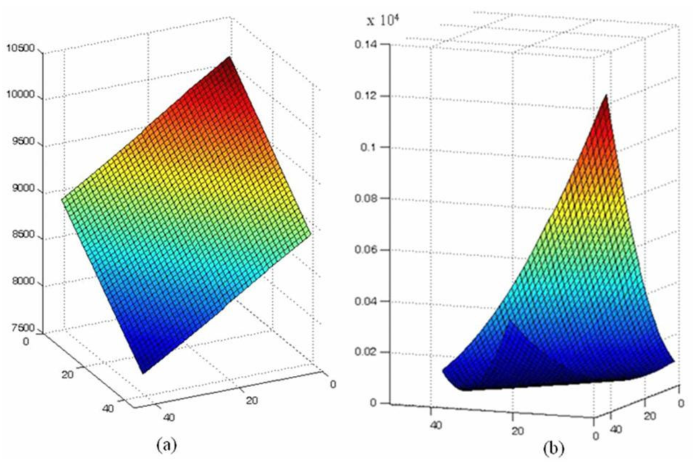


Figure 7.9: Two interval search grids.

Once the algorithm is proposed, several simulations have been carried out in order to verify the effectiveness, and then to make the improvements. Thus, when only the desired coordinates are considered, the robot cannot reach the final point. Figure 7.10 shows that the inputs can minimise the cost function by shifting the robot position to the left.

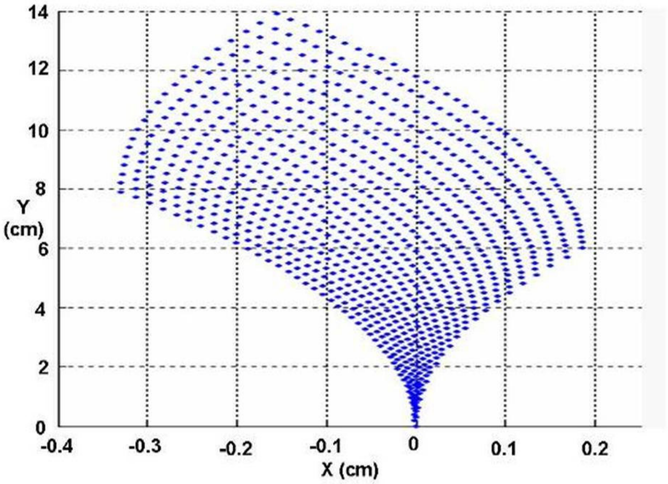


Figure 7.10: Predicted coordinates from speed zero, n=5 and m=3; where n and m are the prediction and control horizons.

The reason for this can be found in Table 7.2, where the right motor has more gain than the left for high speeds.

This problem can be easily solved by considering a straight-line trajectory from the point where the last perception was acquired to the final desired point relating to the local field of perception [Pacheco and Luo, 07D]. Thus, the trajectory should be included into the LMPC cost function. Figure 7.11 shows a simulated result of LMPC for WMR by first using the orientation error as cost function and then the local trajectory distance and the final desired point for the optimisation.

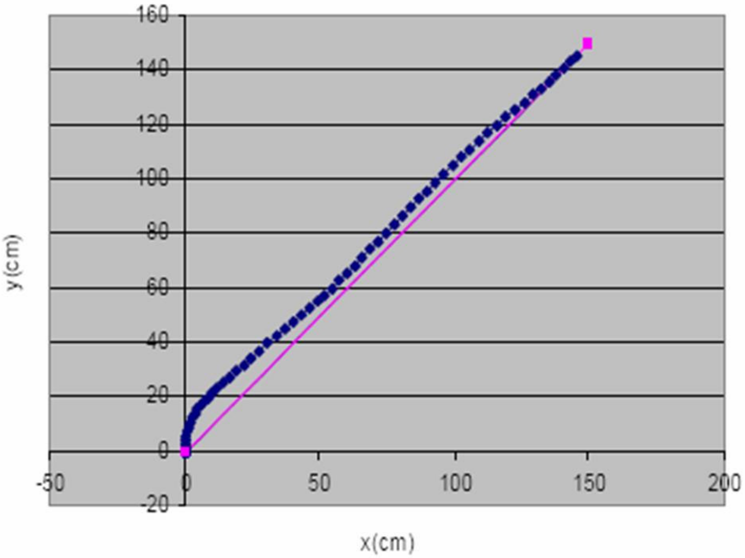


Figure 7.11: LMPC simulated results following a 45º trajectory.

The prediction horizons between 0.5s and 1s were proposed and the computation time for each LMPC step was set to less than 100ms, running in an embedded PC of 700MHz. The simulated results, using a prediction horizon (n=10) and a control horizon (m=5) are shown in Figure 7.12.

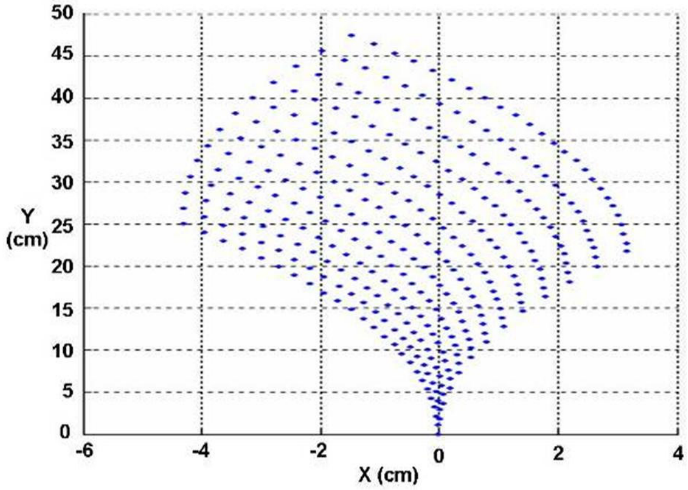


Figure 7.12: Predicted coordinates from speed zero, n=10 and m=5.

The possible coordinates available for prediction when the horizon is shorter, as shown in Figure 7.10, depict a denser possibility of coordinates. The trajectory tracking and final point achievement are other interesting aspects to be analysed. Figure 7.13 shows the results obtained in tracking a straight line of 2 metres by using two different prediction horizons.

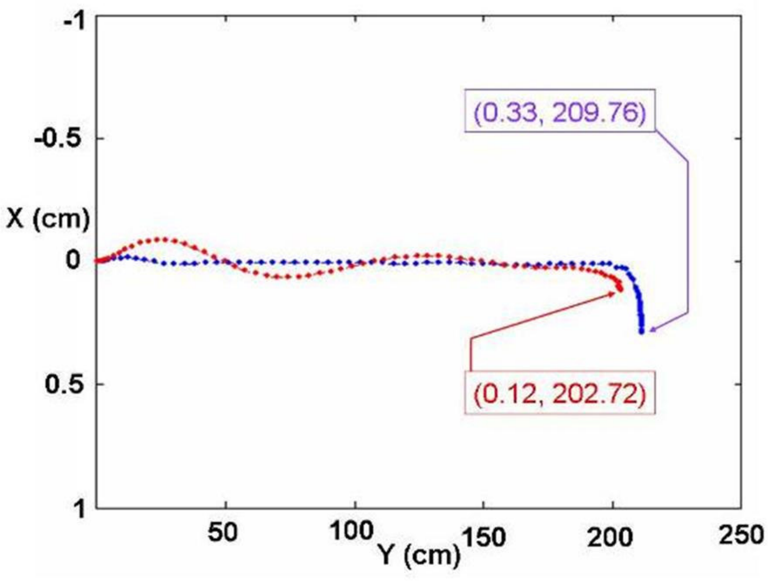


Figure 7.13: Trajectory tracking in red (n=10 and m=5), and in blue (n=5 and m=3). The larger prediction horizon shows a closer final point achievement and a worse trajectory tracking.

Figure 7.14 shows the velocity of both wheels with the two strategies used. The wide prediction strategy shows a softer behaviour due to the bigger control horizon.

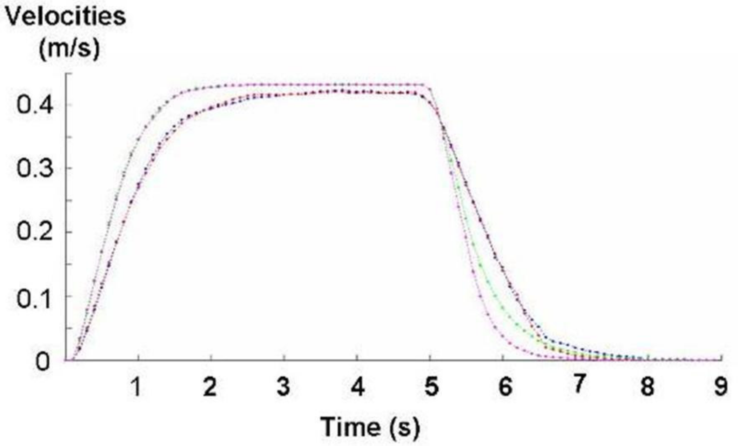


Figure 7.14: Wheel speeds during straight line tracking of 2 meters. The red and blue dots respectively show the right and left speeds (with n=10 and m=5). The magenta and green dot lines depict the right and left speeds (with n=5 and m=3).

Table 7.3 shows the LMPC processing time for different horizons of prediction when complete optimal values search or the gradient method are used.

Table 7.3: LMPC processing times

Horizon of prediction (n)	Complete search method	Gradient descent method
n=5	45ms	16ms
n=8	34ms	10ms
n=10	25ms	7ms

Surprisingly, when the horizon is increased the computing time is decreased. The reason for this is found in the fact that the signal increment is kept fixed within the prediction horizon. Thus, the maximum input value possibilities decrease with larger horizons. Hence for *n=5* there are 1764 possibilities (*42x42*), and for n=10 there are 625 (*25x25*).

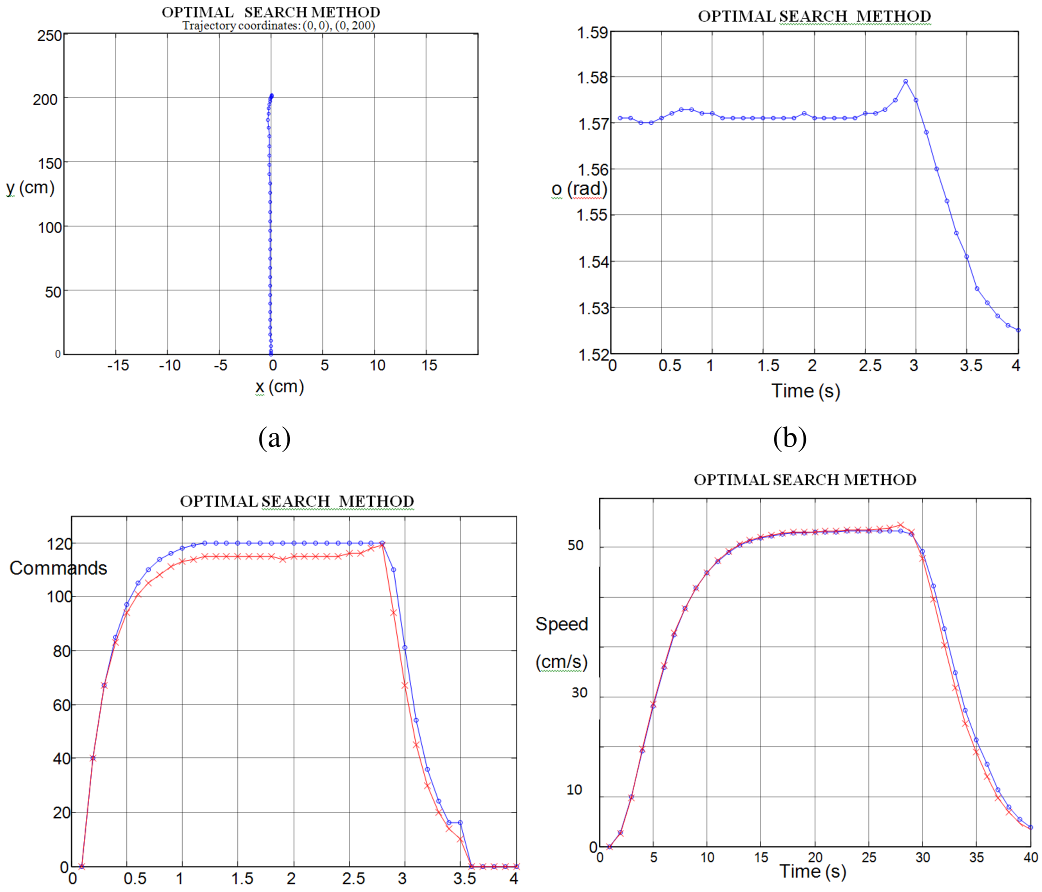
7.4.3 Trajectory-tracking simulation results

This subsection presents the trajectory simulation results obtained when two different cost functions are tested by using optimal or gradient search methodologies. The parameters used in the cost function are:

- Final point distance and trajectory distance
- Final point distance and orientation difference towards desired final point.

7.4.3.1 Trajectory-tracking results using the trajectory distance factor

This subsection shows the trajectory-tracking simulated results obtained when desired point and trajectory distance factors are used as a cost function parameter. Figure 7.15 shows the simulated results obtained when short horizon and optimal search are used. The parameters used in the cost function are the final point distance and trajectory distance. The trajectory is composed by the "*xy*" coordinates, (0, 0) and (0, 200), given in cm.



(a) (b)

(c) (d)

Figure 7.15 Trajectory tracking simulated results for a two meters straight line by using optimal search method. The parameters used in the cost function are the final point distance and trajectory distance. (a) Trajectory-tracking simulated *x* and *y* coordinates are depicted in blue, in red appears the trajectory to be tracked. (b) Trajectory-tracking simulated orientation results. (c) Trajectory-tracking comanded speed results that are depicted in blue for the left wheel and in red for the right wheel. (d) Trajectory-tracking mesured speeds that are depicted in blue for the left wheel and in red for the right wheel.

Figure 7.16 shows the simulated results obtained when short horizon and gradient interval search are used. The parameters used in the cost function are the final point distance and trajectory distance. The trajectory is composed by the "*xy*" coordinates, (0, 0) and (0, 200), given in cm.

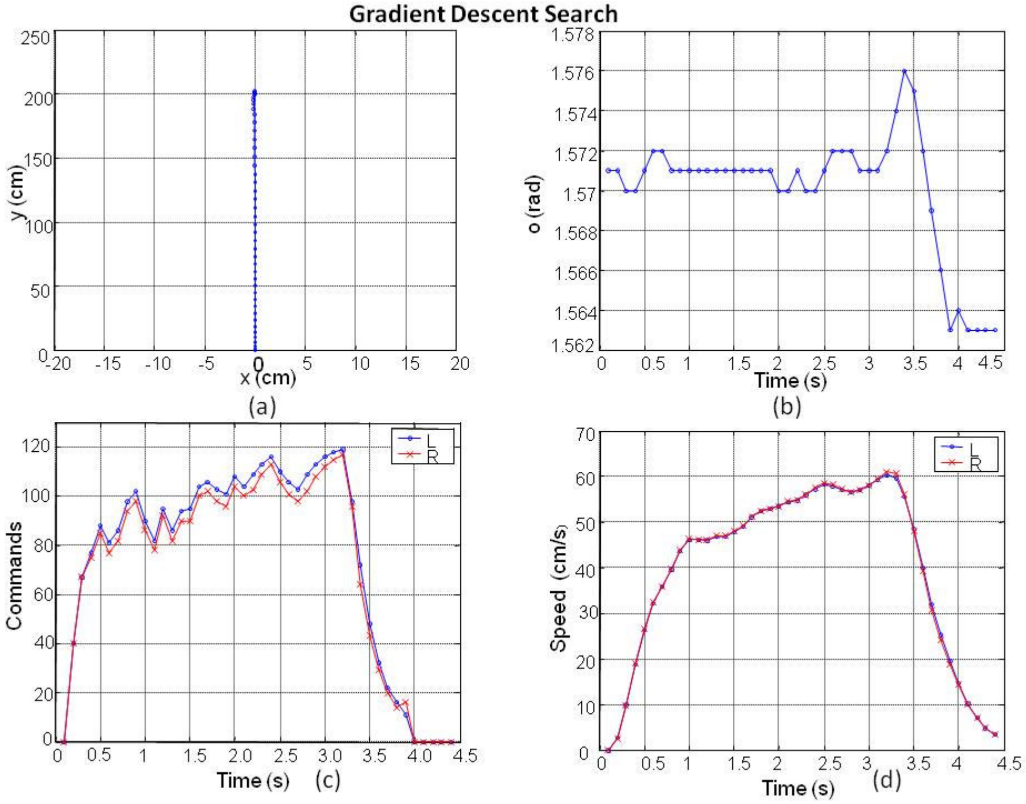


Figure 7.16: Trajectory-tracking simulated results for a two meters straight line by using gradient interval descent search method. The parameters used in the cost function are the final point distance and trajectory distance. (a) Trajectory-tracking simulated *x* and *y* coordinates are depicted in blue, in red appears the trajectory to be tracked. (b) Trajectory-tracking simulated orientation results. (c) Trajectory-tracking comanded speed results that are depicted in blue for the left wheel and in red for the right wheel. (d) Trajectory-tracking mesured speeds that are depicted in blue for the left wheel and in red for the right wheel

Figure 7.17 shows the simulated results obtained when the trajectory is composed by the "*xy*" coordinate, (0, 0), (0, 100) and (-50, 200), given in cm.

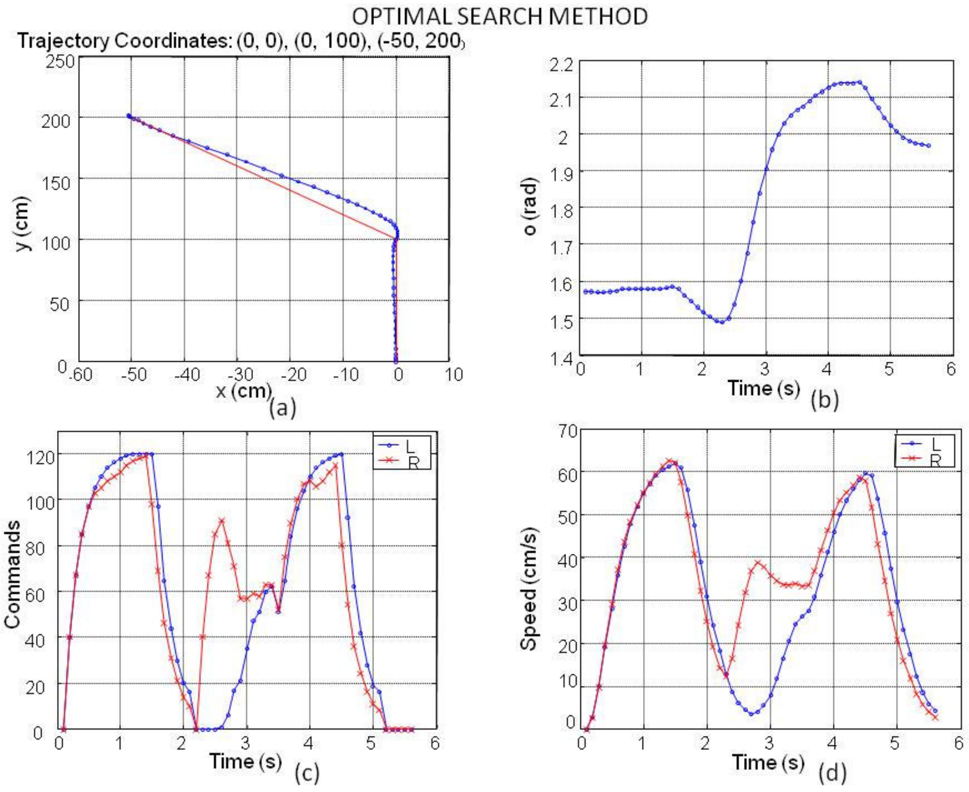


Figure 7.17: Trajectory-tracking simulated results by using optimal search method. The trajectory is composed by the "*xy*" coordinate, (0, 0), (0, 100) and (-50, 200), given in cm.The parameters used in the cost function are the final point distance and trajectory distance. (a) Trajectory-tracking simulated *x* and *y* coordinates are depicted in blue, in red appears the trajectory to be tracked. (b) Trajectory-tracking simulated orientation results. (c) Trajectory-tracking comanded speed results that are depicted in blue for the left wheel and in red for the right wheel. (d) Trajectory-tracking mesured speeds that are depicted in blue for the left wheel and in red for the right wheel

Figure 7.18 shows the simulated results obtained when the trajectory is composed by the "*xy*" coordinate, (0, 0), (0, 100) and (-50, 200), given in cm.

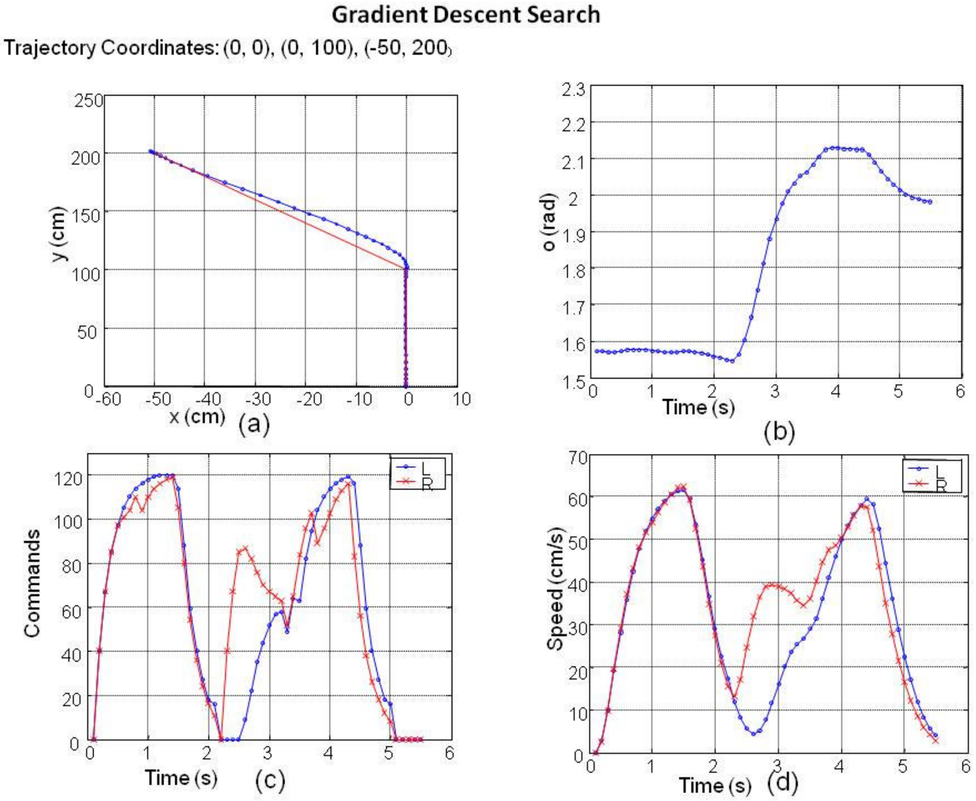


Figure 7.18: Trajectory-tracking simulated results by using gradient descent method. The trajectory is composed by the "*xy*" coordinates, (0, 0), (0, 100) and (-50, 200), given in cm.The parameters used in the cost function are the final point distance and trajectory distance. (a) Trajectory-tracking simulated *x* and *y* coordinates are depicted in blue, in red appears the trajectory to be tracked. (b) Trajectory-tracking simulated orientation results. (c) Trajectory-tracking comanded speed results that are depicted in blue for the left wheel and in red for the right wheel. (d) Trajectory-tracking mesured speeds that are depicted in blue for the left wheel and in red for the right wheel

Figure 7.19 shows the simulated results obtained when the trajectory is composed by the “*xy*” coordinate, (0, 0), (25, 50), (25, 100), (0, 150), and (0, 200) given in cm. The WMR initial coordinates are X and Y equal to zero, and θ equal to 90º.

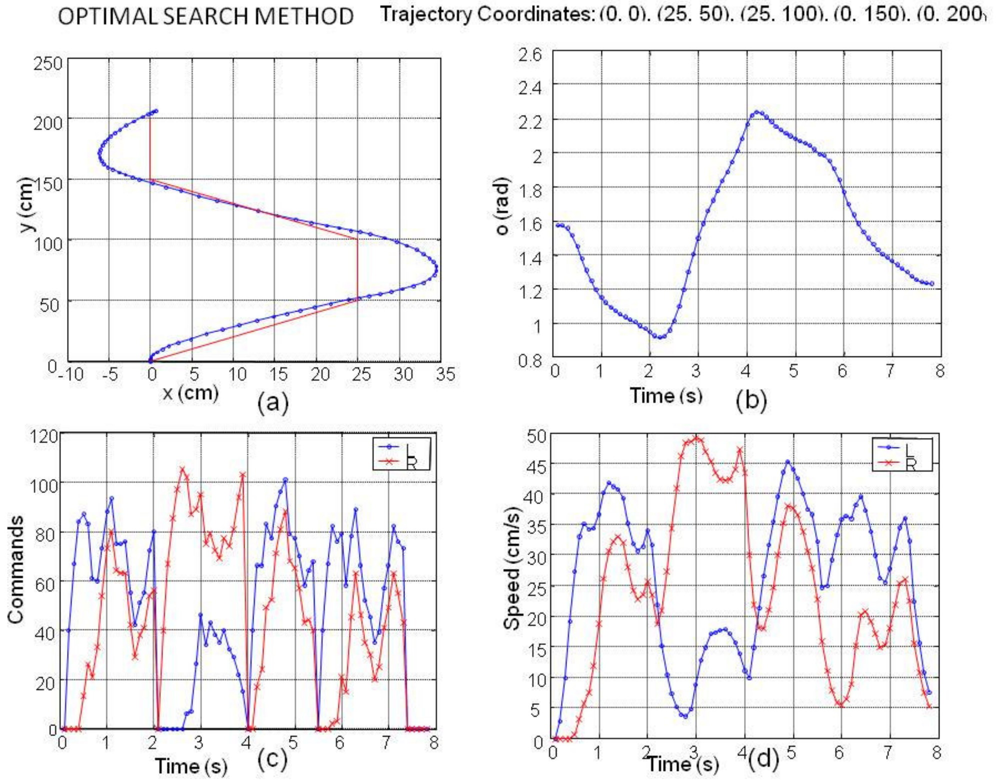


Figure 7.19: Trajectory-tracking simulated results by using optimal search method. The trajectory is composed by the “*xy*” coordinates, (0, 0), (25, 50), (25, 100), (0, 150), and (0, 200), given in cm.The parameters used in the cost function are the final point distance and trajectory distance. (a) Trajectory-tracking simulated *x* and *y* coordinates are depicted in blue, in red appears the trajectory to be tracked. (b) Trajectory-tracking simulated orientation results. (c) Trajectory-tracking comanded speed results that are depicted in blue for the left wheel and in red for the right wheel. (d) Trajectory-tracking mesured speeds that are depicted in blue for the left wheel and in red for the right wheel

Figure 7.20 shows the gradient descent simulated results obtained when the trajectory is composed by the "*xy*" coordinate, (0, 0), (25, 50), (25, 100), (0, 150), and (0, 200) given in cm. The WMR initial coordinates are X and Y equal to zero, and θ equal to 90º.

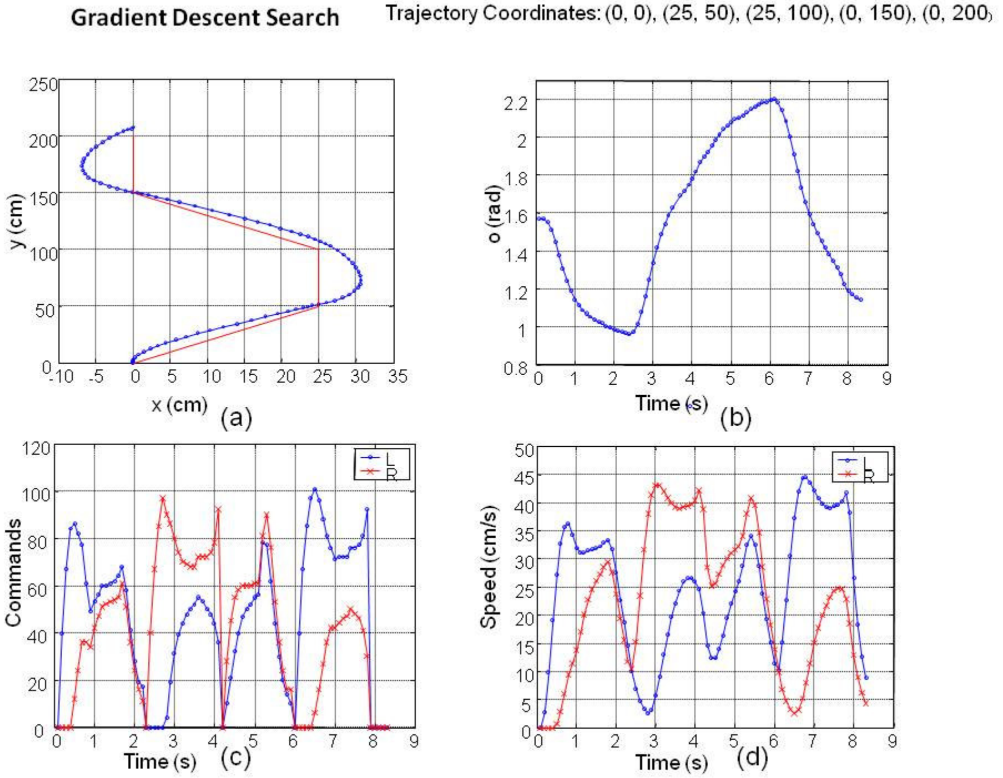


Figure 7.20: Trajectory-tracking simulated results by using gradient descent method. The trajectory is composed by the "*xy*" coordinates, (0, 0), (25, 50), (25, 100), (0, 150), and (0, 200), given in cm. The parameters used in the cost function are the final point distance and trajectory distance. (a) Trajectory-tracking simulated *x* and *y* coordinates are depicted in blue, in red appears the trajectory to be tracked. (b) Trajectory-tracking simulated orientation results. (c) Trajectory-tracking comanded speed results that are depicted in blue for the left wheel and in red for the right wheel. (d) Trajectory-tracking mesured speeds that are depicted in blue for the left wheel and in red for the right wheel

Once the simulation results are obtained, some conclusions about both search methods are obtained. Different trajectories were tested, and consequently the given performances for each trajectory should be analysed.

Straight line trajectory:

- The time performance is better for the optimal search method.
- The control effort is softer for the optimal search method.
- The trajectory-tracking accuracy is better for the gradient descent method. However, both methods provide good results.

Straight line and left turning trajectory:

- The time performance is the same for both methods.
- The control effort is softer for optimal search method.
- The trajectory-tracking accuracy is better for the gradient descent method. However, both methods provide good results.

Right turning, straight, left turning and straight trajectory:

- The time performance is better for the optimal search method.
- The control effort is softer for the optimal search method.
- The trajectory-tracking accuracy is better for the gradient descent method. However, both methods provide good results.

The simulations show that the gradient search method has better accuracy. However, the trajectory-tracking accuracy of the optimal search method is also good. Moreover, optimal search method give better results when time and control effort are analysed.

The gradient descent search method, can be considered as nearly optimal method, see subsection 7.4.2. Motivated by the results obtained and the fact that the gradient descent search can not be considered as an optimal method, experimental cost function parameters tuning is proposed for final point distance and trajectory distance factors.

Next subsection analyses the results obtained by both search methods when final point and orientation distances are used as cost function factors.

7.4.3.2 Trajectory-tracking results using the orientation distance factor

This subsection shows the trajectory-tracking simulated results obtained when desired point and orientation distance factors are used as cost function parameters. Figure 7.21 shows the simulated results obtained when short horizon and optimal search are used. The parameters used in the cost function are the final point distance and trajectory distance. The trajectory is composed by the "*xy*" coordinates, (0, 0) and (0, 200), given in cm.

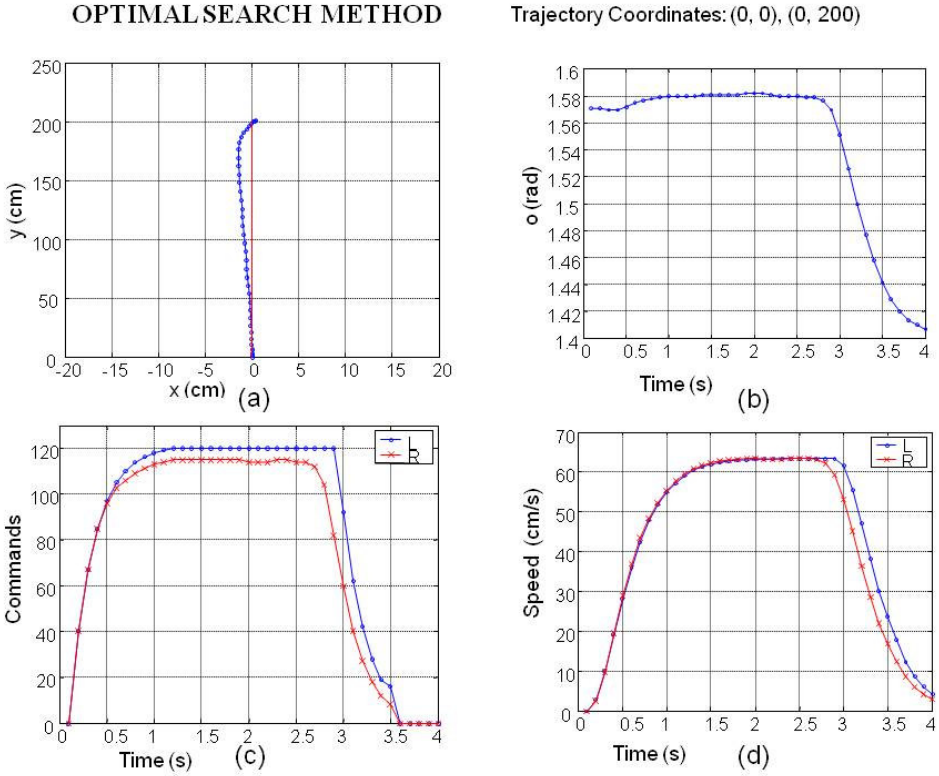


Figure 7.21: Trajectory-tracking simulated results by using optimal search method. The trajectory is composed by the "*xy*" coordinates, (0, 0) and (0, 200), given in cm. The parameters used in the cost function are the final point distance and orientation deviation factors. (a) Trajectory-tracking simulated *x* and *y* coordinates are depicted in blue, in red appears the trajectory to be tracked. (b) Trajectory-tracking simulated orientation results. (c) Trajectory-tracking comanded speed results that are depicted in blue for the left wheel and in red for the right wheel. (d) Trajectory-tracking mesured speeds that are depicted in blue for the left wheel and in red for the right wheel

Figure 7.22 shows the simulated results obtained when short horizon and gradient interval search are used. The parameters used in the cost function are the final point distance and orientation deviation factors. The trajectory is composed by the "*xy*" coordinates, (0, 0) and (0, 200), given in cm.

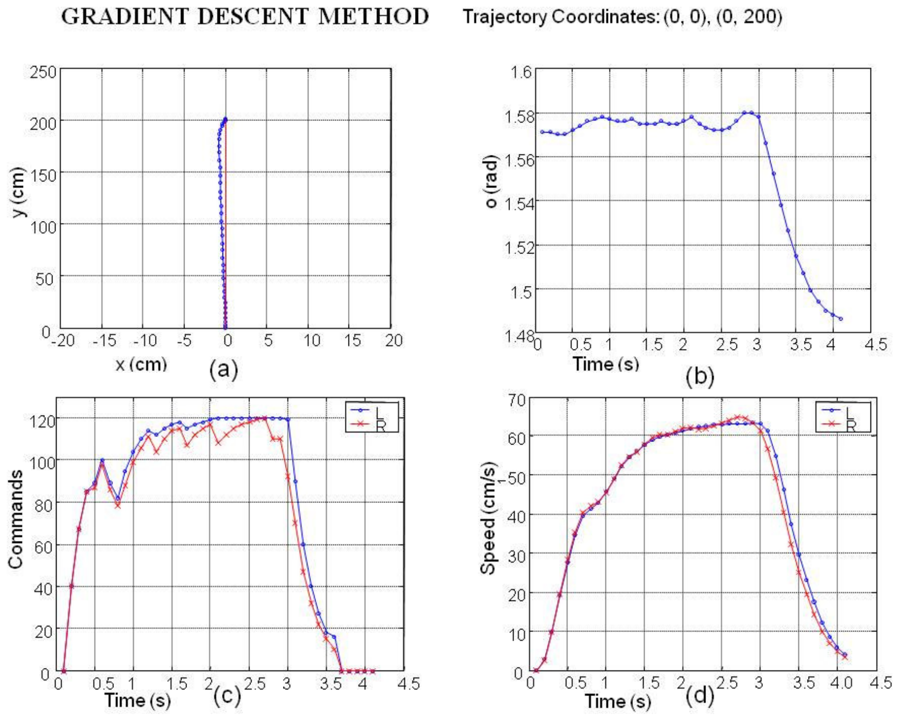


Figure 7.22: Trajectory-tracking simulated results by using gradiend descent search method. The trajectory is composed by the "*xy*" coordinates, (0, 0) and (0, 200), given in cm. The parameters used in the cost function are the final point distance and orientation deviation factors. (a) Trajectory-tracking simulated *x* and *y* coordinates are depicted in blue, in red appears the trajectory to be tracked. (b) Trajectory-tracking simulated orientation results. (c) Trajectory-tracking comanded speed results that are depicted in blue for the left wheel and in red for the right wheel. (d) Trajectory-tracking mesured speeds that are depicted in blue for the left wheel and in red for the right wheel

Figure 7.23 shows the simulated results obtained by the optimal search method when the trajectory is composed by the "*xy*" coordinates, (0, 0), (0, 100) and (-50, 200), given in cm.

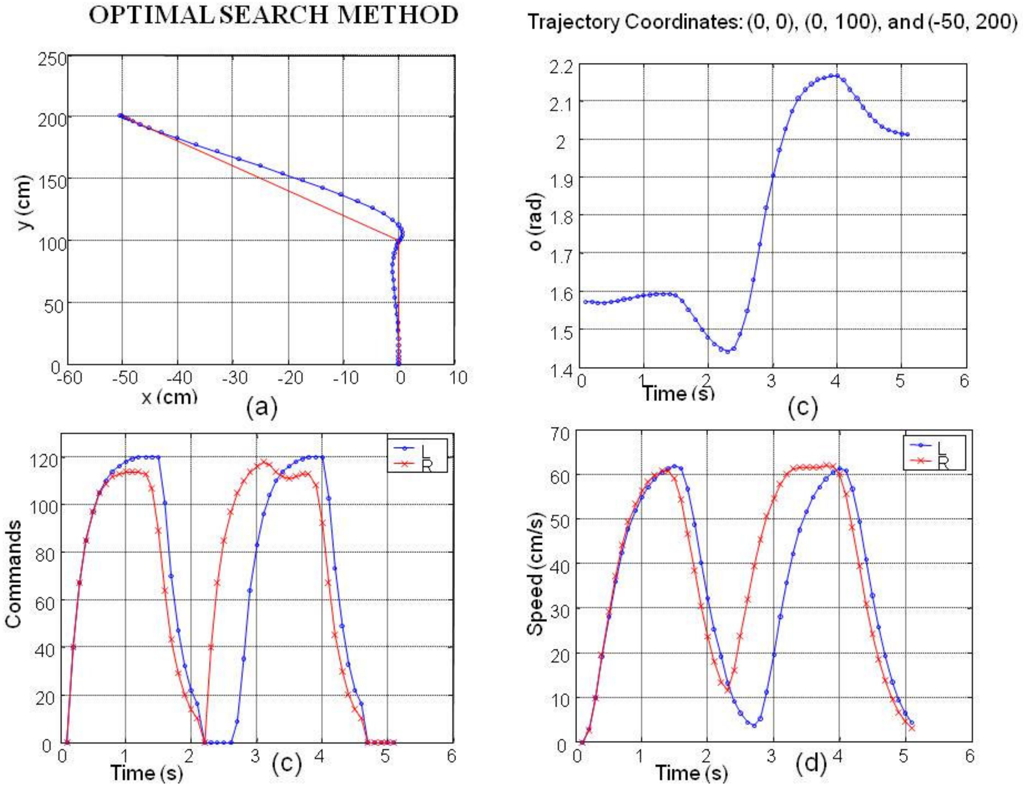
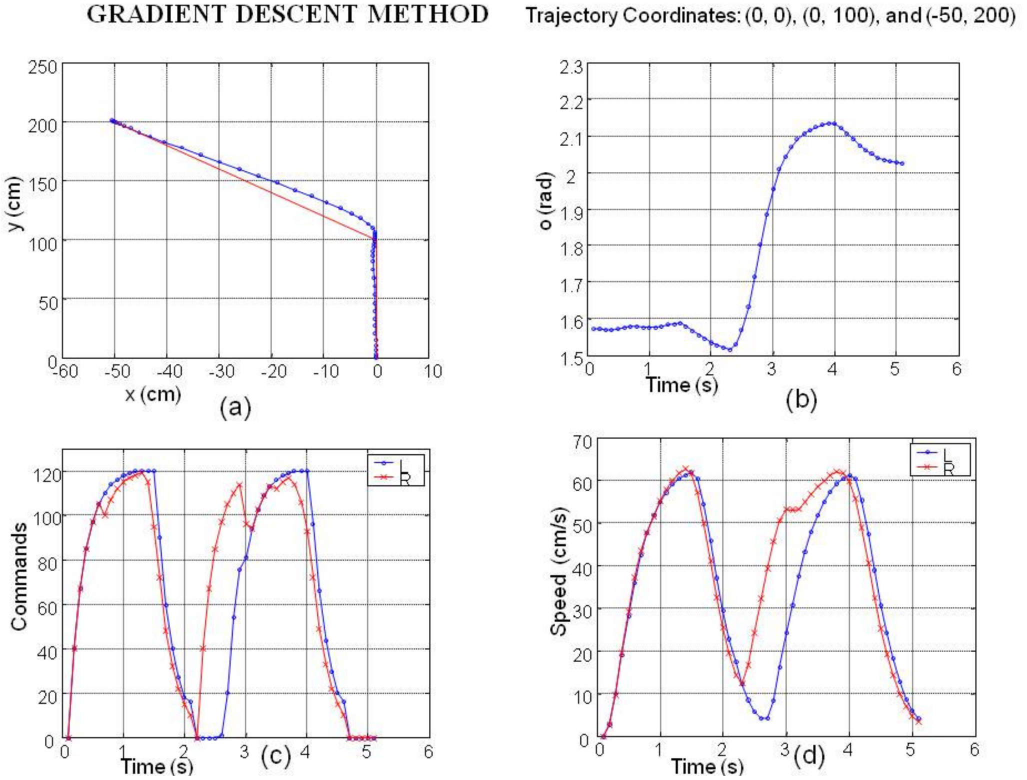


Figure 7.23: Trajectory-tracking simulated results by using optimal search method. The trajectory is composed by the "*xy*" coordinates, (0, 0), (0, 100), and (-50, 200), given in cm. The parameters used in the cost function are the final point distance and orientation deviation factors. (a) Trajectory-tracking simulated *x* and *y* coordinates are depicted in blue, in red appears the trajectory to be tracked. (b) Trajectory-tracking simulated orientation results. (c) Trajectory-tracking comanded speed results that are depicted in blue for the left wheel and in red for the right wheel. (d) Trajectory-tracking mesured speeds that are depicted in blue for the left wheel and in red for the right wheel

Figure 7.24 shows the simulated results obtained by the gradient descent search method when the trajectory is composed by the "*xy*" coordinates, (0, 0), (0, 100) and (-50, 200), given in cm.



GRADIENT DESCENT METHOD Trajectory Coordinates: (0, 0), (0, 100), and (-50, 200)

Figure 7.24: Trajectory-tracking simulated results by using gradiend descent search method. The trajectory is composed by the "*xy*" coordinates, (0, 0), (0, 100), and (-50, 200), given in cm. The parameters used in the cost function are the final point distance and orientation deviation factors. (a) Trajectory-tracking simulated *x* and *y* coordinates are depicted in blue, in red appears the trajectory to be tracked. (b) Trajectory-tracking simulated orientation results. (c) Trajectory-tracking comanded speed results that are depicted in blue for the left wheel and in red for the right wheel. (d) Trajectory-tracking mesured speeds that are depicted in blue for the left wheel and in red for the right wheel

Figure 7.25 shows the simulated results obtained by optimal search method when the trajectory is composed by the "*xy*" coordinates, (0, 0), (25, 50), (25, 100), (0, 150), and (0, 200) given in cm. The WMR initial coordinates are X and Y equal to zero, and θ equal to 90º.

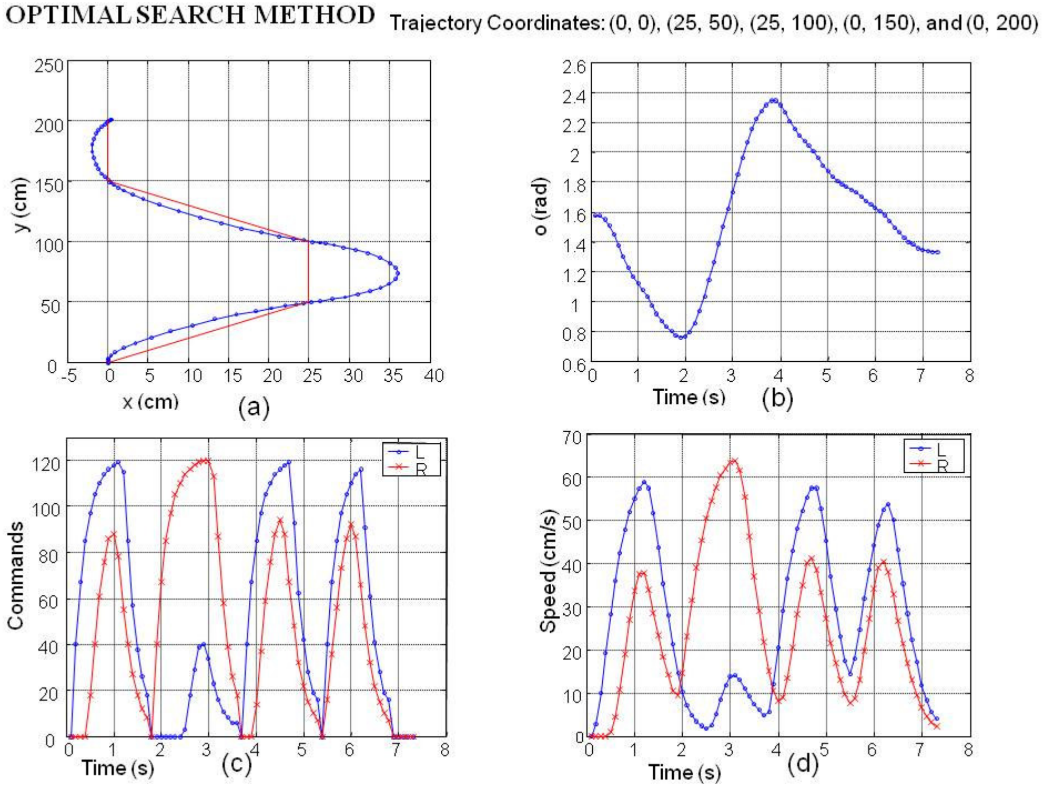


Figure 7.25: Trajectory-tracking simulated results by using optimal search method. The trajectory is composed by the "*xy*" coordinates, (0, 0), (25, 50), (25, 100), (0, 150), and (0, 200), given in cm. The parameters used in the cost function are the final point distance and orientation deviation factors. (a) Trajectory-tracking simulated *x* and *y* coordinates are depicted in blue, in red appears the trajectory to be tracked. (b) Trajectory-tracking simulated orientation results. (c) Trajectory-tracking comanded speed results that are depicted in blue for the left wheel and in red for the right wheel. (d) Trajectory-tracking mesured speeds that are depicted in blue for the left wheel and in red for the right wheel

Figure 7.26 shows the simulated results obtained by using the gradient descent method when the trajectory is composed by the "*xy*" coordinates, (0, 0), (25, 50), (25, 100), (0, 150), and (0, 200) given in cm. The WMR initial coordinates are X and Y equal to zero, and θ equal to 90º.

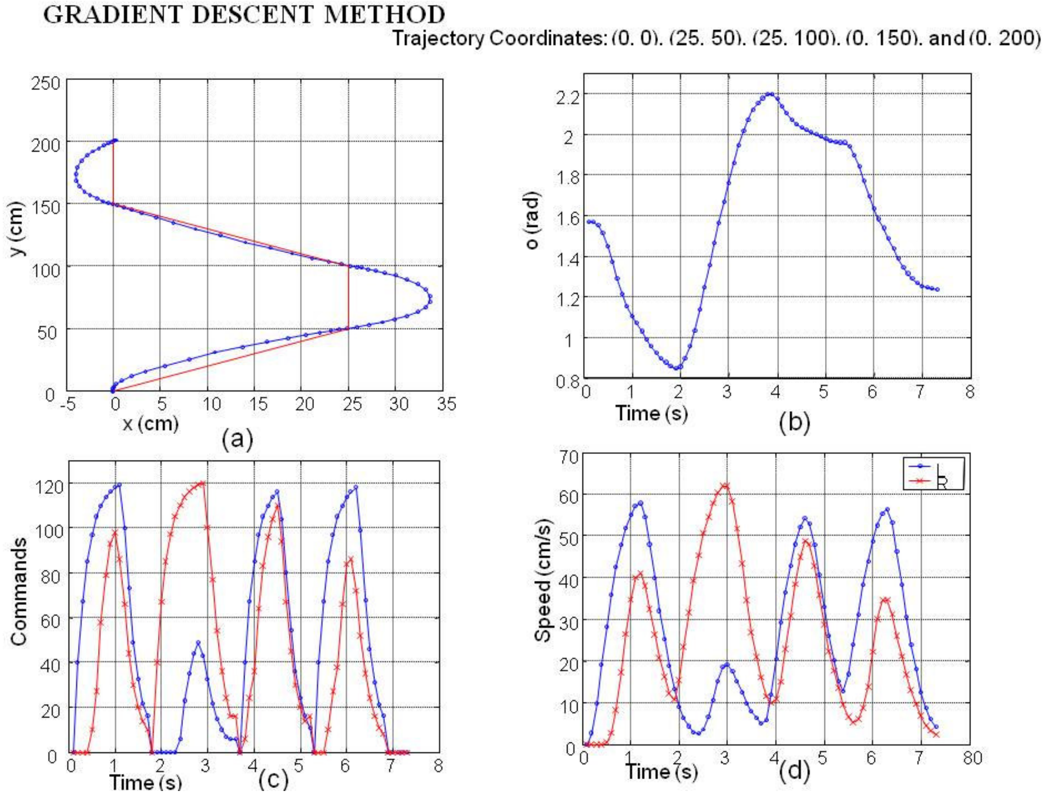


Figure 7.26: Trajectory-tracking simulated results by using the gradiend descent search method. The trajectory is composed by the "*xy*" coordinates, (0, 0), (25, 50), (25, 100), (0, 150), and (0, 200), given in cm. The parameters used in the cost function are the final point distance and orientation deviation factors. (a) Trajectory-tracking simulated *x* and *y* coordinates are depicted in blue, in red appears the trajectory to be tracked. (b) Trajectory-tracking simulated orientation results. (c) Trajectory-tracking comanded speed results that are depicted in blue for the left wheel and in red for the right wheel. (d) Trajectory-tracking mesured speeds that are depicted in blue for the left wheel and in red for the right wheel

Once the simulation results are obtained, some conclusions about both search methods, and control laws are obtained. Different trajectories were tested, and consequently the given performances for each trajectory should be analysed.

Straight line trajectory:

- The time performance is better for the optimal search method.
- The control effort is softer for the optimal search method.
- The trajectory-tracking accuracy is better for the gradient descent method. However, the control law presented in the previous subsection, which uses the final point and trajectory distances, give better results.

Straight line and left turning trajectory:

- The time performance is the same for both methods. Moreover, the control results are better than the one obtained when final point and trajectory distances are used.
- The control effort is softer for optimal search method, but good results are obtained by both methods.
- The trajectory-tracking accuracy is better for the gradient descent method. However, the control law presented in the previous subsection, which use the final point and trajectory distances, give better results.

Right turning, straight, left turning and straight trajectory:

- The time performance is similar for both methods. Moreover, the control results obtained are better than the obtained when final point and trajectory distances are used.
- The control effort is softer for the optimal search method, but good results are obtained by both methods.
- The trajectory-tracking accuracy is better for the gradient descent method. However, the control law presented in the previous subsection, which use the final point and trajectory distances, give better results.

The simulations depicts that the gradient search method has better accuracy, and the optimal search method give better results when time and control effort are analysed. However, when analysing results obtained with the control law, which uses the final point and trajectory tracking distances, worse trajectory-tracking accuracy and better time performances are obtained.

The gradient descent search method, can be considered as a nearly optimal method, see subsection 7.4.2. Motivated, by the results obtained and the fact that the gradient descent search can not be considered as an optimal method, experimental cost function parameters tuning is proposed in the next section. Therefore, next subsection studies the performance of the two control laws simulated in these sections by using optimal search method. Experimental cost function parameter tuning is done by using statistical analysis. Moreover, the results obtained are tested by using a set of different trajectories.

7.5 Trajectory-tracking experimental results: tuning the cost function parameters by factorial Design

In this Section, motivated by the results obtained in the previous simulations developed, the trajectory tracking problem and the cost function parameter weights are analysed by using experimental design. Moreover, the trajectories are considered, accordingly with the machine vision system introduced in Chapter 6, within a constrained field of perception provided by the on-robot sensor system [Pacheco and Luo, 07C]. The first subsection introduces a constrained horizon of perception provided by a monocular camera where artificial potential fields are used in order to obtain the desired coordinates within the field of view of the robot. The approach used in the trajectory-tracking formulation is introduced in the second subsection. The velocity estimation trough the use of Kalman filters is introduced in the third subsection. The importance of the cost function parameter weights is analysed in the remainder subsections by developing the factorial design of experiments. The objective is to analyse the time and the trajectory-tracking accuracy as a function of the cost function factor values.

7.5.1 The local field of perception and the attraction potential fields

In order to test the LMPC by using constrained local perception, the field of view obtained by a monocular camera has been used. The available scene coordinates appear as an image, in which the camera setup and pose knowledge are used, and projective perspective is assumed, to make each pixel coordinate correspond to a 3D scene coordinate [Horn, 98]. Figure 7.27 shows a local map provided by the camera, which corresponds to a field of view with a horizontal angle of 48º, a vertical angle of 37º, *H* set to 109cm and a tilt angle of 32º.

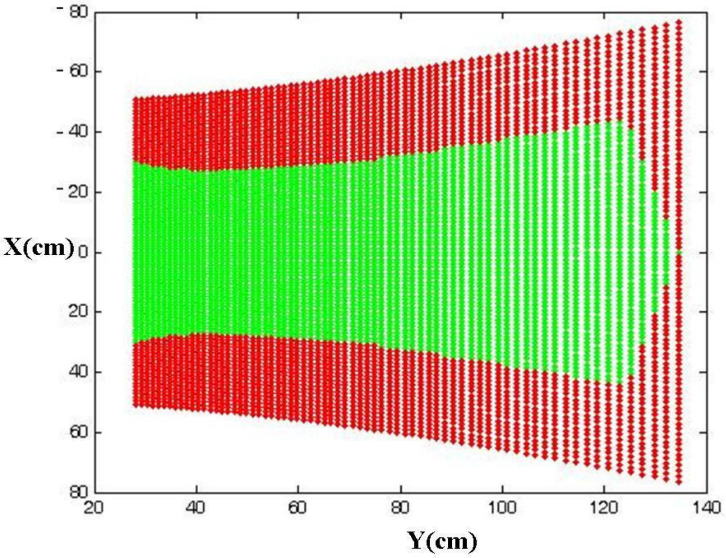


Figure 7.27: Floor local perception grid coordinates relative to the robot position, at the 96x72 size level.

It is pointed out that the available floor coordinates are reduced due to the WP (Wide-Path) of the robot. It should also be noted that for each column position corresponding to scene coordinates Y_j, there are *R* row coordinates $X_{i,j}$. Once perception is introduced, the problem is formulated as finding the optimal cell that brings the WMR close to the desired coordinates (X_d, Y_d) by searching for the closest local desired coordinates $(X_{ld},$

Y_{ld}) within the available local coordinates (X_{ij}, Y_j). In this sense, perception is considered to be a local receding horizon on which the trajectory is planned. The local desired cell is obtained by minimising a cost function *J* that should act as a potential field corridor. Thus, the cost function is minimised by attracting the robot to the desired objective through the free available local cell coordinates. Due to the narrow field of perception, the robot should be orientated towards the goal. Hence, the θ_{lg} value depicts the maximal angle that can be attained within the available local grid. Thus, first the orientation error towards the goal should be minimised:

$$\text{When} \quad \operatorname{atan}\left({}^{Y_d}\!/\!{}_{X_d} \right) = \theta_d > \theta_{\text{lg}}$$

$$J(i,j) = \min\left(\operatorname{abs}\left(\operatorname{atan}\left({}^{Y_1}\!/\!{}_{X_{i1}} \right) - \operatorname{atan}\left({}^{Y_d}\!/\!{}_{X_d} \right) \right) \right) \qquad (7\text{-}10)$$

$$\forall \ 1 \le i \le R \,,\, j = 1$$

Once the WMR is oriented towards the goal, when no obstacles are met, the Euclidean distance is proposed as a potential field:

$$\text{When} \quad \operatorname{atan}\left({}^{Y_d}\!/\!{}_{X_d} \right) = \theta_d \le \theta_{\text{lg}}$$

$$J(i,j) = \min\left(\left(X_{ij} - X_d \right)^2 + \left(Y_j - Y_d \right)^2 \right)^{1/2} \qquad (7\text{-}11)$$

$$\forall \ 1 \le i \le \mathrm{R} \,,\, 1 \le j \le C$$

When obstacles are met, the left and right closer obstacle vertex coordinates, increased by the robot WP, are tested within expression (7-11). Thus, the one that produces minimal cost function is selected [Pacheco and Luo, 07C].

7.5.2 The trajectory-tracking experimental approach by using LMPC methods

The trajectory tracking performance is improved by the adequate choice of a cost function that is derived from (7-7) and consists of a quadratic function containing some of the following four parameters to be minimised:

- The squared Euclidean *approaching point distance* (APD) between the local desired coordinates, provided by the on-robot perception system, and the actual robot position.
- The squared *trajectory deviation distance* (TDD) between the actual robot coordinate and a straight line that goes from the robot coordinates, when the local frame perception was acquired, and the local desired coordinates belonging to the referred frame of perception.
- The third parameter refers to changes allowed to the input signal.
- The last parameter consists of the squared *orientation deviation* (OD) that is used only when the desired orientation is greater than a selected threshold. In this case, other parameters of the cost function are not used.

One consideration that should be taken into account is the different distance magnitudes. In general, the approaching distance could be more than one metre. However, the magnitude of the deviation distance is normally in the order of cm, which becomes effective only when the robot is approaching the final desired point. Hence, when reducing the deviation distance further to less than 1cm is attempted, an increase in the weight value for the deviation distance in the cost function is proposed. The trajectory tracking accuracy is an important aspect to be considered. In this context, the odometer system performance was analysed by measuring the accuracy of the system. It was done by commanding long trajectories along lab corridors. After calibrating the odometer, the results showed that a commanded trajectory of 22m provided averaged final distance errors of less than 0.5m, and angular orientation errors of less than 7º. Hence in this research, local trajectories of less than 1.5m are analysed according with the narrow visual perception provided. Thus, the odometer system errors can be neglected when local trajectories are considered and the odometer system is locally used to compute LMPC trajectory tracking errors. The tested trajectories are obtained from the available set of local map coordinates as shown in Figure 7.27. The LMPC results are analysed when different trajectories tracking are commanded, as depicted in Figure 7.28. Denote E_1 as the average final error, E_2 the maximal average tracking error, E_3 the average tracking error, E_4 the standard deviation of average tracking error. Table 7.4 presents the statistics concerning about the error obtained in cm testing the trajectories shown in Figure 7.28.

Table 7.4: Point to point trajectory tracking statistics

Trajectory	E_1	E_2	E_3	E_4
From (0,0) to (0,130)	4.4cm	0.9cm	0.54cm	0.068
From (0,0) to (34,90)	3.8cm	3.9cm	2.3cm	0.82
From (0,0) to (25,40)	4.5cm	5.3cm	3.9cm	1.96

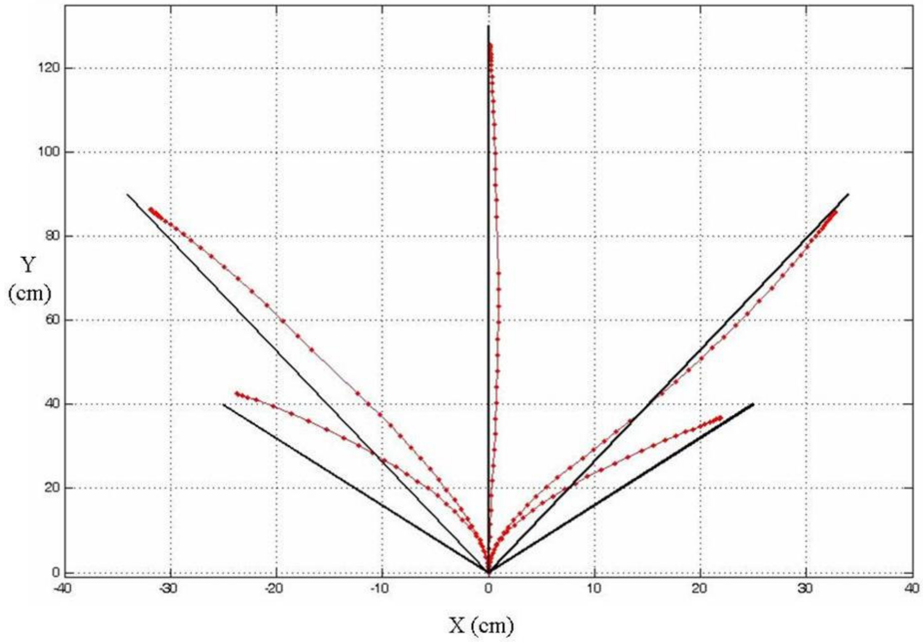


Figure 7.28: Trajectory tracking tested from point to point by using the available local map coordinates provided by the monocular perception system.

It can be seen that the accuracy of trajectory tracking, when a straight line without changing the orientation is commanded, has a deviation error of 0.54cm. However, when a turning action is performed, the error in straight line tracking is bigger as a consequence of the robot dynamics when it is moving forward. The forward movement consists in usually a steering action. Figure 7.28 gives a clue about what is happening. Thus, the major turning angle will produce the major deviation distance. Usually, it is very difficult to reduce the approaching distance to zero, due to the control difficulty of the dead zone for the WMR and to the fact that the final target is considered in the present work as being reached by the robot when the Euclidean approaching distance is less than 5cm. Other interesting results consist in testing the LMPC performance when the trajectory is composed of a set of points to be tracked. In this sense, considering the kind of robot used, a pure rotation is possible by commanding the same speed with a different direction to each motor. Hence, when a trajectory is composed of many points, two possibilities exist: *continuous movement* in an advancing direction, or *discontinuous movement* in which the robot makes the trajectory orientation changes by turning around itself at the beginning of the new straight segment. However, in this research, due to the perception considered, only *continuous movement* in an advancing direction is analysed.

7.5.3 Velocity estimation trough Kalman filter

In the MPC research, Kalman filter has been employed for improving the system performance. In the present work, the technique of Kalman filter is used for the velocity estimation. It is implemented as output prediction estimation by using both inputs and outputs, and the knowledge of the estimated outputs obtained by the system model [Aström and Wittenmark, 88]. The implemented filter is given by the following equations:

$$\tilde{x}(k+1|k)=G\tilde{x}(k|k)+Hu(k) \qquad (7\text{ - }12)$$

$$P(k+1|k)=GP(k|k)G^{T}+R_{1} \qquad (7\text{ - }13)$$

where G, H, C are given by the system model, $\tilde{x}$ is the estimated system output, y is the measured output, and P represents the estimation error variance at the instant k. The initial value P(0/0) can be set to zero. Thus, recursive output estimation can be done, as function of the estimated outputs, the inputs, and the last measured output:

$$K(k+1)=P(k+1|k)C\left(R_{2}+CP(k+1|k)\right)^{-1} \qquad (7-14)$$

$$P(k+1|k+1)=P(k+1|k)-K(k+1)CP(k+1|k) \qquad (7\text{ - }15)$$

$$\tilde{x}(k+1|k+1)=\tilde{x}(k+1|k)+K(k+1)\left(y(k+1)-C\tilde{x}(k+1|k)\right) \qquad (7-16)$$

The parameters R_1 and R_2 are used to model the noise variance empirically done in this work. Figure 7.29 shows the acquired and estimated outputs.

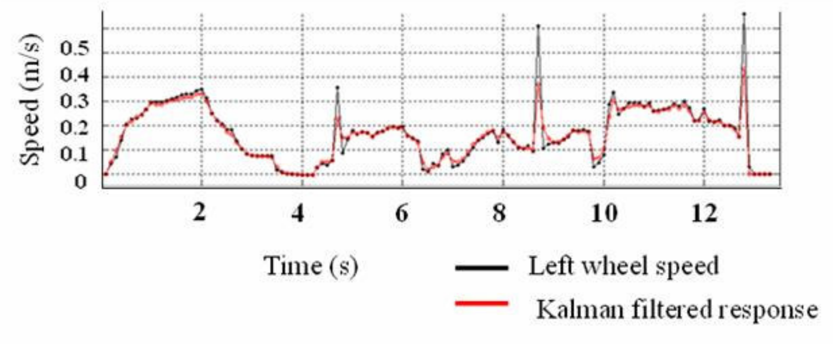


Figure 7.29: Experimental results of the Kalman filter.

As it is shown in Figure 7.29, the speed output results are improved. Therefore the noise is removed from the measured speed outputs. The use of Kalman filters in the MPC methodology is pointed by the concerning research comunity [Camacho and Bordons, 02].

7.5.4 Experimental tuning of APD and TDD factors

Section 7.5.2 has introduced the parameters considered in the cost function used in the LMPC methodology developed in this research. The parameters to be analysed are APD (approaching point distance), TDD (trajectory deviation distance), and OD (orientation deviation). These parameters can be seen as an approach of other results obtained by scientific community (Reeds & Shepp,1990). Therefore, the parameters show the WMR necessary capacity for going forward (APD), and turning (OD or APD). The current and subsequent sections present the results achieved by using factorial design in order to study the LMPC cost function tuning when APD and TDD factors are used. Section 7.5.6 and 7.5.7 depict the results obtained when APD and OD factors are used. Hence, in this section trajectory-tracking performances are analysed by means of the different factor weights applied to APD and TDD factors. The experiments are developed by considering five different kinds of trajectories within the reduced field of view as shown in Figure 7.28. Therefore, straight, wide left turning, less left turning, wide right turning, and less right turning trajectories are tested. Experiments are conducted by using factorial design with two levels of quantitative factors [Box et al., 05]. For each combination of factors two different runs are experimented. The averaged value of the three runs allows statistical analysis for each factor combination. From these standard deviations, the importance of the effects can be determined by using a rough rule that considers the effects when the value differences are similar or greater than 2 or 3 times their standard deviations. In this context, the main effects and lateral effects, related to APD and TDD, are analysed. Figure 7.30 shows the different factor combinations and their influence over the performances to be considered.

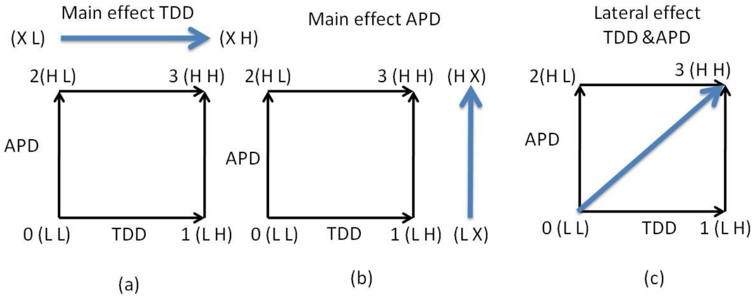


Figure 7.30: The different factor combinations and the influence directions, in which the performances should be analysed.

Thus, the main effect of APD factor, ME_{APD}, can be computed by the following expression:

$$ME_{APD} = \frac{Y_3 + Y_2}{2} - \frac{Y_1 + Y_0}{2} \qquad (7\text{-}17)$$

The performances to be analysed in this research, which are time and trajectory accuracy, are represented by Y. The subscripts depict the different factor combinations. The main effect for TDD factor, ME_{TDD}, is computed by:

$$ME_{TDD} = \frac{Y_3 + Y_1}{2} - \frac{Y_2 + Y_0}{2} \qquad (7\text{-}18)$$

The lateral effects are computed by using the following expression:

$$LE_{APD\_TDD} = Y_3 - Y_0 \qquad (7\text{-}19)$$

The Tables 7.5A, 7.5B, 7.5C, 7.5D and 7.5E, respectively show the measured performance statistics, time, trajectory accuracy and averaged speeds, for straight trajectories, wide and less left turnings and wide and less right turnings.

Table 7.5A: Performance for straight line trajectory

Straight trajectory (x, y) in cm: (0, 0) to (0, 130)				
Time (s)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	4,8	5,0	4,90	0,0200
01 (1)	4,3	4,5	4,40	0,0200
10 (2)	4,3	4,4	4,35	0,0050
11 (3)	4,3	4,2	4,25	0,0050
Average of estimated variances				0,0125
Standart deviation (σ_T)				0,1118

Straight trajectory (x, y) in cm: (0, 0) to (0, 130)				
Trajectory error (cm)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	1,143	0,447	0,795	0,2421
01 (1)	0,189	0,162	0,176	0,0004
10 (2)	1,116	0,349	0,732	0,2940
11 (3)	0,274	0,179	0,226	0,0046
Average of estimated variances				0,1352
Standart deviation (σ_{TE})				0,3678

Straight trajectory (x, y) in cm: (0, 0) to (0, 130)				
Averaged left and right speeds (cm/s)				
APD TDD	Left wheel		Right Wheel	
	Run 1	Run 2	Run 1	Run 2
00 (0)	27,420	25,288	25,706	24,749
01 (1)	29,423	27,287	29,577	27,222
10 (2)	30,293	28,989	29,000	28,758
11 (3)	27,834	32,636	27,934	32,083

Straight trajectory (x, y) in cm: (0, 0) to (0, 130)				
Averaged Speed (cm/s)				
APD TDD				
	Run 1	Run 2	Mean	Variance
00 (0)	26,563	25,019	25,791	1,1927
01 (1)	29,500	27,255	28,377	2,5211
10 (2)	29,647	28,874	29,260	0,2988
11 (3)	27,884	32,360	30,122	10,0151
Average of estimated variances				3,5069
Standart deviation (σ_{AS})				1,8727

It is pointed out that averaged speed results shown are obtained from the averaged left and right speeds.

Table 7.5B: Performance for wide left turning

wide left turning (x, y) in cm: (0, 0) to (-25, 40)				
Time (s)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	3,4	3,3	3,35	0,0050
01 (1)	3,1	3,2	3,15	0,0050
10 (2)	3,0	3,0	3,00	0,0000
11 (3)	3,0	3,1	3,05	0,0050
Average of estimated variances				0,0038
Standart deviation (σ_T)				0,0612

wide left turning (x, y) in cm: (0, 0) to (-25, 40)				
Trajectory error (cm)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	2,694	2,495	2,594	0,0198
01 (1)	3,024	3,177	3,101	0,0117
10 (2)	2,960	3,171	3,066	0,0223
11 (3)	3,078	3,477	3,278	0,0798
Average of estimated variances				0,0334
Standart deviation (σ_{TE})				0,1827

wide left turning (x, y) in cm: (0, 0) to (-25, 40)				
Averaged left and right speeds (cm/s)				
APD TDD	Left wheel		Right wheel	
	Run 1	Run 2	Run 1	Run 2
00 (0)	6,986	7,666	19,123	20,859
01 (1)	7,753	7,474	21,100	20,765
10 (2)	7,777	7,945	21,681	22,271
11 (3)	7,716	6,941	22,216	20,888

wide left turning (x, y) in cm: (0, 0) to (-25, 40)				
Averaged Speed (cm/s)				
APD TDD				
	Run 1	Run 2	Mean	Variance
00 (0)	13,055	14,263	13,659	0,7296
01 (1)	14,427	14,120	14,273	0,0471
10 (2)	14,729	15,108	14,919	0,0718
11 (3)	14,966	13,915	14,440	0,5528
Average of estimated variances				0,3504
Standart deviation (σ_{AS})				0,5919

Table 7.5C: Performance for less left turning

less left turning (x, y) in cm: (0, 0) to (-34, 90)				
Time (s)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	4,0	3,7	3,85	0,0450
01 (1)	4,5	4,8	4,65	0,0450
10 (2)	4,5	4,1	4,30	0,0800
11 (3)	4,2	4,6	4,40	0,0800
Average of estimated variances				0,0625
Standart deviation (σ_T)				0,2500

less left turning (x, y) in cm: (0, 0) to (-34, 90)				
Trajectory error (cm)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	2,652	2,260	2,456	0,0771
01 (1)	1,793	1,768	1,781	0,0003
10 (2)	1,451	2,118	1,784	0,2227
11 (3)	2,063	1,803	1,933	0,0337
Average of estimated variances				0,0835
Standart deviation (σ_{TE})				0,2889

less left turning (x, y) in cm: (0, 0) to (-34, 90)				
Averaged left and right speeds (cm/s)				
APD TDD	Left wheel		Right wheel	
	Run 1	Run 2	Run 1	Run 2
00 (0)	19,037	19,906	25,553	26,841
01 (1)	16,083	17,235	21,183	22,149
10 (2)	17,242	19,376	22,398	25,932
11 (3)	19,086	15,472	24,937	21,770

less left turning (x, y) in cm: (0, 0) to (-34, 90)				
Averaged Speed (cm/s)				
APD TDD				
	Run 1	Run 2	Mean	Variance
00 (0)	22,295	23,374	22,834	0,5816
01 (1)	18,633	19,692	19,163	0,5607
10 (2)	19,820	22,654	21,237	4,0158
11 (3)	22,012	18,621	20,316	5,7477
Average of estimated variances				2,7265
Standart deviation (σ_{AS})				1,6512

Table 7.5D: Performance for wide right turning

wide right turning (x, y) in cm: (0, 0) to (25, 40)				
Time (s)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	2,9	2,5	2,70	0,0800
01 (1)	3,0	3,1	3,05	0,0050
10 (2)	3,1	3,2	3,15	0,0050
11 (3)	2,9	3,6	3,25	0,2450
Average of estimated variances				0,0838
Standart deviation (σ_T)				0,2894

wide right turning (x, y) in cm: (0, 0) to (25, 40)				
Trajectory error (cm)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	3,043	3,041	3,042	0,0000
01 (1)	3,490	3,026	3,258	0,1076
10 (2)	2,961	2,847	2,904	0,0065
11 (3)	2,952	3,210	3,081	0,0333
Average of estimated variances				0,0368
Standart deviation (σ_{TE})				0,1919

wide right turning (x, y) in cm: (0, 0) to (25, 40)				
Averaged left and right speeds (cm/s)				
APD TDD	Left wheel		Right wheel	
	Run 1	Run 2	Run 1	Run 2
00 (0)	22,883	24,762	9,297	10,558
01 (1)	23,290	21,272	9,403	8,606
10 (2)	21,969	20,382	9,703	9,088
11 (3)	23,267	18,722	9,770	8,346

wide right turning (x, y) in cm: (0, 0) to (25, 40)				
Averaged Speed (cm/s)				
APD TDD				
	Run 1	Run 2	Mean	Variance
00 (0)	16,090	17,660	16,875	1,2325
01 (1)	16,347	14,939	15,643	0,9905
10 (2)	15,836	14,735	15,286	0,6061
11 (3)	16,519	13,534	15,026	4,4536
Average of estimated variances				1,8207
Standart deviation (σ_{AE})				1,3493

Table 7.5E: Performance for less right turning

Less right turning (x, y) in cm: (0, 0) to (34, 90)				
Time (s)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	4,9	4,7	4,80	0,0200
01 (1)	4,8	4,7	4,75	0,0050
10 (2)	4,5	4,3	4,40	0,0200
11 (3)	4,5	4,5	4,50	0,0000
Average of estimated variances				0,0113
Standart deviation (σ_T)				0,1061

Less right turning (x, y) in cm: (0, 0) to (34, 90)				
Trajectory error (cm)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	1,632	0,971	1,301	0,2185
01 (1)	0,754	0,955	0,854	0,0202
10 (2)	1,194	1,238	1,216	0,0010
11 (3)	1,004	1,128	1,066	0,0077
Average of estimated variances				0,0618
Standart deviation (σ_{TE})Desviació estàndard				0,2487

Less right turning (x, y) in cm: (0, 0) to (34, 90)				
Averaged left and right speeds (cm/s)				
APD TDD	Left wheel		Right wheel	
	Run 1	Run 2	Run 1	Run 2
00 (0)	22,094	24,216	16,936	19,818
01 (1)	21,729	22,283	17,953	18,429
10 (2)	21,198	25,450	17,584	19,526
11 (3)	22,898	23,949	19,624	19,749

Less right turning (x, y) in cm: (0, 0) to (34, 90)				
Averaged Speed (cm/s)				
APD TDD				
	Run 1	Run 2	Mean	Variance
00 (0)	19,515	22,017	20,766	3,1300
01 (1)	19,841	20,356	20,099	0,1326
10 (2)	19,391	22,488	20,940	4,7957
11 (3)	21,261	21,849	21,555	0,1729
Average of estimated variances				2,0578
Standart deviation (σ_{AE})				1,4345

The experiments developed show the measured performance statistics, time, trajectory accuracy, and averaged speeds, for straight trajectories, wide and less left turnings, and wide and less right turnings. The standard deviation obtained as well as the main and lateral effects are represented in Table 7.6. The time, trajectory error and averaged speed standard deviations are respectively denoted by σ_T, σ_{TE}, and σ_{AS}. Table 7.6 represents the experimental statistic results obtained for the set of proposed trajectories. The standard deviations computed for each kind of trajectory by testing the different factor weights under different runs are also depicted. The main and lateral effects were calculated by using (7-17), (7-18), and (7-19), and the mean values obtained for the different factor combinations.

Table 7.6: Main and lateral effects

Straight trajectory (cm): (0, 0) to (0, 130)				
Parameters		TDD	APD	TDD & APD
Time (s)	σ_T =0.11	-0,30	-0,35	-0,65
Trajectory error (cm)	σ_{TE} =0.37	-0,56	-0,01	-0,57
Speed (cm/s)	σ_{AS} =1.87	1,72	2,61	4,33
Wide left turning (cm): (0, 0) to (-25, 40)				
Parameters		TDD	APD	TDD & APD
Time (s)	σ_T =0.06	-0,07	-0,22	-0,30
Trajectory error (cm)	σ_{TE} =0.18	0,36	0,32	0,68
Speed (cm/s)	σ_{AS} =0.59	0,07	0,71	0,78
Less left turning (cm): (0, 0) to (-34, 90)				
Parameters		TDD	APD	TDD & APD
Time (s)	σ_T =0.25	0,45	0,10	0,55
Trajectory error (cm)	σ_{TE} =0.27	-0,26	-0,26	-0,52
Speed (cm/s)	σ_{AS} =1.65	-2,30	-0,22	-2,52
Wide right turning (cm): (0, 0) to (25, 40)				
Parameters		TDD	APD	TDD & APD
Time (s)	σ_T =0.28	0,22	0,32	0,55
Trajectory error (cm)	σ_{TE} =0.19	0,20	-0,16	0,04
Speed (cm/s)	σ_{AS} =1.35	-0,75	-1,10	-1,85
Less right turning (cm): (0, 0) to (34, 90)				
Parameters		TDD	APD	TDD & APD
Time (s)	σ_T =0.10	0,02	-0,32	-0,30
Trajectory error (cm)	σ_{TE} =0.25	-0,30	0,06	-0,24
Speed (cm/s)	σ_{AS} =1.43	-0,03	0,81	0,79

The different trajectory-tracking performances are analysed in Table 7.6, where the significant main and lateral effects are highlighted. The factorial analysis for straight line trajectories, (σ_T = 0.11s, σ_{TE} = 0.37cm, σ_{AS} = 1.87cm/s), depicts a main APD and TDD effects of -0.35s and -0.3s, and an important lateral effect of -0.65s. The speed lateral effect of 4.3cm/s is also meaningful. Consequently, high values in both factors should improve time and speed performances. The analysis for wide left turning trajectories, (σ_T = 0.06s, σ_{TE} = 0.18cm, σ_{AS} = 0.59cm/s), show main APD effect of "-0.22s", and important lateral effect of -0.3s. However, the lateral effect tends to decrease the trajectory accuracy. A low value for TDD factor and high value for the

APD factor is proposed. The 0.71cm/s speed TDD factor is irrelevant. The factor analysis for less left turning trajectories, (σ_T = 0.25s, σ_{TE} = 0.29cm, σ_{AS} = 1.65cm/s), depicts a possible lateral effect of 0.55s and 0.52cm. However, speed -2.5cm/s lateral effect is not important. The TDD time main effect is also possible (0.45s). In this sense low values are proposed for APD and TDD factors. The analysis for wide right turning trajectories, (σ_T = 0.29s, σ_{TE} = 0.19cm, σ_{AS} = 1.35cm/s), does not provide relevant clues, but small time improvement seems to appear when TDD and APD factors are set to low values. Finally, the factorial analysis for less right turning trajectories, (σ_T = 0.11s, σ_{TE} = 0.25cm, σ_{AS} = 1.43cm/s), depicts APD and lateral effects that decrease the trajectory time with -0.32s and -0.3s. In this sense, high and low values are proposed for APD and TDD factors. Main or lateral effects related to the speed have not been detected.

7.5.5 Experimental time improvements by using APD and TDD trajectory-flexible factors

Once factorial analysis is carried out, this subsection presents the time improvements that can be achieved by using the results obtained. The experiments developed consist in analysing the time performance when a fixed factor cost function or a flexible factor cost function is used. The trajectories to be analysed are formed by straight lines, less right or left turnings, and wide right or left turnings. The fixed factor cost function maintains the high values for APD and TDD factors, while the flexible factor cost function is tested as function of the trajectory to be tracked. Table 7.7 shows the different factors selected as a flexible function of the trajectory to be tracked.

Table 7.7: Factor selection as function of the trajectories

	Trajectories				
Factors	**Straight trajectory**	**Wide left turning**	**Less left turning**	**Wide right turning**	**Less right turning**
APD	1	1	0	0	1
TDD	1	0	0	0	0

The first experiment consists in tracking a trajectory that is composed of four points ((0, 0), (-25, 40), (-25, 120), (0, 160)) given as (x, y) coordinates in cm. The results obtained by using fixed and flexible factor cost function are depicted in table 7.8. Two runs are obtained for each strategy and consequently time performance analysis can be done.

Table 7.8: Results obtained by using fixed or flexible cost function

	Trajectory points: (0,0), (-25,40), (-25,120), (0,160) ((x,y) in cm)					
	Time (s)		Trajectory error (cm)		Averaged Speed (cm/s)	
Experiment	Fixed Law	Flexible Law	Fixed Law	Flexible Law	Fixed Law	Flexible Law
Run 1	10,5	10,3	3,243	3,653	18,209	16,140
Run 2	10,9	9,8	3,194	2,838	16,770	16,632
Mean	10,70	10,05	3,219	3,245	17,489	16,386
Variance	0,0800	0,1250	0,0012	0,3322	1,0354	0,1210
Standart deviation	0,2828	0,3536	0,0346	0,5764	1,0175	0,3479

Figure 7.31 shows the trajectory tracking coordinates, angular position, and speed for the fixed and flexible cost function strategies.

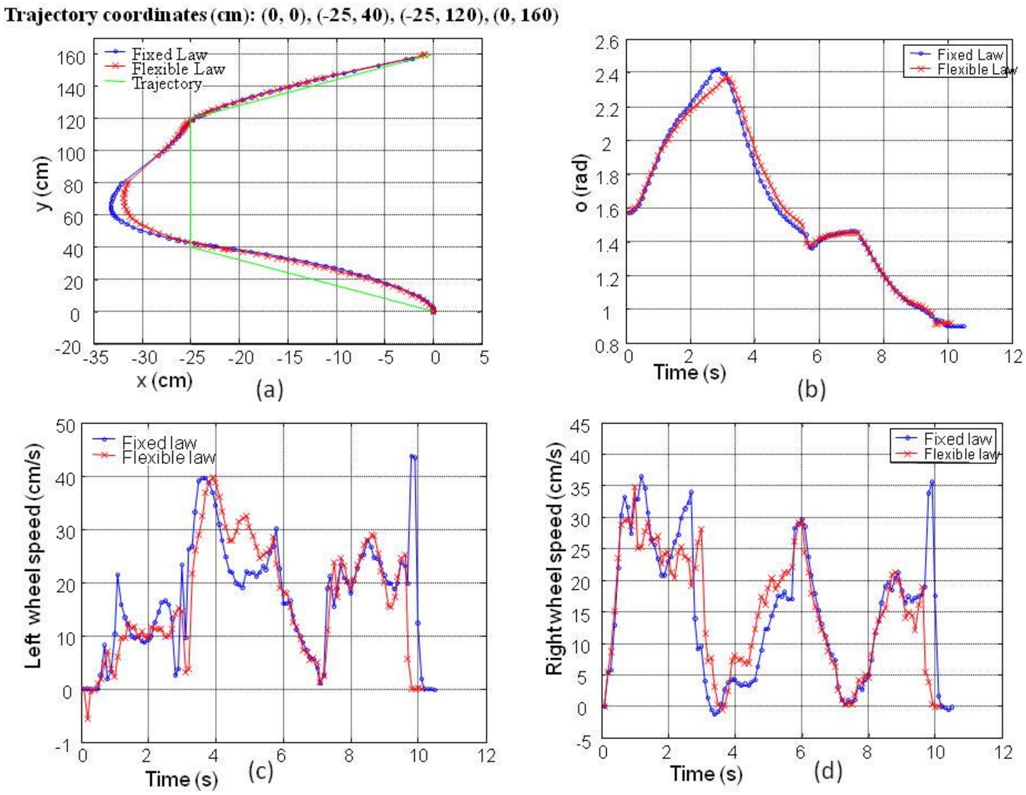


Figure 7.31: (a) Trajectory-tracking experimental results by using flexible or fixed cost function. (b) WMR orientation experimental results by using flexible or fixed cost function. (c) Left wheel speed results by using flexible or fixed cost function. (d) Right wheel speed results by using flexible or fixed cost function.

The averaged standard deviation between the two cost function systems is of 0.32s, and the difference of means are 0.65s. Thus, flexible factor strategy improves about 6% the time performance of the fixed factor strategy. It is remarked that the time improvements seem to be achieved with less control effort and similar trajectory accuracy.

The second experiment consists in tracking a trajectory that is composed of three points ((0, 0), (0, 80), (34, 174)) given as (x, y) coordinates in cm. The results obtained by using fixed and flexible factor cost functions are depicted in table 7.9.

Table 7.9: Results obtained by using fixed or flexible cost function

	Trajectory points: (0,0), (0,80), (34,174) ((x,y) in cm)					
	Time (s)		Trajectory error (cm)		Averaged speed (cm/s)	
Experiment	Fixed law	Flexible law	Fixed law	Flexible law	Fixed law	Flexible law
Run 1	8,2	7,8	0,547	0,371	22,360	20,819
Run 2	8,1	7,8	0,677	0,523	24,603	21,170
Mean	8,15	7,80	0,612	0,447	23,481	20,994
Variance	0,0050	0,0000	0,0084	0,0116	2,5166	0,0616
Standart deviation	0,0707	0,0000	0,0918	0,1075	1,5864	0,2482

Fig. 7.32 shows the trajectory tracking coordinates, angular position, and speed for the fixed and flexible factor cost function strategies.

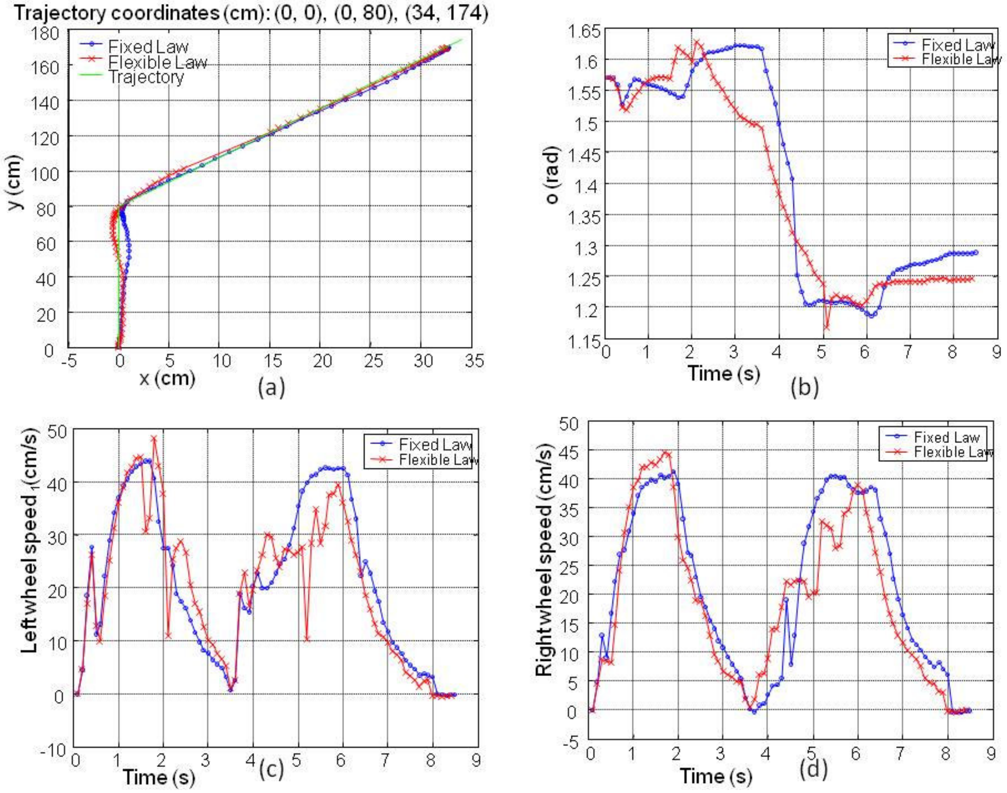


Figure 7.32: (a) Trajectory-tracking experimental results by using flexible or fixed cost function. (b) WMR orientation experimental results by using flexible or fixed cost function. (c) Left wheel speed results by using flexible or fixed cost function. (d) Right wheel speed resultsby using flexible or fixed cost function.

As it is shown in Table 7.9, two runs are obtained for each strategy and consequently trajectory-tracking performance analysis can be done. The averaged time standard deviation between the two cost function systems is of 0.04s, and the difference of means is 0.35s. Thus, in this case flexible factor strategy also improves the time performance of the fixed factor strategy by 5%. Moreover, less trajectory-tracking error, with less averaged speed are produced by using the flexible law cost function.

7.5.6 Experimental tuning of APD and OD factors

This subsection presents the results achieved by using factorial design in order to study the LMPC cost function tuning when APD and OD factors are used. Hence, time and trajectory accuracy performances are analysed by means of the different factor weights. The experiments are developed by considering five different kinds of trajectories within the reduced field of view as shown in Figure 7.28. Therefore, straight, wide left turning, less left turning, wide right turning, and less right turning trajectories are tested. Experiments are conducted by using factorial design with two levels of quantitative factors [Box et al., 05]. For each combination of factors two different runs are experimented. Main and lateral effects are analised by using equations (7-18) and (7-19). The Tables 7.10A, 7.10B, 7.10C, 7.10D and 7.10E, respectively show the measured performance statistics, time, trajectory accuracy, and averaged speeds, for straight trajectories, large and small left turnings, and large and small right turnings.

Table 7.10A: Performance for straight line trajectory

Straight trajectory (x, y) in cm: (0, 0) to (0, 130)				
Time (s)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	4,6	4,7	4,65	0,0050
01 (1)	5,0	4,9	4,95	0,0050
10 (2)	4,8	4,8	4,80	0,0000
11 (3)	4,5	4,6	4,55	0,0050
Average of estimated variances				0,0038
Standart deviation (σ_T)				0,0612

Straight trajectory (x, y) in cm: (0, 0) to (0, 130)				
Trajectory error (cm)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	0,327	0,586	0,457	0,0336
01 (1)	0,227	0,363	0,295	0,0093
10 (2)	0,936	2,816	1,876	1,7680
11 (3)	1,345	1,769	1,557	0,0902
Average of estimated variances				0,4753
Standart deviation (σ_{TE})				0,6894

Straight trajectory (x, y) in cm: (0, 0) to (0, 130)				
Averaged left and right speeds (cm/s)				
APD TDD	Left		Right	
	Run 1	Run 2	Run 1	Run 2
00 (0)	26,777	26,431	26,351	25,933
01 (1)	25,002	26,527	24,702	26,950
10 (2)	27,300	26,357	25,522	23,322
11 (3)	28,961	29,238	26,747	26,819

Straight trajectory (x, y) in cm: (0, 0) to (0, 130)				
Averaged Speed (cm/s)				
APD TDD				
	Run 1	Run 2	Mean	Variance
00 (0)	26,564	26,182	26,373	0,0730
01 (1)	24,852	26,739	25,795	1,7794
10 (2)	26,411	24,840	25,625	1,2348
11 (3)	27,854	28,029	27,941	0,0152
Average of estimated variances				0,7756
Standart deviation (σ_{AE})				0,8807

Table 7.10B: Performance for wide left turning

wide left turning (x, y) in cm: (0, 0) to (-25, 40)				
Time (s)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	3,0	3,0	3,00	0,0000
01 (1)	2,8	2,9	2,85	0,0050
10 (2)	3,0	3,3	3,15	0,0450
11 (3)	3,2	3,0	3,10	0,0200
Average of estimated variances				0,0175
Standart deviation (σ_T)				0,1323

wide left turning (x, y) in cm: (0, 0) to (-25, 40)				
Trajectory error (cm)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	2,012	2,575	2,294	0,1585
01 (1)	2,480	2,420	2,450	0,0018
10 (2)	3,354	3,020	3,187	0,0558
11 (3)	2,203	2,433	2,318	0,0265
Average of estimated variances				0,0606
Standart deviation (σ_{TE})				0,2463

wide left turning (x, y) in cm: (0, 0) to (-25, 40)				
Averaged left and right speeds (cm/s)				
APD TDD	Left		Right	
	Run 1	Run 2	Run 1	Run 2
00 (0)	7,639	7,123	21,219	23,110
01 (1)	8,066	7,707	24,448	23,457
10 (2)	6,342	6,018	24,471	21,921
11 (3)	7,909	7,526	18,947	22,639

wide left turning (x, y) in cm: (0, 0) to (-25, 40)				
Averaged Speed (cm/s)				
APD TDD				
	Run 1	Run 2	Mean	Variance
00 (0)	14,429	15,117	14,773	0,2363
01 (1)	16,257	15,582	15,920	0,2278
10 (2)	15,407	13,970	14,688	1,0325
11 (3)	13,428	15,083	14,255	1,3687
Average of estimated variances				0,7163
Standart deviation (σ_{AE})				0,8464

Table 7.10C: Performance for less left turning

less left turning (x, y) in cm: (0, 0) to (-34, 90)				
Time (s)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	3,8	4,0	3,90	0,0200
01 (1)	3,6	3,7	3,65	0,0050
10 (2)	3,8	3,9	3,85	0,0050
11 (3)	3,8	3,9	3,85	0,0050
Average of estimated variances				0,0088
Standart deviation (σ_T)				0,0935

less left turning (x, y) in cm: (0, 0) to (-34, 90)				
Trajectory error (cm)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	1,152	1,186	1,169	0,0006
01 (1)	1,389	1,324	1,356	0,0022
10 (2)	2,902	3,117	3,009	0,0232
11 (3)	1,562	1,778	1,670	0,0234
Average of estimated variances				0,0123
Standart deviation (σ_{TE})				0,1111

less left turning (x, y) in cm: (0, 0) to (-34, 90)				
Averaged left and right speeds (cm/s)				
APD TDD	Left		Right	
	Run 1	Run 2	Run 1	Run 2
00 (0)	22,215	21,537	26,118	25,300
01 (1)	24,052	22,311	30,352	27,889
10 (2)	20,467	21,989	29,053	30,192
11 (3)	20,494	20,590	28,269	27,700

less left turning (x, y) in cm: (0, 0) to (-34, 90)				
Averaged Speed (cm/s)				
APD TDD				
	Run 1	Run 2	Mean	Variance
00 (0)	24,167	23,419	23,793	0,2798
01 (1)	27,202	25,100	26,151	2,2092
10 (2)	24,760	26,091	25,425	0,8851
11 (3)	24,382	24,145	24,263	0,0280
Average of estimated variances				0,8505
Standart deviation (σ_{AE})				0,9222

Table 7.10D: Performance for wide right turning

wide right turning (x, y) in cm: (0, 0) to (25, 40)				
Time (s)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	2,8	2,6	2,70	0,0200
01 (1)	2,9	3,1	3,00	0,0200
10 (2)	3,3	3,2	3,25	0,0050
11 (3)	3,1	3,2	3,15	0,0050
Average of estimated variances				0,0125
Standart deviation (σ_T)				0,1118

wide right turning (x, y) in cm: (0, 0) to (25, 40)				
Trajectory error (cm)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	2,562	2,705	2,633	0,0102
01 (1)	2,531	2,363	2,447	0,0142
10 (2)	3,344	3,325	3,335	0,0002
11 (3)	2,627	2,663	2,645	0,0006
Average of estimated variances				0,0063
Standart deviation (σ_{TE})				0,0794

wide right turning (x, y) in cm: (0, 0) to (25, 40)				
Averaged left and right speeds (cm/s)				
APD TDD	Left		Right	
	Run 1	Run 2	Run 1	Run 2
00 (0)	24,069	25,292	9,562	10,181
01 (1)	23,313	21,541	10,097	9,053
10 (2)	21,544	22,700	7,462	7,958
11 (3)	20,803	21,476	8,997	8,836

wide right turning (x, y) in cm: (0, 0) to (25, 40)				
Averaged Speed (cm/s)				
APD TDD				
	Run 1	Run 2	Mean	Variance
00 (0)	16,816	17,737	17,276	0,4241
01 (1)	16,705	15,297	16,001	0,9912
10 (2)	14,503	15,329	14,916	0,3411
11 (3)	14,900	15,156	15,028	0,0328
Average of estimated variances				0,4473
Standart deviation (σ_{AE})				0,6688

Table 7.10E: Performance for less right turning

Less right turning (x, y) in cm: (0, 0) to (34, 90)				
Time (s)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	4,4	3,8	4,10	0,1800
01 (1)	4,3	4,2	4,25	0,0050
10 (2)	4,4	4,4	4,40	0,0000
11 (3)	3,9	4,3	4,10	0,0800
Average of estimated variances				0,0663
Standart deviation (σ_T)				0,2574

Less right turning (x, y) in cm: (0, 0) to (34, 90)				
Trajectory error (cm)				
APD TDD	Run 1	Run 2	Mean	Variance
00 (0)	3,396	3,013	3,205	0,0731
01 (1)	1,893	2,261	2,077	0,0676
10 (2)	4,093	4,121	4,107	0,0004
11 (3)	2,387	2,559	2,473	0,0149
Average of estimated variances				0,0390
Standart deviation (σ_{TE})				0,1975

Less right turning (x, y) in cm: (0, 0) to (34, 90)				
Averaged left and right speeds (cm/s)				
APD TDD	Left		Right	
	Run 1	Run 2	Run 1	Run 2
00 (0)	25,889	29,649	17,984	20,883
01 (1)	25,166	24,615	19,120	18,852
10 (2)	26,964	26,378	17,571	17,598
11 (3)	27,220	25,686	20,540	19,048

Less right turning (x, y) in cm: (0, 0) to (34, 90)				
Averaged Speed (cm/s)				
APD TDD				
	Run 1	Run 2	Mean	Variance
00 (0)	21,937	25,266	23,601	5,5428
01 (1)	22,143	21,734	21,938	0,0838
10 (2)	22,268	21,988	22,128	0,0391
11 (3)	23,880	22,367	23,124	1,1446
Average of estimated variances				1,7026
Standart deviation (σ_{AE})				1,3048

The experiments developed show the measured performance statistics, time, trajectory accuracy, and averaged speeds, for straight trajectories, wide and less left turnings, and wide and less right turnings. The standard deviation obtained as well as the main and lateral effects are represented in Table 7.11. The time, trajectory error and averaged speed standard deviations are respectively denoted by σ_T, σ_{TE}, and σ_{AS}. Table 7.11 represents the experimental statistic results obtained for the set of proposed trajectories. The standard deviations computed for each kind of trajectory by testing the different factor weights under different runs are also depicted. The main and lateral effects were calculated by using (7-17), (7-18), and (7-19), and the mean values obtained for the different factor combinations in Table 7.11, the significant results achieved using experimental factorial analysis are highlighted.

Table 7.11: Main and lateral effects

Straight trajectory				
Parameters		OD	APD	APD & OD
Time (s)	$\sigma_T = 0.06$s	0,02	-0,13	-0,10
Trajectory error (cm)	$\sigma_{TE} = 0.69$cm	-0,24	1,34	1,10
Speed (cm/s)	$\sigma_{AS} = 0.88$cm/s	0,87	0,70	1,57
Wide left turning				
Parameters		OD	APD	APD & OD
Time (s)	$\sigma_T = 0.06$s	-0,10	0,20	0,10
Trajectory error (cm)	$\sigma_{TE} = 0.18$cm	0,36	0,38	0,02
Speed (cm/s)	$\sigma_{AS} = 0.59$cm/s	0,36	-0,87	-0,52
Less left turning				
Parameters		OD	APD	APD & OD
Time (s)	$\sigma_T = 0.09$s	-0,12	0,07	-0,05
Trajectory error (cm)	$\sigma_{TE} = 0.11$cm	0,58	1,08	0,50
Speed (cm/s)	$\sigma_{AS} = 0.92$cm/s	0,60	-0,13	0,47
Wide right turning				
Parameters		OD	APD	APD & OD
Time (s)	$\sigma_T = 0.11$s	0,10	0,35	0,45
Trajectory error (cm)	$\sigma_{TE} = 0.08$cm	0,44	0,45	0,01
Speed (cm/s)	$\sigma_{AS} = 0.67$cm/s	-0,58	-1,67	-2,25
Less right turning				
Parameters		OD	APD	APD & OD
Time (s)	$\sigma_T = 0.26$s	-0,07	0,07	0,00
Trajectory error (cm)	$\sigma_{TE} = 0.20$cm	1,38	0,65	-0,73
Speed (cm/s)	$\sigma_{AS} = 0.13$cm/s	-0,33	-0,14	-0,48

The time performance is analysed for the different trajectories. The factorial analysis for straight line trajectories, ($\sigma_T = 0.06$s, $\sigma_{TE} = 0.69$cm, $\sigma_{AS} = 0.88$cm/s), depicts a main APD effects of -0.125s and 1.345cm. However, the lateral effect also decreases the time with less error. The speed lateral effect of 1.57cm/s can also be considered as meaningful. Consequently, high values in both factors should improve time performance, but with a larger speed. The analysis for wide left turning trajectories, ($\sigma_T = 0.06$s, $\sigma_{TE} = 0.18$cm, $\sigma_{AS} = 0.59$cm/s), show main OD effect of -0.1s. A low value for APD factor and high value for the OD factor are proposed. The trajectory error and averaged speed effects are not meaningful. The factor analysis for less left turning trajectories, (σ_T = 0.09s, σ_{TE} = 0.11cm, σ_{AS} = 0.92cm/s), depicts a main OD effect of "-0.125s" and 0.57cm. However, OD speed effect of 0.59cm/s is not significant. APD main effect decreases trajectory-accuracy. In this sense low and high values are proposed for APD and OD factors. The analysis for wide right turning trajectories, (σ_T = 0.11s, σ_{TE} = 0.08cm, σ_{AS} = 0.67cm/s) does not provide relevant clues, but time improvement seems to appear when OD and APD factors are set to a low values. Finally, the factorial analysis for less right turning trajectories, (σ_T = 0.26s, σ_{TE} = 0.20cm, σ_{AS} = 0.13cm/s), depicts lateral effects without increasing the trajectory time. The APD main effect produces a bigger trajectory error of 1.38cm/s. In this sense, high values are proposed for APD and OD factors. Main or lateral effects related to the speed have not been detected.

7.5.7 Experimental time improvements by using APD and OD trajectory-flexible factors

Once factorial analysis is carried out, this subsection presents the time improvements that can be achieved by using the results obtained. The experiments developed consist in analysing the time performance when a fixed factor cost function or a flexible factor cost function is used. The trajectories to be analysed are formed by straight lines, less right or left turnings, and wide right or left turnings. The fixed factor cost function maintains the high values for APD and OD factors, while the flexible factor cost function is tested as function of the trajectory to be tracked. Table 7.12 shows the different factors selected as a function of the trajectory to be tracked.

Table 7.12: Results obtained by using fixed or flexible cost function

	Trajectory				
Factors	**Straight trajectory**	**Wide left turning**	**Less left turning**	**Wide right turning**	**Less right turning**
APD	1	0	0	0	1
OD	1	1	1	0	1

The first experiment consists in tracking a trajectory that is composed of three points ((0, 0), (-25, 40), (-25, 120) given as (x, y) coordinates in cm. The results obtained by using fixed and flexible factor cost function are depicted in table 7.13.

Table 7.13: Results obtained by using fixed or flexible cost function

	Trajectory points: (0,0), (-25, 40), (-25, 120) ((x,y) in cm)					
	Temps (s)		**Mitjana Error (cm)**		**Mitjana Ar (cm/s)**	
Experiment	Fixed law	Flexible law	Fixed law	Flexible law	Fixed law	Flexible law
Run 1	7,2	7,0	3,820	3,007	19,415	17,482
Run 2	7,4	6,6	2,220	3,468	16,509	20,107
Mean	7,30	6,80	3,020	3,237	17,962	18,794
Variance	0,0200	0,0800	1,2794	0,1060	4,2210	3,4453
Standart deviation	0,1414	0,2828	1,1311	0,3256	2,0545	1,8562

Figure 7.33 shows the trajectory tracking coordinates, angular position, and speed for the fixed and flexible cost function strategies.

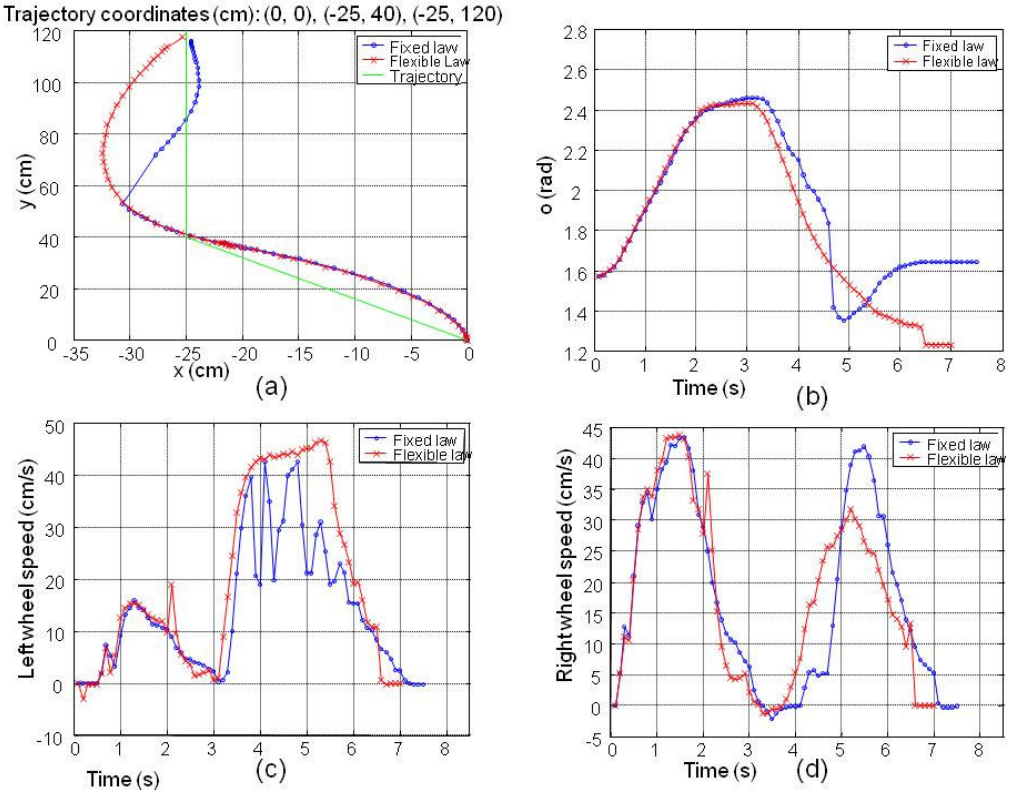


Figure 7.33: (a) Trajectory-tracking experimental results by using flexible or fixed cost function. (b) WMR orientation experimental results by using flexible or fixed cost function. (c) Left wheel speed results by using flexible or fixed cost function. (d) Right wheel speed results by using flexible or fixed cost function.

As it is shown in Table 7.13, two runs are obtained for each strategy and consequently statistical trajectory-tracking performance analysis can be done. The averaged time standard deviation between the two cost function systems is of 0.08s, and the difference of means is 0.5s. Thus, in this case flexible factor strategy improves the time performance of the fixed factor strategy by 7%. No meaningful conclusions can be made about trajectory-error and averaged speed wheels for both control laws. However similar trajectory errors are obtained. Moreover, averaged speeds of flexible control law seem to be less as compared with the results obtained with fixed control law.

The second experiment consist in tracking a trajectory that is also composed of three points ((0, 0), (0, 40), (-34, 130) given as (x, y) coordinates in cm. The results obtained by using fixed and flexible factor cost function are depicted in table 7.14. Two runs are obtained for each strategy and consequently time performance analysis can be done.

Table 7.14: Results obtained by using fixed or flexible cost function

	Trajectory points: (0,0), (-0, 40), (-34, 130) ((x,y) in cm)					
	Temps (s)		**Mitjana Error (cm)**		**Mitjana Ar (cm/s)**	
Experiment	Fixed law	Flexible law	Fixed law	Flexible law	Fixed law	Flexible law
Run 1	6,7	6,0	1,295	1,416	24,434	22,654
Run 2	6,3	5,9	0,822	1,299	21,778	22,763
Mean	6,50	5,95	1,058	1,357	23,106	22,709
Variance	0,0800	0,0050	0,1120	0,0069	3,5272	0,0059
Standart deviation	0,2828	0,0707	0,3347	0,0829	1,8781	0,0771

Figure 7.34 shows the trajectory tracking coordinates, angular position, and speed for the fixed and flexible cost function strategies. Flexible cost function depicts softer orientation and speed changes with a little less of trajectory-tracking accuracy.

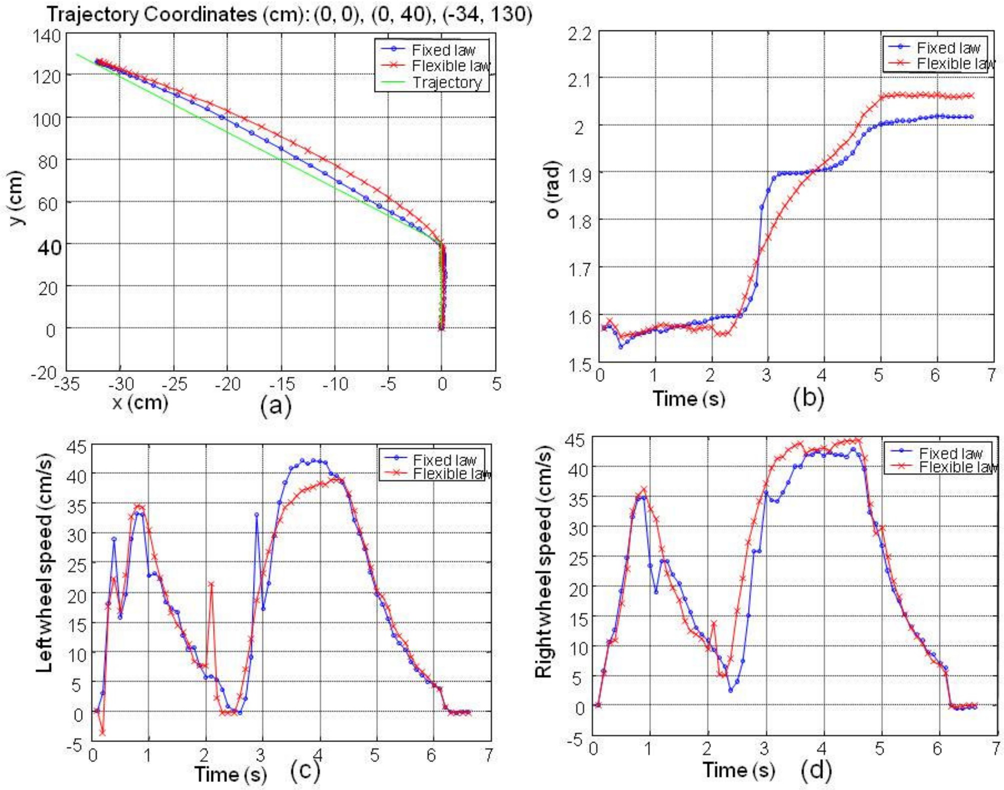


Figure 7.34: (a) Trajectory-tracking experimental results by using flexible or fixed cost function. (b) WMR orientation experimental results by using flexible or fixed cost function. (c) Left wheel speed results by using flexible or fixed cost function. (d) Right wheel speed results by using flexible or fixed cost function.

Analysing the statistic results shown in Table 7.14, it can be observed that the averaged standard deviation between the two cost function systems is of 0.18s, and the difference of means is 0.55s. Thus, in this case flexible factor strategy also improves the time performance of the fixed factor strategy by 8.5%. It seems that better accuracy with more speed is accomplished for fixed law. However, due to the small differences of means for averaged speeds and trajectory errors no conclusion can be drawn concerning about these features.

Finally, the third experiment consists in tracking a trajectory that is composed of three points ((0, 0), (25, 40), (59, 130) given as (x, y) coordinates in cm. The results obtained by using fixed and flexible factor cost function are depicted in table 7.15. Two runs are obtained for each strategy and consequently performance analysis can be done.

Table 7.15: Results obtained by using fixed or flexible cost function

	Trajectory points: (0,0), (25, 40), (59, 130) ((x,y) in cm)					
	Temps (s)		**Mitjana Error (cm)**		**Mitjana Ar (cm/s)**	
Experiment	Fixed law	Flexible law	Fixed law	Flexible law	Fixed law	Flexible law
Run 1	6,9	6,7	2,299	2,818	21,395	21,961
Run 2	7,0	6,6	2,302	2,889	19,933	21,633
Mean	6,95	6,65	2,301	2,853	20,664	21,797
Variance	0,0050	0,0050	0,0000	0,0025	1,0695	0,0538
Standart deviation	0,0707	0,0707	0,0025	0,0503	1,0341	0,2319

Figure 7.35 shows the trajectory tracking coordinates, angular position, and speed for the fixed and flexible cost function strategies.

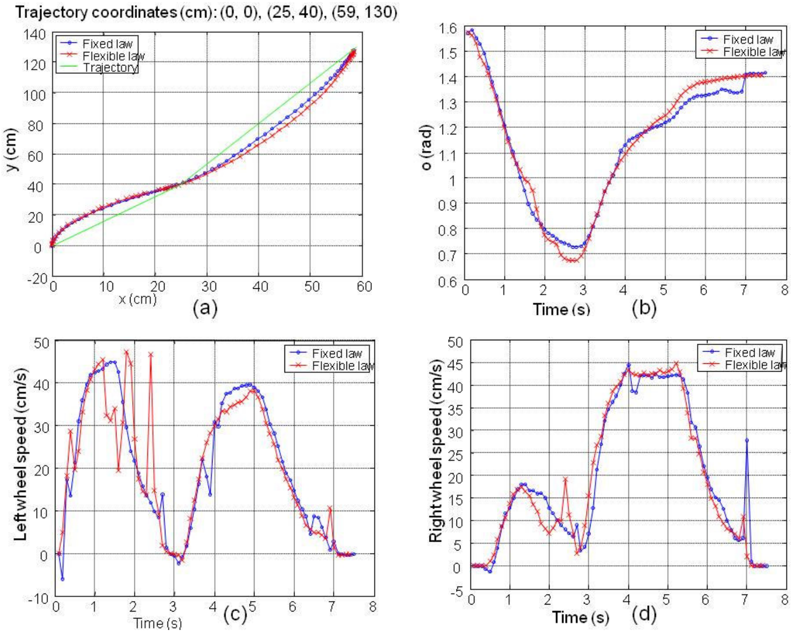


Figure 7.35: (a) Trajectory-tracking experimental results by using flexible or fixed cost function. (b) WMR orientation experimental results by using flexible or fixed cost function. (c) Left wheel speed results by using flexible or fixed cost function. (d) Right wheel speed results by using flexible or fixed cost function.

From the statistic results shown in Table 7.15, it can be observed that the averaged time standard deviation between the two cost function systems is 0.07s, and the difference of means is 0.2s. Thus, in this case flexible factor strategy also improves the time performance of the fixed factor strategy. Trajectory-tracking error analysis depict as a meaningful clue that flexible law error is larger than the fixed one. Moreover, speed of flexible law seems also to be larger.

7.6 Conclusions

The LMPC methodology presented in this Chapter is appropriate for short prediction horizons where local environment information is provided by the on-robot sensor system introduced in Chapter 6. The control strategy proposed consists of short-term actions as going straight or turning. Therefore, the perception system proposal just depicts local environment where just few seconds of path-planning can be done.

The simulation results are used for orienting the experimental research reported. The optimal search method has been selected due to the fact that gradient descent method can be considered as suboptimal. Moreover, short prediction horizon of 0.5s has also been selected. The short horizon can appropriately deal with reactive behaviours that are expected when local navigation is planned. The simulation results show that different cost function factors can be effective for performing the trajectory-tracking of different trajectories that can be accomplished within the reduced field of perception proposed in this research.

The experimental on-robot results depict accurate trajectory-tracking for the different tested trajectories. Moreover, statistical analysis has been performed by using factorial design with two levels of quantitative factors as a way to infer time, trajectory accuracy, and control effort as a function of the cost function factor weights. Therefore, from statistics results obtained through the testing of different trajectories, which can be considered as a representative one of the possible set of trajectories within the perceived local scenario, different weights performances as function of the trajectory to be tracked and performance are obtained. Thus, the use of flexible cost functions, with different factor weights as function of the trajectory to be tracked, can be considered as a good strategy that give better results.

The experimental results were obtained by using optimal search and short horizon of prediction (0.5s). When longer prediction horizons were tested some problems, concerning the motor dead zone, can appear. Therefore, more studies for solving such problems should be developed as future control research. The factorial analysis has depicted the effectiveness and more experiments can also be performed for extending the present research. Moreover, further analysis comparing the LMPC with other control laws can be done.

The next Chapter presents the WMR navigation experiences developed in this research. In this way, the local navigation strategy will use the perception system presented in Chapter 6, and the control system introduced in this Chapter.

Chapter 8

Monocular Local Navigation Results

8.1 Introduction

Research work studying natural agents has presented a new robot navigation paradigm. Hence, corridor planning for natural agents has been presented as a new and useful robot control and planning framework using low level obstacle avoidance and simple control [Butler, 06]. The developed research showed that local animal abilities such as control and perception can be combined with a very simple imposed path-like structure to produce the desired overall motion. Hence, path-planning in partially unknown environments should be short enough to allow local reactive behaviours; consequently long trajectory planning should be flexible in order to deal with short term uncertainties while the global task is accomplished.

This chapter explores this challenge as a navigation strategy for the available WMR platform PRIM that consists of a differential driven mobile robot with a free rotating wheel [Pacheco et al., 08]. In this work, local animal abilities are performed by using a simple perception system, which consists of a monocular camera and an odometer system. The results are obtained by using the camera configuration depicted in Chapter 6. The field of view is constrained to the WMR neighbourhood; thus only a few seconds of trajectory planning can be done. A local narrow perception horizon is used for planning safe trajectories towards the objective. It is relevant to be aware of the narrow field of view considered in this research. In this sense, a new advancing trajectory will produce a new environment description. In the human case it can be understood as an "*illuminated step by step strategy*" (i.e. by using a small torch when we are exploring an unfamiliar dark place where the goal position is achieved by using previous acquired knowledge about the direction that we should explore towards the objective). Therefore, monocular data are proposed as a way to obtain real time local information by building two dimensional occupancy grids through a time integration of the frames. The simple corridor structure is obtained by using local attraction potential fields where the MPC (model predictive control) techniques, presented in Chapter 7, are used for performing accurate trajectory tracking. Therefore, several examples with different indoor scenarios and the path followed are reported.

8.2 The Local Navigation Framework

The navigation methodology is presented in the subsequent subsections by considering its features. The perception horizon is presented as a local map where obstacles should be detected and feasible trajectories should be planned by considering the system constraints such as visual dead zone, robot WP (wide-path) and dynamics. Artificial potential fields are used for attracting the WMR towards the desired configuration. Therefore, the goal approaching strategy is accomplished by searching a desired local point within the perception horizon.

8.2.1 The Perception Horizon

The local visual data provided by the camera are used for planning a feasible trajectory and avoiding the obstacle collision. The available scene coordinates appear as an image, in which each pixel coordinate corresponds to a set of 3D scene coordinates. Hence, scene coordinates can be computed by using camera setup and pose knowledge and assuming projective perspective. In this work, it is supposed that available obstacle positions are provided by using computer vision systems. The monocular perception methodology used in this research can be found in Section 6.4. The use of the local narrow visual information available from the camera is proposed as a local map that has enough information for achieving an objective configuration. The LPMOG can be obtained in real time by using computer vision methods and odometer data; the corresponding preliminary research developed is depicted in Section 6.5. The odometer system is used for knowing the WMR positions and controlling the trajectory tracking. The occupancy probability is divided into only two ranges: free and occupied. Hence, local trajectories along the local grid can approach the robot to the final goal while obstacle collisions are considered. The analysis is focused on some particular indoor environment with flat floor surface; however it can also be applied in outdoor environments.

It is assumed that a feasible LPMOG based on the visual information is provided by computer vision methods previously introduced in Chapter 6. The trajectory tracking is done by using the LMPC strategy introduced in the Chapter 7. Some considerations concerning about the narrow field of view, robot dimension and dynamics are given.

The LPMOG provided by the camera is used for planning a feasible trajectory. In the above case, flat floor is assumed; hence free cells belong to a 2D space. Figure 8.1 shows the camera configuration studied in this work. α, β and φ are angles of the vertical and horizontal field of view and the tilt camera pose, respectively. The vertical coordinate of the camera is represented by H. By using trigonometric relationships, the scene coordinates can be computed [Horn, 98]. Figure 8.2 shows a local map provided by the camera configuration shown in Figure 8.1. The scene grids with low resolution can be used for speeding up the computing process [Gonzalez and Woods, 02]. The results of coordinate maps can be improved by using the calibration techniques that allow removing for instance the radial distortion [Elsayed, 03].

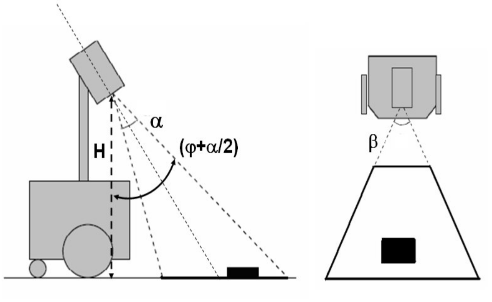


Figure 8.1: Fixed camera configuration and pose (angles α=37°, β=48° and φ=32°,), H=109cm.

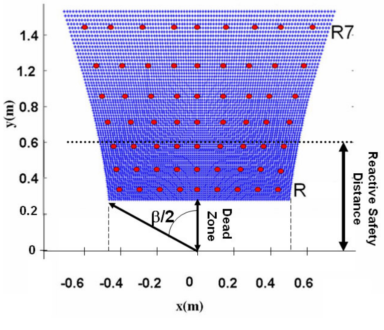


Figure 8.2: Example of local visual perception using 96x72 or 9x7 grids.

The narrow field of view and the fixed camera configuration make necessary that the robot orients towards the desired coordinates and moves in advancing sense. A reactive zone should be considered accordingly to the robot dynamics, in order to plan safe trajectories with obstacle avoidance and safe stop distances. In the perception system analysed, it can be summarised that the robot positions are constrained by the following aspects:

- The visual dead zone
- The robot WP
- The WMR reactive distance

The visual dead zone is inherent to the camera configuration set-up. In the frame 1, a dead zone free of obstacles should be assumed. Furthermore, the trajectories commanded should consider the visual dead zone effects. Hence, the previous acquired images should cover the visual dead zone of the next frame to be processed.

The WP of the robot should also be considered [Schilling, 90]; thus, Figure 8.3 shows that not all the possible orientations allow a trajectory that includes the path-width of the robot.

The WMR reactive distance is related to several aspects:

- The processing time for each frame
- The WMR dynamics
- The obstacle kinematics and dynamics

During the perception computing time, for each new acquired frame, the robot follows the last commanded trajectory. In this sense, the speed of the perception system provides the time of reaction.

The WMR dynamics will set the necessary stop distances or the necessary free space to plan safe turning actions for avoiding collisions.

The previously introduced constraints reduce the available horizon of perception when they are taken into account.

The knowledge of the obstacle kinematics and dynamics are important issues for safe path-planning. Their knowledge can allow increasing or decreasing the WMR safe stop distances and the necessary WP (wide-path). Thus, if static obstacles are supposed, the robot trajectory should be large enough to allow a safe stop distance, which can be considered as a function of the commanded speeds. When mobile obstacles are considered the allowed local map coordinates are reduced, thus mobile obstacle movements should be considered.

The previously commented constrains reduce the horizon of possible coordinates that can be achieved. Figures 8.3 and 8.4 show what is happening. Figure 8.3 depicts an example of unsafe trajectory; the trajectory drawn is planned outside of the field of view. Figure 8.4 shows a feasible trajectory when the visual data allow the WMR reactive distance and the necessary free of obstacle areas; as i.e., the dead zone of the next frame and the robot WP should be considered.

It can be summarised that the maximal distance and turning angle that can be achieved within the field of perception are reduced due to the previously commented constraints.

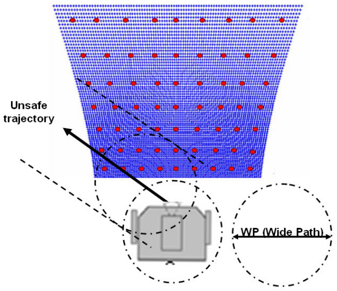


Figure 8.3: Unsafe trajectories, planned within local grid, arise when the heuristic WP concept is not considered.

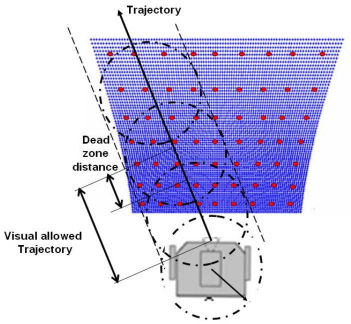


Figure 8.4: The trajectories should include, safe reactive distances, WP, and necessary free of obstacles zone for the next acquired frame.

8.2.2 The local attraction potential fields

The scientific community has carried out several studies in this field. Some approaches use artificial potential fields that attract mobile robot to the desired objective and create repulsive forces for achieving obstacle avoidance [Rimon and Koditschek, 92]. Rimon presented the methodologies for the exact motion planning and control, based on the artificial potential fields where the complete information about the free space and goal are encoded. Some approaches on mobile robots propose the use of potential fields, which satisfy the stability in a Lyapunov sense, in a short prediction horizon [Ögren and Leonard, 05]. In this research it is used the visual information as a dynamic window where the collision avoidance and safe stop can be planned. Thus, the local visual data and the artificial potential fields are used for achieving the Lyapunov stability.

The problem is formulated as searching for the optimal cell that approaches the WMR to the desired coordinates (X_d, Y_d) by finding the closer local desired coordinates (X_{ld}, Y_{ld}). In this sense, the perception is considered as a local receding horizon where trajectory is planned. Hence, a local map with free obstacle coordinates is provided. The local desired cell is obtained by minimising a cost function J, consisting of the distance between the desired coordinates and the available local cell coordinates, as it was shown in Chapter 7. Due to the narrow field of perception there are two possibilities for making the minimization by using the Euclidian and orientation distances, see Figure 8.5. Therefore, expressions (7-10) and (7-11) are used.

As it was depicted in Chapter 7 some constraints should be considered:

- The limitation of the input signal as a function of the local desired points and the WMR dynamics.
- The second constraint is related to the obstacle points.
- The third constraint is a contractive constraint.

Detailed information concerning about these constraints was given in Chapter 7.

Safe collision avoidance should include a heuristic WP, related with the WMR dynamics [Schilling, 90]. The WP is a heuristic concept, which should be large enough to avoid the robot collision, Figure 8.6 shows these concepts. The safe distance can be obtained by:

$$\|X_o - X_{ld}\| > WP/\sin(\pi/2 - \theta_{ld}) \tag{8.1}$$

where the boundary obstacle coordinate is represented by X_o and $\|X_o - X_{ld}\|$ denotes the Euclidean distance between X_o and X_{ld}.

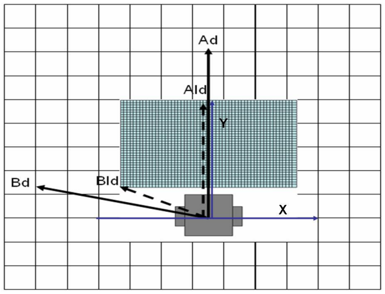


Figure 8.5: Two different desired points *Ad* and *Bd*; *Ald* and *Bld* are computed by using Euclidean and the angular distances.

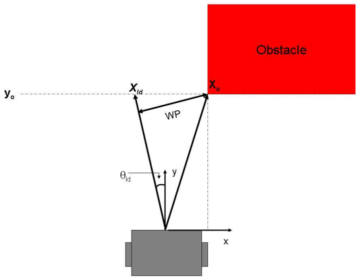


Figure 8.6: WP concept is considered as a necessary distance in order to plan safe trajectories within the local occupancy grid.

8.3 The system of navigation

In this section the navigation system is presented. The perception system is similar to the one introduced in Chapter 6. However, it should be noted that the perception field is increased, by setting the camera pose to 170 cm of high. The benefits effects arise in a better environment description and WMR navigation. In this way, the minimal field of view is increased from nearly two times the robot WP to more than three. Another consideration, arising from the constrained perception used, should be analysed. Due to the reduced field of view the significant information is only in front of the robot. Therefore, short term memory, concerning out of field of view obstacles, is proposed [Schäfer et al., 07]. The short term memory can be used for avoiding repetitive turning actions as well as unsafe trajectories that arise from the narrow field of view. In this sense, the use of the occupancy grid framework, which includes old information of the obstacles, is proposed as a short-term memory that improves WMR navigation. Consequently, the perception memory of the robot is increased through the optical field of view time-integration.

The navigation algorithms are also revised. Thus, due to the available field of view, which is nearly as narrow as three times the robot WP, a simplified navigation algorithm is explained. Therefore, within this reduced field of view, only interaction with two obstacles is possible.

8.3.1 The field of view and the local grid size

The camera height, H, is set to 171 cm. Figure 8.7 shows the robot configuration and the field of view obtained.

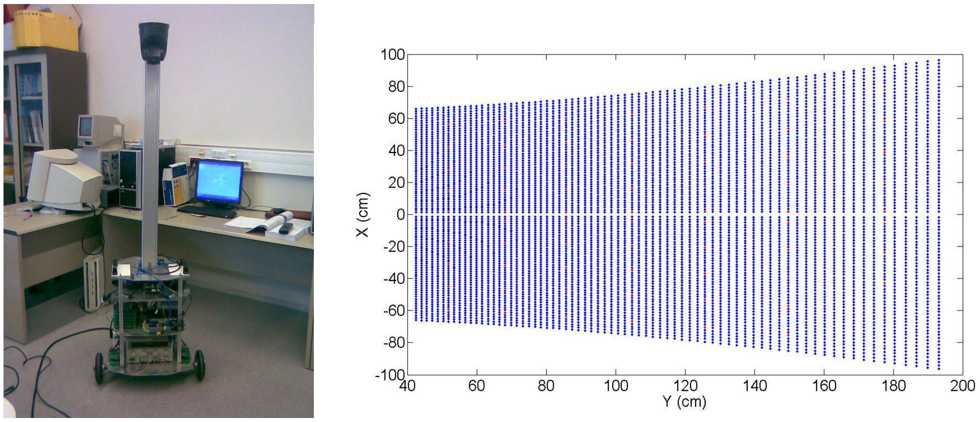


Figure 8.7: WMR camera configuration set-up and floor field of view.

The field of view features:

- Visual dead zone of 42cm.
- Minimal X size dimension of 132cm
- Maximal X size dimension of 193cm
- Y size dimensions from 42cm to 194cm.

The image resolution is 96 x72 pixels. The acquired images are placed in a local occupancy grid of 180 x 120 cells. Figure 8.8 shows the local occupancy grid considered where the acquired image is also depicted.

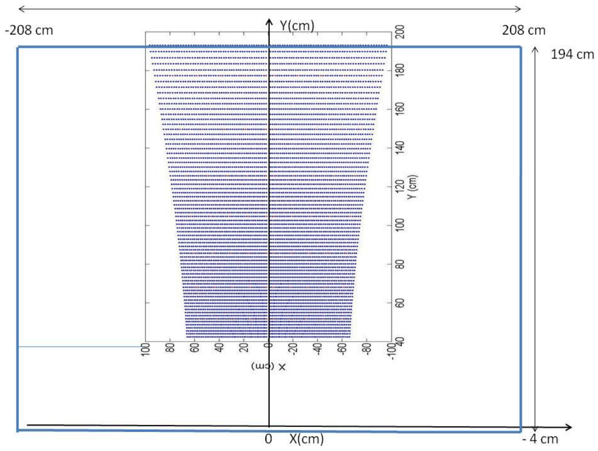


Figure 8.8: Local occupancy grid dimensions and the monocular visual perception obtained.

As it is shown in Fig. 8.8 the local occupancy grid shows a larger scene description. The range of the *X* coordinates is from -208 to 208 cm, and the *Y* goes from -4cm to 194cm. The LPMOG has an area of more than 8m^2. It should allow the closer WMR environment description as well as local navigation towards the desired configuration. The benefits of a local occupancy grid with a bigger size than the camera field of view are:

- A larger horizon of perception
- Better time-integration analysis of the different frames.

- A memory about the out of field of view obstacles

The larger horizon of perception allows planning larger trajectories and consequently a better environment description is also attained. Consequently trajectories are improved by using much more environment knowledge.

The time integration of the different consecutive frames is made within the different LPMOG obtained. It can allow optical flow analysis when the floor model is considered [Kim and Kim, 04]. It is relevant to be aware that floor model discrepancies can be generated by scene obstacles. In this sense a larger occupancy grid can allow a major number obstacles overlapping between the consecutive occupancy grids obtained and a better correlation analysis can be performed, which can allow to infer the obstacle structure or visual odometry.

The field of view of the occupancy grid is of more than 180º; allowing a lateral perception memory of passed obstacles when the different frames are time-integrated. The memory of passed obstacles improves the local scenario knowledge and the search of local goal approaching cells that provide feasible trajectories towards the desired configuration.

Finally an example of an acquired image frame is shown in Figure 8.9. Figure 8.10 depicts the frame integration within the local occupancy grid. The obstacles appear as radiance discontinuities by using the one bit DFF methodology. The robot should avoid the obstacles in advancing direction while approaching trajectories towards the desired coordinates are planned.

Figure 8.9: Compressed image, 96x72, of a typical indoor scenario.

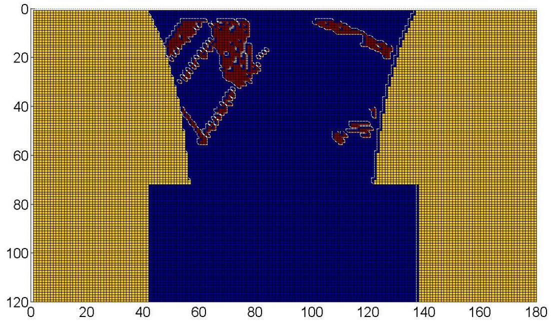


Figure 8.10: Occupancy grid corresponding to the image of the Figure 8.9. The rectangular free zone, depicted in blue, is the necessary free of dead zone of the acquired frame 1.

8.3.2 The navigation algorithms

The navigation strategy is inspired in natural agent behaviours; the developed research showed that local animal abilities such as control and perception can be combined with a very simple imposed path-like structure to produce the desired overall motion [Butler, 06]. Therefore, it is presented the robot control and planning framework by using low level obstacle avoidance and simple control. The field of view perception is constrained to the WMR neighbourhood and is provided by the occupancy grid; thus only a few seconds of trajectory planning can be done. A local narrow perception horizon is used for planning safe trajectories towards the objective. In this sense, a new advancing trajectory will produce a new environment description. In the human case it can be understood as an "*illuminated step by step strategy*" where the closer obstacle vertex is considered. The strategy is implemented by using two steps that include obstacle analysis and navigation. The obstacle analysis is shown in Figure 8.11, and consists of the following statements:

- The procedure analyses the occupancy grid looking for the existence of obstacles in the left or right part. The right vertex is obtained for the closer obstacle in the left part and the left vertex for the closer obstacle in the right part.
- When obstacles are detected in the left part, then their down and right vertex coordinates are computed and considered. It should be pointed out that the analysis is constrained to the left half part of the occupancy grid. Thus, when the right vertex is not obtained it should be considered the left vertex. This last possibility happens when the obstacle is placed in the middle of the occupancy grid.
- When obstacles are detected in the right part, then their down and left vertex coordinates are computed and considered. It should be pointed that the analysis is constrained to the right half part of the occupancy grid. Thus, when the left vertex is not obtained it should be considered the right vertex. This last possibility happens when the obstacle is placed in the middle of the occupancy grid.
- When the closer right obstacle is the same as the left closer obstacle then the obstacle is placed in the middle. In this case, a middle flag is activated.

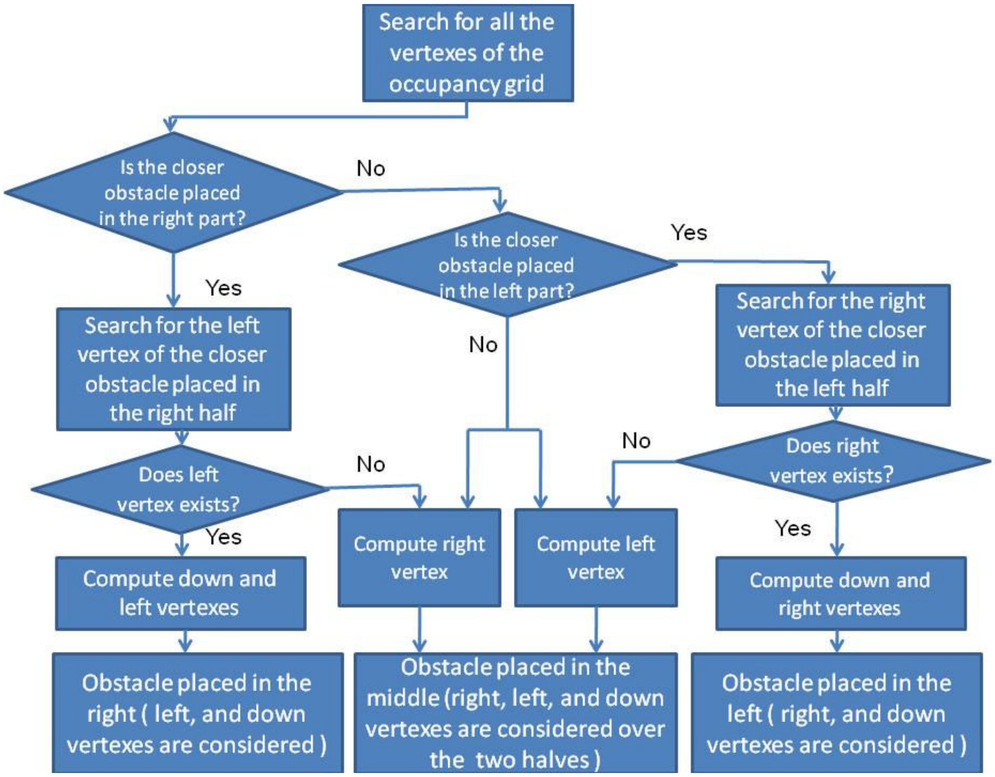


Figure 8.11: The algorithms used for searching the meaningful obstacle vertexes.

Figure 8.12 depicts the vertex considered when just left or right obstacles are met.

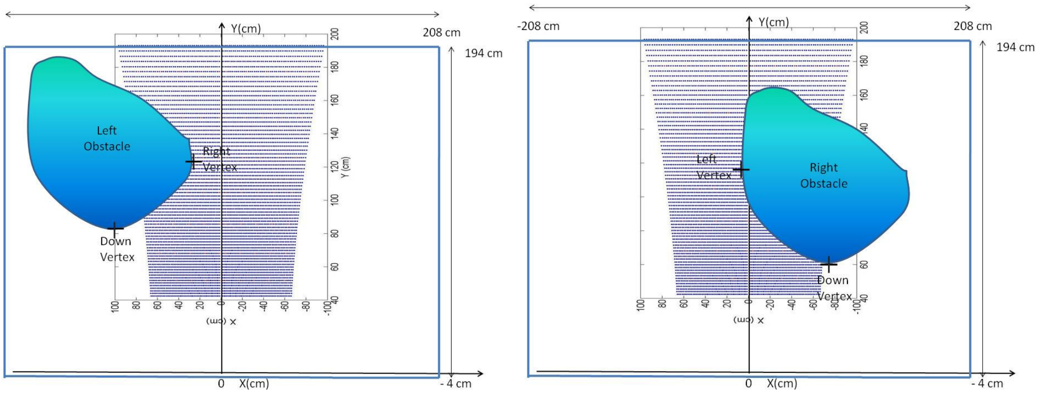


Figure 8.12: Meaningful vertexes considered when right or left obstacles are met.

Once significant closer obstacle vertexes are met path planning can be developed, see Figure 8.13. The navigation algorithms are given by the following statements:

- When obstacles are farther than the desired configuration a flag is activated. Then, the navigation strategy is similar to the free of obstacle case. For the free of obstacle case the strategy selects the best approaching cell of the occupancy grid by computing the potential field within the free cells.
- When the closer obstacle is placed at the middle the WP is applied to the right and left vertexes. When the path is possible through both vertexes; the one that approaches more to the goal is selected. When just one path is possible, the path is followed. If no path is possible the WMR selects between right or left turning as a goal approaching strategy function.
- When the closer obstacle is not placed at the middle but right and left obstacles exist. It is looked for horizontal and vertical possibilities of passing. If the pass is possible the WMR selects the closer right or left vertex as the one to be attained. If the pass is not possible the WMR turns to the left if the closer obstacle is placed into the right. Otherwise the WMR turns to the right.
- When just left obstacles are detected the right and down vertexes are analysed. If the down vertex is too closer to the robot only right vertex will be considered. If the pass is not allowed the robot turns to the right. If the down pass is possible, more than a threshold related with the dynamics of the robot and the capacity of turning, as i.e. 50cm, is considered as a possibility. The WMR selects the best approaching cell where the pass is possible. If WP is not enough the robot selects right or left turning as a goal approaching strategy.
- When just right obstacles are detected the left and down vertexes are analysed. If the down vertex is too closer to the robot only left vertex will be considered. If the pass is not allowed the robot turns to the left. If the down pass is possible it can be considered as another possibility. The WMR selects the best approaching cell where the pass is possible. If WP is not enough the robot selects left or right down turning as a goal approaching strategy.
- The down vertex should be considered as a necessary constraint in order to avoid the collision with the obstacle.
- The WMR remains stopped in other cases not mentioned before.

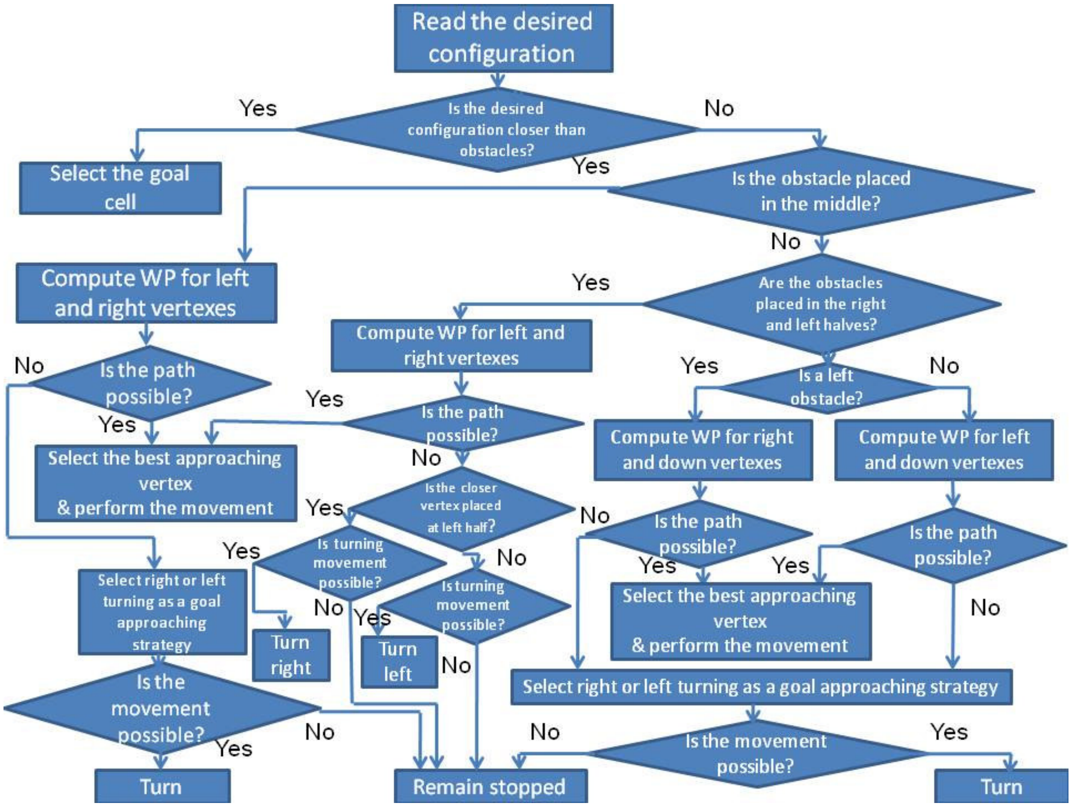


Figure 8.13: The basic navigation algorithms.

Some attaining constraints should be considered, which will be explained before ending this subsection:

- The local desired coordinates, within the occupancy grid, should consider the future dead zone, WMR WP, and reactive distances. A safe distance free of obstacles within the actual occupancy grid should be left.
- A minimal displacement of the robot, as i.e. 10cm, should be attained otherwise the robot should remain stopped.
- As it has been pointed out in the navigation statements the down vertex should be far enough in order to allow the WMR turning. It can be attained by two steps. The *x* coordinate of the desired left or right vertexes is commanded together with the *y* coordinate of the down vertex. Therefore, WMR first turns towards vertex direction and then pass the bigger size dimension of the obstacle.
- In the case of situations, in which it is not permitted enough turning distances or when distance to the goal will be increased the robot should remain stopped.

Figures 8.3 and 8.4 explain the concept of the dead zone, WP and reactive distances. Figure 8.14 explains the necessary constraint of down vertex, for new approaching obstacles, when right or left vertexes should be attained.

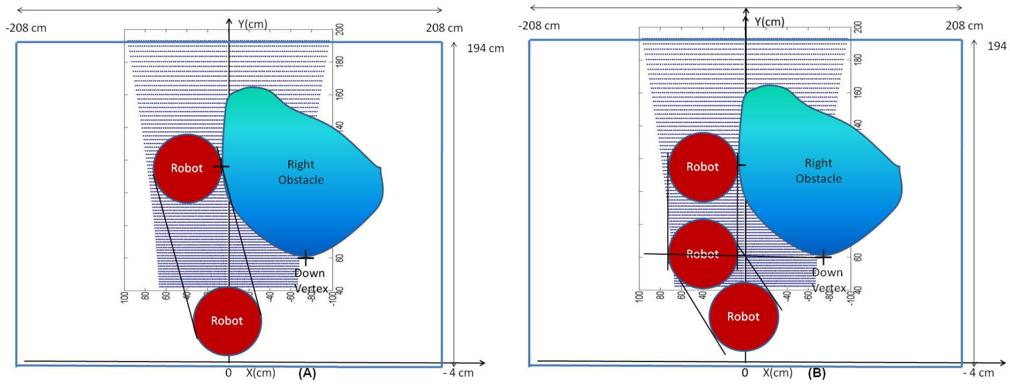


Figure 8.14: It is shown the turning and down vertex necessary constraints when a new obstacle avoidance strategy should be faced.

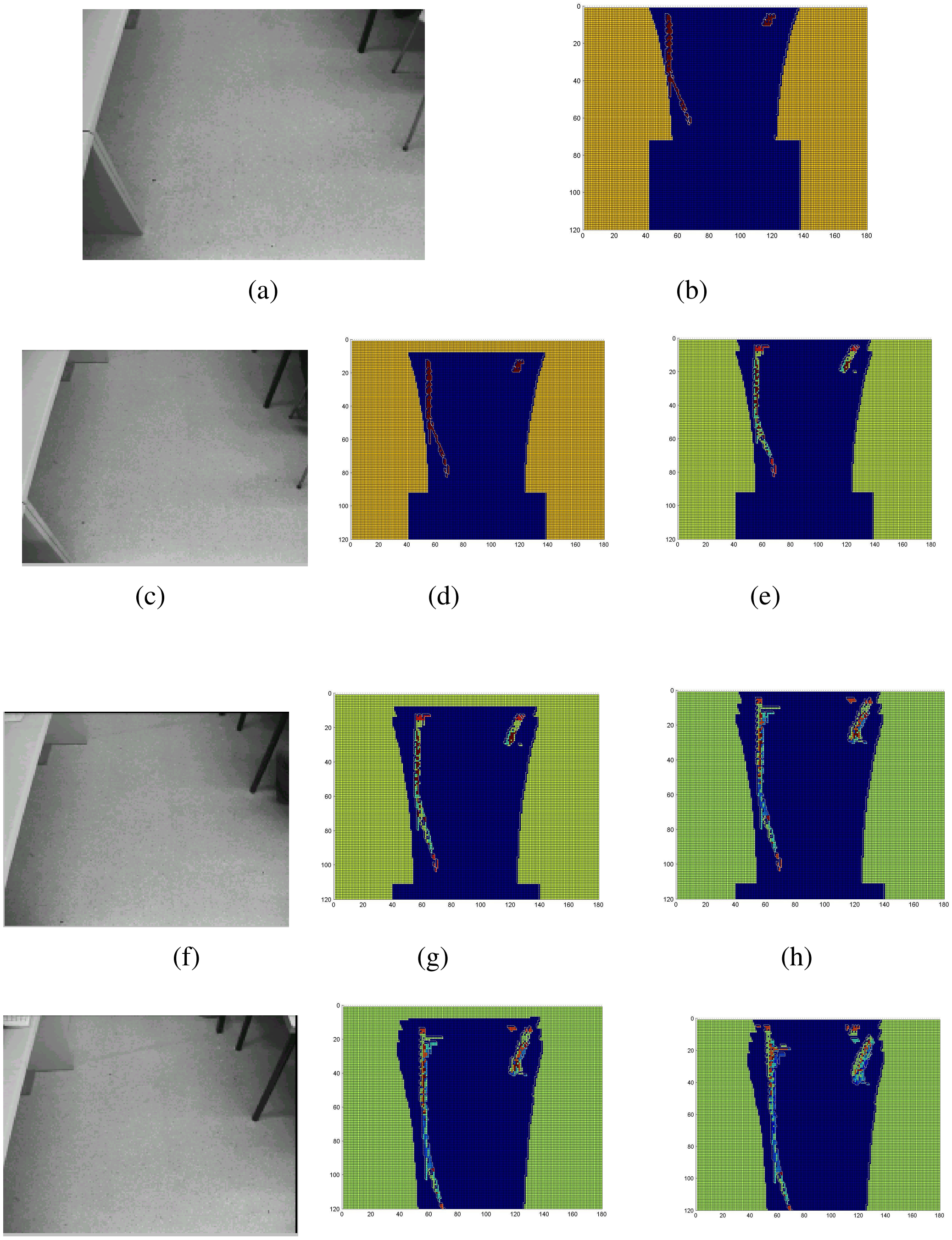
8.4 Experimental results

In this section the navigation strategy is tested. In this sense, the results obtained in the time-integration of the different frames and the navigation are shown in the subsequent subsections.

8.4.1 The time-integration of the frames within the local occupancy grid

The local occupancy grid should provide a bigger local map description where trajectories should be planned and images are time-integrated. The time integration of the frames is analysed from some experimental results. Floor model is used for inferring the predicted obstacle appearance in the next occupancy grid. Thus, the degree of energy match can reveal the true or false obstacles based upon the energy discrepancies. Marks placed on the floor, vertical obstacles, and typical indoor furniture are analysed through time-integration series of acquired frames. Figure 8.15 shows a sequence of acquired images and the occupancy grid obtained. The frames were acquired at positions (x, y, θ): (a) (0, 0, 90º); (c) (0, 21cm, 90.5º); (f) (0, 41cm, 90º); (i) (0, 62cm, 89.5º). The first obtained occupancy grid from position (0, 0, 90º) is shown in (b). The prediction of frame (a) when the position is (c) is shown in (d). In (e) is shown the fusion between the prediction of the last occupancy grid referred to the coordinates (0, 21cm, 90.5º), and the occupancy grid observed from (c). In this way, (e) is referred to the coordinate system of (f) in (g), and the fusion with the perception is shown in (h). Finally (h) is referred to the coordinate system of (i) in (j), and the fusion with the perception of (i) is shown in (k).

The results show coincidences in the table edge due to the aperture problem. The aperture problem arises due to the impossibility of computing OF, see Chapter 3. In this way, matches between the predicted table edges and the edges in the acquired frames show good correspondences, drawn in red colour. However, the vertical edges, which corresponds to the table foots; depict discrepancies when the distance from the floor contact point is increased. Therefore, the prediction uses the floor model and the 3D perspective arises in the acquired frames. Hence, occupancy grids reveal that the vertical edges increase their width along the time due to the lack of good coincidences, except in the floor contact points.



(a) (b)

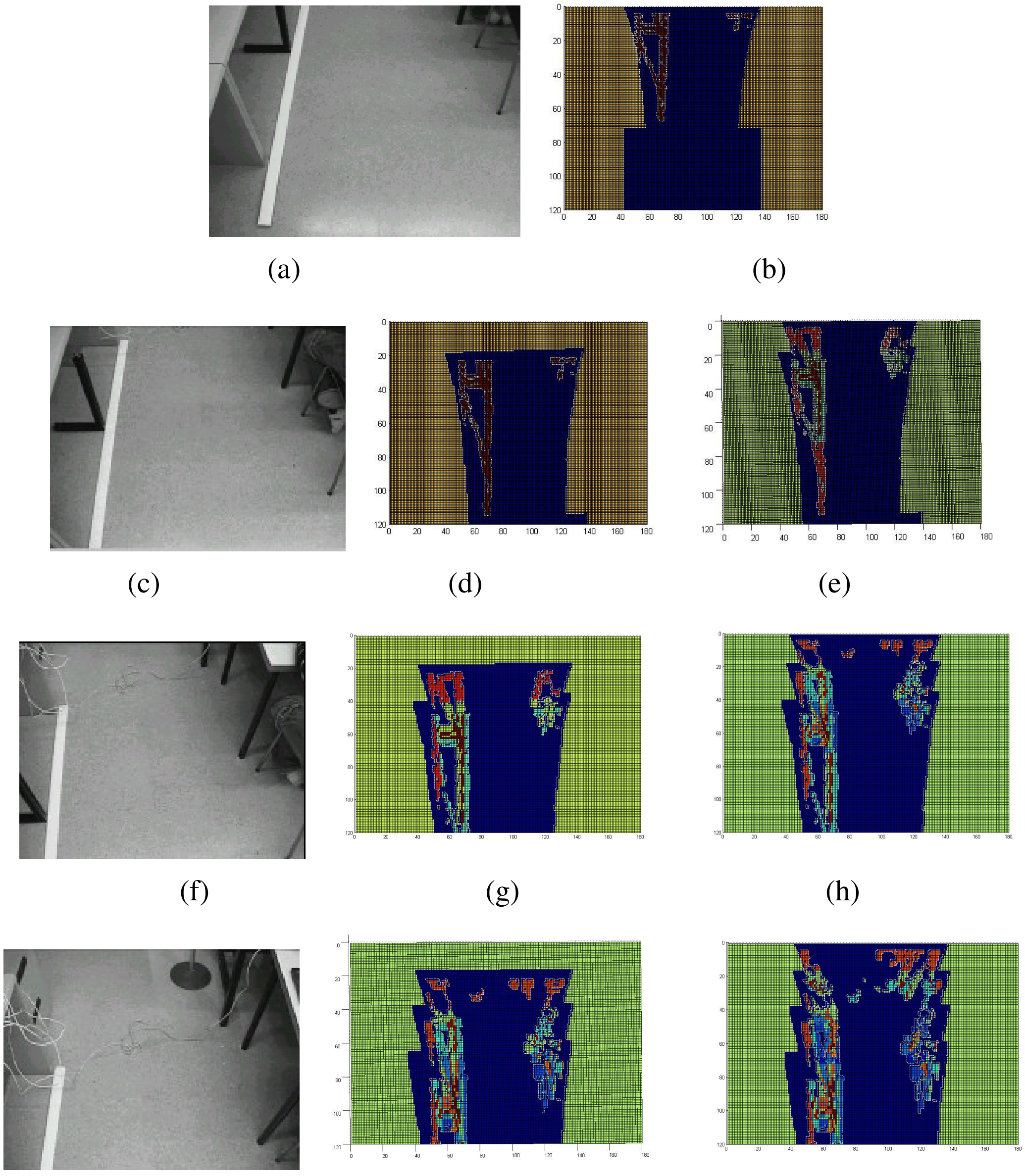
(c) (d) (e)

(f) (g) (h)

(j) (k) (l)

Figure 8.15: The frames acquired at positions (x, y, θ): (a) (0, 0, 90º); (c) (0, 21cm, 90.5º); (f) (0, 41cm, 90º); (i) (0, 62cm, 89.5º). The occupancy grid predictions and the fusions with the acquired frames are also depicted.

Next experiments explore the same scenario but a line placed in the floor is added. Moreover, two different WMR displacements are studied. Figures 8.16 and 8.17 show the results obtained when the different robot displacements are analysed. The expected results should be similar as those obtained in the first experiment. Therefore, floor regions should increase the probability with high values, red colour, and obstacles should decrease their values by approaching to blue colours. The aperture problem is also studied through the table edge and a floor line that are parallel to the WMR displacements. Figure 8.16 shows the results when the displacements are nearly 45cm. The frames were acquired at positions (x, y, θ): (a) (0, 0, 90º); (c) (0, 46cm, 89º); (f) (1cm, 92cm, 88º); (i) (2cm, 137cm, 88º). The first obtained occupancy grid from position (0, 0, 90º) is shown in (b). The prediction of frame (a) when the position is (c) is shown in (d). In (e) is shown the fusion between the prediction of the last occupancy grid referred to the coordinates (0, 46cm, 89º), and the occupancy grid observed from (c). In this way, (e) is referred to the coordinate system of (f) in (g), and the fusion with the perception is shown in (h). Finally (h) is referred to the coordinate system of (i) in (j), and the fusion with the perception of (i) is shown in (k).



(a) (b)

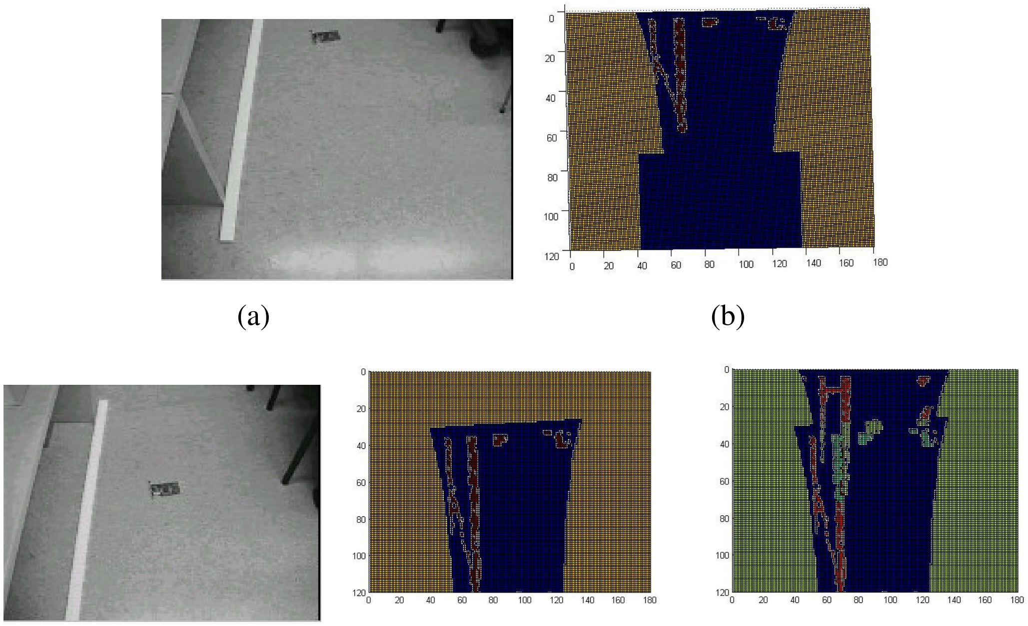
(c) (d) (e)

(f) (g) (h)

(i) (j) (k)

Figure 8.16: The frames acquired at positions (x, y, θ): (a) (0, 0, 90º); (c) (0, 46cm, 89º); (f) (1cm, 92cm, 88º); (i) (2cm, 137cm, 88º). The occupancy grid predictions and the fusions with the acquired frames are also depicted.

Figure 8.17 shows the results when the displacements are of near 70cm.



(a) (b)

(c) (d) (e)

Figure 8.17: The frames acquired at positions (x, y, θ): (a) (0, 0, 90º); (c) (0, 71cm, 88º). The prediction of (b) referred to the coordinates of (c) is shown in (d). The fusion is shown in (e).

The experimental results are analysed; it is shown that the aperture problem should be considered. Moreover, object edge sizes can be increased during the successive overlapping matches. The three experiments depict the results not so good as expected. Therefore, floor objects can have high probability values but also low probability values, drawn in blue colour; see occupancy grids of Figures 8.16 and 817. The source of error may be due to the lack of accurate camera calibration. However other aspects as small flat model deviations, odometer system errors, and 3D blob size should also be considered as a source of discrepancies between the predicted and acquired obstacle positions when high probability values are expected for floor regions and low values are expected for obstacle areas.

The 3D structure cannot be inferred due to the reasons previously depicted. However, the LPMOG can be used as an interesting framework in order to deal with it. Thus, further analysis can be developed by using blob shapes, edges, and featured points that are locally time-integrated within the occupancy grid. Consequently further studies can be developed in future. The aim of this research consists in testing navigation strategies. Therefore, LPMOG is used for such purposes as it is shown in the next sections.

8.4.2 Navigation experimental results

In this subsection the following three navigation experimental results are presented:

- First navigation experiment: static obstacle avoidance
- Second navigation experiment: straight and left turning corridor
- Third navigation experiment: straight and left and right turning corridor

Static obstacles were tested in the three experiments reported in this thesis as a suitable benchmark for obtaining the initial results of the methodology presented.

8.4.2.1 First navigation experiment: static obstacle avoidance

Figure 8.18 shows the lab environment map and the path followed when the WMR starts at the position (0, 0, 90º) towards the desired coordinates (0, 460cm). The scenario contains some static obstacles that the WMR should avoid. The different acquired frames are located within Figure 8.18.

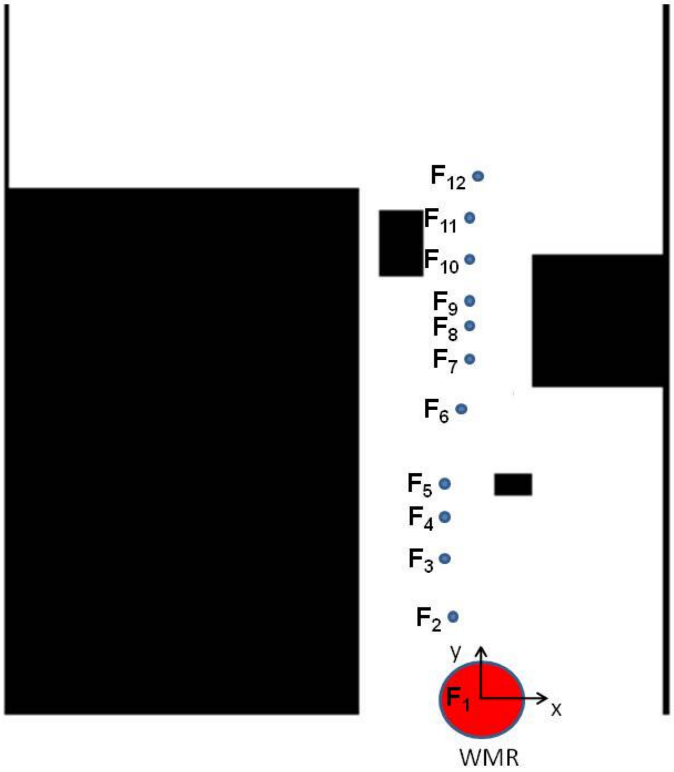


Figure 8.18: Simplified map scenario where the robot trajectory toward the goal is depicted with blue dots. The obstacles are drawn in black.

The real robot environment is showed in the following snapshots, Figure 8.19.a and Figure 8.19.b.

(a) (b)

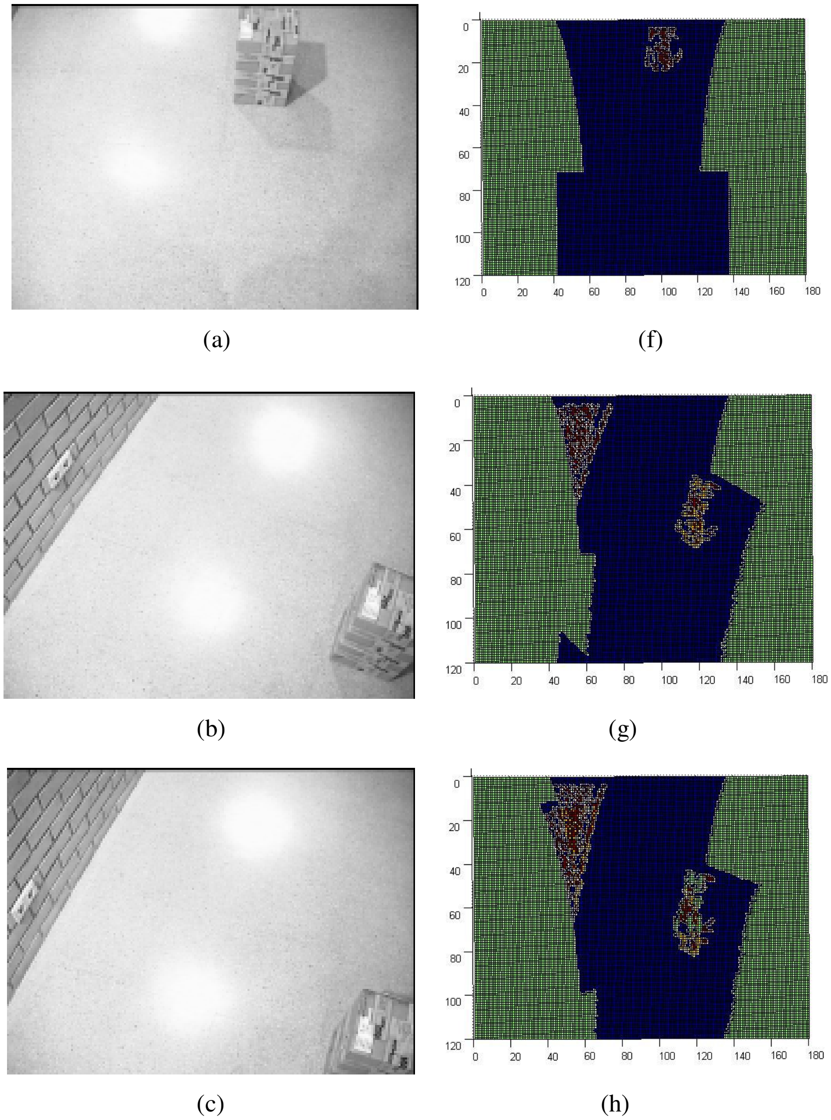
Figures 19.a and 19.b: It is presented the real scenario that has been drawn in Figure 18. It is shown the obstacles placed on the floor that the WMR should avoid.

The robot coordinates achieved during the WMR navigation are depicted in the table 8.1.

Table 8.1: First experiment coordinates from where the monocular frames are acquired

Frame 1	(0, 0, 90º)	Frame 7	(1, 293, 76º)
Frame 2	(-11, 66, 110º)	Frame 8	(4, 315, 88º)
Frame 3	(-21, 108, 99º)	Frame 9	(4, 339, 96º)
Frame 4	(-25, 141, 97º)	Frame 10	(6, 375, 89º)
Frame 5	(-26, 176, 94º)	Frame 11	(4, 415, 100º)
Frame 6	(-8, 248, 83º)	Frame 12	(12, 456, 74º)

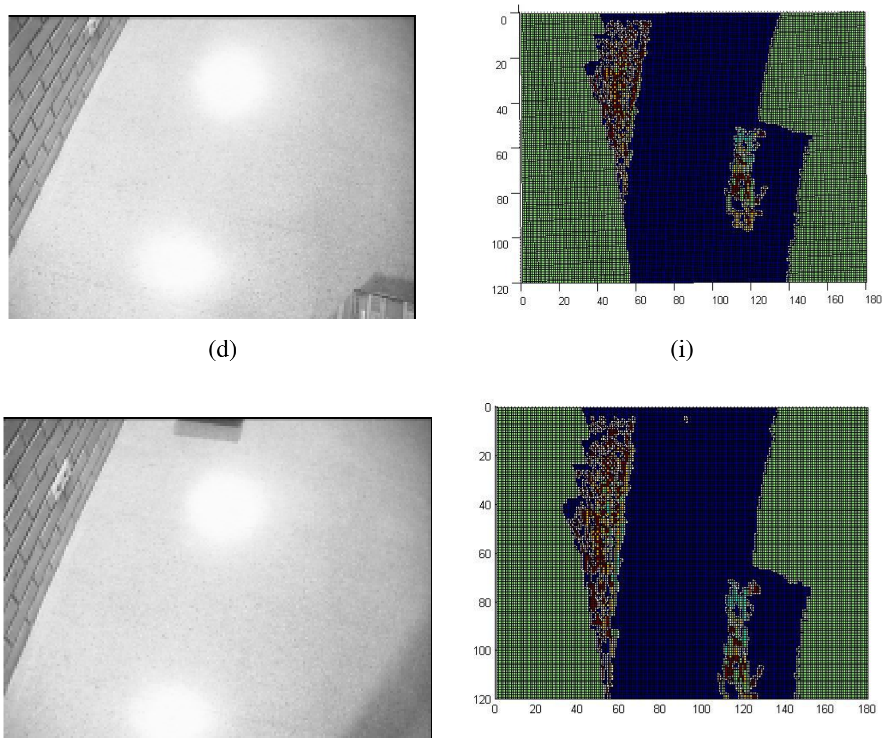
The first 5 frames are acquired during the obstacle avoidance strategy of the first obstacle placed in the middle of the corridor. Figures 20.a, 20.b, 24.c, 20.d and 20.e. show these frames. Figures 20.f, 20.g, 20.h, 20.i, and 20.j show the corresponding occupancy grid obtained by integrating the different frames.



(a) (f)

(b) (g)

(c) (h)



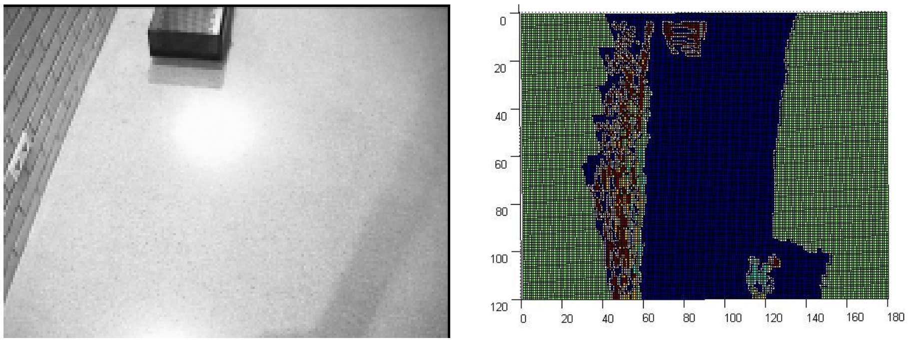
(d) (i)

(e) (j)

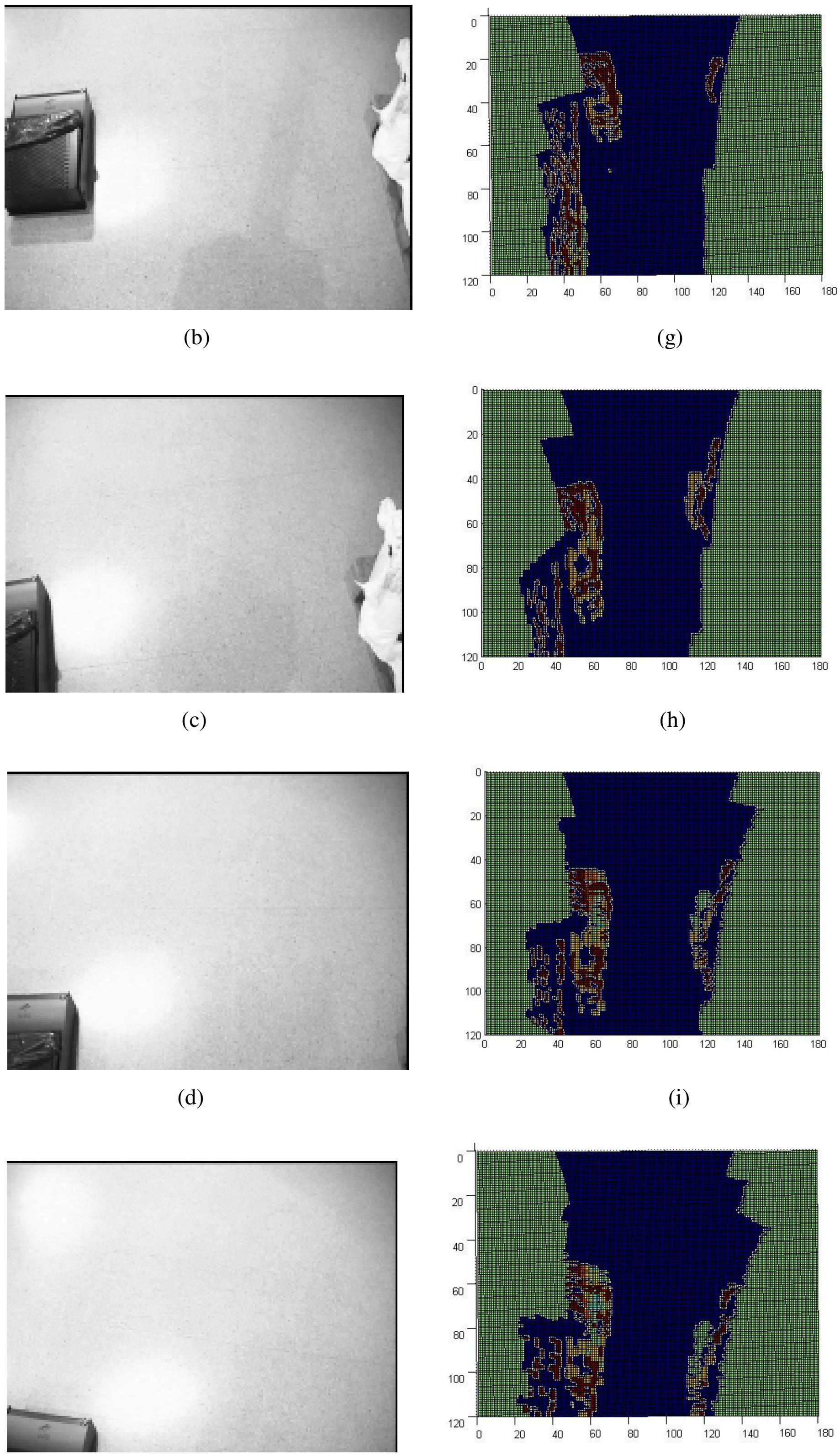
Figure 20: In (a), (b), (c), (d), and (e) are depicted the monocular frames acquired while the WMR is avoiding the first obstacle placed on the direction towards the objective. (f), (g), (h), (i) and (j) show the occupancy grid obtained by integrating the acquired frames.

The LPMOG built by integrating the first 5 frames is shown in Figures 20.f, 20.g, 20.h, 20.i, and 20.j. It is pointed out that in the frame 1 only the front obstacle is perceived. However, when other frames are integrated the left wall is integrated. Therefore the navigation is constrained by both obstacles.

Once the first obstacle is passed the navigation strategy consist in finding a path between the left and right obstacles encountered in the approaching trajectory towards the desired coordinates. Figures 21.a, 21.b, 21.c, 21.d, and 21.e depict these new acquired frames. The occupancy grid is built by integrating the frames from 6 to 10. Figures 21.f, 21.g, 21.h, 21.i, and 21.j represent the new perception horizon where the new appearing obstacles are time integrated. The obstacle avoidance strategy is planned for avoiding the collision with the left and right obstacles encountered.



(a) (f)



(b) (g)

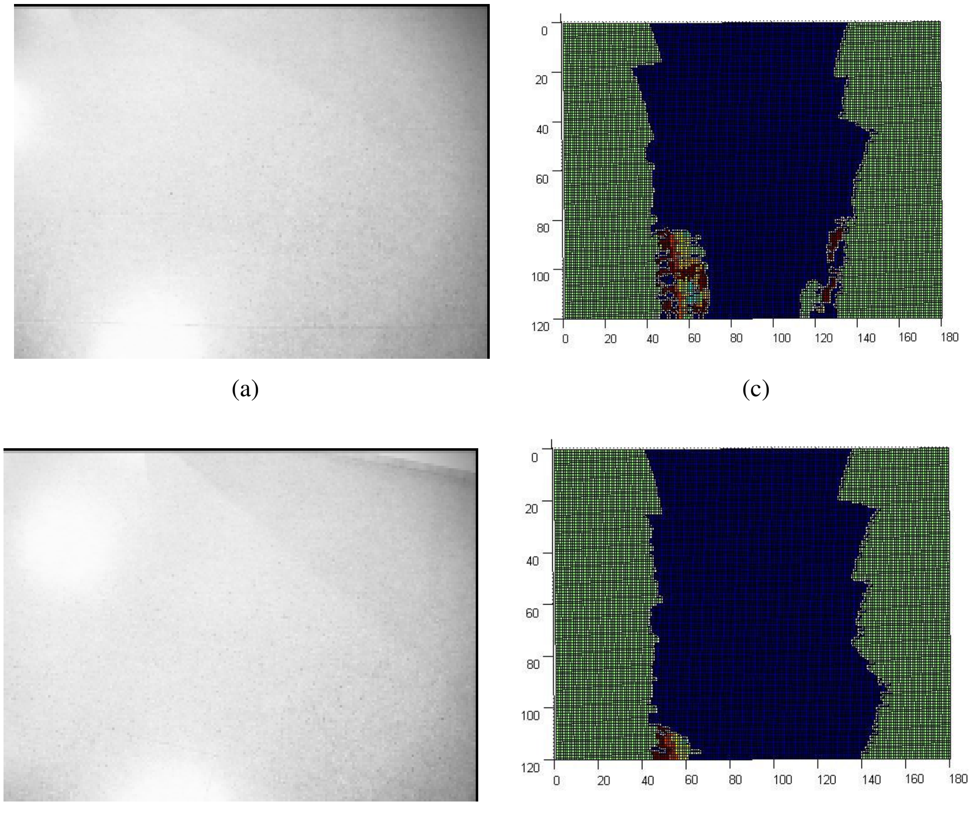
(c) (h)

(d) (i)

(e) (j)

Figure 21: It is shown the set of new frames acquired after the first obstacle avoidance strategy. (f), (g), (h), (i) and (j) show the occupancy grid built by time-integration of the new acquired frames during the approaching navigation towards the goal.

Figure 8.22.a and 8.22.b show the last two acquired frames. It is important to be aware that no obstacle appears in these frames. However, the LPMOG built by integrating these last two frames has memory of out of field of view obstacles. It is shown in Figures 8.22.c and 8.22.d.



(a) (c)

(b) (d)

Figures 8.22: (a) and (b) are the last monocular acquired frames that are free of obstacles. (c) and (d) are the corresponding occupancy grids that integrate the free of obstacle frames.

The LPMOG methodology has provided good results for local WMR navigation when typical static obstacle avoidance has been tested. Therefore memory of recent passed obstacles was obtained. However, the size of obstacles is increased due to the image formation perspective. Obstacle structures by using LPMOG methods have the drawbacks pointed out in section 8.4.1. Hence, matches between predicted and acquired frames are not only constrained to the floor areas due to the size of blobs. Moreover, overlapping mismatches occurred. Hence, lack of accurate camera calibration and dead reckoning errors are considered as the main source of errors. Therefore, the use of the LPMOG as an interesting framework for inferring the obstacle structure by computing qualitative OF is proposed. The use of floor model and the shape analysis is maintained as an attracting issue in which further studies should be developed.

8.4.2.2 Second navigation experiment: straight and left turning corridor

The second lab experiment consist of WMR navigation from coordinates (0, 0, 90º) to (-150cm, 350cm). However a straight path does not exist. The goal should be accomplished by going straight along the lab corridor and then turning to the left. Figure 8.23 shows the lab environment map and the path followed when the WMR starts at initial position towards the desired coordinates. The scenario contains some static obstacles as well as the corridor walls that the WMR should avoid. The different acquired frames are located within Figure 8.23.

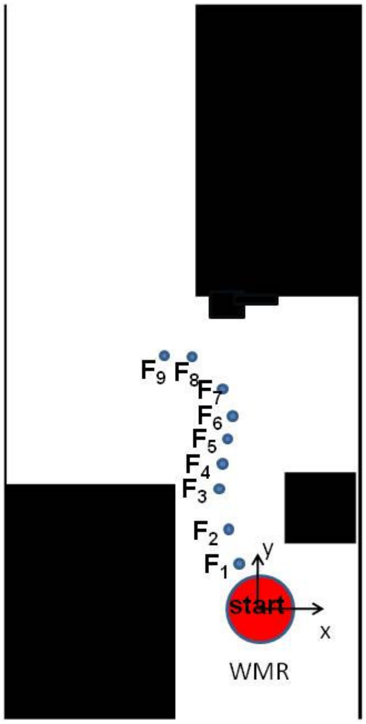


Figure 8.23: Simplified map scenario where the robot trajectory toward the goal is depicted with blue dots. The obstacles are drawn in black.

Figures 8.24.a and 8.24.b show the real robot environment where the second lab experiment has been developed.

(a) (b)

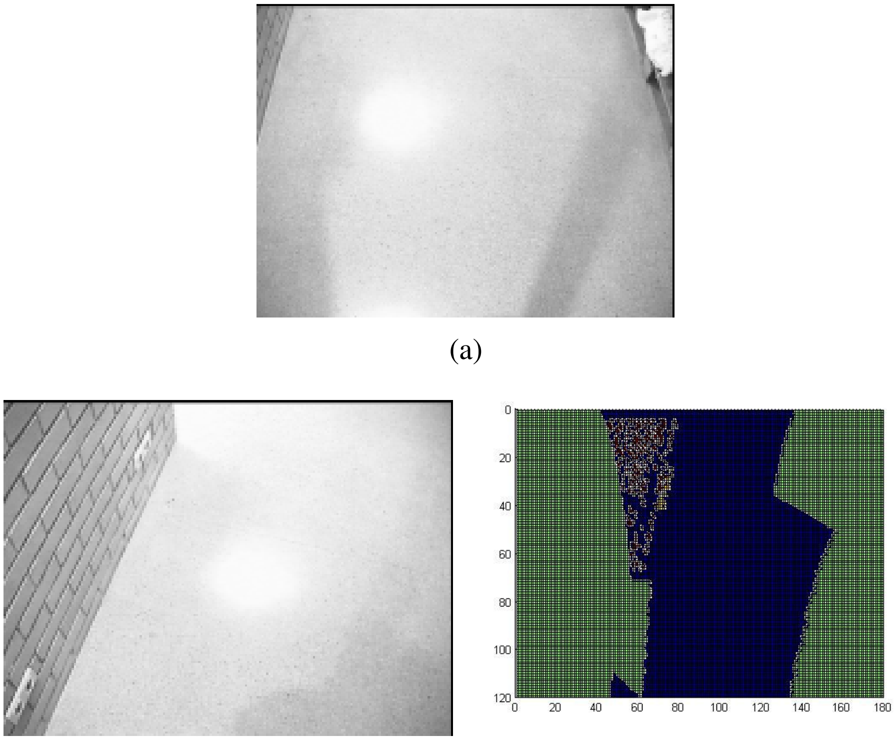
Figure 8.24: (a) It shows the straight corridor navigation where left wall and right obstacles appear. (b) Picture shows the left corner corridor that the WMR should overcome in order to attain the desired coordinates.

The robot coordinates achieved during the WMR navigation are depicted in the table 8.2.

Table 8.2: Second experiment coordinates from where the monocular frames are acquired

Start	(0, 0, 90º)	Frame 5	(-12, 258, 89º)
Frame 1	(-16, 68, 110º)	Frame 6	(-19, 299, 111º)
Frame 2	(-32, 137, 103º)	Frame 7	(-62, 349, 141º)
Frame 3	(-31, 177, 84º)	Frame 8	(-81, 352, 165º)
Frame 4	(-18, 215, 70º)	Frame 9	(-122, -258, 187º)

The navigation strategy is focused by the artificial potential field that pushes the robot to go left and forward. However the WMR navigation is constrained by the left wall as is shown by the pictures of these acquired frames as shown in Figures 8.25.a and 8.25b. The LPMOG is built by integrating the first two frames, as it is shown in Figure 8.25.c. Therefore, after an initial left turning the left wall appears as an obstacle that should be avoided.

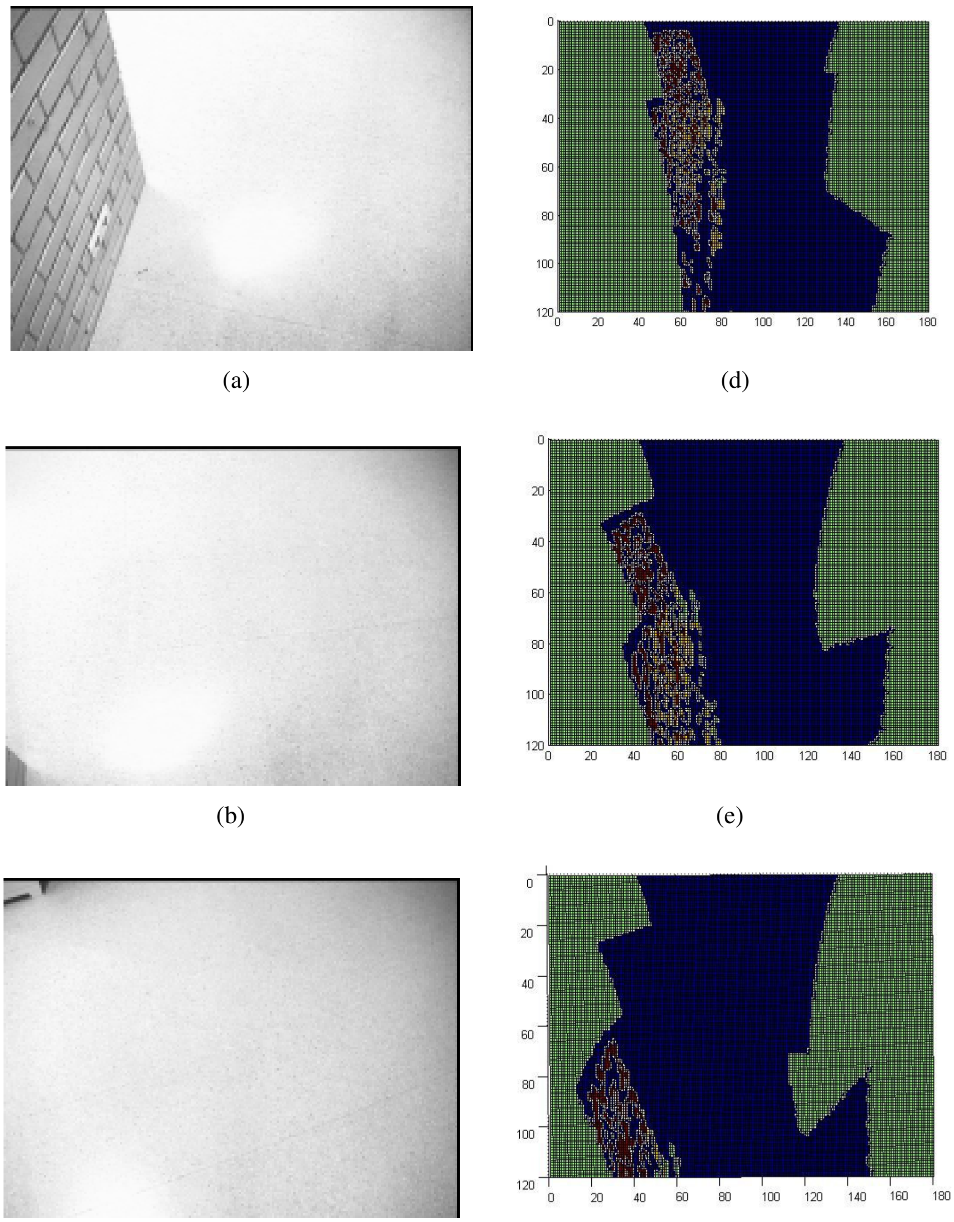


(a)

(b) (c)

Figure 8.25: (a) and (b) are the first two acquired frames depicting the left corridor wall that does not allow the left turning of the robot towards the desired coordinates. Figure 8.25.c depicts the occupancy grid built by the integration of the first two frames where the left corridor wall appears as an obstacle that should be avoided.

Frames “3”, “4” and “5” correspond to the obstacle avoidance of the left wall even when the wall is not present at the acquired frames due to the short term memory of the LPMOG. Figure 8.26.a, 8.26.b, and 8.26.c represent these frames. Figure 8.26.d, 8.26.e, and 8.26.f represent respectively the time-integration of the acquired frames “3”, “4” and “5”. It should be noted that in the corresponding occupancy grids include the left wall as an obstacle that should be avoided even when it is out of the monocular field of view.



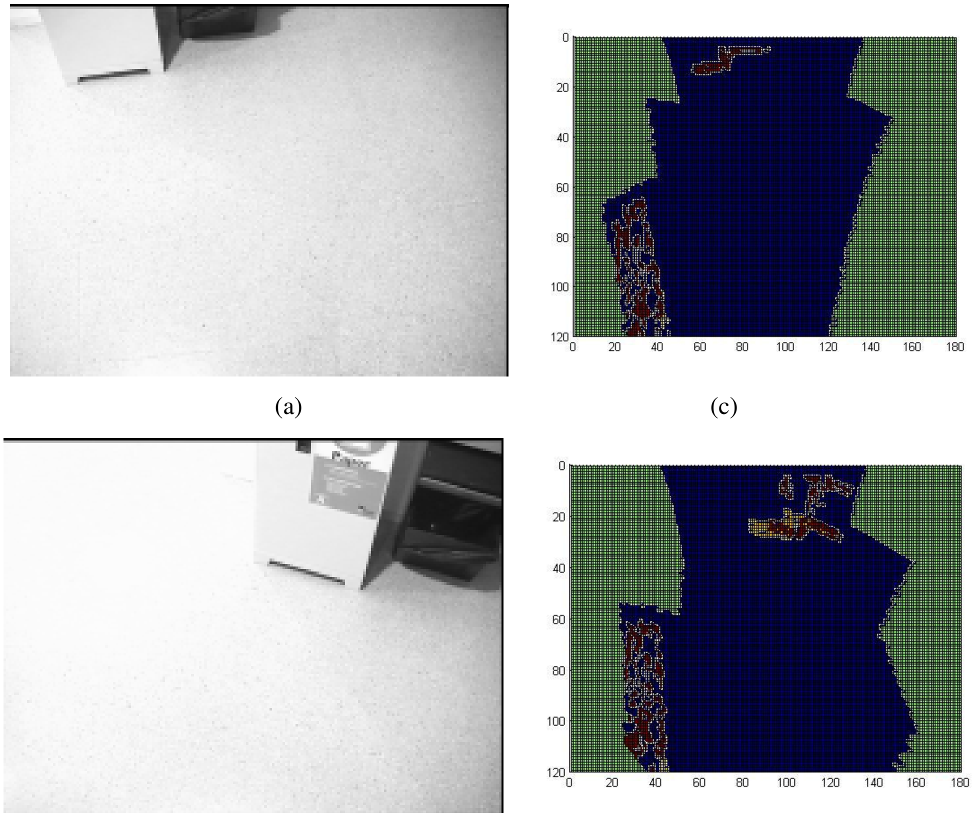
(a) (d)

(b) (e)

(c) (f)

Figures 8.26.a, 8.26.b, and 8.26.c: The acquired frames “3”, “4” and “5” during the WMR navigation. Figures 8.26.d, 8.26.e, and 8.26.f show the corresponding occupancy grids. It is noted that the occupancy grid is aware of the left corridor wall even when the new acquired frame depicts a free of obstacle field of view.

The left wall obstacle avoidance politicy is performed. However, other obstacles, placed in front of the WMR, appear in the frames “6” and “7” as shown in Figures 8.27.a and 8.27.b. The LPMOG integrates the newly found obstacles and has short term memory concerning about the left passed wall. In this sense, due to the 3D perspective of the left wall the obstacle physical size is increased. Figures 8.27.c and 8.27.d show the above fact. In this context the WMR navigation towards the goal is constrained by both obstacles.



(a) (c)

(b) (d)

Figures 8.27.a and 8.27.b: Front placed obstacles appear in the images acquired in the frames “6” and “7”. Figures 8.27.c and 8.27.d depict the occupancy grids that were built by integrating frames “6” and “7”. They still have memory of the left wall.

Frame “8” depicts the right obstacle that corresponds to the front obstacle of the previous frames. It is obtained after the obstacle avoidance strategy developed by the WMR. Figures 8.28.a and 8.28.b depict the frames “8” and “9”. Figure 8.28.c represents the LPMOG obtained after integrating frame “9”. The WMR navigation is constrained by left and right obstacles, but last frame corresponds to a free of obstacles perception allowing the WMR navigation towards the desired objective.

(a) (b) (c)

Figure 8.28.a and 8.28.b depict the frames “8” and “9”. Figure 8.28.c shows the occupancy grid obtained after the integration of frame “9”; it has memory of the past obstacles.

The second experiment reported a local navigation through indoor corridor; where going straight and turning left were accomplished by using an attraction potential field. The LPMOG methodology has provided good results for local WMR navigation when typical indoor navigation with static obstacles has been tested. Therefore memory of recent passed obstacles was obtained. The drawbacks are the same of the first experiment. Navigation failures consist of larger trajectories due to the fact that obstacles size is increased by the image formation perspective. Obstacle structures by using LPMOG probabilities have the similar difficulties that were pointed out in the first experiment and in section 8.4.1. Hence, floor segmentation using statistical results is not suitable for providing larger values to the floor areas. Therefore predicted and acquired radiance discontinuities of the frames are not only constrained to the floor areas due to the nature of obstacles and size of the blobs. In this sense, overlapping mismatches occurred. Therefore, lack of accurate camera calibration and dead reckoning errors are also pointed as the main source of errors in order to use occupancy grid statistics for segmenting the floor and inferring the obstacle structure.

8.4.2.3 Third navigation experiment: straight, left and right turning, and straight corridor

The third lab experiment consist in WMR navigation from coordinates (0, 0, 90º) to (-250cm, 1000cm); it is the larger navigation experiment that has been tested. Therefore, straight path does not exist. The goal should be accomplished by going straight along the lab corridor, then turning to the left, then to the right, and finally going straight ahead until the goal is reached. Figure 8.29 shows the simplified lab environment map and the path followed when the WMR starts at initial position towards the desired coordinates. The scenario contains some static obstacles as well as the corridor walls that the WMR should avoid. The different acquired frames are also localised in Figure 8.29.

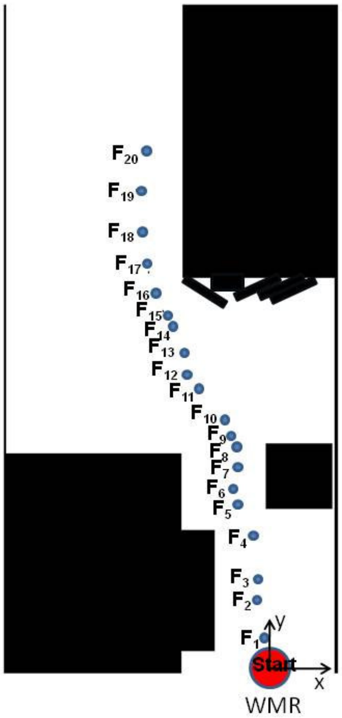


Figure 8.29: Simplified map scenario where the robot trajectory toward the goal is depicted with blue dots. The obstacles are drawn in black.

Figure 8.30.a, 8.30.b, 8.30.c, and 8.30.d show the real robot environment where the third experiment has been developed. Figure 8.30.a and 8.30.b depict the starting WMR position where the corridor and obstacles are shown. Figure 8.30.c represents the field of perception of the WMR in the proximity of the first left corner. Finally, Figure 8.30.d shows the last part of the scenario after right turning. It should be pointed that during the last part of the WMR navigation there are some radiance discontinuities placed on the floor.

(a) (b)

(c) (d)

Figure 8.30.a, b, c, and d: The scenario where the third experimented has been developed. It is the larger trajectory tested. Indoor corridor navigation includes straight, left turning, right turning, and straight trajectories.

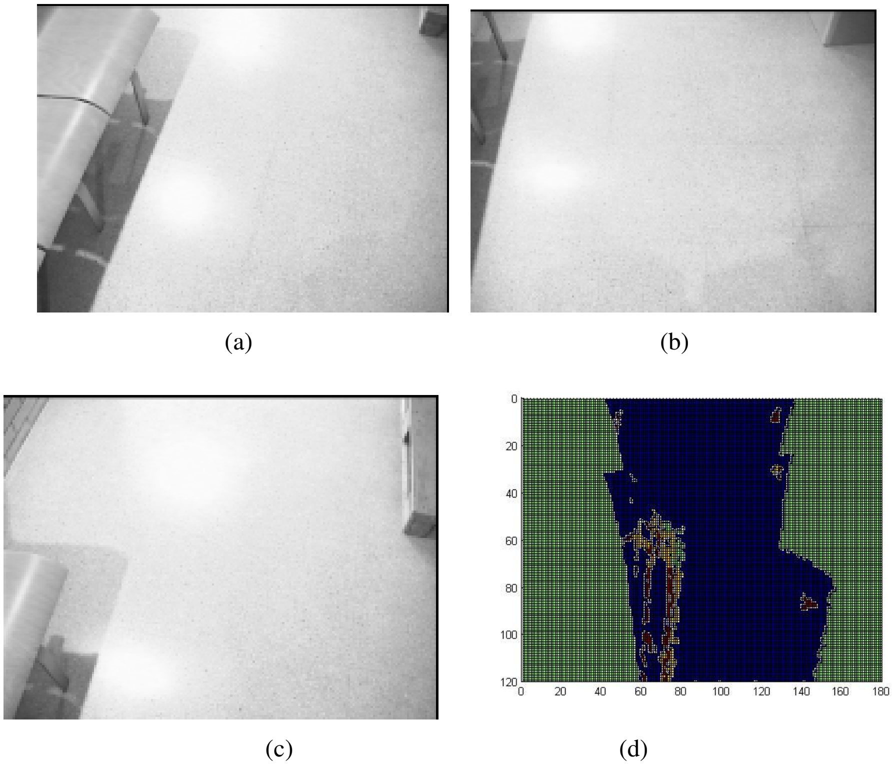
The WMR navigation should avoid the static obstacles while goal approaching is produced. The artificial potential fields push the WMR towards going straight and left. However due to the existence of obstacles placed in the scenario the navigation is constrained. Thus, the WMR movements consist in going straight, turning left, turning right, and going straight. The robot coordinates achieved starting at coordinates (0, 0, 90º) during such navigation are depicted in the table 8.3; their localisations are depicted in Figure 8.29. From these coordinates are acquired the corresponding monocular frames.

The artificial potential field pushes the WMR to go forward and left. However, static obstacles are placed in the left. The first 3 frames are acquired during the obstacle avoidance strategy of the first obstacle placed at the left of the corridor. Figures 8.31.a, 8.31.b, and 8.31.c show these frames.

Table 8.3: Third experiment coordinates from where the monocular frames are acquired

Frame 1	(-10, 56, 108º)	Frame 11	(-81, 480, 118º)
Frame 2	(-20, 126, 103º)	Frame 12	(-126, 517, 152º)
Frame 3	(-18, 166, 85º)	Frame 13	(-145, 543, 113º)
Frame 4	(-32, 242, 110º)	Frame 14	(-150, 581, 96º)
Frame 5	(-61, 305, 116º)	Frame 15	(-183, 631, 136º)
Frame 6	(-62, 337, 78º)	Frame 16	(-194, 658, 94º)
Frame 7	(-54, 374, 80º)	Frame 17	(-220, 728, 123º)
Frame 8	(-58, 414, 104º)	Frame 18	(-239, 802, 111º)
Frame 9	(-67, 431, 130º)	Frame 19	(-242, 878, 99º)
Frame 10	(-76, 460, 90º)	Frame 20	(-231, 952, 89º)

Figure 8.31.d shows the LPMOG corresponding to the integration of the first three frames where the left obstacles should be avoided by considering also the right obstacles.



(a) (b)

(c) (d)

Figures 8.31.a, 8.31.b, and 8.31.c: Left obstacles appear in the first acquired frames. The navigation is constrained by these obstacles. Figure 8.31.d shows the integration of the first three frames. The left and right obstacles are constraining the WMR movements towards the objective.

The WMR avoid the collision with the left obstacle by right turning. The right part of the corridor has also obstacles. Therefore, the monocular perception of the frame “4” is mainly focused on the right obstacle. It is noted that the left obstacle appears out of it. Figure 8.32.a shows the frame “4”. The LPMOG that integrates frame “4” includes the right side obstacle and has short term memory of the left side obstacles. Figure 8.32.b shows the occupancy grid. The WMR navigation consists in going forward while the collision with right and left side obstacles is avoided.

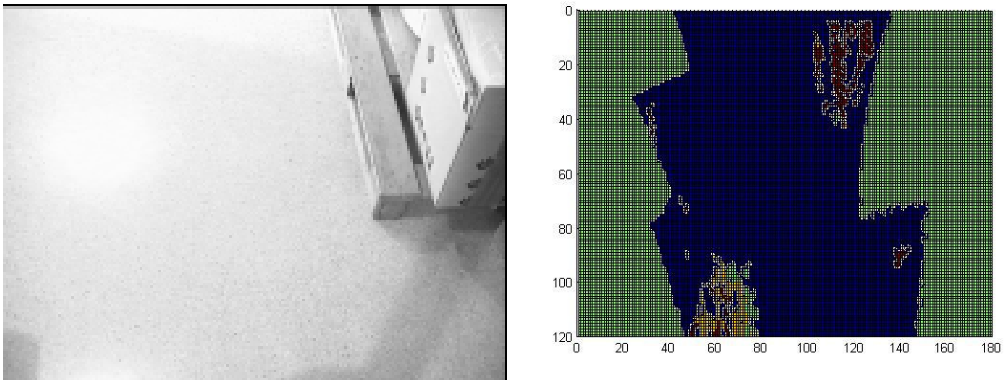


Figure 8.32.a: Frame “4” depicts the obstacles placed on the right side of the corridor. Figure 8.32.b shows the corresponding occupancy grid where appears the right side obstacle and the left side obstacles due to the integration of the previous acquired frames.

The WMR navigation tends to avoid the right obstacle turning to the left, as a consequence the corridor wall appears at frame “5”, see Fig 8.33.a. Figure 8.33.b shows the corresponding LPMOG where the right obstacle memory and the recent wall perception are collected together.

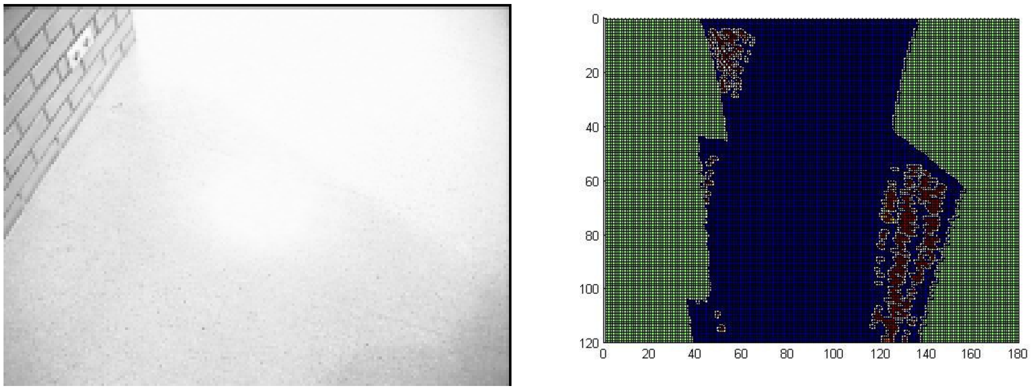
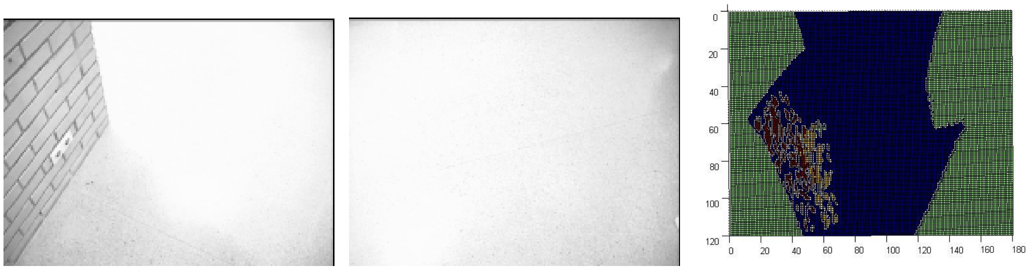


Figure 8.33.a: The left side corridor wall appears when the WMR turns towards left. Figure 8.33.b shows the occupancy grid corresponding to the frame “5” integration. Short term memory of the right obstacles is attained.

The WMR navigation avoids collision with left wall that appears also at frame “6” however at frame “7” a free of obstacles perception is obtained. These acquired frames are shown in Figures 8.34.a and 8.34.b. Figure 8.34.c shows the LPMOG obtained by integrating frames “6” and “7”. It can be observed that the memory of the right wall persist, however due to 3D perspective of the left wall the real obstacle size is increased.



(a) (b) (c)

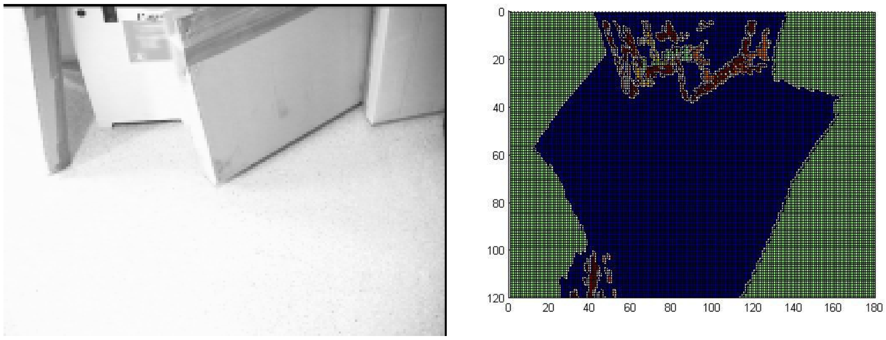
Figure 8.34.a and 8.34.b: The left wall and a free of obstacles scenario are shown in these frames. Figure 8.34.c depicts the occupancy grid obtained by integrating the frames “6” and “7”. The obstacle shape of the left side wall is oversized due to the 3D perspective.

The WMR navigation consists in avoiding the oversized left wall, consequently the robot moves in advancing sense. Frames “8” and “9” register the presence of obstacles placed in the front door faced by the WMR during the obstacle avoidance of the left wall. Figures 8.35.a and 8.35.b show these frames. The LPMOG is obtained when the frame “9” is integrated as depicted in Figure 8.35.c. The oversized left wall and the front obstacles should be avoided.

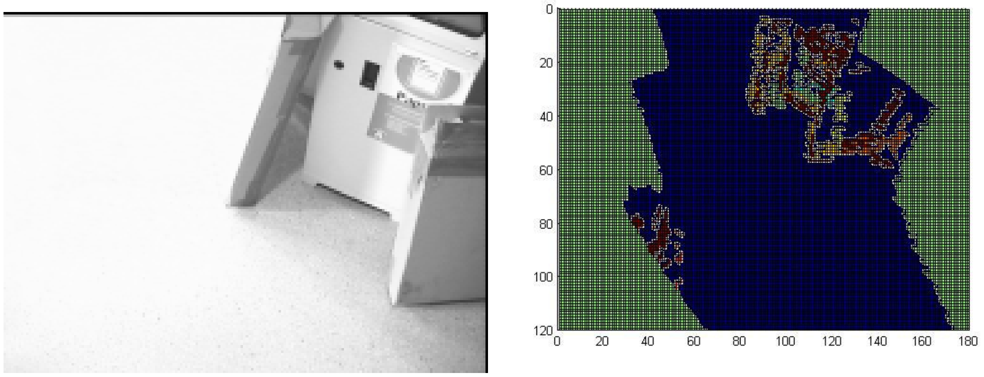
(a) (b) (c)

Figure 8.35.a and 8.35.b: Frames “8” and “9” depict the presence of front obstacles. Figure 8.35.c shows the occupancy grid obtained after the integration of the frame “9”.

It is shown that the presence of right and front obstacles will focus on the WMR navigation. Therefore artificial potential field pushes the WMR to go forward and left. Thus, the obstacle avoidance navigation consists in avoiding the collision by left turning. Figure 8.36.a and 8.36.b show frames “10” and “11” where the front obstacles appear during the left turning WMR navigation. Figures 8.36.c and 8.36.d show the corresponding LPMOG after the integration of the frames “11” and “12”. It can be seen that the front obstacles are placed within the grid while the memory of the left side wall is preserved during the left turning WMR navigation.



(a) (c)



(b) (d)

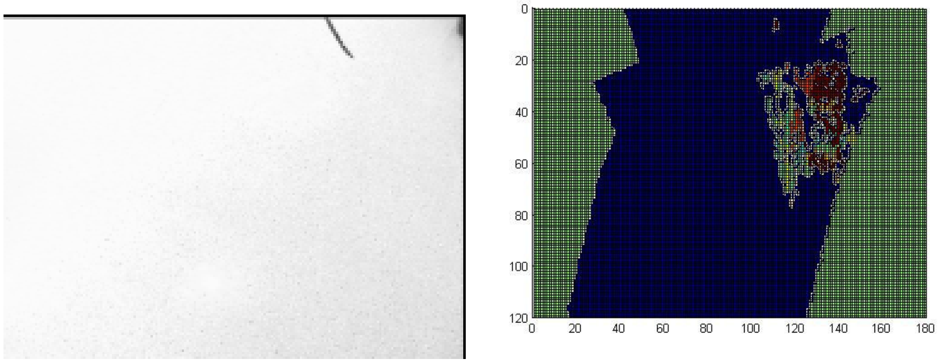
Figures 8.36.a and 8.36.b: Monocular perception of frames “11” and “12”. In Figures 8.36.c and 8.36.d are time integrated within the occupancy grid framework. The left side wall and front obstacles appear as obstacles.

Figures 8.37.a and 8.37.b show the frames “13” and “14”, which corresponds to right turning while collision with right side obstacles is kept. Figure 8.37.c shows the LPMOG obtained by time-integration of the acquired frame “14”.

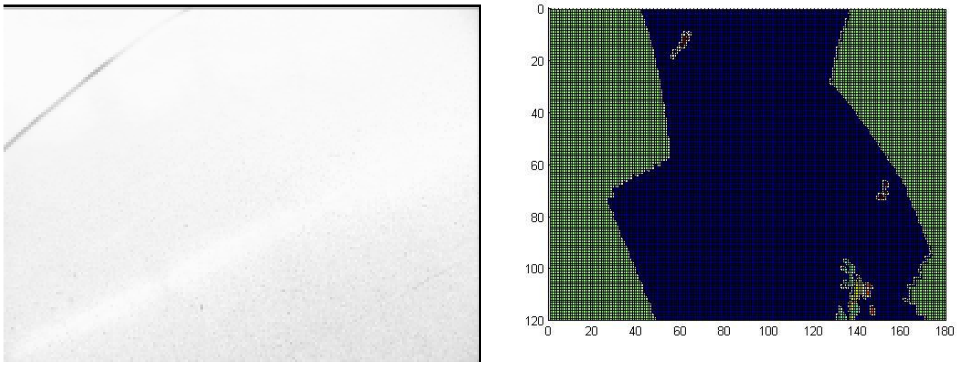
(a) (b) (c)

Figures 8.37.a and 8.37.b: Acquired frames when right turning navigation is done. Figure 8.37.c shows the time integration of the frame “14”.

After right turning, WMR navigation is focused on a new corner and corridor part. Figures 8.38.a and 8.38.b show frames “15” and “16” obtained while the WMR is right turning. In these frames lines places at the floor appear. It is noted that by using OPUOF methodology these marks are considered as obstacles. The LPMOG analysis should detect these radiances discontinuities as floor areas. Figures 8.38.c and 8.38.d show the occupancy grid obtained by integrating frames “16” and “17”.



(a) (c)

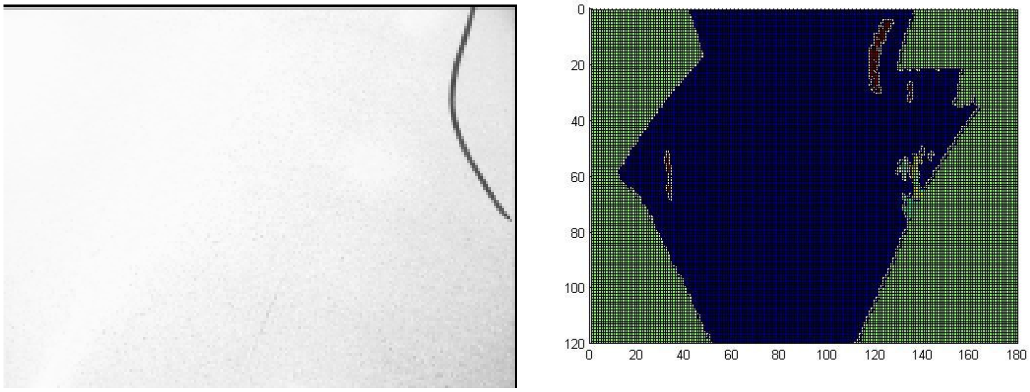


(b) (d)

Figure 8.38.a and 8.38.b: Monocular acquired frames during right turning movement. Figures 8.38.c and 8.38.d depict the occupancy grid obtained after the integration of frames “16” and “17”. It is important to be aware that the right corner appears as an obstacle due to the short-term memory.

It is noted that the field of view of these frames does not have data concerning about the right corner, however this information remains in the LPMOG as memorised past frames.

Once the WMR has turned right the artificial potential field pulls it to advance and left turning. Figure 8.39.a shows the acquired frame “17”. Figure 8.39.b depicts the corresponding LPMOG where the actual perception is integrated with the recent data corresponding to the previous frames.

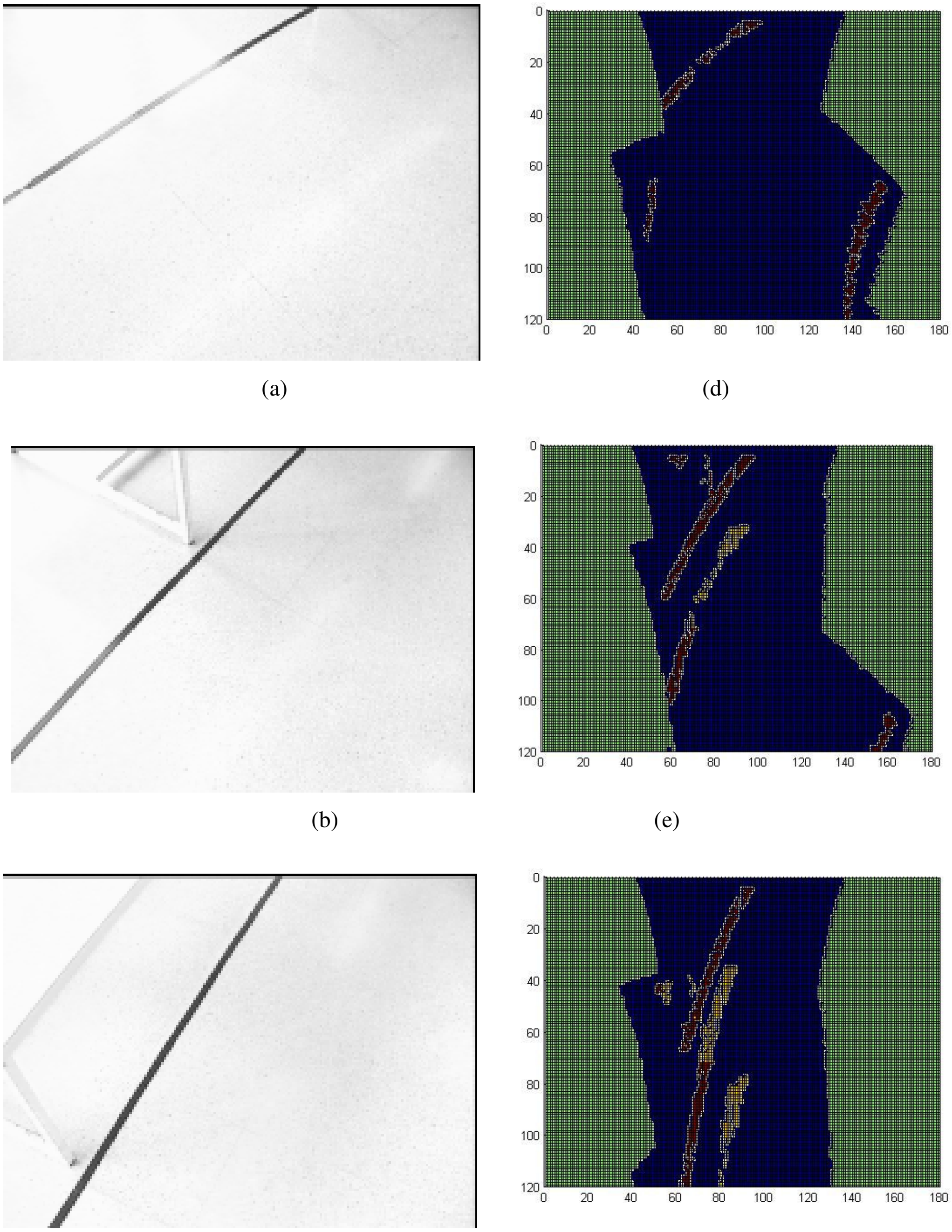


(a) (b)

Figure 8.39.a: Acquired frame “17”. Figure 8.39.b depicts the occupancy grid obtained with the integration of the frame “17” with the previous acquired frames. It is noted that the marks placed on the floor appear as obstacles that can be removed by using OF methods.

Frames “16” and “17” have marks placed at the floor. These marks are considered as obstacles in the research developed in this thesis. The WMR navigation is constrained by these marks. Figures 8.38.b and 8.39.b depict the LPMOG obtained with the integration of frames “16” and “17”.

The WMR navigation is constrained by the floor marks. Figures 8.40.a, 8.40.b, and 8.40.c show frames “18”, “19” and “20” that were acquired during the robot navigation. The artificial potential field pulls the WMR to go to the left and straight ahead, however the lines placed on the floor constraint the WMR navigation. Figures 8.40.d, 8.40.e, and 8.40.f show the LPMOG obtained by integration of the concerning frames. It should be pointed out the lack of accuracy in the overlapping of the different frames.



(a) (d)

(b) (e)

(c) (f)

Figures 8.40.a, 8.40.b, and 8.40.c: The monocular acquired frames “18”, “19” and “20” depict marks placed on the floor. Figures 8.40.d, 8.40.e, and 8.40.f show the occupancy grid that arise with the integration of the frames “18”, “19” and “20”.

The navigation strategy was successfully tested in the third experiment. The local desired configuration was 12.5m away, and the objective was accomplished by using a single artificial attraction field.

However, it is important to be aware of the lack of accuracy in the overlapping, of the marks placed on the floor, between predicted positions and acquired frames. Some clues about the reason of this drawback can be found as follows:

- The odometer system has errors due to the wheel slippage. These deviations are bigger when WMR movements contain turning actions.
- The camera calibration is not exact. Moreover small movements of the mechanical system used to hold the camera are produced during the WMR navigation.

Further LPMOG analysis is proposed as future work in order to solve the mismatches between the radiance discontinuities of regions belonging to the floor.

8.5 Conclusions

In this Chapter local navigation oriented to goal achievement was presented and successfully tested for static obstacles in typical indoor scenarios. The navigation algorithms are based on the closer obstacles coordinates obtained by the visual perception system and the artificial attraction field that push the WMR towards the desired coordinates. Therefore, the monocular machine vision system, presented in Chapter 6, is used for obtaining the significant obstacle vertex coordinates within an occupancy grid. The LPMOG methodology introduced in Chapter 6 was also tested for removing false obstacles but mismatches were obtained due to the reasons described in sections 8.3 and 8.4. Trajectory tracking is pursued by using the LMPC techniques depicted in Chapter 7.

Local navigation, which avoids static obstacles while goal approaching coordinates are achieved, has been successfully tested for the three experiments presented: static obstacle avoidance; straight and left turning corridor; and straight corridor, left and right turning and straight corridor. The navigation strategy presented has used artificial potential fields that attract the WMR towards the desired coordinates. The monocular system of perception used consists of a single camera. The use of the LPMOG, where the sequence of frames, is time-integrated allows a short term memory increasing the field of perception and consequently a better performance is achieved. In this context, the short-term memory provided by the LPMOG was depicted as an important way of increasing the field of perception. Thus, memory of out of field pass obstacles, and perception of visual dead zone are attained. The short-term memory benefits were clearly depicted in the experiments presented in the previous section.

In this work the knowledge of the environment is given by selecting effective potential fields that push the WMR towards the goal while static obstacle knowledge is provided by time-integration of monocular data. From each LPMOG the best approaching coordinates to the goal are obtained.

LMPC control techniques have been used to track segment lines. Concretely, each tracking segment is given by the straight line, which goes from the point from where the last perception was done to the local desired coordinates to be achieved within the field of perception. In this context, the local desired coordinates, to be achieved within the field of perception, are obtained by using the navigation strategy presented in section 8.3.2. Thus LMPC trajectory-tracking consists of a straight line actualised at each new perception frame. Odometer system is used for controlling WMR speeds and trajectory tracking and has achieved excellent results for local trajectory-tracking. However, dead reckoning errors are important issues in order to constraint the dimensions of the local trajectories that can be performed. In this way, the presented methodology is effective for trajectories not more than 30m. Therefore, the Chapter 7 experiences showed that a commanded trajectory of 22m provided averaged final distance errors of less than 0.5m, and angular orientation errors of less than 7º.

The methodology has been successfully tested with indoor scenarios with static obstacles. However some drawbacks should be pointed out:

- The monocular vision system is not effective for some scenarios as for instance those being lack of radiance or texture. The use of active vision systems can solve the above problem. The use of other sensor systems, such as sonar rings or laser rangefinders, should also be considered. Moreover, sensor fusion can be achieved by using the occupancy grid methodology. The use of deeper fields of view can be considered as an important issue for dealing with dynamic obstacles.
- The 3D perspective for radiance discrepancies of objects not belonged to the floor is an important drawback. Thus, the shape of the obstacle is increased. Consequently the navigation path is increased. Moreover, oversized obstacles can produce no path possibilities even when a real path is possible. It is important to be aware that such problems are not inherent of computer vision systems. Similar problems were reported by using other techniques as laser rangefinders or stereo vision systems [Coue et al., 06][Braillon et al., 06]. However, the use of computer vision systems provides a richer environment description, and consequently more clues can be obtained for recovering the 3D scenario structure.
- The overlapping mismatches for objects with contact parts on the floor should be analysed more carefully. In this sense a better camera calibration and the use of the floor contact coordinates as a way to improve the odometer system should be studied. OF analysis by using the floor model and the LPMOG framework is proposed as future work.

The LPMOG framework can be used for obtaining 3D obstacle structure by computing qualitative OF. Therefore, there is not any limitation concerning about the number of frames that can be time-integrated. Moreover, edges, shapes, and significant points can be locally analysed. An important goal will consist in finding a set of parameters for inferring 3D structure. It is proposed to search for a set of parameters that could be independent from the source of errors reported in this section. 3D knowledge can afford several benefits:

- To reduce the trajectories and wide-path.
- Visual Odometry.
- Landmark detection.

It can be summarised that the local navigation strategy presented can be used with global navigation strategies and complementary sensor systems. Thus, dead reckoning problems can be set to zero when feasible landmarks are detected. Moreover the use of complementary sensors systems can prevent failures of the system developed in this work. Therefore, the use of multilayer occupancy grids can allow dealing with dynamic obstacles in a sensor fusion context.

Chapter 9

Conclusions

9.1 Introduction

This thesis deals with WMR in a wide sense in accordance with the key requirements to enable further progress in mobile robotics. Therefore, the WMR was used to explore new methodologies related with monocular machine vision systems, control systems, trajectory-tracking, local mapping, and navigation issues. Furthermore, the experience acquired has also been transmitted to the student's community.

The research was done by first analysing the state of the art that corresponds to WMR navigation using computer vision. The 3D machine vision techniques were also studied. Monocular techniques were deeply studied, and especially attention was paid to DFF methods. The motivation to study DFF is based on the robustness and good results obtained by using such methodology. In addition, DFF methods are only used in few WMR applications. The control strategy is focused in methods that use experimental models as a way to obtain the system dynamics. The use of MPC techniques is proposed as an interesting methodology in order to develop the research. Therefore, LMPC was proposed in order to deal with the available on-robot local perception and the reactive behaviours that should be achieved under dynamics environments. The WMR perception systems, which consist of a monocular camera that describes just the neighbourhood of the robot and the odometer system, focus the experimental work that was developed.

Consequently, the principal contributions embrace different robotic areas as monocular computer vision, model based MPC, trajectory-tracking, local mapping, and navigation. The experimental results were presented in Chapter 6, Chapter 7, and Chapter 8 respectively. The main contributions that arise from this thesis are summarised in the next section.

9.2 Contributions

The different contributions are briefly commented in this section. In this sense, the different results achieved, within this research and their relationship with the different research areas, are listed in the following paragraphs:

- The monocular perception performance was analysed within the camera configuration used in this work. The camera pose produced a reduced field of view perception where WMR neighbourhood was depicted in order to achieve local information. The use of DFF techniques obtained robust and feasible results. Therefore, OPUOF method was reported as a suitable monocular machine vision system for WMR perception when flat floor and homogeneous floor radiance are expected.
- The floor radiance discontinuities were also analysed. OF methods were studied. However, due to the special camera configuration used in this work, magnification changes were expected even for patches belonging to the same

object. Thus, classical OF methods are not reliable. Hence, the present proposal for obstacle structure knowledge consists in using the LPMOG framework, where the floor model and the odometer data are used for analysing the predicted discontinuities and the new obtained discontinuities as a clue that will allow inferring 3D information. The LPMOG methodology is introduced as a probabilistic framework where sequences of images can be time-integrated. Time-integration can be used as a suitable system for analysing qualitative OF when radiance discontinuities are produced.

- The LMPC was presented as an appropriate methodology for short prediction horizons when local environment information is provided by the on-robot sensor system. The control strategy proposed consists of short-term actions as going straight or turning. Therefore, the perception system proposal just depicts local environment where just few seconds of path-planning can be done.
- The LMPC simulation results are used to direct the experimental research reported. The optimal search method has been selected due to the fact that gradient descent method can be considered as suboptimal. The short horizons can dealt appropriately with the expected reactive behaviours when local navigation is planned. The simulation results show that different cost function factors can be effective to perform the trajectory-tracking of the different local trajectories.
- The experimental on-robot LMPC results depict accurate trajectory-tracking for the different tested trajectories. Cost functions parameters were studied. APD factor is useful for going straight; while OD and TDD factors are suitable for turning actions. In this context APD with OD or APD with TDD were studied as different control law possibilities. Moreover, statistical analysis has been performed by using factorial design with two levels of quantitative factors as a way to infer time, trajectory accuracy, and control effort as a function of the cost function factor weights. Therefore, from statistical results obtained by testing a set of different representative trajectories within the perceived local scenario, different weights performances as function of the trajectory to be tracked and performance are obtained. Thus, the use of flexible cost functions, with different factor weights as function of the trajectory to be tracked, can be considered as a good strategy that allows better results when compared with fixed cost functions.
- The navigation strategy developed was similar to the corridor planning methodology tested with natural agents. The developed research showed that local animal abilities such as control and perception can be combined with a very simple imposed path-like structure to produce the desired overall motion. In this thesis, the corridor structure was given by artificial attraction potential fields while local control and perception were implemented by using LMPC and monocular perception methodologies.
- Path-planning in partially unknown environments should be short enough to allow local reactive behaviours. The trajectory planning should be flexible for dealing with short term uncertainties while the global task is accomplished. In this work, a new advancing trajectory generated a new environment description, due to the narrow field of view considered in this research. The present contribution can be understood in the human case as an "*illuminated step by step strategy*" (i.e. by using a small torch when we are exploring an unfamiliar

dark place where the goal position is achieved by using previous acquired knowledge that consists in the direction that we should explore towards the objective).

- In this context, the navigation algorithms are based on the significantly closer obstacle vertexes, obtained from the constrained field of perception by analysing the significant segmented blobs. Moreover, navigation and local path planning algorithms were developed. It can be considered as another important contribution of this thesis.
- Finally, the local map construction, where the navigation was accomplished, is another important contribution. Therefore, the use of the LPMOG where the sequence of frames is time-integrated allows a short term memory by increasing the field of perception and consequently a better performance is achieved. The LPMOG was not only used for obtaining the best approaching coordinates to the goal but also the occupancy grid provided a local map, where the dimensions are 4m by 2m, and local navigation can be planned while obstacles are avoided and approaching to the desired configuration trajectories was found.

9.3 Publications

Part of the results, obtained in the research presented, was published in specialised international conferences, journals, and books. The research subjects embrace different areas as the work presented. In this context, computer vision, WMR control, trajectory-tracking, navigation, and local-mapping are the main related issues that were published. The publications are listed in the following:

International Journals:

Mobile Robot Experimental Modeling and Control Strategies using Sensor Fusion. Pacheco L., Luo N., ***Control Engineering and Applied Informatics,*** Vol. 8, I(3), pp. 47-55, ISSN-1454-8658, September 2006.

Mobile Robot Local Predictive Control using a Visual Perception Horizon. Pacheco L., Luo N., ***International Journal of Factory Automation, Robotics and Soft Computing,*** I(2), pp. 73-81, ISSN-1828-6984, April 2007.

Mobile Robot Local Predictive Control under Perception Constraints. Pacheco L., Luo N., Ferrer J., ***International Journal of Factory Automation, Robotics and Soft Computing,*** I(4), pp. 40-48, ISSN-1828-6984, October 2007.

Local Model Predictive Control Experiences with Differential Driven Wheeled Mobile Robots. Pacheco L., Luo N., Ferrer J. ***Control Engineering and Applied Informatics,*** Vol. 10, I(2), pp. 59-67, ISSN-1454-8658, June 2008.

Interdisciplinary knowledge Integration through an Applied Mobile Robotics Course. Pacheco L., Luo N., Ferrer I., Cufí X., ***International Journal of Engineering Education***, (in press).

Book Chapters:

Mobile Robot Local Predictive Control Using a Visual Perception Horizon. Pacheco, L., Luo, N., (Eds) Pennachio S., International Society for Advanced Research, ***Emerging Technologies, Robotics and Control Systems***, Vol. 2, pp. 217-225, ISBN: 978-88-901928-2-1, Palermo (Italy), June 2007.

Constrained Monocular Obstacle Perception with just one frame. Pacheco, L., Cufí, X., Cobos, J., Lecture Notes in Computer Science***, Pattern Recognition and Image Analysis,*** (Eds.) Marti J., et al., Springer-Verlag, Vol. 1, pp. 611-619, ISBN: 978-3-540-72846-7, Berlin (Germany), June 2007.

Trajectory planning with a control horizon based on a narrow local grid perception: A practical approach for WMR with monocular visual data. Pacheco, Ll, Luo, N., Lecture Notes in Control and Information Sciences, ***Robot Motion and Control,*** (Eds.) Kozlowski, K., Springer-Verlag, pp. 99-106, ISBN: 978-1-84628-973-6, London (UK), 2007.

Mobile Robot Local Predictive Control Under Perception Constraints. Pacheco, Ll, Luo, N., Ferrer, J., International Society for Advanced Research, ***Recent advances in Control Systems, Robotics and Automation***, (Eds) Pennachio S., pp. 103-111, ISBN: 978-88-901928-3-8, Palermo (Italy), January 2008.

Predictive Control with Local Visual Data. Pacheco, L., Luo, N., Cufí, X., ***Robotics, Automation and Control,*** Artificial Intelligence Series, (Eds.) Pecherkova, P., et al., pp. 289-306, ISBN: 978-953-7619-18-3, I-Tech Education and Publishing KG, Vienna (Austria), October 2008.

Mobile Robot Navigation Strategies Oriented to Goal Achievement with Local Monocular Data. Pacheco, L., Luo, N., Cufí, X., ***Robot Vision: Strategies, Algorithms, and Motion Planning,*** (Eds) Daiki Itô, pp. 1-118, ISBN: 978-1-60692-091-6, Nova Science Publishers, Inc., New York (USA), 2009.

Conferences:

Calibration of a camera, the control of mobile robots using DFF techniques. Pacheco, L., Cufi, X., Cifuentes, J., ***IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics***, Vol. 2, pp. 135-140, ISBN: 973-713-047-2, 2004.

Experimental modelling and Control Strategies on an open mobile robot platform PRIM. Pacheco, L., Luo N., Arbusé, R., ***IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics,*** Vol. 2, pp. 225-230, ISBN: 1-4244-0360-X, 2006.

PRIM an open mobile platform. Motivation, present and future trends. Pacheco, L., Batlle, J., Cufi, X., Arbusé, R., ***IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics,*** Vol. 2, pp. 231-236, ISBN: 1-4244-0360-X, 2006.

Constrained Monocular Obstacle Perception with just one frame. Pacheco, L., Cufí, X., Cobos, J., ***3rd Iberian Conference on Pattern Recognition and Image Analysis, IbPRIA'07***, Vol. 1, pp. 611-619, ISBN: 978-3-540-72846-7, 2007.

Predictive control by local visual data. Pacheco, L., Luo, N., ***4th International Conference on Informatics in Control, Automation, and Robotics,*** **ICINCO'07**, Vol. 1, pp. 259- 266, ISBN: 978-972-8865-87-0, 2007.

Trajectory planning with a control horizon based on a narrow local grid perception: A practical approach for WMR with monocular visual data. Pacheco, L., Luo, N., **6th** ***International Workshop on Robot Motion and Control, ROMOCO'07,*** pp. 99-106, ISBN: 978-1-84628-973-6, 2007.

A Multidisciplinary Knowledge Integration Experience Through an Applied Robot Mobile Course. Pacheco L., Luo N., Cufí, X., Ferrer I., ***Proc. Inter. Technology, Education and Development Conference, INTED 2008*,** ISBN: 978-84-612-0190-7, Valencia 2008.

Control Education within a Multidisciplinary Summer Course on Applied Mobile Robotics. Pacheco L., Luo N., Ferrer I., Cufí X., ***Proc. 17th IFAC World Congress*,** pp. 11660-11665, ISBN: 978-1-1234-7890-2, Seoul, Korea, 2008.

Local Model Predictive Control Experiences with Differential Driven Wheeled Mobile Robots. Pacheco L., Luo N., Ferrer J., ***IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics**,* Vol. 2, pp. 377-382, ISBN: 978-1-4244-2576-0, 2008.

Knowledge Integration within an Applied Mobile Robot Summer Course. Pacheco L., Cufí X., Luo N., Ferrer I., Batlle J., ***IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics**,* Vol. 2, pp. 310-315, ISBN: 978-1-4244-2576-0, 2008.

WMR Navigation using Local Potential Field Corridors and Narrow Local Occupancy Grid Perception. Pacheco L., Cufí X., Luo N., Cobos J., ***IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics**,* Vol. 2, pp. 304-309, ISBN: 978-1-4244-2576-0, 2008.

Local WMR Navigation with Monocular Data. Pacheco L., Luo N., Cufí X., Cobos J., ". ***Proc. 4th Inter. Conference on Autonomous Robots and Agents, ICARA 2009**,* pp. 584-589, ISBN: 978-1-4244-2713-0, 2009.

A Monocular Occupancy Grid for Local WMR Navigation. Pacheco L., Cufí X., Luo N., Cobos J. ***6th International Conference on Informatics in Control, Automation, and Robotics, ICINCO'09***, (accepted).

9.4 Future Work

The development of this research has led to new problems and topics that need further work. The future work will embrace different research areas as a consequence of the research results of this thesis. The principal research lines are listed below:

- The 3D structure is an important objective to be accomplished. The results achieved show that probabilistic results are not enough in order to obtain the obstacle structure. Therefore, the results obtained for getting 3D information have some mismatches when overlapping areas between predicted and obtained blobs are analysed. These errors can be generated by the following sources:
    - Odometry errors.
    - Camera calibration errors.

  Thus, future research should include other features as blob shape, size, edge evolution, and featured corners as additional clues for inferring obstacle structure within the methodology presented. The use of vertical models is also suggested. The future work will be addressed to solve the above problems. In this way, the occupancy grid framework can be used for obtaining 3D obstacle structure. Therefore, there is not limitation concerning to the number of frames that can be time-integrated. The future goal will consist in finding a set of parameters in order for inferring 3D structure. It is searched for a set of parameters that could be independent from the source of errors reported in this section.

- The 3D understanding will afford several benefits that can be summarised as follows:
    - Reduction of the trajectories and wide-path.
    - Visual odometry.
    - Landmark detection.

  The trajectories can be reduced due to the fact that the shape of the obstacles will be reduced. Consequently, shorter paths will improve the WMR path-planning. Visual odometry is another interesting issue that should be studied. Therefore, the wheel slippage effects can be decreased. Furthermore, the structure knowledge should be used for detecting feasible landmarks that set to zero the incremental error produced by the odometer system.

- Despite the good results obtained by using LMPC strategies more improvements should be attained. In this way, when longer prediction horizons were tested some problems concerning the motor dead zone can appear. Therefore, more studies for solving such problems should be developed as a future control research. Moreover, further studies on LMPC should be done in order to analyse its relative performance with respect to other control laws or to test the cost function performance when other factors are used. For instance, factorial analysis by using a cost function with the three factors (APD, OD, TDD) is also proposed as future work.

- To develop a framework that will allow splitting experiments based on control from computer science results. Therefore, it is proposed to implement the presented methods by developing flexible software tools that allow vision methods to be tested and locally readable and virtual obstacle maps to be created taking into account the robot's position and the selected camera configuration. The use of virtual visual information can be useful for testing the robot in synthetic environments and for simulating different camera configurations.

- The navigation strategies should be tested by using other camera configurations that afford benefits, as for instance for reducing the dead zone of the perception system.

- Finally, the occupancy grid is an important framework that allows sensor fusion by considering multiple layers. The use of other sensor systems, as for instance sonar or range-finders, is an important goal that should be accomplished. Therefore, the vision system can have drawbacks when texturesless environments are occurred. To solve such problems sensor fusion by using the occupancy grid framework is proposed. Moreover, the use of multiple layers was reported by scientific community as a feasible way in order to discriminate static obstacles from dynamic obstacles. Therefore, mobile obstacles can be tracked in specific layers. In this way, the use of sensors with wider fields of view, which can allow larger scenario descriptions for analysing obstacle movements, is proposed for dealing with dynamic obstacles.

References

[Adelson et al., 84]
Adelson E. H., Anderson C. H., Bergen J. R., Burt P. J., Ogden J. M., Pyramid Methods in Image Processing, *RCA Engineer*, 29-6, pp. 33-41, Nov/Dec. 1984.

[Alamo et al., 03]
Alamo, T., Bravo, J. M., Camacho, E., F., Guaranteed State Estimation by Zonotopes, *IEEE Conf. on Decision and Control*, Vol. 6, pp. 5831-5836, 2003.

[Aloimonos and Duric, 92]
Aloimonos Y., Duric Z., Active Egomotion Estimation: a Qualitative Approach, *Proc. ECCV*, Ligure, Italy, pp. 497-510, 1992.

[Astron and Wittenmark, 88]
Astron K. J., Wittenmark, B., Computed Controlled Systems. Theory and Design, *Prentice Hall Ed.*, 1988.

[Atiya and Hager, 93]
Atiya, S., Hager, G. D., Real Time Vision-Based Robot Localitzation, *IEEE Trans. Robotics and Automation*, vol. 9, no. 6, pp. 785-800, Dec. 1993.

[Barrow et al., 94]
Barrow J. L., Fleet D. J., Beauchemin S. S., Performance of Optical Flow Techniques, *ICJV* 12:1, pp. 43-77, 1994.

[Batlle et al., 95]
Batlle, J., Garcia, R., Forest, J., Cufi, X., Pacheco, Ll., The robot prototype ROGER " Robot operational generic for research and experimenting", *IAV 95, Int. Aut. Vehicles*, Participation with the robot in a field demonstration, Helsinki, 1995.

[Bernardino and Santos-Victor, 98]
Bernardino, A., Santos-Victor, J., Visual Behaviours for Binocular Tracking, *Robotics and Autonomous Systems*, vol. 25, no. 3-4, pp. 137-146, Nov. 1998.

[Borenstein and Koren, 89]
Borenstein, J., Koren, Y., Real-Time Obstacle Avoidance for Fast Mobile Robots, *IEEE Trans. System, Man, and Cybernetics*, vol. 19, no. 5, oo. 1179-1187, 1989.

[Born, 65]
Born, M., Wolf, E., *Principles of Optics*, London: Pergamon, 1965.

[Boyd and Vanderberghe, 04]
Boyd, S., Vandenberghe, L., Convex Optimization, *Cambridge University Press*, 2004.

[Box et al., 05]
Box, G. E. P., Hunter, J. S., Hunter, W. G., Statistics for Experimenters, Wiley Series in Probability and Statistics, D. J. Balding et al. (Eds.), New Jersey, USA, 2005.

[Braillon et al., 06]
Braillon, C., Usher, K., Pradalier, C., Crowley, J.L., Laugier, C., Fusion of Stereo and Optical Flow Data using Occupancy Grids, *Proc. IEEE Int. Transportation Systems Conference*, pp. 1240-1245, 2006.

[Brockett, 83]
Brockett R. W., Asymtotic stability and feedback stabilization. *In: Differential Geometric Control Theory. (R. W. Brockett, R. S. Millman and H. S. Sussman, Eds.* Birkhauser. Boston. pp. 181-191. 1983).

[Bruhn et al., 02]
Bruhn A., Weickert J., Schnörr C., "Combining the advantages of local and global optimal flow methods", *in Proc. Pattern Recognition, SPRINGER LCNS*, Sep. 2002, pp. 454-462.

[Bruhn et al., 05]
Bruhn A., Weickert J., Feddern C., Kohlberger T., Schnörr C., Variational Optical Flow Computation in Real Time, *IEEE Trans. On Image Proc.,* Vol. 14, May 2005.

[Butler, 06]
Butler Z., Corridor planning for natural agents, *Proc. ICRA*, pp. 499-504, 2006.

[Camacho and Bordons, 02]
Camacho, E., F., Bordons, C., Model Predictive Control, *Springer-Verlag, London*, 2002.

[Campbell, 04]
Campbell J., Sukthankar R., Nourbakhsh I., Techniques for Evaluating Optical Flow in Extreme Terrain, *Proc. of IROS,* 2004.

[Chen and Allgöwer, 97]
Chen, H; Allgöwer, F., A Quasi-infinite Horizon Nonlinear Model Predictive Scheme for Constrained Nonlinear Systems, *In Proc. of the 1997 European Control Conference*, 1997.

[Choi and Yun, 99]
Choi, T., Yun, J., Accurate 3-D Shape Recovery using Curved Window Focus Measure, Image Processing, *ICIP'99, Proceedings International Conference on*, Vol. 3, Page(s): 910-914, 1999.

[Coue et al., 06]
Coue, C., Pradalier, C., Laugier, C., Fraichard, T., Bessiere, P., Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application, *The International Journal of Robotics Research,* January 2006.

[Cox, 94]
Cox, I. J., Modeling a Dynamic Environment Using a Bayesian Multiple Hypothesis Approach, *Artificial Intelligence*, vol. 66, no 2, pp. 331-344, Apr. 1994.

[Cumani et al., 04]
Cumani A., Denasi S., Guiducci A., Quaglia G. (2004). *Integrating Monocular Vision and Odometry for SLAM*. WSEAS Transactions on Computers, vol. 3, no. 3, pp. 625-630.

[Dellaert et al., 99]
Dellaert, F., Fox, D., Burgard, W., Thrun, S., Monte Carlo Localization for Mobile Robots, *Proc. IEEE Int'l Conf. Robotics and Automatization*, pp. 1322-28, May 1999.

[Dellart, 00]
Dellaert F., Seitz S. M., Thorpe C. E., Thrun S., Structure from Motion without Correspondence, *IEEE Conf. on CVPR*, June 2000.

[Dennis and Shnabel, 96]
Dennis, J. E., Shnabel, R. S., Numerical Methods for Unconstrained Optimization and Nonlinear Equations, *Society for Industrial and Applied Mathematics*, 1996.

[Denk and Schmidt, 03]
Denk, J., Schmidt, G., Syntesis of Walking Primitive Databases for Biped Robots in 3D-Environments, *Proc. IEEE Int'l Conf. on Robotics and Aut. (ICRA03)*, pp. 887-892.

[DeSouza and Kak, 02]
DeSouza,G. N., Kak, A. C., Vision for Mobile Robot Navigation: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, VOL. 24, NO. 2, Feb. 2002.

[Dev et al., 97]
Dev, A., Kröse, B., Groen, F., Navigation of a Mobile Robot on the Temporal Development of the Optic Flow, *Proc. IEEE Int'l Conf. Intelligent Robots and Systems*, pp. 558-563, Sept. 1997.

[Dhond et al., 89]
Dhond, U. R., Aggarwal, J. K., Estructure from Stereo- A Review, *IEEE Trans. On System, Man, and Cybernetics*, vol. 19, no. 6, November/December, 1989.

[Elfes, 89]
Elfes, A., Using occupancy grids for mobile robot perception and navigation, *IEEE Computer,* 22(6), pp. 46-57.

[Elsayed, 03]
Elsayed E. Hemayed, "A Survey of Camera Self-Calibration," *avss*, p. 351*, IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'03)*, 2003.

[Ens and Lawrence, 91]
Ens, J., Lawrence, P., 1991, A matrix based method for determining depth from focus, *In Proc. of IEEE Conf. On Computer Vision and Pattern Recognition,* pp. 600-609.

[Ens and Lawrence, 93]
Ens, J., Lawrence, P., An Investigation of Methods for Determining Depth from Focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 15, No 2, February 1993.

[Favaro and Soatto, 02]
Favaro, P., Soatto, S., Learning Shape from Defocus, *European Conference on Computer Vision*, EECCV 2002, pp. 735-742, Copenhaguen 2002.

[Fernandez and Casals, 97]
Fernandez, J., Casals, A., Autonomous Navigation in Ill-Structured Outdoor Environments, *Proc. IEEE Int'l Conf. Intelligents Robots and Systems*, pp. 395-400, Sept. 1997.

[Fleet and Jepson, 90]
D. J. Fleet, A. D. Jepson, *Computation of Component Image Velocity from Local Phase Information, Int.* J. Comp. Vision 5, pp. 74-104, 1990.

[Fox et al., 97]
Fox D., Burgard W., Thun S., The dynamic window approach to collision avoidance, *IEEE Robot. Autom*. Mag. Vol. 4, pp. 23-33, Mar. 1997.

[Garcia et al., 00]

Garcia, J., Sánchez, J. M., Orriols, X., and Binefa, X.,Sánchez, Chromatic Aberration and Depth Extraction, *Pattern Recognittion (ICPR00)*, Vol. 1, September, 2000, Barcelona.

[Gaussier et al., 97]

Gaussier, P., Joulain, C., Zrehen, Revel, A., Visual Navigation in a Open Environment without Map, *Proc. IEEE Int'l Conf. Intelligent Robots and Systems*, pp. 545-550, Sept. 1997.

[Ghosal and Vaneck, 96]

Ghosal S., Vaneck P.C., "Scalable algorithm for discontinuous optical flow estimation", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 18, pp. 181-194, Feb. 1996.

[Gluer and Schmidt, 00]

Gluer, D. Schmidt, G, A new approach for context based exception handling in autonomous mobile service robots, *Proc. IEEE International Conference on Robotics and Automation,*Vol. 4, pp.: 3272-3277, 2000

[Gonzalez and Woods, 02]

R. C. Gonzalez, R. E. Woods, Digital Image Processing, *Prentice Hall Int. Ed., Second Edition,* 2002.

[Gu and Hu, 02]

D. Gu, H. Hu, Neural predictive control for a car-like mobile robot, Robotics and Autonomous Systems 39 (2), 73-86, 2002.

[Gupta et al., 05]

Gupta G. S., Messom C. H., Demidenko S., Real-time identification and predictive control of fast mobile robots using global vision sensor, *IEEE Trans. On Instr. and Measurement,* Vol. 54, No. 1, Feb. 2005.

[Hager and Hutchinson, 94]

Hager, G., Hutchinson, S., Visual Servoing: Achivements, Aplications and Open Problems, *IEEE Int. Conf. Robotics and Automation*, workshop M-5, 1994.

[Hashima et al., 97]

Hashima, M., Hasegawa, F., Kanda, S., Maruyama, T., Uchiyama, T., Localization and Obstacle Detection for a Robot Carrying Food Trays, *Proc. IEEE Int'l Conf. Intelligent Robots and Systems,* pp. 345-351, Sept. 1997.

[Heeger, 88]

D. J. Heeger, Optical Flow using Spatiotemporal Filters, *Int. J. Computer Vision* 1, pp. 279-302, 1988.

[Hindman and Hauser, 92]

Hindman, R., Hauser, J., Maneuver modified trajectory tracking. *Proc. of MTNS'96, Inter. Sym. on the Mathematical Theory of Networks and Systems,* St Louis, MO, USA 1992).

[Hiura et al., 98]

Hiura, S., Matsuyama, T., Depth Measurement by the Multi-Focus Camera, *Computer Vision and Pattern Recognition, 1998, Proceedings, IEEE Computer Society Conference on,* Page(s): 953-959, 1998.

[Horn and Schunck, 81]
Horn, B. K. P., Schunck B. G., Determinig optical flow, *Artificial Intelligence*, Vol. 17, pp. 185-203, 1981.

[Horn, 89]
Horn, B. K. P., Brooks, M. J., Shape from Shading, *MIT Press,* Cambridge, Mass., 1989.

[Horn, 98]
Horn, B. K. P., Robot Vision, *McGraw –Hill Book Co*., Twelfth printing, 1998.

[Huber and Kortenkamp, 95]
Huber, E., Kortenkamp, D., Using Stereo Vision to Pursue Moving Agent with a Mobile Robot, *Proc.IEEE Int'l Conf. Robotics and Automation*, vol. 3, pp. 2340-2346, May 1995.

[Izquierdo et al., 04]
Izquierdo, J., Cifuentes, J., Arbuser, R., Pacheco, Ll., "The Project Prim, 18th *IAVE World Volunteer Conference Barcelona*, page 73, Aug. 2004.

[Jarvis, 83]
Jarvis, R. A., A perspective on range finding techniques for computer vision. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-5, No. 2, pp. 122-139, March 1983.

[Jones et al., 97]
Jones, S. D., Andresen, C., Crowley, J. L., Appearance Based Processes for Visual Navigation, *Proc. IEEE Int'l Conf. Intelligent Robots and Systems*, pp. 551-557, Sept. 1997.

[Joulian et al., 97]
Joulian, C., Gaussier, P., Revel, A., Gas, B., Learning to Build Visual Categories from Perception-Action Associations, *Proc. IEEE Int'l Conf. Intelligent Robots and Systems*, pp. 857-864, Sept. 1997.

[Kakuba and Arenas, 87]
Kakuba, M. R., Arenas, A. E., Position Verification of a Mobile Robot Using Standart Pattern, *IEEE J. Robotics and Automation,* vol. 3, no. 6, pp. 505-516, Dec. 1987.

[Kalman, 60]
Kalman, R. E., A new approach to linear filtering and prediction problems, *Trans. Of ASME, Journal of Basic Engineering*, Vol. 87, pp. 35-43, 1960.

[Kim and Nevatia 98]
Kim, D., Nevatia, R., Recognition and Localization of Generic Objects for Indoor Navigation Using Functionality, *Image and Vision Computing*, vol. 16, no. 11, pp. 729-743, Aug. 1998.

[Kim and Kim, 04]
Kim Y. G., Kim H., Layered Ground Floor Detection for Vision-based Mobile Robot Navigation. *Proc. IEEE ICRA*, pp. 13-18, 2004.

[Klancar and Skrjanc, 07]
Klancar, G., Skrjanc, I., Tracking-error model-based predictive control for mobile robots in real time, in: Robotics and Autonomous Systems 55, 460-469, 2007.

[Kosaka and Kak, 92]

Kosaka, A., and Kak, A. C., Fast Vision-Guided Mobile Robot Navigation Using Prediction of Uncertainties, *Computer Vision, Graphics, and Image Processing-Image Understanding*, vol. 56, no. 3, pp. 271-329, 1992.

[Krotkov, 86]

Krotkov, E. 1986. Focusing. MS-CIS-86-22. Grasp lab 63. Dept. of Computer and Information Science. Univesity of Pennsylvania.

[Kuipers and Byun, 91]

Kuipers, B. J., Byun, Y. T., A Robot Exploration and Mapping Strategy Based on a Semantic Hierachy of Spatial Representation, *J. Robotics and Autonomous Systems*, vol. 8, pp. 47-63, 1991.

[Küne et al., 05]

Küne F., Lages W., Da Silva J., Point stabilization of mobile robots with nonlinear model predictive control, *Proc. IEEE Int. Conf. On Mechatronic and Automation*, pp. 1163-1168, July 2005.

[Lenain et al., 05]

Lenain, R., Thuilot, B., Cariou, C., Martinet, P., Model Predictive Control for Vehicle Guidance in Presence of Sliding: Application to Farm Vehicles Path Tracking, in: International Conference of Robotics and Autonomous systems ICRA, Barcelona, Spain, vol. 2, pp. 885-890, 2005.

[Limon et al., 05]

Limon, D., Bravo, J. M., Alamo, T., Camacho, E., F., On the computation of invariant sets based on interval arithmetic, *IEE Proc. Control Theory and Applications*, Vol. 152, May 2005.

[Lju, 89]

Lju, L., System Identification: Theory for the User, *ed., Prentice Hall*, 1989.

[Lozano-Perez, 83]

Lozano-Perez, T., Spatial planning: A configuration space approach, *IEEE Trans. Computers*, Vol. C-32, pp. 108-120, 1983.

[Lucas and Kanade, 81]

Lucas B., Kanade T., An iterative image registration technique with application to stereo vision, *in Proc. Int. Joint Conf. Artificial Intelligence*, Aug. 1981, pp. 674-679.

[Luo, 01]

Luo, N., Advanced Control Techniques*, lecturer notes,* DEIA, University of Girona, 2001.

[Maciejowski, 02]

Maciejowski, J. M., Predictive Control with Constraints, *Ed. Prentice Hall*, 2002.

[M'Closkey and Murray, 97]

M'Closkey, R. T., R. M. Murray, R. M., Driftless Nonlinear Control Systems Using Homogeneus Feedback. (*IEEE Trans. On Aut. Control,* Vol. 42, No. 5, May 1997).

[Michalska and Mayne, 93]

Michalska, H., Mayne, D. Q., R, Robust receding horizon control of constrained nonlinear systems, *IEEE Trans. on Automatic Control*, Vol. 38, pp. 1623-1633, Nov. 1993.

[Miura and Shirai, 92]
Miura, J., Shirai Y., Hierachical Vision-Motion Planning with Uncertainty: Local Path Planning and Global Route Selection, *Proc. IEEE Int'l Conf. Robotics and Automation*, July 1992.

[Miura and Shirai, 93]
Miura, J., Shirai Y., An Uncertainty Model of Stereo Vision and Its Application to Vision-Motion Planning of Robots, *Proc. 13th Int'l Joint Conf. Artificial Intelligence*, Aug. 1993.

[Miura et al., 06]
J. Miura, J., Negishi, Y., Shirai, Y., Adaptive Robot Speed Control by Considering Map and Motion Uncertainty, in: Robotics and Autonomous Systems, 54 (2) (2006) 110-117, 2006.

[Moravec and Elfes, 85]
Moravec, H. P., Elfes, A., High Resolution Maps from Wide Angle Sonar, *Proc. IEEE Int'l Conf. Robotics and Automation*, pp, 116-121, 1985.

[Moravec, 88]
Moravec, H. P., Sensor fusion in certainty grids for mobile robots, *AI Magazine* 9(2): 61-74, 1988.

[Muñoz et al., 03]
Muñoz, X., Freixenet, J., Cufi, X., Martí, J, (2003), Active regions for colour texture segmentation integrating region and boundary information, *Proc, Inter. Conf. on Image Processing,* Vol. 3, pp. 453-456.

[Murray et al., 03]
Murray R. M., Aström K. J., Boyd S. P., Brockett R. W., Stein G., Future Directions in Control in an Information-Rich World*, IEEE Control Systems Magazine*, pp. 20-33, April 2003.

[Nakamura and Asada, 95]
Nakamura, T., Asada, M., Motion Sketch: Acquisition on Visual Motion Guided Behaviours, *Proc. 14th Int'l Joint Conf. Artificial Intelligence*, vol. 1, pp. 126-132, Aug. 1995.

[Nayar et al., 90]
Nayar, S. K., Ikeuchi, K., Kanade, T., Determinig Shape and Reflectance of Hybrid Surfaces by Photometric Sampling. *IEEE Transaction on Robotics and Automation*. VOL. 6, no 4. August 1990.

[Nayar, 92]
Nayar, S. K., Shape from Focus System for Rough Surfaces, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Champaign, Illinois, pp. 302-308, June 1992

[Nayar et al., 96]
Nayar, S. K., Watanabe, M., Noguchi, M., Real-Time Focus Range Sensor, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, NO. 12, December 1996.

[Negahdaripour, 02]
Negahdaripour, S., Optical Flow-Based 3-D Motion Estimation and Positioning, Navigation, Image Mosaiking, and Image Compression. *Lecture Notes on*, May 2002.

[Norton, 86]
Norton, J. P., An Introduction to Identification, *ed., Academic Press*, New York, 1986.

[Nourbakhsh et al., 97]
Nourbakhsh I. R., Andre D., Tomasi C., Genesereth M. R., Mobile Robot Obstacle Avoidance Via Depth From Focus, *Robotics and Autonomous Systems*, Vol. 22, pp. 151-58, June, 1997.

[Noborio and Schmidt, 96]
Noborio, H., Schmidt, G., Mobile Robot Navigation under Sensor and Localization Uncertainties, *Autonomous Mobile Systems*, pp.: 118 – 127, ISBN:3-540-61751-5, Publisher Springer-Verlag London, UK, 1996.

[Ogata, 93]
Ogata, K., Modern Control Engineering, *Prentice Hall Inter.*, 1993.

[Ogata, 96]
Ogata, K., Discrete Time Control Systems, *Prentice Hall Inter.*, Second Ed. 1996.

[Ögren and Leonard, 05]
Ögren P., Leonard N., A convergent dynamic window approach to obstacle avoidance, *IEEE Trans. Robotics*, Vol. 21, no 2, April 2005.

[Ollero and Amidi, 91]
Ollero, A., Amidi, O., Predictive Path Tracking of Mobile Robots. Application to the CMU Navlab, in: Proceedings of 5th International Conference on Advanced Robotics, Robots in Unstructured Environments, ICAR'91, vol. 2, pp. 1081-1086, 1991.

[Olson and Matties, 98]
Olson, C., Matthies, L., Maximum Likelihood Rover Localization by Matching Range Maps, *Proc. IEEE Int'l Conf. Robotics and Automation*, vol. 1, pp. 272-277, May 1998.

[Pacheco et al., 01]
Pacheco, LL., Cufi, X., Garcia, R., Autonomous Navigation of the Robot Mobile Roger Using Control Fuzzy, (writen in Spanish), *Seminar of Industrial Electronics, Automatics and Instrumentation*, Matanzas (Cuba), Sept. 2001.

[Pacheco et al., 04]
Pacheco, Ll., Cufi, X., Cifuentes, J., Calibration of a camera, the control of mobile robots using DFF techniques, *IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics*, Vol. 2, pp. 135-140, 2004.

[Pacheco and Luo, 06]
Pacheco, L., Luo, N., Mobile robot experimental modelling and control strategies using sensor fusion, in: Control Engineering and Applied Informatics 8 (3), 47-55, 2006.

[Pacheco, et al., 06-A]
Pacheco, Ll, Luo N., Arbusé, R., Experimental modelling and Control Strategies on an open mobile robot platform PRIM, *IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics*, Vol. 2, pp. 225-230, 2006.

[Pacheco, et al., 06-B]

Pacheco, Ll, Batlle, J., Cufi, X., Arbusé, R., PRIM an open mobile platform. Motivation, present and future trends, *IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics,* Vol. 2, pp. 231-236, 2006.

[Pacheco, et al., 06C]

Pacheco, Ll., Cufí, X., Batlle, J., PRIM: An Open Mobile Robot Platform, Recerca en automàtica, visió i robòtica, *Editorial: Barcelona Digital*, pp. 273-279, 2006.

[Pacheco, et al., 07]

Pacheco, Ll, Cufí, X., Cobos, J., Constrained Monocular Obstacle Perception with just one frame, *3rd Iberian Conference on Pattern Recognition and Image Analysis, IbPRIA'07*, Girona, Spain, June 2007. Published in Lecture Notes in Computer Science.

[Pacheco and Luo, 07A]

Pacheco, Ll, Luo, N., Predictive control by local visual data, *4th International Conference on Informatics in Control, Automation, and Robotics,* 9-12 May, 2007, Angers, France.

[Pacheco and Luo, 07B]

Pacheco, Ll, Luo, N., Mobile robot experimental modelling and control strategies using sensor fusion, *special issue of the journal "Control Engineering and Applied Informatics*, 2007.

[Pacheco and Luo, 07C]

Pacheco, Ll, Luo, N., Trajectory planning with a control horizon based on a narrow local grid perception: A practical approach for WMR with monocular visual data, 6th *International Workshop on Robot Motion and Control*, ROMOCO'07, Bukowy Dworek, Poland, 11-13 June, Published in Lecture Notes in Control and Information Sciences.

[Pacheco and Luo, 07D]

L. Pacheco, N. Luo, Mobile Robot Local Predictive Control Using a Visual Perception Horizon, Int. Journal of Factory Autom. Robotics and Soft Computing 1 (2), 73-81, 2007.

[Pacheco and Luo, 07D]

Pacheco, Ll, Luo, N., Mobile Robot Local Predictive Control Using a Visual Perception Horizon, *International society for advanced research, International Scientific book recent advances in control systems, robotics, and automation*.

[Pacheco et al., 08A]

Pacheco L., Luo N., Ferrer I., Cufí X., Control Education within a Multidisciplinary Summer Course on Applied Mobile Robotics, *Proc. 17th IFAC World Congress*, Seoul, Korea, pp. 11660-11665, 2008.

[Pan et al., 95]

Pan, J., Pack, D. J., Kosaka, A., Kack, A. C., FUZZY-NAV: A Vision-Based Robot Navigation Architecture Using Fuzzy Inference for Uncertainty-Reasoning, *Proc. IEEE World Congress Neuronal Networks*, vol. 2, pp. 602-607, july 1995.

[Pentland, 85]

Pentland, A. P. 1985. A new sense for depth of field. Proceedings of International Joint *Conference on Artificial Intelligence,* pp. 988-994.

[Pentland, 87]

Pentland, A. P., 1987, A New Sense for Depth of Field, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 523-531, July 1987.

[Pierce and Kuipers, 94]

Pierce, D., Kuipers, B., Learning to Explore and Build Maps, *Proc. 12th Nat'l Conf. Artificial Intelligence,* vol. 2, pp. 1264-1271, 1994.

[Pourboughrat and Karlsson, 02]

Pourboughrat, F., Karlsson, M. P., Adaptive control of dynamic mobile robots with nonholomic constraints, in: Computers and Electrical Engineering, 28, 241-253, 2002.

[Primbs et al., 99]

Primbs, J. A., Nevistic, V., Doyle, J. C., Nonlinear optimal control: A control Lyapunov function and receding horizon perspective, in: Asian J. Control, 2 (1), 14-24, 1999.

[Qin and Badgwell, 03]

Qin, S.J., Badgwell, T. A., *Control Engineering Practice*, pp. 733-764, 2003.

[Raimondi and Melluso, 05]

Raimondi, F. M., Melluso, M., A new fuzzy dynamics controller for autonomous vehicles with nonholonomic constraints, in: Robotics and Autonomous Systems 52, 115-131, 2005.

[Rajagolapan and Chaudhuri, 97]

Rajagopalan, A. N., Chaudhuri, S., A Variatonal Approach to Recovering Depth From Defocused Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 19, NO. 10, Page(s): 1158-1164. October 1997.

[Rajagolapan and Chaudhuri, 99]

Rajagopalan, A. N., Chaudhuri, S., An MRF Model-Based Approach to Simultaneous Recovery of Depth and Restauration from Defocused Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, NO. 7, July 1999.

[Rakovic et al., 03]

Rakovic, S. V., Kerrigan, E. C., Kouramas, K., Mayne, D. Q., Approximation of the minimal Robustly Positively Invariant Set of Discrete-time LTI Systems with Persistent State Disturbances, *IEEE Conf. on Decision and Control*, Vol. 4, pp. 3917-3918, 2003.

[Rao et al., 03]

Rao, C.V., Rawling, J. B., Mayne, D. Q., Constrained State Estimation for Nonlinear Discrete-time Systems: Stability and Moving Horizon Aproximations, *IEEE Trans. on Automatic Control,* Vol. 48, pp. 246-258, Feb. 2003.

[Rawling and Muske, 93]

Rawling, J. B., Muske, K. R., The Stability of Constrained Receding Horizon Control, *IEEE Trans. On Automatic Control*, Vol. 38, pp. 1512-1516, 1993.

[Reeds and Shepp, 90]

Reeds, J. A., Shepp, L. A., Optimal paths for a car that goes forwards and backwards, *Pacific Journal of Mathematics*, vol. 145, 1990.

[Ribes and Borrelly, 97]

Ribes, P., Borrelly, J., Underwater Pipe Inspection Task Using Visual Servoing Techniques, *Proc. IEEE Int'l Conf. Intelligent Robots and Systems*, pp. 63-68, Sept. 1997.

[Richalet et al., 76]

Richalet, J., Rault, A., Testud, J. L., Papon, J., Algorithmic control of industrial processes, *In Proceedings of the 4th IFAC symposium on identification and system parameter estimation*, pp. 1119-1167, 1976.

[Ridao et al., 98]

Ridao, P., Pacheco, Ll., Cufi, X., Forest, J., Sensorial and Navigation System for a Mobile Robot (ROGER), *Intelligent Autonomous Vehicles IFAC'98*, Madrid, Mars 1998.

[Rimon and Koditschek, 92]

Rimon E., Koditschek D., Exact robot navigation using artificial potential functions, *IEEE Trans. Robot Autom.*, vol. 8, no5, pp. 501-518, Oct. 1992.

[Saleh and Subotic, 85]

Saleh, B. E. A., Subotic, N. S., Time variant Filtering of Signals in the Mixed Time-Frequency Domain, *IEEE Trans. ASSP* vol. 33, pp. 1479-1485, Dec. 1985.

[Salvi et al., 96]

Salvi, J., Pacheco, Ll., Garcia, R., ROGER: A mobile robot for research and experimentation, *ISRAM 96*, pp. 745-750, 1996.

[Santos-Victor et al., 93]

Santos-Victor, J., Sandini, G., Curotto, F., Garibaldi, S., Divergent Stereo for Robot Navigation: Learning from Bees, *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 1993.

[Sarkar et al., 94]

Sarkar, N., Yun, X., Kumar, V., Control of mechanical systems with rolling constraints: Application to dynamic control of a mobile robot, in: The International Journal of Robotic Research 13 (1), 55-69, 1994.

[Schäfer et al., 07]

Schäfer H., Proetzsch M., Berns K., Obstacle Detection in Mobile Outdoor Robots, *Proc. Inter. Conf. on Informatics in Control, Autom. and Robotics*, pp. 141-148, 2007.

[Schechner and Kiryati, 98]

Schechner, Y. Y., Kiryati, N., Depth from Defocus vs. Stereo: How Different Really are They?, *Proceedings Fourteeenth International Conference on Pattern Recognition*, vol. 2, 1998.

[Schilling, 90]

Schilling, R. J., Fundamental of Robotics*, Prentice-Hall International*, 1990.

[Simmons and Koenig, 95]

Simmons, R., Koenig, S., Probabilistic Robot Navigation in Partially Observable Environments, *Proc. Int'l Joint Conf. Artificial Intelligence,* pp. 1080-1087, Aug. 1995.

[Sørdalen and Canudas de Wit, 93]

Sørdalen, O. J., Canudas de Wit, C., Exponential Stabilization of Mobile Robots: Extension to Path Following, *IEEE Trans. On Robotics and Automation*, Vol. 9, NO. 6, December 1993.

[Subbarao, 87]

Subbarao, M. Direct Recovery of Depth-map I: Differential methods. *Proceedings of the IEEE Computer Society Workshop on Computer Vision*. Pp. 58-65, Miami Beach, Florida, Nov 1987.

[Subbarao, 91]

Subbarao, M., Spatial- Domain Convolution/Deconvolution Transform, *Tech. Report* No. 91.07.03, Stony Brook, New York, 1991.

[Subbarao et al., 92]

Subbarao, M., Choi, T., Nikzad, A., Focusing Techniques, *Tech. Report 92.09.04*, Stony Brook, NewYork 1992.

[Subbarao, 93]

Subbarao, M, Automatic Ranging and Automatic Focusing, *U.S.A. Patent No.* 5231443, July 1993.

[Subbarao, 94]

Subbarao, M., Continuous Focusing of Moving Objects Using DFD1F, *Proceedings of SPIE IS Symposium on Electronic Imaging Science and Technology*, Feb. 1994, San Francisco.

[Subbarao and Choi, 95]

Subbarao, M., Choi, T., Accurate Recovery of Three-Dimensional Shape from Image Focus, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 3, March 1995.

[Sugihara, 88]

Sugihara, K., Some Location Problems for Robot Navigation Using a Single Camera, *Computer, Vision, Graphics, and Image Processing*, vol. 42, pp. 112-129, 1988.

[Surya, 94]

Surya, G., Three Dimensional Scene Recovery from Image Defocus, *PHD thesis*, Stony Brook, New York, december 1994.

[Teel et al., 95]

Teel, A. R., Murray, R. M., Walsh, G. C., Nonholonomic control systems: from steering to stabilization with sinusoids, *Int. J. Contr.*, vol. 62, no. 4, pp. 849-870, 1995

[Tsumura, 86]

Tsumura, T., Survey of Automated Guided Vehicles in Japanese Factory, Proc. *Int'l Conf. Robotics and Automation*, pp. 1329- 1334, Apr. 1986.

[Thrun, 98]

Thrun, S., Learning Metric-Topological Maps for Indoor Robot Mobile Navigation, *Artificial Intelligence*, vol. 99, no. 1, pp. 21-71, Feb. 1998.

[Thrun, 00]

Thrun, S., Probabilistic Algorithms in Robotics, *Technical Report CMU-CS*-00-126, Carnegie Mellon Univ., 2000.

[Thrun, 02]

Thrun S. (2002*). Robotic mapping: a survey. Exploring Artificial Intelligence in the New Millennium, Morgan Kaufmann*, San Mateo.

[Unnikrishnan, 02]

Unnikrishnan R., Kelly A., Mosaicing large Cyclic Environments for Visual Navigation in Autonomous Vehicles, *IEEE ICRA*, Vol. 4, pp. 4299-4306, May 2002.

[Van Overschee and Moor, 96]

Van Overschee, P., Moor, B., Subspace Identification for Linear Systems: Theory, Implementation, *Appl., ed., Kluwer Academic*, 1996.

[Vehí, et al, 01]

Vehí, J., Luo, N., Rodellar, J., Armengol J., Digital control via interval analysis, *Nonlinear Analysis: Theory, Methods & Applications*, 47, 203-212, 2001.

[Verri and Poggio, 87]

Verri A., Poggio T., Against Quantitative Optical Flow, *Proc. IEEE ICCV*, London, pp. 171-180, 1987.

[Wan, 07]

Wan, J. (2007). *Computational reliable approaches of contractive MPC for discrete-time systems.* PhD Thesis, University of Girona.

[Wang and Allgower, 05]

Wang, G. L., Allgower, F., Flatness-based Optimal non Causal Output Transitions for Constrained Nonlinear Systems: Case Study on an Isothermal Continuously Stirred Tank Reactor, *IEEE Proc. Control Theory and Applications*, Vol. 152, pp. 105-112, 2005.

[Wang et al., 05]

Wang, Y., Bahrami, S., Zhu, S., Perceptual scale space and its applications, *Proc. of IEEE Conf. on Computer Vision, ICCV05*, Vol. 1, pp. 58-65, 2005.

[Watanabe and Nayar, 96]

Watanabe, M., Nayar, S. K., Telecentric Optics for Computacional Vision, *Proc.Of European Conference on Computer Vision (ECCV'96)*, April 1996.

[Watanabe and Nayar, 98]

Watanabe, M., Nayar, S. K., Rational Filters for Passive Depth from Defocus, *International Journal of Computer Vision, Kluwer Academic Publishers*, Vol. 27 (3), pp: 203-225, 1998.

[Wilcox et al., 92]

Wilcox, B., Matthies, L., Gennery, D., Copper, B., Nguyen, T., Litwin, T., Mishkin, A., Stone, H., Robotic Vehicles for Planetary Exploration, *Proc. IEEE Int'l Conf. Robotics and Automations,* pp. 175-180, May 1992.

[Wilson and Shafer, 94]

Wilson, R. G., Shafer, S., Modeling and Calibration of Automated Zoom Lenses, *Technical Report CMU-RI-TR-94-03*, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, Pa., Jan. 1994.

[Woodham et al., 80]

Woodham, J. R., Photometric Method for Determining Surface Orientation from Multiple Images*, Optical Engineering*, pages 139-144, 1980.

[Xiong and Shafer, 94]
Xiong, Y., Shafer, S. A., Variable Window Gabor Filters and Their Use in Focus and Correspondence, *CMU-RI-TR-94-06, Technical Report*, Robotics Institute, Carnegie Mellon University, Pittsburg (Pennsylvania), March, 1994.

[Yagi et al., 94]
Yagi, Y., Kawato, S., Tsuji, Real-Time Omnidirectional Image Sensor (COPIS) for Vision-Guided Navigation, *IEEE Trans. Robotics and Automation*, vol. 10, no. 1, pp. 11-22, Feb. 1994.

[Yamauchi and Beer, 96]
Yamauchi, B., Beer, R., Spatial Learning for Navigation in Dynamic Environments, *IEEE Trans. System, Man, and Cybernetics*, Part B, vol. 26, no. 3, pp. 496-505, June 1996.

[Zheng et al., 91]
Zheng, J. Y., Barth, M., Tsuji, S., Autonomous Landmark Selection for Route Recognition by a Mobile Robot, *Proc. IEEE Int'l Conf. Robotics and Automation*, pp. 2004-2009, 1991.

[Zimmer, 94]
Zimmer, G., State Observation by on-line minimization, *Int. J. Control,* No. 60, pp. 595-606, 1994.