TÉCNICO LISBOA

UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

Design and Experimental Validation of a PCA-based 2D Localization System for Mobile Robots in Unstructured Environments

Fernando Paulo Neves da Fonseca Cardoso Carreira

Supervisor: Doctor Carlos Baptista Cardeira
Co-Supervisor: Doctor João Manuel Ferreira Calado

Thesis approved in public session to obtain the PhD Degree in
Mechanical Engineering

Jury final classification: Pass with distinction

2018

TÉCNICO
LISBOA

UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

Design and Experimental Validation of a PCA-based 2D Localization System for Mobile Robots in Unstructured Environments

Fernando Paulo Neves da Fonseca Cardoso Carreira

Supervisor: Doctor Carlos Baptista Cardeira
Co-Supervisor: Doctor João Manuel Ferreira Calado

Thesis approved in public session to obtain the PhD Degree in
Mechanical Engineering

Jury final classification: Pass with distinction

Jury
Chairperson: Doctor João Miguel da Costa Sousa
Instituto Superior Técnico, Universidade de Lisboa
Members of the Committee:
Doctor António Paulo Gomes Mendes Moreira
Faculdade de Engenharia da Universidade do Porto
Doctor Paulo Jorge Coelho Ramalho Oliveira
Instituto Superior Técnico da Universidade de Lisboa
Doctor Vítor Manuel Ferreira dos Santos
Universidade de Aveiro
Doctor Carlos Baptista Cardeira
Instituto Superior Técnico da Universidade de Lisboa
(Orientador Científico)

2018

Abstract

This work proposes and experimentally validates an indoor mobile robot localization system, to operate in unstructured indoor environments, using only on-board sensors. The proposed approach is a self-localization system architecture composed by three modules: i) an attitude estimation filter, based on compass and odometry data; ii) a 2D non-linear positioning system based on ceiling vision; iii) a 2D position estimator to merge the absolute non-linear positioning system and odometry, through a linear filter.

The positioning system resorts to Principal Component Analysis (PCA) of grayscale or depth images, acquired by a camera looking upwards to the ceiling, where no beacons or special features are present. The use of the PCA has the advantage of avoiding the need of extraction features and obtain the global position with low complexity algorithms. During operation, the principal components of the acquired images are compared with previously registered images present in a reduced dimension on-board image database.

A Linear Parameter Varying (LPV) model for differential drive mobile robot position kinematics is derived without the need of local linearization. This LPV allows the use of a globally stable estimator, avoiding the drawbacks associated to Extended Kalman Filters (EKF). The 2D linear position and velocity sub-optimal estimations are provided by a Kalman Filter (KF), for the LPV model, thus guaranteeing stability. The optimal estimates of mobile robot attitude are also provided by a KF.

To improve the PCA-based positioning system for varying illumination conditions, the algorithm was extended to the use of depth images, captured by a structured light 3D scanner. This extension allows the system to deal with missing data present in the depth images. The algorithm shows to be robust even if parts of data are missing.

To validate the proposed algorithms, a set of tests was carried out in a real environment, using a mobile robot with differential drive kinematics, where: i) the global stability is illustrated based on multiple re-localization experiments; ii) the PCA-based positioning system is validated; iii) the angular and linear slippages estimation are shown to be effective under real operation; iv) the images reconstruction are shown and the localization robustness is validated even if parts of data are missing.

The self-localization system is integrated in a navigation system to perform autonomous motion toward to goals positions. The motion control is ensured by a Dipolar Navigation Function (DNF) applied along a roadmap, based on a Generalized Voronoi Diagram (GVD).

With the purpose of extending the attitude estimator, to implement the sensor fusion with more motion sensors, a new class of estimators was developed based on Complementary Filter (CF) design. The proposed method addresses the design to combine the measurements provided from several sensors according to the frequency bands of each one. The approach was applied to tackle a generic three sensors fusion problem, allowing the design of mobile robot attitude estimators to merge the data from a rate gyroscope, a digital compass, and odometry.

Keywords: Mobile robots, self-localization, principal component analysis, Kalman filters, complementary filters.

Resumo

Este trabalho propõe e valida experimentalmente um sistema de localização de robôs móveis que navegam em ambientes interiores não estruturados, utilizando apenas sensores de bordo. A abordagem consiste num sistema de auto-localização composto por três módulos: i) um filtro para estimação de atitude, utilizando os dados de uma bússola digital e da odometria, ii) um sensor de posicionamento não-linear 2D baseado na visão do teto; iii) um estimador de posição 2D para fundir o sistema de posicionamento absoluto não-linear com a odometria, através de um filtro linear.

O sistema de posicionamento recorre à análise de componentes principais (PCA) de imagens em escala de cinzentos ou de profundidade, adquiridas por uma câmara a apontar para o teto, sem balizas ou caraterísticas locais especiais. A utilização da PCA tem a vantagem de evitar a necessidade de extração de caraterísticas locais e obter a posição global com algoritmos de baixa complexidade. Durante o funcionamento, os componentes principais das imagens adquiridas são comparados com os das previamente registadas numa base de dados de imagens de dimensão reduzida.

Neste sistema é utilizado um modelo linear de parâmetro variável (LPV) para a cinemática da posição do robô móvel de tração diferencial, sem linearização local. Este LPV permite o uso de um estimador globalmente estável, evitando os inconvenientes associados aos filtros de Kalman estendidos (EKF). As estimativas subótimas da posição linear 2D e da velocidade são fornecidas por um filtro de Kalman (KF) para o modelo LPV. As estimativas ótimas de atitude e deslizamento angular também são fornecidas por um KF.

Para melhorar o sistema de posicionamento baseado em PCA em diferentes condições de iluminação é estendido o algoritmo para utilizar imagens de profundidade capturadas por um sensor 3D baseado em luz estruturada. O método apresenta robustez com imagens de profundidade corrompidas.

Para se validar os algoritmos propostos são realizados testes num ambiente real, utilizando um robô móvel com cinemática diferencial, onde: i) a estabilidade global é apresentada com base em múltiplas experiências de relocalização; ii) o sistema de posicionamento baseado em PCA é validado; iii) a estimativa de deslizamentos angulares e lineares mostra-se eficaz sob operação real; iv) a reconstrução de imagens é apresentada e a robustez de localização é validada, considerando a adição de dados corrompidos aleatórios.

O sistema de auto-localização é integrado num sistema de navegação para desenvolver missões. O controlo de movimento é assegurado por uma função de navegação dipolar (DNF) aplicada ao longo de um diagrama de Voronoi generalizado (GVD).

Com a finalidade de estender o estimador de atitude para implementar a fusão de mais sensores de movimento é desenvolvida uma nova classe de estimadores com base em filtros complementares (CF) e aborda-se a forma de combinar as medidas fornecidas por diversos sensores, de acordo com as faixas de frequência utilizadas por cada um deles. A abordagem é aplicada para um problema genérico de fusão de três sensores e são projetados vários estimadores de atitude de robôs móveis, fundindo os dados de um giroscópio, uma bússola digital e a odometria.

Palavras-chave: Robôs móveis, auto-localização, análise de componentes principais, filtros de Kalman, filtros complementares.

Acknowledgements

The realization of a PhD is always compared to a long trip, with moments of enthusiasm and motivation, and others of disturbance or dismay. Fortunately, during this work several people crossed with my way, and, directly or indirectly contribute to this journey, people how I am very gratefull.

First of all, to my advisers, Professors Carlos Cardeira and João Calado for the orientations, scientific support, suggestions, corrections and encouragement words that were given throughout this research project.

Special thanks go to Professor Paulo Oliveira, for all the knowledge, scientific support, suggestions and criticisms, as well as encouragement words throughout this work.

I would also like to thank the remaining members of the Technical Monitoring Committee, Professors Vítor Santos and João Sousa for their suggestions, which were very usefull in the preparation of this document.

To Professors José Sá da Costa, Jorge Martins, Paulo Oliveira and Pedro Lima, through the knowledge transmitted in the classes of the PhD curricular units.

To Engineer Camilo Christo, for collaboration in the results validation tests and technical support realized in the robots electronic components.

To Mr. Luís Raposeiro for the prototypes structures construction used in the experimental tests.

To master students João Rodrigues and João Silva, I thank the collaboration in the space mapping tasks where tests were carried out and the measurements that allowed to validate the implemented algorithms.

I would also like to thank Duarte Valério and Tiago Silva for the transmitted knowledge to write technical documents using LaTex, from which was written all the publications and technical reports along the work, including this thesis.

Special thanks go to Rui Vaz and César Simões for the correctness of the texts in English, and suggestions for their improvement.

I am gratefull to my colleague Francisco Campos for the support, suggestions and corrections of some texts.

My colleagues, Mário Mendes, João Sabino and Cláudia Casaca, for their constant support and encouragement.

I also thank my family, especially my parents, sister, parents in-laws, brothers-in-law and some friends who, in one way or another, were helped me to carry out this work.

Finally, a special thanks to my wife Carla Carreira, for the infinite support, patience, understanding, encouragement and dedication in the difficult and absence hours, over this years.

To my dear son Francisco Carreira, thank you for the wonderful smiles that have given me an increased motivation in the last months of work.

To all, my gratefullness!

Agradecimentos

A realização de uma tese de doutoramento é sempre comparada a uma longa viagem, com momentos de entusiasmo e motivação, e outros de perturbações ou desânimo. Felizmente, durante a realização deste trabalho diversas pessoas cruzaram o meu caminho, que de forma mais direta ou indireta contribuíram para a sua realização, e às quais estou deveras agradecido.

Em primeiro lugar aos meus orientadores, Professores Carlos Cardeira e João Calado pelas orientações, apoio científico, sugestões, correções e palavras de incentivo que foram dadas ao longo deste projecto de investigação.

Um agradecimento especial ao Professor Paulo Oliveira, por todo o saber, apoio científico, sugestões e críticas, assim como as palavras de incentivo transmitidas ao longo da realização do trabalho.

Gostaria também de agradecer aos restantes membros da Comissão de Acompanhamento Técnico, Professores Vítor Santos e João Sousa pelas sugestões transmitidas, que foram muito úteis na elaboração deste documento.

Aos Professores José Sá da Costa, Jorge Martins, Paulo Oliveira e Pedro Lima pelo conhecimento transmitido nas aulas das unidades curriculares do doutoramento.

Ao Engenheiro Camilo Christo, pela colaboração nos primeiros testes de validação de resultados e no apoio técnico realizado aos componentes eletrónicos dos robôs.

Ao Sr. Luís Raposeiro pela construção da estrutura dos protótipos utilizados nos ensaios experimentais.

Aos alunos de mestrado João Rodrigues e João Silva, agradeço a colaboração no mapeamento do espaço onde se realizaram os ensaios e nas medições que permitiram validar os estimadores.

Gostaria também de agradecer ao Duarte Valério e Tiago Silva pelo conhecimento transmitido na escrita de documentos técnicos utilizando LaTex, a partir do qual foi possível escrever todas as publicações e relatórios realizados ao longo do trabalho, incluindo esta dissertação.

Um obrigado especial ao Rui Vaz e ao César Simões pela correcção de alguns textos em inglês, e sugestões de melhoria para os mesmos.

Um agradecimento ao meu colega Francisco Campos pelo apoio, sugestões e correções de alguns textos.

Aos meus colegas de trabalho Mário Mendes, João Sabino e Cláudia Casaca pelo constante apoio e encorajamento.

Agradeço também à minha família, especialmente aos meus pais, irmã, sogros, cunhados e a alguns amigos que, de uma forma ou de outra, foram contribuindo para a motivação na realização do trabalho.

Finalmente, um especial agradecimento à minha esposa Carla Carreira, pelo infinito apoio, paciência, compreensão, encorajamento e dedicação nas horas difíceis e de ausência, ao longo destes anos.

Ao meu querido filho Francisco Carreira, agradeço os maravilhosos sorrisos que me transmitiram uma motivação acrescida nos últimos meses de trabalho.

A todos, os meus agradecimentos!

Table of contents

List of figures

List of tables

Publications

The research work presented in this document led to the following scientific publications

International journal

- Carreira, F., Calado, J.M.F., Cardeira, C., Oliveira, P., 2015. Enhanced PCA-based localization using depth maps with missing data. Journal of Intelligent & Robotic Systems 77, 341-360. DOI: 10.1007/s10846-013-0013-6.

National journal

- Carreira, F., Christo, C., Valério, D., Ramalho, M., Cardeira, C., Calado, J.M.F., Oliveira, P., 2012. Experimental Validation of a PCA-Based Localization System for Mobile Robots in Unstructured Environments, in Revista Robótica, n. 88, 2012, pp. 12-17 (selected from Proceedings of Robótica 2012: 12th International Conference on Autonomous Robot Systems and Competitions, Guimarães, April 11, 2012).

International conferences

- Carreira, F., Calado, J.M.F., Cardeira, C., 2011. A mobile robot navigation planning in a human populated environment, in: Proceedings of Robótica 2011, the 11th International Conference on Autonomous Robot Systems and Competitions, Lisboa, Portugal. pp. 15-20. URL: http://www.dem.ist.utl.pt/cardeira/papers/Robotica2011Proceedings_Fcarreira.pdf.

- Carreira, F., Christo, C., Valério, D., Ramalho, M., Cardeira, C., Calado, J.M.F., Oliveira, P., 2012. Experimental validation of a PCA-based localization system for mobile robots in unstructured environments, in: Proceedings of Robótica 2012, the 12th International Conference on Autonomous Robot Systems and Competitions, Guimarães, Portugal. pp. 69-74. URL: http://www1.dem.ist.utl.pt/cardeira/Mesh/Ca12a.pdf

- Carreira, F., Christo, C., Valério, D., Ramalho, M., Cardeira, C., Calado, J.M.F., Oliveira, P., 2012. 2D PCA-based localization for mobile robots in unstructured environments, in: Proceedings of IROS 2012, the IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Vilamoura, Portugal. pp. 3767-3868. DOI: 10.1109/IROS.2012.6386272.

- Carreira, F., Calado, J.M.F., Cardeira, C., Oliveira, P., 2013. Enhanced PCA-based localization using depth maps with missing data, in: Proceedings of ICARSC 2013, the 13th International Conference on Autonomous Robot Systems and Competitions, Lisboa, Portugal. pp. 56-63. DOI: 978-989-97531-2-9.

- Rodrigues, J., Carreira, F., Calado, J.M.F., Cardeira, C., Oliveira, P., 2013. A bayesian grid method PCA-based for mobile robots localization in unstructured environments, in: Proceedings of ICAR 2013, the 16th International Conference on Advanced Robotics, Montevideo, Uruguay. DOI: 10.1109/ICAR.2013.6766487.

- Rodrigues, J., Cardeira, C., Carreira, F., Calado, J.M.F., Oliveira, P., 2013. Experimental validation of a visual odometry system for indoor unstructured environments. In: Proceedings of ICAR 2013, the 16th International Conference on Advanced Robotics. IEEE, Montevideo, Uruguay. DOI: 10.1109/ICAR.2013.6766496

- Carreira, F., Calado, J.M.F., Cardeira, C., Oliveira, P., 2015. Complementary filter design with three frequency bands: Robot attitude estimation, in: Proceedings of ICARSC 2015, the 15th IEEE International Conference on Autonomous Robot Systems and Competitions, IEEE, Vila Real, Portugal. pp. 168-173. DOI: 10.1109/ICARSC.2015.33.

- Carreira, F., Calado, J.M.F., Cardeira, C., Oliveira, P., 2018. Navigation System for Mobile Robots Using PCA-based Localization System from Ceiling Depth Images: Experimental Validation, in: Proceedings of Controlo 2018, the 13th APCA

International Conference on Control and Soft Computing, IEEE, Ponta Delgada, S. Miguel Island, Azores, Portugal. (accepted for publication).

Technical report

- Carreira, F., Christo, C., Valério, D., Ramalho, M., Cardeira, C., Calado, J.M.F., Oliveira, P., June 2012. Experimental validation of a PCA-based localization system for mobile robots in unstructured environments. Tech. rep., IDMEC/CSI. URL: http://www1.dem.ist.utl.pt/cardeira/Mesh/Ca12b.pdf.

Acronyms and abbreviations

APF	**A**rtificial **P**otential **F**ields
BPF	**B**and-**P**ass **F**ilter
CF	**C**omplementary **F**ilter
CF$^{\mathbf{3}}$	**C**omplementary **F**ilter with **3** inputs
CF$^{\mathbf{m}}$	**C**omplementary **F**ilter with **m**ultiple inputs
CKF$^{\mathbf{3}}$	**C**omplementary **K**alman **F**ilter with **3** inputs
DCKF$^{\mathbf{3}}$	**D**iscrete-time **C**omplementary **K**alman **F**ilter with **3** inputs
DKF$^{\mathbf{3}}$	**D**iscrete-time **K**alman **F**ilter with **3** inputs
DNF	**D**ipolar **N**avigation **F**unction
EKF	**E**xtended **K**alman **F**ilter
GVD	**G**eneralized **V**oronoi **D**iagram
HPF	**H**igh-**P**ass **F**ilter
HVAC	**H**eating, **V**entilation and **A**ir **C**onditioning
HOG	**H**istograms **O**riented **G**radient
IR	**I**nfra**R**ed
LPF	**L**ow-**P**ass **F**ilter
LT	**L**aplace **T**ransform
LPV	**L**inear **P**arameter **V**arying
KF	**K**alman **F**ilter
KF$^{\mathbf{3}}$	**K**alman **F**ilter with **3** inputs
KL	**K**arhunen–**L**oéve
MEMS	**M**icro-**E**lectro-**M**echanical **S**ystems
MMSE	**M**inimum **M**ean **S**quare **E**rror
MSE	**M**ean **S**quare **E**rror

MCL	**M**onte **C**arlo **L**ocalization
ML	**M**arkov **L**ocalization
NF	**N**avigation **F**unction
PCA	**P**rincipal **C**omponent **A**nalysis
RANSAC	**RAN**dom **SA**mple **C**onsensus
RGB	**R**ed-**G**reen-**B**lue
RGB-D	**R**ed-**G**reen-**B**lue and **D**epth
SIFT	**S**cale **I**nvariant **F**eactures **T**ransform
SLAM	**S**imultaneous **L**ocalization **A**nd **M**apping
SURF	**S**peeded **U**p **R**obust **F**eatures
TF	**T**ransfer **F**unction
UAV	**U**nmanned **A**erial **V**ehicle
USB	**U**niversal **S**erial **B**us
UV	**U**nderwater **V**ehicle
VGA	**V**ideo **G**raphics **A**rray
ZOH	**Z**ero **O**rder **H**old

Notation

Optimal attitude and slippage estimation

$\mathbf{B}$	input matrix
$\mathbf{F}$	transition matrix
$\mathbf{H}$	observation matrix
k	discrete instants of time
K	Kalman gains
$\mathbf{K}$	Kalman gain matrix
l	distance between wheels
r	wheels radius
s_a	angular slippage
T	discrete sampling time
$\mathbf{x}$	states vector
$\hat{\mathbf{x}}$	estimated states vector
y	observation output
z	discrete Laplace operator
α_r	right wheel accumulated angle
α_l	left wheel accumulated angle
ψ	mobile robot attitude
$\hat{\psi}$	estimated attitude
ψ_c	measured attitude on the compass
ψ_o	measured attitude from the encoders' odometry
ω	mobile robot angular speed
$\hat{\omega}$	estimated angular speed

ω_o	odometry angular speed
μ	process noise
γ	observation noise
σ	noise covariance

Complementary filters for attitude estimation

a,b	filters transfer function parameters
b	bias existing in the rate gyro measurements
e	angular error error caused by the odometry slippage
f_c	1^{st} order filters cut-off frequency
$\mathbf{B}_{\mathbf{CF^3}}$	CF^3 input matrix
$\mathbf{B}_{\mathbf{CKF^3}}$	CKF^3 input matrix
$\mathbf{B}_{\mathbf{KF^3}}$	KF^3 input matrix
$\mathbf{D}_{\mathbf{KF^3}}$	KF^3 feedforward matrix
$F_i(s)$	transfer function relating each sensor with the corresponding filter output
$\mathbf{F}_{\mathbf{CF^3}}$	CF^3 transition matrix
$\mathbf{F}_{\mathbf{CKF^3}}$	CKF^3 transition matrix
$\mathbf{F}_{\mathbf{KF^3}}$	KF^3 transition matrix
$\mathbf{H}_{\mathbf{CF^3}}$	CF^3 observation matrix
$\mathbf{H}_{\mathbf{CKF^3}}$	CKF^3 observation matrix
$\mathbf{H}_{\mathbf{KF^3}}$	KF^3 observation matrix
K	filters gains matrix
i	index of each complementary filter that composes the estimator
k	discrete instants of time
$\mathbf{K}_{\mathbf{CF^3}}$	CF^3 gain matrix
$\mathbf{K}_{\mathbf{CKF^3}}$	CKF^3 Kalman gain matrix
$\mathbf{K}_{\mathbf{KF^3}}$	KF^3 Kalman gain matrix
m	number of sensors and corresponding complementary filter inputs
n	filters order number
p_{T_i}	poles of the i filter transfer function
$\mathbf{P}$	estimation error covariance matrix

$\mathbf{Q}$	covariance noise associated with high-pass (process noise)
$\mathbf{R}$	covariance noise associated with low-pass and band-pass filters
s	Laplace transform operator
t	time
T	discrete sampling time
$T_i(s)$	transfer function relates $\psi(s)$ with $\hat{\psi}(s)$, in each filter
$\mathbf{x}$	states vector
$\hat{\mathbf{x}}$	estimated states vector
z_{T_i}	zeros of the i filter transfer function
z	discrete Laplace operator
$\mathbf{z}$	output vector (measurement noise)
ψ	measured position: mobile robot attitude
$\dot{\psi}$	measured velocity: mobile robot angular velocity
$\ddot{\psi}$	measured acceleration: mobile robot angular acceleration
$\hat{\psi}$	estimated position: mobile robot attitude estimated
$\dot{\hat{\psi}}$	estimated velocity: mobile robot angular velocity
$\ddot{\hat{\psi}}$	estimated acceleration: mobile robot angular acceleration
ψ_c	angular measurement obtained from a digital compass
ψ_o	angular measurement obtained from odometry
$\psi(s)$	Laplace transformation of $\psi(t)$
$\omega(s)$	Laplace transformation of $\dot{\psi}(t)$
$\alpha(s)$	Laplace transformation of $\ddot{\psi}(t)$
$\hat{\psi}(s)$	Laplace transformation of $\hat{\psi}(t)$
ω_r	angular velocity measurement obtained from a rate gyro
τ	1$^{\text{st}}$ order filters time constant

PCA-based positioning system

c, C	non-corrupted signals auxiliary counters
f	observation noise coefficient
i	signals index
$\mathbf{l}$	missing data indicator vector

$\mathbf{m_x}$	PCA database ensemble mean
M	captured images number to the PCA database
n	selected eigenvalues first number
N	stored signals dimension on the PCA database
N_x	image width
N_y	image height
r_{PCA}	PCA residual
$\mathbf{R_{xx}}$	PCA error covariance matrix
$\mathbf{R}$	PCA-based positioning system covariance (observation noise)
t_k	time instants
$\mathbf{x}$	stochastic signal (image data)
$\mathbf{x_i}$	vector with signals extracted from each image
$\mathbf{u_j}$	eigenvector of each component
$\mathbf{U}$	eigenvectors matrix
$\mathbf{U_n}$	eigenvectors matrix of the first n components
$\mathbf{v}$	projection of $\mathbf{x}$ in the eigenspace
$\mathbf{v_i}$	eigenvector of each image stored in the PCA database
x_i,y_i	x,y coordinate of each image stored in the PCA database
x_{PCA},y_{PCA}	x,y coordinate obtained from the PCA-based positioning system
λ_j	eigenvalue of each component

2D position estimator

$\mathbf{F}(\omega)$	transition matrix, for a parameter ω
$\mathbf{G}(\omega)$	stochastic noise matrix, for a parameter ω
$\mathbf{H}$	observation matrix
k	discrete instants of time
$\mathbf{K}$	Kalman gain matrix
$\mathcal{O}$	observability matrix
$\mathbf{Q_l}$	process noise covariance matrix
$\mathbf{P}$	estimation uncertainty
$\mathbf{R_l}$	observation covariance matrix

T	discrete sampling time
u	common mode input speed
$\hat{u}$	estimated common mode speed
$\hat{s}_l$	estimated linear slippage
x,y	mobile robot position in x,y axis, respectively
$\hat{x}$,$\hat{y}$	estimated position in x,y axis, respectively
$\hat{\dot{x}}$,$\hat{\dot{y}}$	estimated linear speed in x,y axis, respectively
$\mathbf{x}$	states vector
$\hat{\mathbf{x}}$	estimated states vector
$\mathbf{y}$	output vector
ω	angular mode input speed
$\mu_\mathbf{l}$	process noise
$\gamma_\mathbf{l}$	observation noise
σ_μ^2	process noise covariance
σ_γ^2	observation noise covariance

Generalized Voronoi diagram

$d(q)$	distance from cell q to the closer obstacle
$d_a(q)$	distance between a position q and any (a) obstacle (excluding the closest)
$d_i(q)$	distance between a position q and the closer (i) obstacle
i,j,a	Index
$\mathcal{F}$	Set of positions in the configuration space
F_{ij}	Set of points common to both, i and j, Voronoi regions
GVD	Set of all two-equidistant points (GVD points)
$\mathcal{O}$	Obstacles
$\mathbf{p}$	Position in the configuration space
$\mathbf{p_g}$	Goal pose
$\mathbf{p_s}$	Start pose
$\mathbf{q}$	Position in the configuration space
$\mathbf{q_g}$	Goal position
$\mathbf{q_s}$	Start position

$\mathbf{q_g'}$	Point of the GVD closest to the goal position
$\mathbf{q_s'}$	Point of the GVD closest to the start position
$\mathcal{Q}$	Configuration space
$\mathcal{Q}_{free}$	Free configuration space
$\mathcal{QO}$	Obstacles in the configuration space
$R(\mathbf{q})$	Positions intercepted with the robot space
$\mathcal{W}$	Workspace
$\mathcal{WO}$	Obstacles contained into the workspace
S_{ij}	Set of points equidistant to obstacles i and j
RM	Set of points that define the roadmap

Navigation functions

i,j	Obstacles index
H_{nh}	Pseudo-obstacles for motion with non-holonomic constraints
k	Navigation function parameter (integer positive)
M	Number of obstacles
$\mathbf{q}$	Robot position
$\mathbf{q_0}$	Circular world center position
$\mathbf{q_d}$	Local goal position
$\mathbf{q_i}$	Position of the i obstacle
$\mathbf{p}$	Robot pose
$\mathbf{q_d}$	Desired pose
r_r	Robot radius
r_0	Circular world radius
r_i	Radius of the i obstacle
x	Generic function parameter
x_0,y_0	Circular world center coordinates
x_i,y_i	Coordinates of the i obstacle
x_r,y_r	Robot position coordinates
β	Repulsive function
β_0	Circular world repulsive function

β_i	*i* obstacle repulsive function
γ_d	Attractive function
ψ	Mobile robot attitude
ψ_d	Desired attitude
ε	Positive coefficient
ε_{nh}	Positive coefficient for motion with non-holonomic constraints
ρ_0	Distance between the robot and the center of the circular world
ρ_i	Distance between the robot and the center of the *i* obstacle
σ	Diffeomorphism function
σ_d	Function that ensure only one minimum, located in the goal
φ	Navigation function
$\hat{\varphi}$	APF function based on robot, goal and obstacles positions

Pose control

e	Distance error between the robot and the goal position
e_x,e_y	Position error in cartesian coordinates
$\mathbf{e}$	Error state vector
u	Commanded linear speed
u_{max}	Maximum robot linear speed
V,V_1,V_2	Lyapunov functions
x_r, y_r	Mobile robot position
x_g,y_g	Mobile robot goal position
α	Angle between the robot frame and the distance vector
θ	Angle of the goal vector with respect with inertial frame
ψ_r	Mobile robot attitude
ω	Commanded angular speed
ω_{max}	Maximum robot angular speed
γ,k	Control law parameters
λ,h	Lyapunov parameters

Bayesian PCA-based positioning system

d	cells dimension
i,l	cell index
k	discrete instants of time
u	commanded linear speed
$\mathbf{m}$	environment grip-map
M	number of cells in the environment grip-map
M_x	number of cells in the x axis
M_y	number of cells in the y axis
p	probability
P	corrected probability
p_l	probability to the robot moves from the cell l
$\mathbf{x}$	state vector
x_l,y_l	cartesian coordinates of the cell l
$\bar{P}$	predicted probability
q	positive coefficient
$\mathbf{z}$	observation state
η	normalizer factor
ψ	mobile robot attitude

1

Introduction

The development of robotics, instrumentation and control in the last decades has led many universities and robotic companies to develop robots for the execution of a variety of tasks. Nowadays, the application of robots is widespread and it is possible to find them in many areas, such as: logistics (Kelly et al., 2007), industry (Bogh et al., 2014; Sprunk et al., 2016), demining (Habib, 2008; Colon et al., 2007), offices (Nourbakhsh et al., 1995; Simmons and Koenig, 1995), hospitals (Carreira et al., 2006; Ilias et al., 2014; Gonçalves and Arsenio, 2015), museums (Burgard et al., 1999; Samejima et al., 2015), zoo parks (Evers et al., 2014), among others. When compared with manual vehicles, mobile robots help improving the quality and efficiency of services, allowing people to spend more time with critical tasks. Moreover, unlike humans, mobile robots provide 24/7 services.

Many robots execute tasks, like following a person or vacuuming a room with a random path, which can be done using an approach known as reactive navigation. In these tasks, no knowledge about the robot's location in the environment is required, therefore, it can navigate without a map. In contrast, other tasks, such as transporting loads in warehouses (Kelly et al., 2007) or meals in hospitals (Carreira et al., 2006), executing inspection tasks (Sprunk et al., 2016), interacting with children (Gonçalves and Arsenio, 2015), require the knowledge about their pose (position and attitude) in the environment. To perform these tasks, the robot needs to move between specific positions in the environment. Here, a non trivial challenge emerges: how to drive the robot toward a given target, and maintain the knowledge about where it is, where it wants to go and how to get there, without any external support?

The work presented in this thesis is concerned with a navigation system for indoor mobile robots, that move under a ceiling with rich information (e.g. Heating, Ventilation and Air Conditioning (HVAC) units, electrical and security systems, etc.). In this approach, the navigation is based on the knowledge the robot has about the environment and its global pose inside a map, previously created in a manual acquisition process and installed on-board the robot.

Thus, in contrast with behaviour-based navigation, where the robot motion depends on the perception around, the design of a navigation system based on a map is more challenging, because the mobile robot must possess skills in four key points (Siegwart and Nourbakhsh, 2004, p. 193):

- Perception: ability to interpret useful and important data from sensors;
- Localization: ability to estimate its position in the environment;
- Cognition: ability to decide what it needs to do to achieve the goal;
- Motion control: ability to follow the desired path.

Clearly, the mobile robot localization is a crucial task in a map-based navigation system. To achieve an accurate localization, the perception system design has an important role. In order to estimate its pose, robots make use of sensors that provide information about the environment and their own motion.

Nowadays, the satellite systems are the standard solution to obtain the global robot pose in outdoor environments (Siegwart and Nourbakhsh, 2004, p. 101), and has been implemented in many outdoor autonomous vehicles (Thrun et al., 2006; Levinson and Thrun, 2010). However, when no external positioning system is available, e.g. indoor, underwater or outer planets, the robot pose estimation problem becomes more complex.

This means that the environment where the robot operates may have impact on the perception system to be implemented. Moreover, sensors are limited in perceiving the environment and extract only a small part of the available information. Thus, the perception is often performed by more than one sensor, which information is merged.

The work presented in this thesis proposes an alternative approach for robots' navigation in indoor environments with rich information ceilings (Fig. 1.1). It only uses on-board sensors, without the need to structure the environment. The self-localization system for unstructured environments has the following assumptions:

i) The robot pose is estimated only from the on-board sensors raw data (no external sensor or landmark are required);

ii) No choice or coding are needed in the image processing algorithm, i.e. it is not necessary to segment objects, special structures (lines, circles, corners, etc.), or any elements of the image;

iii) No calibration is required prior to the use of the image processing algorithm to guarantee that features are recognized or segmented;

iv) No feature tracking algorithms are needed, simplifying the algorithm complexity.

.

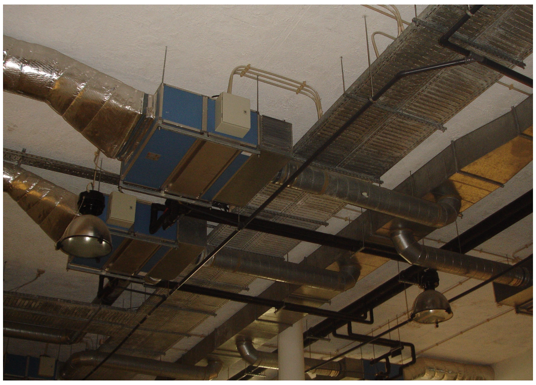

Figure 1.1: Ceiling view of the environment.

The main objectives of this thesis are:

- To implement a self-localization system for differential drive robots that work in indoor environments under a ceiling with rich information (e.g. industrial ceiling) and estimate the pose from on-board sensors raw data without to structure the environment;
- To compute the robot position from ceiling images PCA, namely the Karhunen-Loève (KL) transform, and achieve a good performance in different lighting conditions, using ceiling depth images corrupted with missing data in unstructured environments;

- To design a globally stable self-localization system to estimate the differential drive robot pose from the ceiling images PCA and the sensor fusion with two linear KF. The position and the attitude estimation is decoupled, avoiding the kinematic model linearization or multi-model implementation;
- To implement a navigation system that moves a differential drive robot along a clearance and smooth path towards desired destinations, while is self-locating by the PCA of ceiling images and merges the sensor information with the two KF;
- To validate the implemented system with experimental tests, using a differential drive robot in an indoor real environment with an industrial-like ceiling and, consecutively, to experimentally validate the PCA-based positioning system proposed by Oliveira (2007).

1.1 Motivation

To navigate from one point to another, indoor mobile robots need to self-localize in the environment. Typically, the localization problem consists in finding the robot's pose in the map, based on the information obtained from the environment through sensors. In indoor environments, when no external localization system is available, the mobile robot pose is estimated with on-board sensors only. To perform this task mobile robots are usually equipped with different types of sensors, like encoders, compasses, rate gyros, accelerometers, sonars, laser rangefinders, 3D time-of-light cameras, vision, etc.

Among the available sensors that can be used for mobile robots self-localization, vision cameras are one of the most powerful sensors, due to the large amount of information that it provides about the environment (Siegwart and Nourbakhsh, 2004, p. 89). Because of this, vision has been widely used in mobile robot self-localization systems (See DeSouza and Kak (2002); Bonin-Font et al. (2008) and references therein). Although the general approach is to install forward looking or panoramic cameras (Do et al., 2015; Nakazato et al., 2015), some robots use cameras looking upwards to estimate its global pose with information extracted from the ceiling (Dellaert et al., 1999a; Thrun et al., 2001; Hwang and Song, 2011). Ceiling view has the advantage that images are not affected by scale and, usually, the scenario is less susceptible of changes than the looking to an horizontal plane. Moreover, industrial and similar environments often have static systems (e.g. HVAC, electrical, security, etc.)

with components installed in ceiling, which becomes a rich information source about the environment, to be used by the mobile robot self-localization systems.

The most common approach to obtain environment information is to extract features such as geometric primitives (e.g. lines, circles, etc.) and environment objects (edges, doors, etc.), or to use artificial beacons (Yuan et al., 2016; Shih and Ku, 2016; Dias et al., 2015; Choi et al., 2011; Jeong and Lee, 2005). However, feature extraction, object recognition or beacon detection are performed by complex algorithms that add computational cost to the system.

In contrast, the Principal Component Analysis (PCA) method allows the implementation of simpler approaches due to the high compression of the environment. The use of PCA in mobile robots for self-localization has been explored in Maeda et al. (1997); Artač et al. (2002); Payá et al. (2007), using frontal and panoramic cameras. Oliveira (2007) proposes the use of PCA as a positioning system in a terrain reference navigation of Underwater Vehicles (UV), using signals extracted by a sonar point downward. However, the results were only obtained in simulation.

In addition to the problems related with the processing information extracted from the environment, the fusion between this information and the mobile robot motion is another challenge to the self-localization systems. Usually, localization systems are designed using Bayes filters. The Kalman Filter (KF) is a classical technique for implementing Bayes filters in linear systems (Thrun et al., 2005, pp. 40-45). However, mobile robots kinematics (e.g. differential drive) are in general nonlinear, preventing the use of a KF, and so, the achievement of its optimal estimations. To overcome this problem, many localization systems use the Extended Kalman Filter (EKF), which results can diverge in consequence of wrong linearization or noise sensor (Gutmann and Fox, 2002). Other probabilistic algorithms have been proposed to estimate the mobile robot localization, merging the information extracted from the environment with the odometry, using Monte Carlo Localization (MCL) (Gutmann and Fox, 2002; Theodoridis et al., 2013) or Markov Localization (ML) approaches (Murray and Little, 2000; Almansa-Valverde et al., 2012). However, such approaches do not provide optimal estimates and, to improve the estimations accuracy, it is necessary to increase the complexity to a multi-modal estimation.

The present work proposes an innovative method to perform mobile robot navigation tasks indoors, and under ceilings with rich information, such as industry or similar environments. This is achieved only with on-board sensors (no external sensors or landmarks are required), and without the need to structure the environment.

The proposed self-localization system resorts to PCA of images acquired by a video or depth camera installed on-board, looking upwards to the ceiling (see Fig. 1.1). The image data computed by PCA does not depend on any predefined structure of the environment. Clearly, there should always be some information available to distinguish one location to another, i.e., the captured images must be different along the environment. However, with the PCA data, no previous assumptions on the predefined structure of the environment need to be considered. The PCA data analysis corresponds to the computation of the most representative data components that will turn each dataset unique. Due the great PCA compression ratio, the database of images stored on-board has a reduced size, when compared with the total number of images considered.

To deal with different lighting conditions, this work also extends the PCA-based positioning system for indoor mobile robots to use depth images captured from a ceiling with rich information, instead the usual images captured by video cameras. The depth images implementation avoids the limitation related to different environment illumination conditions, but the signals are usually corrupted by missing data. Thus, when depth images are considered, the self-localization system implements a PCA extension that reconstruct the captured corrupted signals.

Beyond the problems of image processing for self-localization, another challenge is to deal with the fusion of the position from the PCA-based system with the odometry data given by the robot kinematics. The mobile robot kinematics (e.g. differential drive) is, usually, non-linear, preventing the access to optimal estimates provided by a KF. To tackle this problem, many localization systems use the EKF, that is a not optimal estimator and has stability limitations. Even though, under special conditions, this solution can present reasonable performance, it may diverge due to wrong linearization or sensor noise.

The stability of the robot pose estimations is especially important so that they can be used on the navigation systems. In this work, the position and attitude estimation is decoupled into two linear estimators. Thus, the robot attitude estimates are achieved with optimal performance, through the implementation of a KF. For the position estimation, the implementation of a Linear Parameter Varying (LPV) model for differential drive kinematics allows the use of a time-varying KF, that also guarantees stable estimates. This approach avoids the non-linear model implementation and the related issues mentioned above.

The proposed self-localization system is implemented in a mobile robot platform with kinematics of a differential drive car and validated with a large set of experiments in an indoor environment. During these experiments, the robot is self-located using only the computation of the raw data of vision or depth images captured from the ceiling and motion sensors installed on-board, achieving stable position and attitude estimates. The self-localization system is also integrated in a navigation system, allowing an autonomous motion from an initial position towards a desired destination. Along the mission, the robot moves using only the information provided by on-board sensors.

1.2 Related work

1.2.1 Perception and localization

When no external global or local positioning system is available, e.g. indoors, underwater, or outer planets, the problem of estimating the position and attitude of an autonomous mobile robot becomes more complex. In this case, the robot must estimate its pose relying only on on-board sensors. The design of robots navigation systems, based on information from sensors installed on-board, has been a challenge for many researchers. Usual inputs to the localization systems are the measurements provided by sensors installed on-board, like compasses, gyroscopes, accelerometers, cameras, laser scanners, time of flight or structure light cameras, odometers, etc. These measures have been used to compute the pose of the robot, and nowadays can be found many different approaches to perform this task.

1.2.1.1 Vision based location systems

Vision sensors are the most powerful robotic sensors, due to the large amount of information that cameras extract from the environment (Siegwart and Nourbakhsh, 2004, p. 117). For this reason, many mobile robots have been developed to use visual information, captured by cameras installed on-board, as the main sensorial data in the navigation systems (see Thrun et al. (2005); Corke (2011), and references there in).

The most common approach to capture visual information about the environment is to install cameras looking around the robot. The easier vision perception system is achieved installing a monocular camera looking ahead. However, taking into account the available cameras and the corresponding installation design in the robots, other approaches can be implemented to increase the visual information that can be captured from the environment, as eye-fish cameras (Han et al., 2013), stereo vision (Murray and Little, 2000; Zhou et al., 2015; Nakazato et al., 2015) or omnidirectional vision (Maohai et al., 2013; Do et al., 2015).

Many vision systems compute the robot pose from features extracted from the environment (e.g. corners, lines, circles, etc.) or using landmarks (edges, doors, etc.). The use of features allows the construction of a more compact environment model (Scaramuzza et al., 2009; Gil et al., 2010; Maohai et al., 2013; Zhou et al., 2015), when compared with the raw data of captured images. The features extraction from images is performed using algorithms and descriptors like Scale-Invariant Features Transform (SIFT) (Lowe, 2004), Speeded Up Robust Features (SURF) (Bay et al., 2006) or Histograms Oriented Gradient (HOG) (Dalal and Triggs, 2005), among others. In practice, the descriptor analyse the captured images, aiming to extract from them a set of interest points, that may be useful to structure the environment. Hence, with different images taken from the same localization, different interest points may appear, distorting the environment perception. The RANdom SAmple Consensus (RANSAC) (Fischler and Bolles, 1981) is an usual implemented algorithm to keep track on features while disregarding outliers but, in practice, all these strategies rely on some structure of the environment (Zhou et al., 2015; Bacca et al., 2011; Loevsky and Shimshoni, 2010). In addition, such algorithms require complex computation to extract and select the features, and often need to be combined to get the interest information about the environment structure.

As was mentioned above, although the most common approach is to install cameras in mobile robots to look around (similar to humans view), aiming to get its global pose in the environment, others use a different approach, with a single camera looking upward to the ceiling. The use of vision from the ceiling has the advantage that images can be considered without scaling, i.e. a 2D image problem results. Moreover, ceiling vision is usually more static than the horizontal view, i.e., ceilings are less susceptible of scenario changes and are not affected by moving obstacles around themselves.

Dellaert et al. (1999a) developed a mobile robot localization system based on ceiling vision, where the information provided from the center of the image is used in a probabilistic algorithm to improve the vision robustness, when it passes under the lights. A ceiling vision-based navigation system was developed by Jeong and Lee (2005), where corner points extracted from the Red-Green-Blue (RGB) image are used as points of interest. Jo et al. (2012) extract landmarks (lines and circles) from ceiling images captured from a monocular camera pointed upward. In this work, the addition of a sonar was implemented to measure the ceiling distance, allowing the mobile robot localization to move with ceilings at different heights. Hwang and Song (2011) developed a Simultaneous Localization And Mapping (SLAM) system for a mobile robot, where the corners, lamps, and door features detected from the upward-looking camera were used as the landmarks. Huang et al. (2012) implemented a mobile robot localization system using artificial landmarks (patterns with geometric figures) installed on the ceiling of an indoor environment. A similar approach was developed by Shih and Ku (2016), but in this work the landmarks have alphanumeric characters and the mobile robot has a map with the neighborhood relations between them. Dias et al. (2015) uses lamps, pipelines and HVAC boxes, installed on the ceiling, as landmarks to extract features (circles and rectangles) that are used to localize the mobile robot.

This work proposes the design of a self-localization system using ceiling vision (with the mentioned above advantages), but the implemented algorithm works in unstructured environments. The only requirements are that the ceiling be static and have enough information to capture different images in each location.

1.2.1.2 RGB-D images based localization systems

Despite the advantages of using vision sensors, cameras are very sensitive to the environment lighting conditions, which have a negative impact on the robustness of localization systems (Siegwart and Nourbakhsh, 2004, p. 89).

To avoid the aforementioned problem, some localization systems are based on Time-of-Flight cameras or 3D laser scanners. These sensors present a more robust system able to cope with different light conditions and have the advantage of providing depth information in two directions (horizontal and vertical), increasing the amount of information captured on the environment, when compared to the information provided from a set of sonars or 2D laser scanners (Weingarten et al., 2004; Almansa-Valverde et al., 2012; Surmann et al., 2003).

In this last decade, new devices with a RGB and Depth (RGB-D) cameras have been developed for video games, e.g Kinect from PrimeSense and Microsoft. Due to its low price and a straightforward way to be connected to a computer, the Kinect device became popular in mobile robotics community, creating several applications in this area.

To reduce the amount of depth information captured by the Kinect, several approaches divide the depth image as an array of distance sensors (Theodoridis et al., 2013; Correa et al., 2012). Once the environment information in depth images is more than the provided by a sonar array, this solution provides better results (Theodoridis et al., 2013). Biswas and Veloso (2012) also show that the mobile robot achieves better localization accuracy using cloud points extracted from the Kinect than when a laser rangefinder is used, due to the greater amount of data extracted from the environment. Rodrigues et al. (2013a) use all depth images data captured from the ceiling to compute the mobile robot visual odometry. However, depth information can be used in other applications. For example, Stowers et al. (2011) implemented a Kinect in a quadrotor, pointing to the ground, using the depth data to control the altitude during the flight.

Given that Kinect captures visual and depth information, the data provided by RGB-D images can be combined to extract the feature points and be used in an autonomous robot navigation (Huang et al., 2012) or to detect artificial landmarks (Ganganath and Leung, 2012). Datasets with RGB-D images provided by a Kinect device were captured by Sturm et al. (2012), aiming the design of mobile robots localization systems in indoor environments. The tracking of features extracted from consecutive RGB-D frames can be used to build 3D models of the environment for mobile robots navigation (Endres et al., 2012; Henry et al., 2014; Endres et al., 2014). The features tracking of RGB-D images have been also implemented in visual odometry to estimate the position of a quadrotor (Huang et al., 2011), to detect and track people (Jafari et al., 2014) or to navigate avoiding obstacles (Nguyen et al., 2016). Yuan et al. (2016) use the RGB-D images to detect artificial landmarks around the robot, composed by patterns with color cards.

Despite the wide variety of robot implementations that use data provided from a Kinect, such approach compute those amount of information to extract specific features from the environment. As mentioned above, this computation requires complex algorithms to structure the environment. In addition, some approaches use those data to construct dense maps, with consequences in the complexity of the implemented algorithms.

This work implements an opposite approach, where the raw data of depth images provided by a Kinect is used to build a compact model, without the need to extract specific characteristics from the environment. Thus, in this work the mobile robot is localized in the environment from a raw data linear transformation of ceiling depth images.

1.2.1.3 Principal component analysis

The most common approaches to design mobile robot self-localization systems are based on the structured environments to create the corresponding map. These approaches need to extract specific features from the captured data, such as: points, lines, planes or structures, which are performed by complex algorithms. The use of algorithms to create a localization system without the need of specific features extraction, i.e, in an unstructured environment, becomes an alternative approach to improve the performance of this task.

The PCA (Jolliffe, 2002) is an efficient algorithm that converts the database into an orthogonal space, creating a database with an high compression ratio, when compared with the amount of captured data, and without the need of specific features extraction. In addition, the PCA is computed using only linear computation, allowing to achieve an optimal approximation of the stochastic signals, minimizing the mean square error. This characteristics, make the PCA an useful tool to be implemented in applications with many signal processing such as, data compression (Pustianu et al., 2011), pedestrians and objects detection (Malagón-Borja and Fuentes, 2009), faces recognition (Budiharto et al., 2011), patterns in rotated images (Heras et al., 1997), among others.

Thus, the PCA allows the development of localization systems that do not depend on any predefined structure, i.e, does not need to detect any specific features about the environment to create a compact map. Since feature based techniques require complex algorithms, some researchers have been working to find methods that make this process more efficient, namely through a dimensionality reduction of the image database captured from frontal or omnidirectional cameras with the PCA.

A mobile robot localization system using eigenspaces analysis was developed by Maeda et al. (1997), where the database was created with images captured at different positions of the environment by a rotating camera looking around robot. In this approach, the mobile robot position was achieved testing candidate images, rotating the camera to test images with different points of view. Aiming to improve the localization with lighting conditions, Jogan et al. (2002) implements a system with a set of gradient-filters. The system was tested along a straight line, capturing images from a panoramic camera, considering different lighting conditions. Artač et al. (2002) propose the use of an incremental PCA for on-line learning and recognition of visual images. The method was validated using panoramic images captured from a camera located in a mobile robot, exploring the same path repeatedly. This method was implemented by Jogan et al. (2003) for exploration and localization of a mobile robot, using panoramic images. To build the map, a robot moves along a grid, capturing panoramic image that are shifted to consider multiple rotated versions. In this approach a set of filters was implemented to deal with lighting changes. In the experimental tests, the robot makes new relocations each time it moves 70 cm in the goal direction. To reduce the number of panoramic images stored in the PCA database, Steinbauer and Bischof (2005) proposes to capture images in a certain reference and, during the motion, to compare the acquired images in the same direction. Although it refers the use of a compass angle to rotate an image into the orientation where the database were captured, in this work a bank with multiple Kalman Filter was implemented to estimate the robot attitude and to perform this task. Payá et al. (2007) also implement the incremental PCA to create a pre-defined routes navigation system. This approach was experimentally validated using two mobile robots, where the leader captures images to create the route and the second robot follows it, comparing the eigenvectors of captured images with the current stored eigenspace.

Oliveira (2007) proposes the use of PCA as a positioning system of an UV with a sonar pointed downward. In this approach, a set of mosaics are captured from the sonar with ground information to create the PCA database. During a mission, the PCA positioning is computed from mosaics signals acquired by the sonar and aligned with the grid orientation. The PCA positioning is merged in a multi-model for different observations noises to estimate the UV position. However, the results was only obtained in simulation.

The PCA was implemented by Pustianu et al. (2011) in a mobile robot control using a face recognition algorithm. However, in this approach the face classifier was implemented by a neural network, being the PCA only applied to reduce the amount of information sent to the neural network classifier. Cao and Hashimoto (2013) developed a mobile robots control algorithm to follow a person, which is recognized resorting to the PCA. In this approach, the PCA database is build with the links length obtained by skeleton information extracted from the RGB-D images, captured by the Microsoft Kinect software.

A very common problem in depth sensors, including the Kinect depth sensor, is the existence of missing data in signals, caused by InfraRed (IR) beams that are not well reflected and not return to the depth sensor receiver. When depth signals are used to extract features from the environment, only the available information is taken into account. This means that missing data present in the signals is commonly neglected. For example, Huang et al. (2011) use the depth data to obtain the features position extracted from the RGB-D. However, when the missing data occurs in depth images, the corresponding landmarks are discarded. A visual odometry using depth images is proposed by Rodrigues et al. (2013a), but in this work the computation is only performed with non corrupted data.

However, when all raw data is needed to percept the environment, as happen with images comparison through the PCA algorithm, the existence of missing data can distort the results. In the case of the localization systems, this decreases their robustness, leading to erroneous localization. Oliveira and Gomes (2010) implement a method using the PCA methodology to avoid the problem of missing data in signals and its performance is compared with other algorithms. In their work it is concluded that the PCA algorithm can be extended to reconstruct signals corrupted with missing data, and it presents a better performance in an extended range of missing data.

This work implements a PCA-based positioning system for a mobile robot using ceiling vision, inspired in the Oliveira (2007) approach. Thus, the system has the ceiling vision advantages: the ceiling has a fixed distance to the robot and is not affected by moving obstacles or scenario changes around it. Moreover, the use of PCA in ceiling depth images allows to work in different lighting conditions without the need of additional filters.

1.2.2 Estimation

1.2.2.1 Sensor fusion

The problem of mobile robots navigation, without the need of any external information, requires the measurements estimation from the sensors installed on-board. Nowadays, with the increasing development of Micro-Electro-Mechanical Systems (MEMS), there are several sensors available to estimate measurements in mobile robots (e.g. compass, rate gyro, accelerometers, etc.), beyond the odometry measure provided by encoders, usually connected to the drive wheel motors. Aiming to obtain robust systems, it is usual to increase the sensors number to estimate the same variable. However, when more than one sensor are used to estimate the same variable, the design of a sensor fusion system to merge the signals provided by different measurements must be developed. See Brown and Hwang (1997); Auger et al. (2013); Cavallo et al. (2014) and references therein.

The classical sensor fusion technique, for linear systems corrupted by stochastic white noise uncertainties with measurements also corrupted by white noise, is the celebrated KF. This method solves a Minimum Mean Square error (MMSE) estimation problem, providing optimal estimates, based on a set of Kalman gains (Kalman, 1960; Kalman and Bucy, 1961; Brown and Hwang, 1997; Auger et al., 2013). KF has been applied to estimate variables, merging the signals provided by several sensors (Mitsantisuk et al., 2012; Zhang et al., 2015; Li et al., 2012b; Suh, 2006).

However, KF requires a complete characterization of the process and observed noises, a task that may be difficult, or not suited to specific problems. The main difficulty associated with this method is the complexity in the identification of a good model for sensors and the robotic system. In alternative to the use of KF for the data fusion, other researches merge the signals resorting to Complementary filtering. This approach allows to merge the signals corrupted by stationary noise with different frequency bandwidths to produce more precise signals in the time domain. The main advantages of Complementary Filters (CF) are the design simplicity and the more intuitive parameters tuning. CFs estimate variables considering signals provided by two sensors in distinct and complementary frequency bands, without the need of characterizing the stationary white noise present (Higgins, 1975). According to Pascoal et al. (2000), CF was used by Brown (1973) in a navigation system, and has been further used integrated in several systems.

Optimal results can be obtained for CF based on measurements from two sensors, in the case where the noise is stationary. Moreover, the estimators design based on sensors working in different bands is often useful to develop redundant measurement systems to fuse signals provided by sensors with different dynamics. Complementary filtering allows the estimator fine-tune for the frequency band where the sensors provide better performance. For instance, to estimate the tilt angle relative to gravity, an inclinometer and a rate gyro were used by Craig (2009). In this work, the rate gyro provides a measure of the angular velocity with a nice flat frequency response to about 50 Hz. However, the angular position, if obtained from integration, is highly affected by the bias error, giving quickly an unacceptable, ever increasing error drift on the position signal. The inclinometer measures tilt angle relative to gravity, does not suffer from a drift problem, but has a low bandwidth (0.5 Hz to 6 Hz) which is too slow for many robotic applications. A CF may hence be used for estimating both angular position and angular velocity over a larger bandwidth with negligible drift. An adaptative-gain CF was developed by Calusdian et al. (2011), where the gain of the two frequency bands is tuned according to the dynamics of the system.

The use of CF has been successfully applied in mobile robot navigation systems, performing the sensor fusion through two frequency bands complementarity, composed by a Low-Pass Filter (LPF) and an High-Pass Filter (HPF), both with the same cut-off frequency: UV (Pascoal et al., 2000; Batista et al., 2010b), Unmanned Aerial Vehicle (UAV) (Euston et al., 2008; Kubelka and Reinstein, 2012; Wang et al., 2014), autonomous helicopter (Baerveldt and Klang, 1997) or quadrotor (Lu et al., 2014).

In spite of the CF be widely implemented to estimate measurements, typically it is used to merge two signals. In this work an approach to merge signals provided by several sensors is proposed, where the frequency spectrum is split according with the number of measurements that are merged.

1.2.2.2 Localization estimators

The self-localization systems is, usually, designed using probabilistic algorithms that merge signals provided from environment perception with robot motion, taking into account the corresponding measurements uncertainty (Thrun et al., 2005). As discussed above, KF (Kalman, 1960; Kalman and Bucy, 1961) is a powerful technique to achieve sensor fusion and provides optimal estimates, implementing Bayes filters (Siegwart and Nourbakhsh, 2004, p. 227; Thrun et al., 2005, pp. 19-20). This mathematical tool produces optimal states estimations in linear systems, based on measurements corrupted with white gaussian error, as a train moving along a railway line (Faragher, 2012). However, when 2D localization is considered, mobile robot kinematics (e.g. differential drive) are, in general, non linear. This fact prevents the direct use of a KF in the mobile robots localization systems. Notice that, although KF provides optimal estimates, it only can be applied to linear systems.

To tackle this problem, many localization systems extend the KF approach, applying a local linearization to the model to estimate the robot pose (Smith and Cheeseman, 1986). The EKF have been implemented to merge the odometry with features and landmarks extracted from the environment, to estimate the robot pose. Teslić et al. (2011) estimate the self-localization of a mobile robot in a structured environments, where the correction was performed by line segments of the global map, obtained from the measurements of a laser range finder. Jo et al. (2012) merge points of interest extracted from ceiling images with the odometry. Hwang and Song (2011) use landmarks position (corners, lamps, and door features) extracted by a monocular ceiling camera to estimate the mobile robot pose. Yuan et al. (2016) use artificial landmarks, detected by a RGB-D camera.

The use of EKF was proposed to solve the SLAM problem to estimate the mobile robot pose and, at the same time, the features position extracted from the environment (Durrant-Whyte and Bailey, 2006; Bailey and Durrant-Whyte, 2006). However, due to the known divergences caused by wrong linearizations, the use of EKF in SLAM can lead to distorted maps. To overcome this problem, special conditions for EKF-SLAM have been developed to improve the estimation results. For example, Moutarlier and Chatila (1990) proposed the robot re-localization before the states update, to reduce the estimation divergences caused by the EKF linearization; Leonard and Durrant-Whyte (1991) only updated the robot or features states when the obtained estimation about features localization or robot pose were under a predefined uncertainty.

Although the EKF provides reasonable results in many applications, the linearization around the estimated variable, through the jacobian matrix, makes it a non-optimal estimator and often leads to divergence and inconsistency problems (Bailey et al., 2006). However, this limitation due to the linearization is known and accepted by the scientific community (Thrun et al., 2005, pp. 54-64).

To avoid the use of linearization, Pascoal et al. (2000) propose the design of a navigation system for surface or underwater vehicles, composed by two CF to estimate the attitude and the position, respectively. In a similar approach, Batista et al. (2010a) propose a navigation system design to estimate of position and velocity of an UV, using an time-varying optimal estimator based on the angular velocity provided from an attitude and heading reference system. The use of two cascaded estimators, based on KF, was implemented by Barbosa et al. (2015) to estimate the relative localization of a mobile robot to a landmark. In this approach, a LPV using the instantaneous angular rate, provided from an attitude estimator, is merged to estimate the robot position. The estimated measurements provided from the localization system are used to control the robot motion in a docking station system.

On the other hand, MCL estimates the mobile robot pose applying a Bayes filter to a set of particles that are initially randomly distributed over the entire environment (Dellaert et al., 1999b; Fox et al., 1999). In spite of the complex computation, MCL has become popular for providing more accurate results than the EKF. Benchmark tests were presented by Gutmann and Fox (2002) and by Ganganath and Leung (2012). However, the better performance of the MCL is achieved at the cost of estimator's increased complexity. In this approach, the complexity reduction can be implemented by reducing the number of particles. Gutmann and Fox (2002) propose the use of an adaptative number of particles and Maohai et al. (2013) select the better particles using a genetic algorithm.

A MCL algorithm was implemented by Biswas and Veloso (2012) in a mobile robot navigation system, where a 2D map was built from depth data captured by a Kinect sensor. Theodoridis et al. (2013) apply a MCL to estimate the global position of a robotic wheelchair, comparing the results when sonar or depth information are merged with the odometry. The results shows better performance when depth information is used. A particle filter is implemented in an omnidirectional mobile robot with panoramic vision by Maohai et al. (2013). A global localization in a static environment using bluetooth beacons was implemented by Raghavan et al. (2010), where the position provided by the bluetooth beacons trilateration is merged with the mobile robot model, through the MCL algorithm.

Rowekamper et al. (2012) combine MCL with particles sampling and a scan matching algorithm with purpose to achieve accurate poses in a mobile robot localization system. An estimator that merge the particles filter with a KF was developed by Karlsson and Gustafsson (2003), and was tested with experimental data from an underwater navigation system. Results present a performance near the optimal, according with the Cramèr Rao lower bound obtained, but with very computational demanding. A Kalman-particle filter has been proposed by Pham et al. (2003) to combine an EKF with the robustness of the MCL. However, a large number of particles still must be used and even the simulations results presented a reduced number of divergences. Aiming to improve the SLAM robustness, the combination of particle filter with an EKF was proposed by Montemerlo et al. (2002, 2003). In this approach, the particle filter is used to estimate the robot motion and the EKF is employed to update the probability of the samples, which represent the candidates for the robot and landmarks positions.

Despite the MCL being a more accurate algorithm than EKF, its accuracy is dependent on the particles number, which increases the system complexity. Furthermore, given that the particles only “survive” near the mobile robot, the estimation can converge to incorrect values if the particles move away from the real pose.

In spite of the known divergences caused by wrong linearization, EKF remains a very popular approach to perform SLAM tasks. Moreover, the EKF-SLAM complexity becomes a limitation to the number of features, and therefore the map size, due the fact that covariance matrices are quadratic with the number of landmarks. On the other hand, FastSLAM is presented as an approach to improve the stability, but the complexity is related with the number of particles. In addition to the challenges, an issue that remains is that solutions rely on landmarks or any other features that the robot must sense in the environment. In practice, there is no guarantee that the same features will be recognized on subsequent visits of the robot to the same location.

The use of a multiple-model adaptation estimator composed by a set of KF was proposed by Oliveira (2007) to estimate the UV position. In this work, the bank of KF was implemented for different terrain characteristics, using the positioning system based on the PCA algorithm. A multiple-model estimator was implemented by Gaspar et al. (2011) to estimate a marine mammal position, where a set of KF was applied with different angular rates, using the 2D horizontal constant-turn model with constant speed model presented by Li and Jilkov (2003). The same constant-turn and speed model was implemented by Song et al. (2012) in an interactive multiple model to estimate the mobile robot position. However, this approach requires the installation of an external system to extract the possible modes from trilateration and the use of multiple-modes to achieve appropriate estimates, failing when only one single measurement is applied in a KF.

This work proposes an approach to indoor mobile robots self-localization, where the robot position and attitude estimation is decoupled into two different linear KF, whose estimates are computed using only the information provided by on-board sensors. Thus, an optimal KF estimates the robot attitude and its angular velocity. The 2D robot position is estimated from one time-varying KF, using the estimated instantaneous angular velocity in a LPV model. Thus, the position and attitude have globally stable estimates from linear KF.

1.2.3 Navigation

1.2.3.1 Path planning

Path planing is a key task in a map-based mobile robot navigation. This task defines a set of waypoints between the actual pose and the goal, using the knowledge about a map, where the environment (e.g. walls, fixed and moving obstacles, etc.) is represented. Thus, for a given environment, an infinitude of possible paths can be defined to connect the start and goal positions. Each path is defined by a set of waypoints in the free space. Thus, to find the shortest path between the start and goal positions, several algorithms have been developed: Dijkstra (Dijkstra, 1959), A* (Hart et al., 1968), D* (Stentz, 1994), D* Lite (Koenig and Likhachev, 2002), among others.

In spite of a simple and common way to represent the environment being through an occupancy grid, the definition of a path involves the selection of waypoints, from a very large set of possibilities, covering all free space. This drawback may become critical if the environment is large and the size of the grid greatly increases. Moreover, the shorter paths between start and goal positions, usually, are defined too close to the obstacles, causing collision or security risks.

An approach to simplify the planning task is to reduce the possible paths in a roadmap, defined within the free space. This approach is similar to the highway systems, by covering the free space that define the possible paths in a territory. In this approach, the robots take the highway near the starting point, and follow it until they are close to the goal position (Choset et al., 2005, p. 108). Thus, the path is defined by a set of points that connect the start and goal positions through the roadmap.

One approach to design a roadmap and, simultaneously, to ensure a safety path is using a Generalized Voronoi Diagram (GVD). This roadmap is defined by a set of points that maximize the distance to the obstacles. Thus, in several approaches the design of GVD is made using mobile robots with a controller to keep them in a equidistant position to the obstacles at their side (Kim et al., 2010; Arunkumar et al., 2014).

When the obstacles and free space are represented using an occupancy grid, it is common to build a GVD-based roadmap using a grid GVD. Barraquand and Latombe (1991) propose the GVD definition using a wavefront algorithm, being the GVD defined by the set of points where the "waves" propagated by the obstacles and environment boundary meet. Similar approaches were implemented by Yang and Hong (2007); Kuderer et al. (2014); Garrido and Moreno (2015). Corke (2011) proposes an algorithm to create a GVD from an occupancy grid, where a roadmap is computed by a morphological image processing algorithm.

In other approach, when the obstacles are represented by points, the roadmap can be computed using triangulation (Dong et al., 2010). Bhattacharya and Gavrilova (2008) compute an initial Voronoi, using the Deluanay triangulation algorithm, from a cloud of points that define the obstacles boundary. The final GVD-based roadmap is obtained after the removal of unnecessary lines.

The path is chosen by searching for one that, using the GVD, minimize the distance from the start to goal positions. Usually, it is computed from algorithms as Dijkstra (Wein et al., 2005; Bhattacharya and Gavrilova, 2008; Dong et al., 2010) or A* (Ok et al., 2013).

1.2.3.2 Navigation based on artificial potential fields

Artificial Potential Fields (APF) is a technique that combines path planning and obstacle avoidance, inspired by the attraction and repulsion of bodies in magnetic fields. The idea of a robot moving through a potential field was presented by Khatib (1986), being the path defined by a force vector that results from attractive and repulsive fields. Thus, the obstacles are defined as high potential points that repel the desired path, while the goal position is a point with lower potential that attracts it (Arkin, 1987; Ge and Cui, 2002; Bing et al., 2011; Li et al., 2012a). Carreira et al. (2011) applied potential fields to define a safe and reliable mobile robot path in an human populated environment.

Despite the simplicity of implementation, potential fields have the drawbacks of local minimums and oscillations. Although it can be implemented conceptually as a global navigation system, in practice, potential fields are considered as a local planning method, being often integrated in an hybrid system, with a global planning algorithm.

1.2.3.3 Navigation based on navigation function

To avoid the problem of local minima in APF, Rimon and Koditschek (1992) developed a new class, denoted as Navigation Functions (NF). This approach has the advantage to plan and control the mobile robot along an environment with obstacles, ensuring an unique minimum located in the target.

The NF were presented to control the pose of a non-holonomic mobile robot, while is avoiding the collision (Tanner and Kyriakopoulos, 2000; Tanner et al., 2001). Multi-robots NF were proposed to control teams of differential drive robots (Loizou and Kyriakopoulos, 2003; Dimarogonas et al., 2006; Widyotriatmo and Hong, 2011). In these approaches each robot acts as a potential obstacle to the others, while it is attracted towards its goal. Such approach is extended to control teams composed by holonomic and non-holonomic robots (Loizou and Kyriakopoulos, 2008). Arslan and Koditschek (2016) use a power diagram (GVD with additive weights) to create a convex free space in an environment populated with circular obstacles.

The classical navigation function proposes the use of repulsive fields sensed in all free space emitted by circular obstacles, inside a circular environment that repels the robot motion towards the goal. However, other approaches can be found. The use of bump functions allow design controllers where the world boundary or obstacles are only sensed when the robot is close. This approach was proposed to simulate the navigation of UAV teams in a combat zone by Chen et al. (2007). Pradhan et al. (2011) propose the implementation of elliptical fields around moving obstacles to predict their motion in the environment.

The navigation function is applied to circular worlds. When the worlds are non-circular, what happens in most of the environments, it is usual to transform the configuration space to a circular shape (Conner et al., 2003; Iizuka et al., 2014). An alternative approach was presented by Pradhan et al. (2013) to allow an easy implementation and avoiding this complex operation. In this approach, a set of consecutive worlds are generated along a corridor and circular obstacles are applied representing the walls, to avoid the collision. Thus, a goal is defined in each world in order that the robot crosses it, and gets in the next world.

1.3 Thesis contributions

The mains contributions of this Thesis are described bellow:

The first contribution is the design of a self-localization system to estimate the global pose of a differential drive robot that navigates in an indoor environment under a ceiling with rich information (e.g. industrial-like environment). This is achieved only with raw data of grayscale or depth images captured from the ceiling, without the need of extract specific features about the environment structures, and merged with other on-board sensors, through stable linear estimators. Conceptually, the proposed self-localization system architecture is composed by three modules (Fig. 1.2):

i) an optimal attitude estimator that provides the mobile robot estimated attitude and the estimated angular slippage, relying on data from motion sensors installed on-board the robot;

ii) a PCA-based non-linear positioning system that provides measurements on the absolute 2D position. This system will process (grayscale or depth) images acquired from the ceiling (Fig. 1.1) with the information from a data-set previously collected and pre-processed;

iii) a sub-optimal 2D position estimator, based on a LPV model for a differential drive mobile robot, parameterized by the instantaneous angular rate estimate, that provides error bounded 2D robot position estimates.

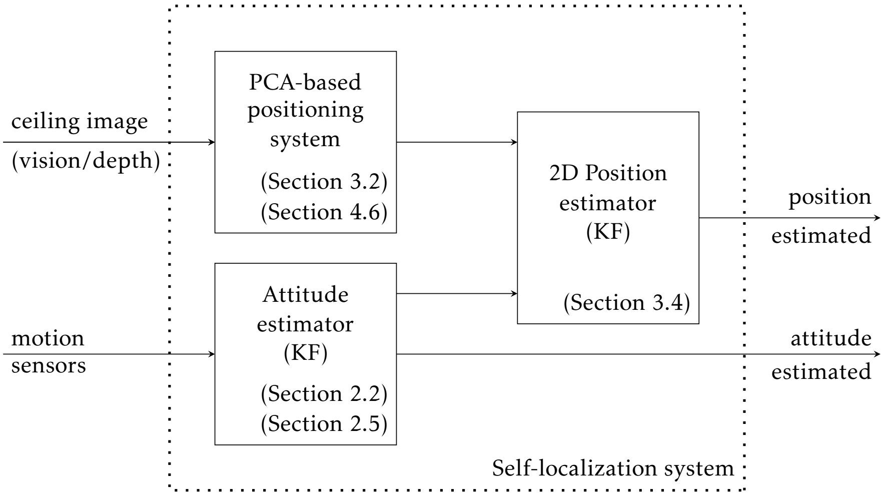

Figure 1.2: Conceptual architecture of the self-localization system.

This leads to the second contribution that is the experimental validation of the PCA-based positioning system proposed by Oliveira (2007), to estimate the position of an UV with a sonar pointed downwards, which results were obtained with simulation.

The third contribution consists in the PCA-based positioning system extension to reconstruct ceiling depth images, corrupted with missing data. The use of this extension eliminates the corrupted data present in the depth images and improves the positioning system robustness to deal with different lighting conditions.

The fourth contribution is the design of estimators to merge several sensors, based on complementary filtering. The approach was detailed and analysed to estimate the mobile robot attitude, merging the signals provided from three sensors.

The fifth contribution is the self-localization system integration in an navigation architecture with environment configuration, path planning and motion controller.

Finally, a last contribution is the large set of experiments, performed with a mobile robot with differential drive kinematics in a real indoor environment, that allows to validate the developed self-localization system.

1.3.1 System advantages

The proposed system has the following advantages:

- the mobile robot is able to self-locate in an indoor environment, using only on-board sensors, i.e., no external sensors or landmarks are required to achieve its global pose in the environment;
- the self-localization system has a positioning system based on PCA images (grayscale or depth) captured from the ceiling, that avoids the need of specific features extraction or beacons detection from the environment. The only requirements are that the robots should work in a building with static ceilings where rich information can be found (e.g. HVAC, electrical or security systems, etc.) and the images must be different in each location.
- the PCA method compresses the data with high compression ratio, when compared with the dimension of raw data captured from the ceiling. Thus, the database of images stored on-board the mobile robot is of reduced size, when compared with the total number of images considered;
- the images captured using a ceiling vision can be considered without scaling, and are less affected by scenery changes or moving objects around the robot, than the horizontal view;
- the PCA-based 2D positioning system implementation with ceiling depth images avoids the limitation related to different environment illumination conditions. Thus, when depth images are considered in the self-localization system, a PCA extension to reconstruct corrupted signals is implemented, improving the system robustness and preventing wrong locations caused by the existence of missing data;

- the mobile robot position and attitude are estimated using only linear KFs, with global stable error dynamics, avoiding the need of local linearization, or multi-modal estimators, normally employed in common approaches of mobile robot probabilistic location. The self-localization system is composed by two linear KF to estimate the attitude and the 2D position of the robot, respectively;
- the attitude is estimated by a linear KF, that merges the raw data provided from several sensors installed on-board, preserving the complementary filter propriety, with optimal performance;
- the position is estimated by a linear time-varying KF, that resorts to a LPV model to merge the differential drive kinematic model with the global position obtained from the PCA-based 2D positioning system;
- the self-localization system is integrated in a closed loop to perform navigation tasks. The robot motion resorts to navigation functions along a set of waypoints, extracted from a GVD-based roadmap.

1.4 Thesis layout

This thesis is organized as follows:

- This chapter presented some of the approaches usually implemented in mobile robot navigation systems, giving a special focus to the environment perception and the self-localization systems, using sensor fusion. This chapter concludes by introducing the concept of the self-localization system developed in this work and detailing its main advantages over the competing systems.
- Chapter 2 presents a set of mobile robot optimal attitude estimators. The section 2.2 details the KF design to estimate the attitude, merging the signals provided from a digital compass with the odometry. This estimator will be used to analyse the proposed self-localization system performance, using grayscale images (chapter 3) and depth images (chapters 4 and 6). In addition, a set of new attitude estimators based on CF design to merge the signals provided by three sensors, considering different inputs combinations, is detailed in section 2.3. Simulation results using these attitude

estimators are presented in section 2.4. The design of some equivalent discrete-time attitude estimators, aiming the implementation in a mobile robot platform to perform navigation tasks (chapter 5) is detailed in section 2.5. The section 2.6 illustrates experimental results with dead-reckoning localization.

- In chapter 3, the proposed self-localization system architecture is detailed and experimentally validated. The PCA-based positioning system, using grayscale images captured from the ceiling, is detailed in section 3.2 and motivation results are illustrated in section 3.3. The 2D position linear estimator design, using a LPV model for a differential drive mobile robot, is detailed in section 3.4. This chapter presents, in section 3.5, a set of experimental tests to validate the developed self-localization system robustness, integrating the attitude estimator detailed in section 2.2.
- Chapter 4 presents a PCA extension to reconstruct signals corrupted by missing data (section 4.3). For performance analysis purposes, experimental depth image reconstruction results are shown in section 4.4, using the PCA algorithm in depth images corrupted with different ratios of missing data. In section 4.5, motivation results using corrupted depth images in the extended PCA-based positioning system are illustrated. This PCA extension is integrated in the self-localization system, with the attitude estimator detailed in section 2.2 and the 2D position estimator presented in section 3.4. This chapter presents, in section 4.6, a set of experimental tests to validate the self-localization system using ceiling depth images, corrupted with missing data.
- In chapter 5, the self-localization system is integrated in a close loop to perform navigation tasks (section 5.2). The section 5.3 details the roadmap based on GVD construction, according with the grid information about the ceiling images mapping. The GVD-based roadmap is used to extract a set of waypoints along the path. The motion controller based in navigation functions is detailed in section 5.4. The experimental results to validate the navigation system in a real indoor environment are presented in section 5.5. These tests are performed using a self-localization system composed by the extended PCA-based positioning system for depth images, detailed in chapter 4, the attitude estimator addressed in section 2.5, and the 2D position estimator presented in section 3.4.

- Chapter 6 presents some implementations using PCA-based 2D localization. Thus, in section 6.2, the self-localization system architecture for ceiling depth images, presented in section 4.6 is integrated in a closed loop control system and used to control the mobile robot pose in a real environment. In section 6.3, an alternative PCA-based positioning system, resorting to a Bayesian probabilistic grid-map is detailed. This bayesian PCA-based 2D positioning system is integrated in a self-localization system, composed by the attitude estimator detailed in section 2.2 and the 2D position estimator presented in section 3.4, and is experimentally validated.

- Finally, chapter 7 details and discuss the main conclusions about the developed work (section 7.1) and presents some pointers for future work (section 7.2).

2

Design of mobile robots attitude estimators

2.1 Introduction

The design of a navigation system requires the attitude estimation based on information provided from sensors installed on-board. The mobile robot attitude can be estimated from measurements provided by a set of the available sensors (e.g. compass, rate gyro, accelerometers, encoders, etc.). Thus, from an accurate attitude estimation, it is possible to perform dead-reckoning, i.e., to estimate the position using only the knowledge about motion sensing. With the increasing availability of sensors to estimate this variable, sensor fusion techniques are often applied to merge the signals provided from more than one sensor. When the sensor fusion is applied to attitude estimation, the sensor fusion improves the accuracy of sensor signals and, consequently, the dead-reckoning accuracy. Notice that sensor fusion is not only applied in robotics, being also an increasingly common approach in the control loops design, in general.

The KF is one classical sensor fusion technique for linear systems corrupted by white noise. This estimator merges the signals provided from several sensors with a set of Kalman gains, computed by a MMSE algorithm and, thus, provides optimal estimates (Mitsantisuk et al., 2012; Zhang et al., 2015).

Alternatively to the use of KF, the estimator design based on Complementary filtering is often used to perform the sensor fusion. The CF design allows the fusion of signals corrupted by stationary noise, through filters with different frequency bandwidths to produce more precise signals in the time domain. The main advantages of CF are the simplicity to design the estimator and the more intuitive parameters tuning. The CF has been widely applied in two signals fusion, combining a Low-Pass Filter (LPF) with an High-Pass Filter (HPF) (Pascoal et al., 2000; Euston et al., 2008; Kubelka and Reinstein, 2012; Lu et al., 2014).

Nowadays there is a large number of commercially available sensors that can provide measurements related to attitude. The existence of more than two sensors in a sensor fusion approach allows a more accurate estimate or avoid erroneous estimation results, thus providing redundancy and a more robust or reliable estimate.

This chapter presents some approaches to design mobile robots attitude estimators, based on sensor fusion, resorting to KF and CF techniques. These estimators can be implemented in mobile robots navigation systems for dead-reckoning localization or integrated in a global self-localization system, when aided by a system that can estimate the global position in the environment (Carreira et al., 2012a,b, 2013, 2015b).

Thus, this chapter is organized as follows:

- The design of a KF for mobile robot attitude estimation, merging the signals provided from a digital compass with the odometry sensors, is presented in section 2.2. This estimator will be integrated in the self-localization systems presented in chapters 3 and 4, that use ceiling grayscale and depth images to obtain the robot position, respectively. Later, this estimator will also be integrated in some PCA-based localization systems implementations detailed in chapter 6.
- In section 2.3, alternative approaches to design estimators based on CF to estimate variables is presented and analysed. These new class of estimators are designed to merge the signals provided by several sensors based on complementary filtering. Aiming to fuse signals provided from three sensors to estimate the attitude of a mobile robot, this class is analysed in detail, taking into account the sensors that are merged. This class is complemented with the design of a Kalman filter, keeping the complementarity propriety. The results are analysed considering a simulation scenery in section 2.4.

- With the purpose of implementing the optimal CF-based attitude estimators in a mobile robot platform, where the signals are processed by a digital processor, section 2.5 presents the design of equivalent discrete-time CF estimators.
- To validate the estimators design, the discrete-time CF are implemented in a real mobile robot platform and its pose is estimated by dead-reckoning, in section 2.6, using real data extracted from three sensors installed on-board. This attitude estimator will be integrated in a self-localization system, using ceiling depth images, and applied in the navigation system detailed in the chapter 5.
- Finally, section 2.7 presents some conclusions about the designed estimators, presenting same advantages and disadvantages of the different approaches.

2.2 Kalman filter design for optimal attitude and angular slippage estimation

This section presents the design of the KF for mobile robot attitude estimation that will be implemented and experimentally validated in the PCA-based self-localization system, using vision images captured from the ceiling, detailed in the chapter 3 (Carreira et al., 2012a,b). In addition, this KF for attitude estimation will be integrated in the extended PCA-based self-localization system detailed in chapter 4, that computes the robot position from ceiling depth images corrupted with missing data (Carreira et al., 2015a), and in other implementations presented in chapter 6.

When a differential drive mobile robot is moving, disturbances due to uncertainties on the wheels radius, incorrect distance between wheels, asymmetries on the axis, etc. cause slippage between the attitude that can be obtained from odometry and reality. In order to estimate the attitude of the robot, considering these phenomena, this section proposes the implementation of an attitude and bias estimator designed by Pascoal et al. (2000). Following this approach, a new state (s_a) is adopted in the attitude kinematic model, extending it to explicitly take into account the angular slippage.

Thus, the kinematic model that describes the attitude, taking into account the angular slippage is:

$$\dot{\psi} = \omega + s_a + \mu_1 \tag{2.1}$$

$$\dot{s}_a = 0 + \mu_2 \tag{2.2}$$

where s_a is the angular slippage and ω is the angular velocity obtained from the odometry, i.e. $\omega = \omega_o$. It is assumed that s_a is constant or slowly varying i.e. $\dot{s}_a = 0$) and the noise in the odometry and the slippage velocity are assumed to be disturbed by zero-mean uncorrelated white Gaussian noise, $\mu_i \sim N(0, \sigma_i^2)$. Expressing the previous equations in a state-space form, with state vector $\mathbf{x} = \begin{bmatrix} \psi & s_a \end{bmatrix}^T$, the dynamic model is defined as:

$$\dot{\mathbf{x}} = \overbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}^{\mathbf{F}} \mathbf{x} + \overbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}^{\mathbf{B}} \omega + \overbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}^{\mathbf{G}} \mu_{\mathbf{a}} \tag{2.3}$$

$$z = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{H}} \mathbf{x} + \gamma_a \tag{2.4}$$

where $\mu_{\mathbf{a}} = \begin{bmatrix} \mu_1 & \mu_2 \end{bmatrix}^T$ is the corresponding discrete process noise and γ_a is the white Gaussian noise present in the compass measurements ψ_c.

Applying the KF design approach, the classical estimator model is obtained:

$$\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{B}\omega + \mathbf{K}(z - \mathbf{H}\hat{\mathbf{x}}) \tag{2.5}$$

$$\hat{z} = \mathbf{H}\hat{\mathbf{x}} \tag{2.6}$$

where $\mathbf{K} = \begin{bmatrix} K_1 & K_2 \end{bmatrix}^T$ is a the Kalman gain matrix, computed through the stochastic characterization of the uncertainty present in the sensors measurements. Being $\mathbf{R}$ the covariance of the white Gaussian noise associated with the attitude sensor (measurement noise), and $\mathbf{Q}$ the covariance of the white Gaussian noise associated with the odometry (process noise), the optimal gains can be obtained as usual, i.e.

$$\mathbf{K} = \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1} \tag{2.7}$$

where $\mathbf{P}$ is the covariance of the estimation error, solution of the algebraic Riccati equation,

$$\mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T - \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P} + \mathbf{Q} = 0. \tag{2.8}$$

Figure 2.1 presents the corresponding attitude estimator block diagram where ψ is the attitude measured in the compass, ψ_c, and $\dot{\psi}$ is the angular velocity provided from the encoders' odometry, ω_o.

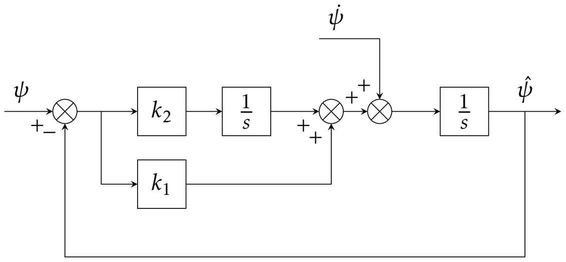

Figure 2.1: Kalman filter block diagram for attitude estimation.

The KF to provide optimal estimations requires the computation of the optimal gains for the matrix **K**, resorting to a classical Kalman filter design. This can be achieved through the stochastic characterization of the uncertainty present in the sensors measurements.

Since the attitude estimator is to be implemented in a digital processor, a discrete model must be obtained. Following the assumption that the angular velocity ω is constant between sampling instants, i.e. Zero Order Hold (ZOH) assumption, the discrete-time linear model results:

$$\mathbf{x}(k+1) = \overbrace{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}^{\mathbf{F}} \mathbf{x}(k) + \overbrace{\begin{bmatrix} T \\ 0 \end{bmatrix}}^{\mathbf{B}} \omega(k) + \overbrace{\begin{bmatrix} T & \frac{T^2}{2} \\ 0 & T \end{bmatrix}}^{\mathbf{G}} \mu(k) \tag{2.9}$$

$$z(k) = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{H}} \mathbf{x}(k) + \gamma(k) \tag{2.10}$$

where T is the sampling time. The corresponding KF for discrete-time optimal estimation is represented by the compact structure given by:

$$\hat{\mathbf{x}}(k+1) = \mathbf{F}\hat{\mathbf{x}}(k) + \mathbf{B}\omega(k) + \mathbf{K}[z(k) - \mathbf{H}\hat{\mathbf{x}}(k)] \tag{2.11}$$

where $\mathbf{K} = \begin{bmatrix} K_1 & K_2 \end{bmatrix}^T$ are the Kalman gains, computed based on the discrete Kalman filter design as follows:

$$\mathbf{K} = \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1} \tag{2.12}$$

being **P** computed by solving the discrete-time algebraic Riccati equation (Brown and Hwang, 1997).

The robot angular velocity can be computed from the odometry numerical differential as the follow:

$$\omega_o(k) = \left(\frac{(\alpha_r(k) - \alpha_r(k-1)) - (\alpha_l(k) - \alpha_l(k-1))}{T} \right) \frac{r}{l} \tag{2.13}$$

where r is the wheels radius, l is the distance between wheels, α_r and α_l are the right and left accumulated wheels angles, respectively, computed from the encoder pulses count. This estimator, relying on the numerical differentiation of instantaneous odometric readings, is represented on the block diagram of Fig. 2.2.

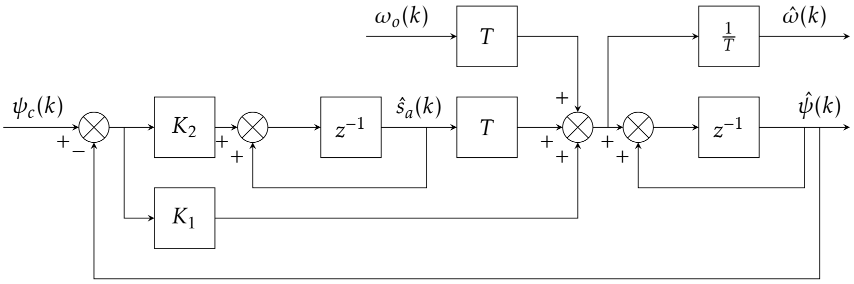

Figure 2.2: Block diagram of attitude estimator with numerical differentiation.

The use of a numerical differentiation on the instantaneous sensor readings must be avoided, as this operation amplifies the noise present. Since the accumulated angles of the wheels (α_r and α_l) are directly available from the hardware, the attitude measurement can be calculated by:

$$\psi_o(k) = (\alpha_r - \alpha_l)\frac{r}{l} \tag{2.14}$$

This change in the odometry signal leads to the design of an equivalent KF, as shown in the Fig. 2.3. The estimate of the angular velocity will be used in the LPV, equation (3.26), being obtained through a numerical differentiation of the estimated attitude of the robot:

$$\hat{\omega}(k) = \frac{\hat{\psi}(k) - \hat{\psi}(k-1)}{T} \tag{2.15}$$

This solution, although calculated with a numerical difference, is filtered by the attitude estimator. Thus, the attitude estimator uses the raw data provided from the encoders,

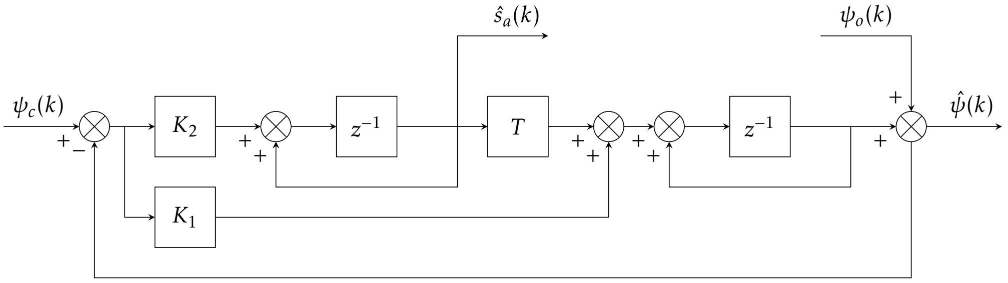

Figure 2.3: Block diagram of attitude estimator without numerical difference.

instead of measurements obtained from their direct numerical difference, which operation increase the sensors noise.

2.3 Complementary filters design for attitude estimation: continuous-time

2.3.1 Complementary filter design for multiple sensors fusion - CF$^{\mathrm{m}}$

With the great development of MEMS, an increasing number of sensors are available in the market, becoming easier and cheaper to purchase sensors to be implemented in measuring systems. Moreover, considering that each sensor has its advantages and limitations, it is usual the use of more than one sensor to estimate the same variable. When redundant measurement systems are applied to estimate the same variable, CF is typically implemented in sensor fusion. CF have been widely used to merge two signals, using a LPF and a HPF, with complementary frequency bands. One of the methods used to design CF is the use of 1$^{\text{st}}$ order filters, whose Transfer Functions (TF) that relates the measured variable with the estimated are given by the following equations (Brown and Hwang, 1997):

$$T_1(s) = \frac{1}{\tau s + 1} \tag{2.16}$$

$$T_2(s) = \frac{\tau s}{\tau s + 1} \tag{2.17}$$

where $T_1(s)$ and $T_2(s)$ are the LPF and the HPF transfer function, respectively, and τ is the time constant, inversely proportional to the cut-off frequency, $\tau = 1/f_c$. The Bode diagrams corresponding to the equations (2.16)–(2.17) are depicted in Fig. 2.4.

However, when it is intended to combine signals provided from more than two sensors,

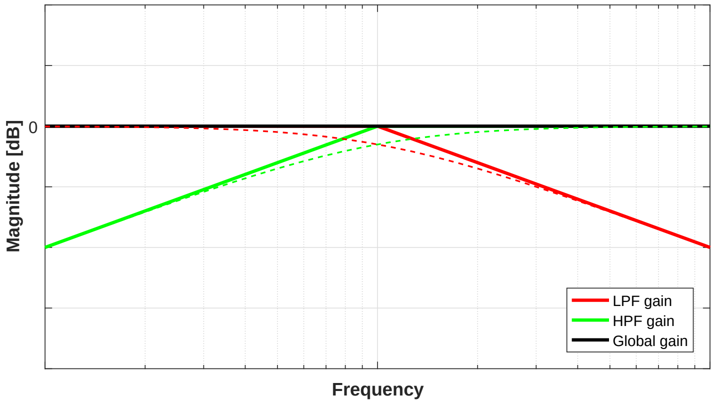

Figure 2.4: CF Bode diagram with two inputs.

using the complementary filter techniques, it becomes necessary to implement estimators with more than two inputs. This section presents an approach to design CF with multiple inputs (CF$^{\mathrm{m}}$), increasing the number of possible combinations to implement the sensor fusion. This method allows the CF design to estimate a variable, taking into account the number and the type of available sensors. Thus, the approach presented in this section proposes an estimators class inspired on a CF, denoted as CF$^{\mathrm{m}}$, that merges signals provided by several (m) sensors, using only part of frequency spectrum from each one. The implementation of all sensors together will cover the whole spectrum, providing distortion-less estimates of the unknown quantities. This means that one sensor complements others in frequency domain, thus the name complementary. The CF$^{\mathrm{m}}$ estimator considers m sensors signals merged in different, yet complementary frequency bands. The frequency spectrum is split in m bands. The complementarity is achieved if the estimator output has an unitary magnitude gain over whole frequency spectrum.

Consider the CF design to estimate a variable $\psi(t)$, abbreviated in the sequel by ψ, combining m filters with frequency bands complementary between each other. Lets consider $\psi(s)$ the Laplace Transform (LT) of the signal ψ, and $T_i(s)$, $i = 1,...,m$ the TF of each filter that composes the CF estimator, the global TF that relates the measured variable $\psi(s)$ with the estimated $\hat{\psi}(s)$ is:

$$\hat{\psi}(s) = \psi(s)\sum_{i=1}^{m} T_i(s) \tag{2.18}$$

Considering that each filter has a complementary frequency band with each other, it is a required condition that the sum of TF has unity gain for all frequency spectrum.

$$\sum_{i=1}^{m} T_i(s) = T_1(s) + T_2(s) + ... + T_m(s) = 1 \tag{2.19}$$

Consider the follow generic TF that relates to the measurement variable, $\psi(s)$, with the estimated $\hat{\psi}(s)$,

$$T(s) = \frac{\hat{\psi}(s)}{\psi(s)} = \frac{b_n s^n + b_{n-1} s^{n-1} + ... + b_0}{a_n s^n + a_{n-1} s^{n-1} + ... + a_0} \tag{2.20}$$

Thus, to ensure the equality presented in equation (2.19), the numerator and denominator coefficients, must be equal, i.e., $a_0 = b_0, a_1 = b_1, ..., a_n = b_n$. To design a CF that receives signals provided by m sensors, the equation (2.20) will take the order $n = m - 1$, with purpose to be decomposed in m filters TF, such that the sum ensures the necessary equality of equation (2.19). It is assumed, in this estimator, that the higher order coefficients has a unitary value, $a_n = b_n = 1$. Attending to this assumptions, the proposed CF^m is composed by a combination of m filters with complementary frequency bands, where the TF corresponding to each filter is given below:

$$T_1(s) = \frac{a_0}{s^n + a_{n-1} s^{n-1} + ... + a_1 s^1 + a_0} \tag{2.21}$$

$$T_2(s) = \frac{a_1 s^1}{s^n + a_{n-1} s^{n-1} + ... + a_1 s^1 + a_0} \tag{2.22}$$

$$\vdots$$

$$T_{m-1}(s) = \frac{a_{n-1} s^{n-1}}{s^n + a_{n-1} s^{n-1} + ... + a_1 s^1 + a_0} \tag{2.23}$$

$$T_m(s) = \frac{s^n}{s^n + a_{n-1} s^{n-1} + ... + a_1 s^1 + a_0} \tag{2.24}$$

where $T_1(s) + T_2(s) + ... + T_{m-1}(s) + T_m(s) = 1$.

Figures 2.5–2.7 presents the Bode diagrams corresponding to the TF given by equations (2.21)–(2.24), for different inputs number. Thus, as it is possible to observe the Bode diagrams depicted in both figures, the equation (2.21) represents a LPF, the equation (2.24) represents a HPF and the remaining equations (2.22)–(2.23) represents Band-Pass Filters (BPF). Both filters were designed with the same order number, according with number of CF inputs $n = m - 1$. Analysing the Bode diagrams it is possible to observe that the CF

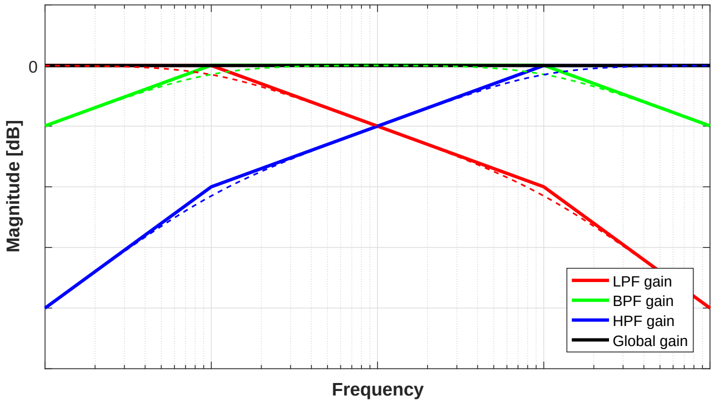

Figure 2.5: CF Bode diagrams with three inputs ($m = 3$).

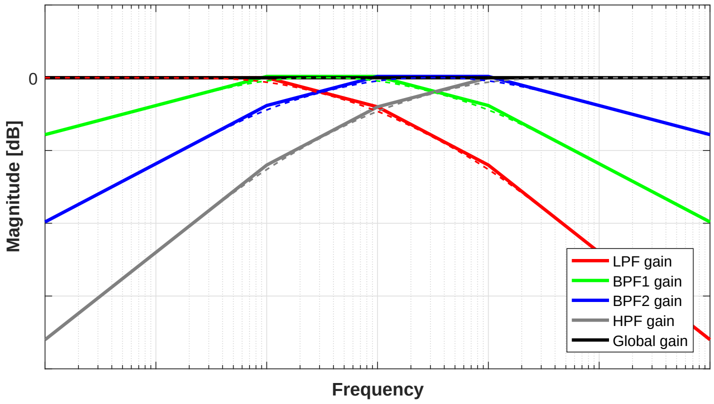

Figure 2.6: CF Bode diagrams with four inputs ($m = 4$).

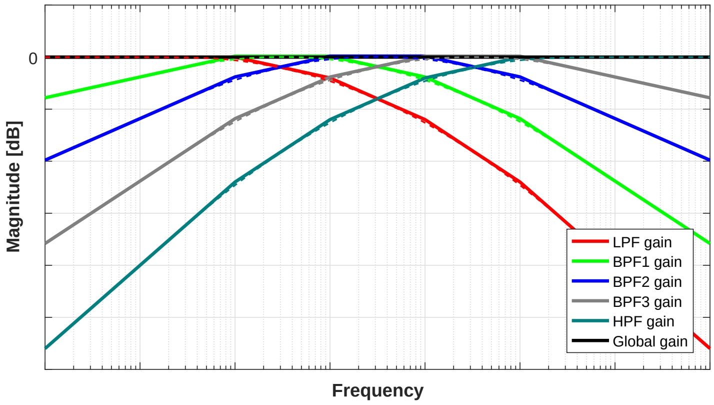

Figure 2.7: CF Bode diagrams with five inputs ($m = 5$).

covers all frequencies spectrum with unity gain, whatever the inputs number used in sensor fusion.

The sensor fusion implementation with more measured signals has several advantages.

In addition to increasing the number of signals used in sensor fusion, allows to adjust the sensors to different CF frequency bands, availing the best that each one can contribute to the estimated variable. Nowadays there are different sensors, with inherent characteristics to each one, where the use of their signals in different frequency bands leads them to produce different estimates. For example, consider a sensor fusion system design where the estimation can be obtained through the signals provided from three types of sensors: position (ψ), velocity ($\dot{\psi}$) and acceleration ($\ddot{\psi}$), that are available to be implemented. In spite of, in real applications it has been usual to implement certain measurements combinations in the LPF and the HPF, when a CF is used to merge two signals with these three measurements types it is possible to implement a maximum of nine different combinations (see Table 2.1).

Table 2.1: CF sensor fusion combinations.

no	LPF	HPF
1	ψ	ψ
2	$\dot{\psi}$	ψ
3	$\ddot{\psi}$	ψ
4	ψ	$\dot{\psi}$
5	$\dot{\psi}$	$\dot{\psi}$
6	$\ddot{\psi}$	$\dot{\psi}$
7	ψ	$\dot{\psi}$
8	$\dot{\psi}$	$\ddot{\psi}$
9	$\ddot{\psi}$	$\ddot{\psi}$

However, with the implementation of a CF with three inputs, as it is depicted in Table 2.2, there are more possibilities for combining them in the different frequency bands. In section 2.3.2 the design of CF with three inputs will be analysed with more detail, presenting the block diagrams of some of these combinations.

As seems obvious, when the number of signals in a sensor fusion increases, the combinations possibilities increases significantly. Table 2.3 shows the maximum CF combinations that can be possible to design, given the sensors' number that is intended to merge, relating with the different measurements types that can be obtained from them.

Table 2.2: CF^3 sensor fusion combinations.

no	LPF	BPF	HPF
1	ψ	ψ	ψ
2	$\dot{\psi}$	ψ	ψ
3	$\ddot{\psi}$	ψ	ψ
4	ψ	$\dot{\psi}$	ψ
5	$\dot{\psi}$	$\dot{\psi}$	ψ
6	$\ddot{\psi}$	$\dot{\psi}$	ψ
7	ψ	$\ddot{\psi}$	ψ
8	$\dot{\psi}$	$\ddot{\psi}$	ψ
9	$\ddot{\psi}$	$\ddot{\psi}$	ψ
10	ψ	ψ	$\dot{\psi}$
11	$\dot{\psi}$	ψ	$\dot{\psi}$
12	$\ddot{\psi}$	ψ	$\dot{\psi}$
13	ψ	$\dot{\psi}$	$\dot{\psi}$
14	$\dot{\psi}$	$\dot{\psi}$	$\dot{\psi}$
15	$\ddot{\psi}$	$\dot{\psi}$	$\dot{\psi}$
16	ψ	$\ddot{\psi}$	$\dot{\psi}$
17	$\dot{\psi}$	$\ddot{\psi}$	$\dot{\psi}$
18	$\ddot{\psi}$	$\ddot{\psi}$	$\dot{\psi}$
19	ψ	ψ	$\ddot{\psi}$
20	$\dot{\psi}$	ψ	$\ddot{\psi}$
21	$\ddot{\psi}$	ψ	$\ddot{\psi}$
22	ψ	$\dot{\psi}$	$\ddot{\psi}$
23	$\dot{\psi}$	$\dot{\psi}$	$\ddot{\psi}$
24	$\ddot{\psi}$	$\dot{\psi}$	$\ddot{\psi}$
25	ψ	$\ddot{\psi}$	$\ddot{\psi}$
26	$\dot{\psi}$	$\ddot{\psi}$	$\ddot{\psi}$
27	$\ddot{\psi}$	$\ddot{\psi}$	$\ddot{\psi}$

Table 2.3: Maximum CF$^{\mathrm{m}}$ possible combinations taking into account the available sensors types and the inputs' number.

CF inputs	2 **sensor types**	3 **sensor types**
2 inputs	$2^2 = 4$	$3^2 = 9$
3 inputs	$2^3 = 8$	$3^3 = 27$
4 inputs	$2^4 = 16$	$3^4 = 81$
5 inputs	$2^5 = 32$	$3^5 = 243$
$\vdots$	$\vdots$	$\vdots$
m inputs	2^m	3^m

2.3.2 Complementary filter design with three frequency bands - CF^3

In the section 2.3.1, an approach was presented to merge signals provided by a set of sensors, based on complementary filtering methodology. This section aims the design and analysis of a CF, merging the signals provided by three sensors. This estimator, denoted as CF^3, is characterized by having three inputs, which signals are merged in complementary frequency bands, as referred above (Carreira et al., 2015b). The CF^3 is composed by a LPF, a BPF and a HPF. Complementarity is achieved if the estimator output has an unitary magnitude gain over whole frequency spectrum.

Lets focus on the CF^3 design to estimate the attitude of a mobile robot with three on-board sensors. Assuming that the sensors provide measurements related to the attitude, the filters are complementary if the following equality holds:

$$T_1(s) + T_2(s) + T_3(s) = 1 \tag{2.25}$$

where $T_1(s)$, $T_2(s)$ and $T_3(s)$ are the transfer functions from each of the sensors $i = 1,...,3$ to the robot attitude ψ, i.e. the CF^3 output.

Furthermore, being K_1 and K_2 positive parameters, then, one possible TF that relates the LT of the measured attitude $\psi(s)$ with its estimate $\hat{\psi}(s)$, is:

$$\hat{\psi}(s) = \frac{s^2 + K_1 s + K_2}{s^2 + K_1 s + K_2}\psi(s) \tag{2.26}$$

This second order transfer function can be decomposed into three transfer functions, ensuring the equality given by equation (2.25), for the purpose of obtaining the transfer function that relates each input with the CF^3 output:

$$\hat{\psi}(s) = T_1(s)\psi(s) + T_2(s)\psi(s) + T_3(s)\psi(s) \tag{2.27}$$

where $T_1(s)$ is a HPF, $T_2(s)$ a BPF, and $T_3(s)$ a LPF, characterized respectively by the following transfer functions:

$$T_1(s) = \frac{s^2}{s^2 + K_1 s + K_2} \tag{2.28}$$

$$T_2(s) = \frac{K_1 s}{s^2 + K_1 s + K_2} \tag{2.29}$$

$$T_3(s) = \frac{K_2}{s^2 + K_1 s + K_2}. \tag{2.30}$$

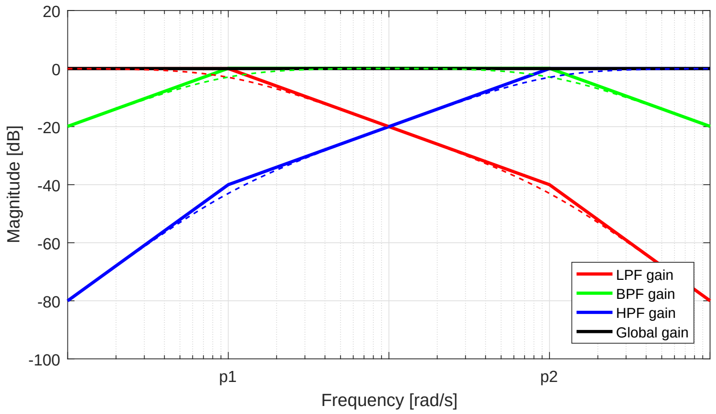

Figure 2.8: CF^3 Bode diagram.

Figure 2.8 shows the Bode diagrams corresponding to the transfer functions given by equations (2.28)–(2.30). Unitary gain along the whole frequency spectrum is thus obtained. All filters have the same two eigenvalues, i.e. the roots of the characteristic equation, given by:

$$p_{T_{1,2,3}} = \left\{ -\frac{K_1 \pm \sqrt{K_1^2 - 4K_2}}{2} \right\} \tag{2.31}$$

and the zeros for each filter can be obtained as follows,

$$z_{T_1} = \{0\ ,\ 0\} \quad , z_{T_2} = \{0\} \quad , z_{T_3} = \{\} \tag{2.32}$$

Reorganizing the transfer function given by equation (2.26), the model of a CF^3 could be analytically represented as the differential equation:

$$\ddot{\hat{\psi}} = \ddot{\psi} + K_1(\dot{\psi} - \dot{\hat{\psi}}) + K_2(\psi - \hat{\psi}), \tag{2.33}$$

with the corresponding block diagram depicted in Fig. 2.9.

Notice that the CF^3 has three inputs, allowing to merge the measurements provided by three sensors, each one corresponding to a different physical quantity, that can be denominated generically as (attitude) position (ψ), velocity ($\dot{\psi}$), and acceleration ($\ddot{\psi}$). The position signal is filtered by the LPF, the velocity by the BPF, and the acceleration by the HPF. The filters gains can be computed based on its parameters without the need to know the stochastic noise and any system physical model. Hence, to implement the CF^3, it is only

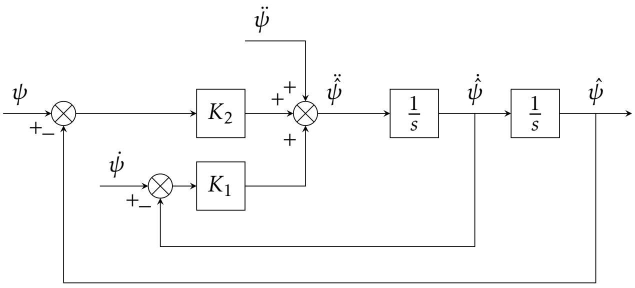

Figure 2.9: CF^3 block diagram.

needed to take into account the CF property along the frequency spectrum, and the characteristic of the individual sensors used, allocating the sensor signals to the corresponding filter input.

To analyse if the obtained CF^3 is optimal, the model given by equation (2.33) has been written with a space-state representation, considering the state and output vectors $\hat{\mathbf{x}} = \begin{bmatrix} \hat{\psi} & \hat{\dot{\psi}} \end{bmatrix}^T$ and $\mathbf{z} = \hat{\mathbf{x}}$, respectively:

$$\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{B}\ddot{\psi} + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \tag{2.34}$$

$$\mathbf{z} = \mathbf{H}\mathbf{x} \tag{2.35}$$

with:

$$\mathbf{F}_{\mathbf{CF^3}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{B}_{\mathbf{CF^3}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \mathbf{H}_{\mathbf{CF^3}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.36}$$

and the CF^3 gain matrix is given by:

$$\mathbf{K}_{\mathbf{CF^3}} = \begin{bmatrix} 0 & 0 \\ K_2 & K_1 \end{bmatrix}. \tag{2.37}$$

Notice that the gain matrix $\mathbf{K}_{\mathbf{CF^3}}$ is sparse, corresponding to a non-convex estimation problem. The most important consequence is that optimality is lost in general, i.e. tuning the gains K_1 and K_2 does not render the CF^3 optimal. To attain optimality, a linear Kalman filter must be designed for the problem at hand. The design of the equivalent optimal estimator will be presented in section 2.3.3.

2.3.2.1 Alternative CF^3 structures

The sensor fusion approach based on the CF^3 depicted in Fig. 2.9 considers that each sensor measures a different physical quantity, related with the same variable. In this case, a position variable is estimated based on position, velocity and acceleration measurements. However, as it was discussed in section 2.3.1, the sensor fusion based on complementary filtering allows the possibility of different combinations according with the available sensors. For example, looking to the 1st line of Table 2.2, it is possible to observe a sensor fusion considering three position signals. This fusion combination can be obtained by changing the architecture model depicted in Fig 2.9 by other equivalent, with 3 inputs position, which is shown in Fig. 2.10

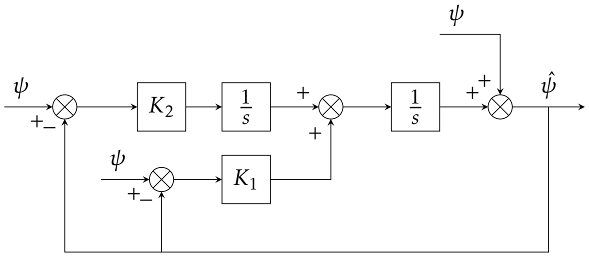

Figure 2.10: CF^3 block diagram with three inputs (LPF: ψ, BPF:ψ, HPF:ψ).

As it was detailed in Table 2.2, this approach allows to obtain other alternative CF^3 structures, tacking into account the sensors intended for use in the sensor fusion. Now, considering a practical application which aims to estimate the attitude angle of a mobile robot, merging the signals providing by three sensors, with two different physical quantities, i.e., one velocity plus two position sensors.

Analysing the Table 2.2, it is possible to find three combinations with one velocity and two position signals inputs. Figures 2.11, 2.12 and 2.13 show the architecture of the combinations number 2, 4 and 12, respectively. All of these CF^3 models are represented by the same transfer functions, equation (2.27), and have the same frequency response. Thus, it is possible to conclude that the proposed CF^3 design methodology also allows to obtain CF^3 equivalent estimators, changing the filter structure, while keeping the transfer functions of each filter.

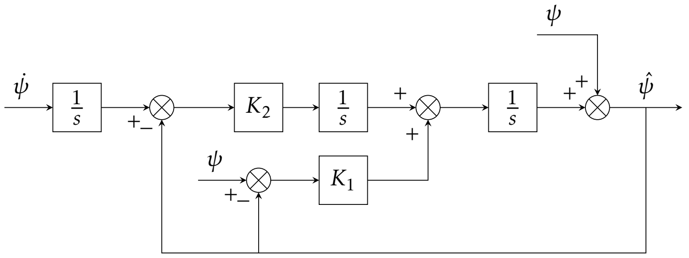

Figure 2.11: CF^3 block diagram with three inputs (LPF: $\dot{\psi}$, BPF:ψ, HPF:ψ).

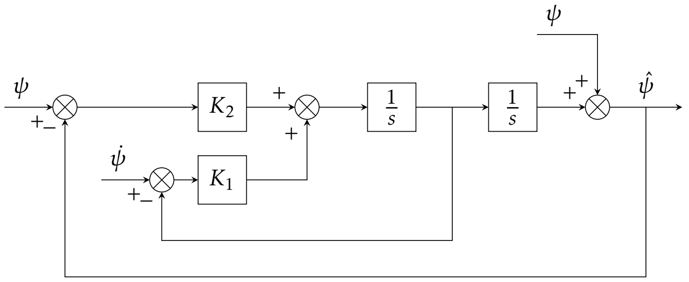

Figure 2.12: CF^3 block diagram with three inputs (LPF: ψ, BPF:$\dot{\psi}$, HPF:ψ).

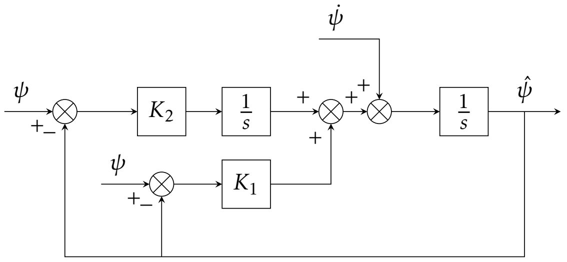

Figure 2.13: CF^3 block diagram with three inputs (LPF: ψ, BPF:ψ, HPF:$\dot{\psi}$).

2.3.3 Optimal complementary Kalman filter design with three frequency bands - CKF^3

The section 2.3.2 presents an estimator design to merge the signals provided by three sensors. Although the parameters can be easily tuned according with the desired frequency bands, looking to the gain matrix $\mathbf{K}_{\mathbf{CF}^3}$ shown in equation (2.37), it is possible to observe that it is sparse, with a corresponding non-convex estimation problem (Carreira et al., 2015b). To attain optimality, a linear Complementary Kalman Filter (CKF^3), equivalent to the CF^3, is proposed in this section, for the problem at hand. In that case, the gain matrix has the same dimensions than the presented in equation (2.37) but the $\mathbf{K}_{\mathbf{CKF}^3}$ matrix is full. Such approach allows the design of a fusion solution that best suits the sensors

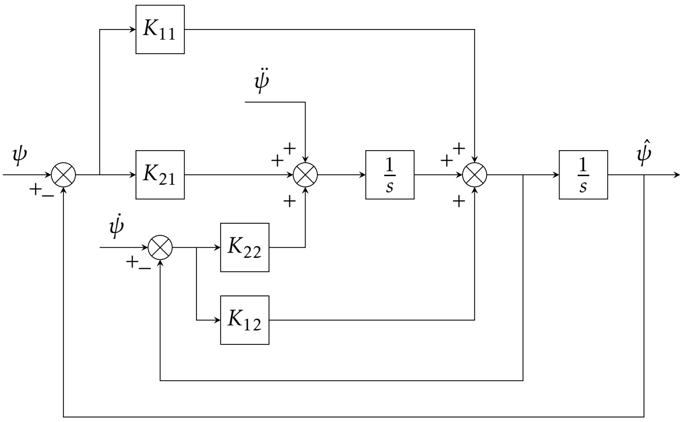

Figure 2.14: CKF^3 block diagram.

characteristics, preserving the corresponding filter frequency band, while providing optimal estimates.

Pursuing the classical Kalman filter design approach, a full Kalman gain matrix $\mathbf{K}_{\mathbf{CKF^3}}$ is obtained, with four gains to be determined, i.e.,

$$\mathbf{K}_{\mathbf{CKF^3}} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}. \tag{2.38}$$

Using (2.38) in the model, equations (2.34)–(2.36) leads to the CKF^3 filter shown in Fig. 2.14. Notice that two more gains are now present in this model, when compared with the suboptimal model depicted in Fig. 2.9. Analysing the CKF^3 structure presented in Fig. 2.14, it can be concluded that the transfer functions relating the signal from each sensor with the corresponding filter output are as follows:

$$F_1(s) = \frac{\hat{\psi}(s)}{\alpha(s)} = \frac{1}{(1+K_{12})s^2 + (K_{11}+k_{22})s + K_{21}} \tag{2.39}$$

$$F_2(s) = \frac{\hat{\psi}(s)}{\omega(s)} = \frac{K_{12}s + K_{22}}{(1+K_{12})s^2 + (K_{11}+K_{22})s + K_{21}} \tag{2.40}$$

$$F_3(s) = \frac{\hat{\psi}(s)}{\psi(s)} = \frac{K_{11}s + K_{21}}{(1+K_{12})s^2 + (K_{11}+K_{22})s + K_{21}} \tag{2.41}$$

where $\omega(s)$ and $\alpha(s)$ are the LT of signals $\dot{\psi}$ and $\ddot{\psi}$, respectively. The global TF will be given by the following equation:

$$\begin{aligned}\hat{\psi}(s) &= F_1(s)\alpha(s) + F_2(s)\omega(s) + F_3(s)\psi(s) = \\ &= F_1(s)s^2\psi(s) + F_2(s)s\psi(s) + F_3(s)\psi(s)\end{aligned} \tag{2.42}$$

Finally, equation (2.42) can be written as equation (2.27), being $T_1(s) = F_1(s)s^2$, $T_2(s) = F_2(s)s$ and $T_3(s) = F_3(s)$. Moreover, analysing the transfer functions of each filter, equations (2.39)–(2.41), it can be concluded that the condition expressed by equation (2.25) is also satisfied resulting a global transfer function as follows:

$$\hat{\psi}(s) = \frac{(1+K_{12})s^2 + (K_{11}+K_{22})s + K_{21}}{(1+K_{12})s^2 + (K_{11}+K_{22})s + K_{21}} \psi(s) \tag{2.43}$$

In conclusion, the use of the structure associated to the linear KF leads also to a CF, with the advantage of attaining optimal performance, preserving the minimal representation and the stability.

Figure 2.15 shows the CKF^3 block diagram and Fig. 2.16 the corresponding Bode diagram. Once again, three filters working in complementary frequency bands and with unitary global gain along the whole frequency spectrum are observed.

The Bode diagrams are designed according to the gain matrix, since there is a

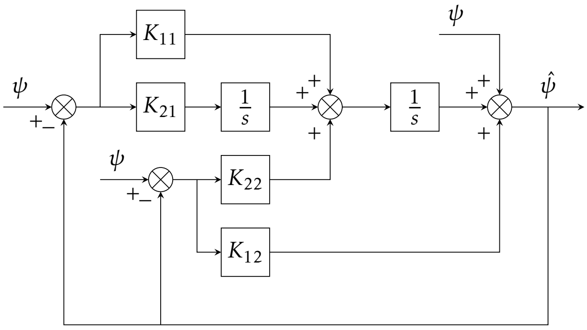

Figure 2.15: CKF^3 block diagram with three inputs (LPF: ψ, BPF:ψ, HPF:ψ).

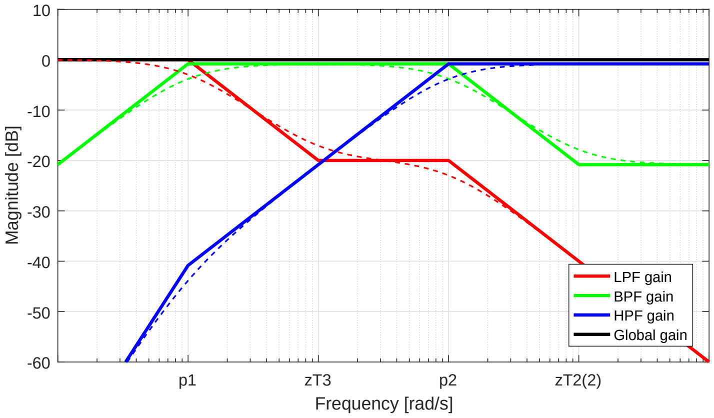

Figure 2.16: CKF^3 Bode diagram.

relationship between the filter gains and the corresponding system eigenvalues and zeros, which can be obtained from the transfer functions of each filter. Thus, all filters have the same two eigenvalues, given by:

$$p_{T_{1,2,3}} = \left\{ -\frac{K_{11} + K_{22} \pm \sqrt{(K_{11} + K_{22})^2 - 4(1 + K_{12})K_{21}}}{2(1 + K_{12})} \right\} \tag{2.44}$$

and the zeros for each filter can be obtained as follows:

$$z_{T_1} = \{0\ ,\ 0\} \quad , z_{T_2} = \left\{0\ ,\ -\frac{K_{22}}{K_{12}}\right\} \quad , z_{T_3} = \left\{-\frac{K_{21}}{K_{11}}\right\}. \tag{2.45}$$

For the CKF^3 to provide optimal estimations, it requires the computation of the optimal gains for the matrix $\mathbf{K}_{\mathbf{CKF}^3}$, resorting to a classical KF design. This can be achieved through the stochastic characterization of the uncertainty present in the sensors measurements. Being **R** the covariance of the white Gaussian noise associated with the LPF and BPF (measurement noise), and **Q** the covariance of the white Gaussian noise associated with the HPF (process noise), the CKF^3 optimal gains can be obtained as usual, i.e., by the equation (2.7), where **P** is the covariance of the estimation error, solution of the algebraic Riccati equation, computed by equation (2.8).

2.3.3.1 Alternative CKF^3 structures

As with the CF^3, the CKF^3 also allows to obtain alternative structures to merge the signal provided by the same three sensors (see Fig. 2.17–2.19). The implementation of these alternative structures enable to merge the signals from each sensor, considering different combinations of frequencies bands. These structures are all represented by the same

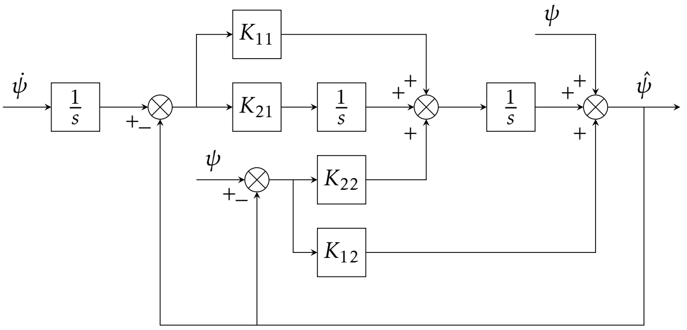

Figure 2.17: CKF^3 block diagram with three inputs (LPF: $\dot{\psi}$, BPF:ψ, HPF:ψ).

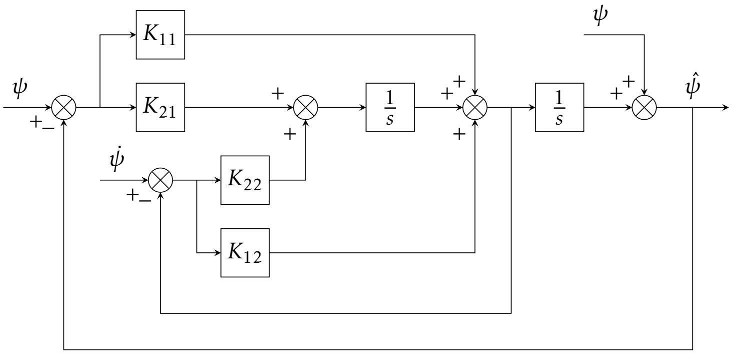

Figure 2.18: CKF^3 block diagram with three inputs (LPF: ψ, BPF:$\dot{\psi}$, HPF:ψ).

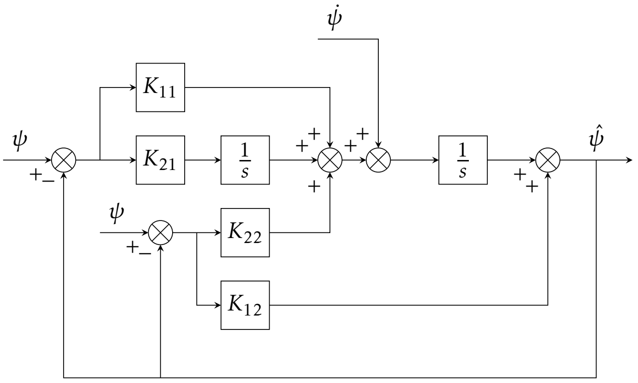

Figure 2.19: CKF^3 block diagram with three inputs (LPF: ψ, BPF:ψ, HPF:$\dot{\psi}$).

transfer functions, the same equation (2.43), and the same frequency response, as the structure developed above and presented in Fig. 2.14.

2.3.4 Optimal Kalman filter design - KF^3

This section presents the optimal estimator design to merge the signals provided by three sensors, based in Kalman filtering. This estimator, denoted as KF^3, is based in the sensory system global model, and is developed through the physical variables of each sensor implemented in the particular estimation process.

Thus, lets consider an estimator design to estimate the attitude of an indoor mobile robot, merging the signals provided by a digital compass, odometry and rate gyro. Analysing the sensors measuring physical variables, it is possible to obtain the equations that relates the mobile robot attitude (ψ) and angular velocity ($\dot{\psi}$) with the signals provided by each sensor:

$$\psi_c = \psi \tag{2.46}$$

$$\psi_o = \psi + e \tag{2.47}$$

$$\omega_r = \dot{\psi} - b \tag{2.48}$$

where ψ_c, ψ_o and ω_r are signals provided by the digital compass, odometry and rate gyro, respectively, e is error caused by the odometry slippage, and b is the bias existing in the rate gyro.

The sensor models presented in equations (2.46)–(2.48) can be written in a space-state representation, considering the state and output vectors $\mathbf{x} = \begin{bmatrix} \psi & b & e \end{bmatrix}^T$ and $\mathbf{z} = \begin{bmatrix} \psi_c & \psi_o \end{bmatrix}^T$, respectively, and $\omega = \omega_r$:

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{B}\omega + \mu \tag{2.49}$$

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{D}\omega + \gamma \tag{2.50}$$

with:

$$\mathbf{F}_{KF^3} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{B}_{KF^3} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{H}_{KF^3} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \mathbf{D}_{KF^3} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.51}$$

Pursuing the Kalman filter design approach, the classical estimator model is obtained:

$$\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{B}\omega + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \tag{2.52}$$

$$\hat{\mathbf{z}} = \mathbf{H}\hat{\mathbf{x}} \tag{2.53}$$

where $\mathbf{K}$ is a full Kalman gain matrix, with six gains, as shown below:

$$\mathbf{K}_{KF^3} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \\ K_{31} & K_{32} \end{bmatrix}. \tag{2.54}$$

Figure 2.20 presents the KF^3 block diagram, that can be obtained from the equations (2.49)–(2.54).

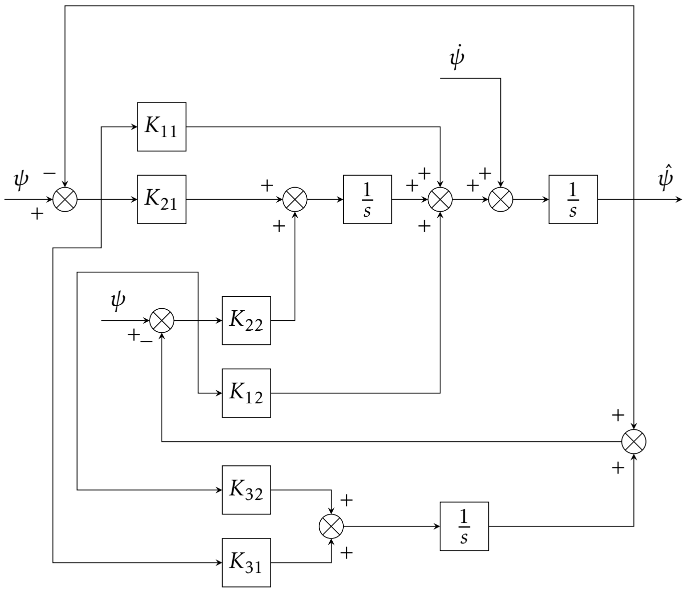

Figure 2.20: KF^3 block diagram with three inputs (LPF: ψ, BPF:ψ, HPF:$\dot{\psi}$).

Analysing the KF^3 structure presented in Fig. 2.20, it can be concluded that the transfer functions relating the signal from each sensor with the corresponding filter output are as follows:

$$F_1(s) = \frac{\hat{\psi}(s)}{\psi(s)} = \frac{K_{11}s^2 + (K_{11}K_{32} - K_{21}K_{31} + K_{22})s + K_{21}K_{32} - K_{22}K_{31}}{den(s)} \tag{2.55}$$

$$F_2(s) = \frac{\hat{\psi}(s)}{\psi(s)} = \frac{K_{12}s^2 + K_{22}s}{(den(s))} \tag{2.56}$$

$$F_3(s) = \frac{\hat{\psi}(s)}{\omega(s)} = \frac{s^2 + K_{32}s}{(den(s))} \tag{2.57}$$

with $den(s)$ given as follows:

$$den(s) = s^3 + (K_{11} + K_{12} + K_{32})s^2 + (K_{11}K_{32} - K_{21}k_{31} + K_{22} + K_{22})s + K_{21}K_{32} - K_{22}K_{31} \tag{2.58}$$

resulting in the global transfer function given by the following equation,

$$\begin{aligned} \hat{\psi}(s) &= F_1(s)\psi(s) + F_2(s)\psi(s) + F_3(s)\omega(s) = \\ &= F_1(s)\psi(s) + F_2(s)\psi(s) + F_3(s)s\psi(s) \end{aligned} \tag{2.59}$$

Lets defined $T_1(s) = F_1(s)$, $T_2(s) = F_2(s)$ and $T_3(s) = F_3(s)s$, the global transfer functions in equation (2.59) can be rewritten as follows:

$$\hat{\psi}(s) = T_1(s)\psi(s) + T_2(s)\psi(s) + T_3(s)\psi(s) \tag{2.60}$$

$$\begin{aligned}\frac{\hat{\psi}(s)}{\psi(s)} &= \frac{s^3 + (K_{11} + K_{12} + K_{32})s^2}{den(s)} + \frac{(K_{11}k_{32} - K_{21}K_{31} + K_{22} + K_{22})s}{den(s)} + \\ &+ \frac{K_{21}k_{32} - K_{22}K_{31}}{den(s)} = 1\end{aligned} \tag{2.61}$$

Thus, it is possible to conclude that KF^3 is designed with a KF structure, but it is also a complementary filter, preserving the complementary equality presented in equation (2.25).

Looking to the Bode diagram presented in Fig. 2.21 it is possible to observe that, with KF^3 estimator, once again, the three filters work in complementary frequency bands and with unitary global gain along the whole frequency spectrum.

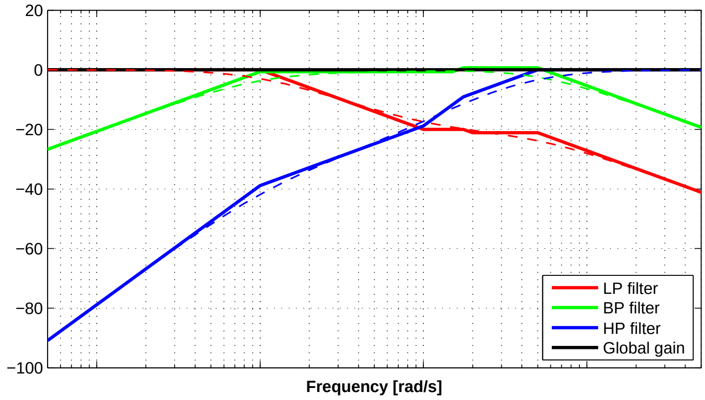

Figure 2.21: KF^3 Bode diagram.

The KF^3 Bode diagrams are, once again, designed according to the gain matrix, since there is a relationship between the filter gains and the corresponding system eigenvalues and zeros, which can be obtained from the transfer functions of each filter. Thus, all filters have the same three eigenvalues that are obtaining solving the cubic polynomial presented in equation (2.58) and the zeros for each filter can be obtained as follows:

$$\begin{aligned}z_{T_1} &= \left\{\frac{-(K_{11}K_{32} - K_{21}K_{31} + K_{22})}{2K_{11}} \pm \frac{\sqrt{(K_{11}K_{32} - K_{21}K_{31} + K_{22})^2 - 4K_{11}(K_{21}K_{32} - K_{22}K_{31})}}{2K_{11}}\right\}, \\ z_{T_2} &= \left\{0\,,\ -\frac{K_{22}}{K_{12}}\right\} \quad, z_{T_3} = \{0\,,\ 0\,,\ -K_{32}\}.\end{aligned} \tag{2.62}$$

Once KF^3 is designed through the Kalman filter approach, the optimal gains for **K** are computed based on white Gaussian noise associated with sensors. Being **R** the covariance of the white Gaussian noise associated with the LPF and BPF (measurement noise), and **Q** the covariance of the white Gaussian noise associated with the HPF (process noise), the KF^3 optimal gains can be obtained as usual, i.e., by the equation (2.7), where **P** is the covariance of the estimation error, solution of the algebraic Riccati equation, computed by equation (2.8).

2.3.4.1 Alternative KF^3 structures

Similarly to CF^3 and CKF^3, the KF^3 also allows to obtain alternative structures to merge the signal provided by the same three sensors (see Fig. 2.22–2.23). The implementation of these alternative structures allows to merge the signals from each sensor, considering different combinations of frequency bands. These structures are all represented by the same transfer functions, equation (2.61), and the same frequency response, as the structure developed above and presented in Fig. 2.20.

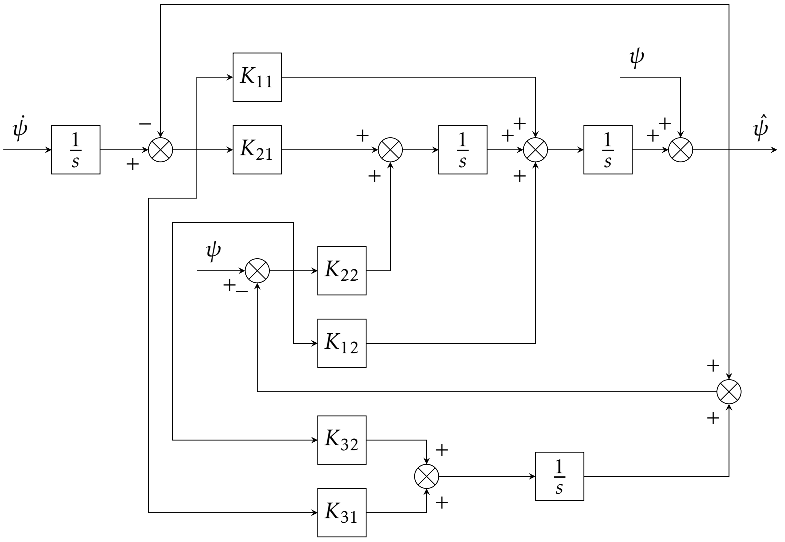

Figure 2.22: KF^3 block diagram with three inputs (LPF: $\dot{\psi}$, BPF:ψ, HPF:ψ).

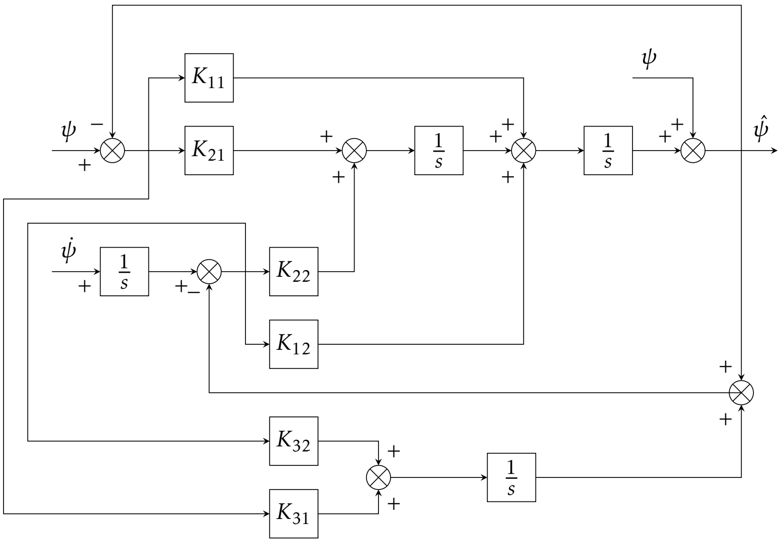

Figure 2.23: KF^3 block diagram with three inputs (LPF: ψ, BPF:$\dot{\psi}$, HPF:ψ).

2.4 Simulation results

The proposed estimators developed in section 2.3 were tested and simulated for the attitude estimation of a mobile robot. The estimated attitude was obtained from the measures provided by different combinations of three sensors with different characteristics, namely a rate gyro, a compass, and odometry. The simulated trajectory has a shape of a square with rounded corners. The robot kept straight during 600 s and turned smoothly 90° to the left during 10 s. After four times, the robot was back to the initial position having performed the round corners square trajectory. In the performed simulations, the following noises associated with the sensors were added to the model:

- Rate gyro - Gaussian noise $\mu_g \sim N(0, 2\times10^{-3})$ and a bias of $0.8°/s$. However, the bias can be compensated, without disturbing the estimator input signal, because it has been observed in a real experiment that after the temperature stabilization that occurs in 20 min the rate gyro bias remains approximately constant (see Fig. 2.24);
- Compass - Gaussian noise $\mu_c \sim N(0, 2\times10^{-3})$ and a magnetic perturbation of $20\sin(\psi_c)$;
- Odometry - Gaussian noise $\mu_o \sim N(0, 4\times10^{-6})$ and a bias of $0.5°/s$. Such a value could change in a real control loop implementation.

Once the sensors stochastic characteristics were known, the attitude estimation was simulated considering several fusion alternatives, with the corresponding optimal gains.

For the two sensor fusion case (CF), the attitude was estimated by the classical KF with two inputs, thus optimal but with only a subset of the sensors available. For the three sensor fusion case, the CF^3 (Fig. 2.11-2.13), the CKF^3 (Fig. 2.17-2.19) and KF^3 (Fig. 2.20 and Fig 2.22-2.23) solutions were exploited. In the last two cases the estimators parameters were tuned, resulting in optimal gains. Tables 2.4 and 2.5 show the Mean Square Error (MSE) for the different combinations, considering the classical CF structure (with two inputs) and the proposed CF^3, CKF^3 and KF^3 structure with three inputs, respectively.

Table 2.4: MSE for different CF - simulation results.

LPF	HPF	MSE [rad^2]
$\dot{\psi}_r$	ψ_o	2.01×10^{-3}
$\dot{\psi}_r$	ψ_c	2.02×10^{-3}
ψ_c	ψ_o	2.62×10^{-2}
ψ_c	$\dot{\psi}_r$	2.53×10^{-2}
ψ_o	ψ_c	1.41×10^{2}
ψ_o	$\dot{\psi}_r$	1.41×10^{2}

Table 2.5: MSE for different CF^3, CKF^3 and KF^3 combinations - simulation results.

LPF	BPF	HPF	MSE [rad^2]		
			CF^3	CKF^3	KF^3
$\dot{\psi}_r$	ψ_c	ψ_o	2.22×10^{-3}	1.91×10^{-3}	1.86×10^{-3}
$\dot{\psi}_r$	ψ_o	ψ_c	2.99×10^{-3}	1.91×10^{-3}	1.86×10^{-3}
ψ_c	$\dot{\psi}_r$	ψ_o	3.22×10^{-2}	2.70×10^{-2}	2.69×10^{-2}
ψ_c	ψ_o	$\dot{\psi}_r$	3.05×10^{-2}	2.61×10^{-2}	2.62×10^{-2}
ψ_o	$\dot{\psi}_r$	ψ_c	1.46×10^{2}	1.46×10^{2}	1.46×10^{2}
ψ_o	ψ_c	$\dot{\psi}_r$	1.46×10^{2}	1.46×10^{2}	1.46×10^{2}

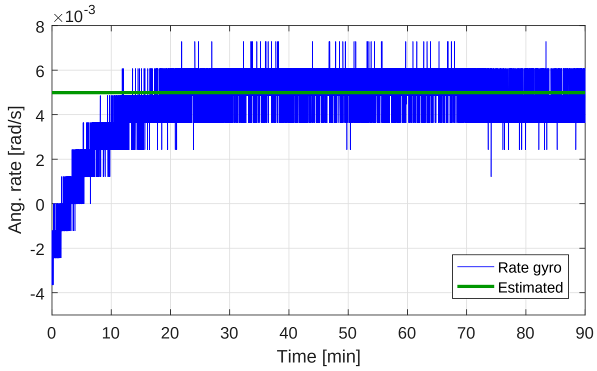

Figure 2.24: Rate gyro bias - experimental results.

Analysing Table 2.4 it can be observed that the CF inputs combination that provides estimations with less MSE is the one that has the rate gyro signal as input of the LPF and the odometry as input of the HPF. The same combination adding now the compass as input of the BPF, thus resulting in a CKF^3 or KF^3, is the best solution. See Table 2.5 where a slightly better performance is achieved. Comparing the results presented in Table 2.5, it is possible to conclude that KF^3 provides better performance than CKF^3. Furthermore, the addition of more one sensor increases the estimator robustness when in presence of erroneous signals or sensor faults.

2.5 Complementary filters for attitude estimation: discrete-time

Lets consider the purpose of implementing the developed CKF^3 and KF^3 in a digital processor to estimate the mobile robot attitude, merging the signals provided by three sensors: compass, encoders and rate gyro. The first two sensors provide attitude (“position” measurement) and the last sensor provides angle rate (“velocity” measurement) Once the sensor fusion is performed by a digital processor, it is assumed that the signals are constant between two sampling times (ZOH assumption). To that purpose, the discrete-time complementary filters design will be presented in this section.

2.5.1 Optimal discrete-time complementary Kalman filter design - DCKF^3

Aiming the sensor fusion implementation with CKF^3 in a digital processor, a Discrete-time Complementary Kalman Filter (DCKF^3) to estimate the variable must be developed.

The DCKF^3 design is developed considering the sensors physical measurements that will be merged and keeps the TF complementarity, providing optimal estimates. The step invariant discrete-time linear model, for a sampling time T, is given by:

$$\hat{\mathbf{x}}(k+1) = \mathbf{F}\hat{\mathbf{x}}(k) + \mathbf{B}\omega(k) + \mathbf{K}(\mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}(k)) \tag{2.63}$$

$$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) \tag{2.64}$$

where $\omega(k)$ is the sampled angle rate, denoted previously as “velocity”, and the system matrices are:

$$\mathbf{F}_{DCKF^3} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \mathbf{B}_{DCKF^3} = \begin{bmatrix} T \\ 0 \end{bmatrix}, \mathbf{H}_{DCKF^3} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \tag{2.65}$$

corresponding to the block diagram depicted in Fig. 2.25. The discrete Kalman gain **K** is a full matrix, with a structure similar to that represented in equation (2.38), computed now based on the discrete KF methodology, through the equation (2.12).

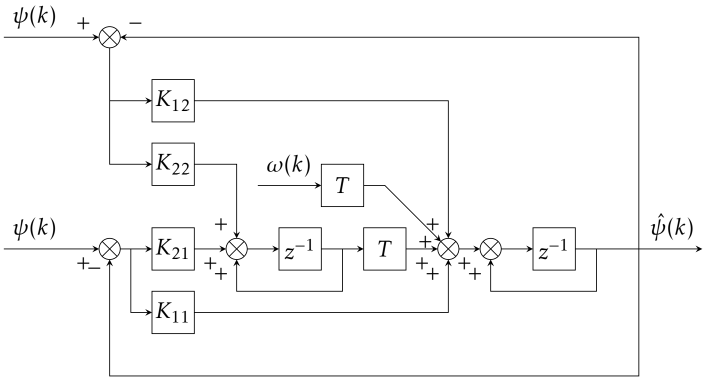

Figure 2.25: DCKF^3 block diagram (LPF: ψ, BPF:ψ, HPF:ω).

Analysing equations (2.63)–(2.65), it can be observed that the DCKF^3 model was obtained from an estimator that receives $\begin{bmatrix} \psi(k) & \psi(k) & \omega(k) \end{bmatrix}^T$ signals in the LPF, BPF and HPF inputs, respectively. However, as shown in section 2.4, for some noise conditions, the best solution to perform the sensor fusion may need different input signals. In this case, the signals provided by the rate gyro, compass and odometry sensors corresponding to have a DCKF^3 structure that receives $\begin{bmatrix} \omega(k) & \psi(k) & \psi(k) \end{bmatrix}^T$ as inputs. Formally, two of these quantities are measurements in the Kalman filter and one is an input. Hence, an equivalent structure has been designed to cope with the connection of the rate gyro, compass and odometry sensors in the correspondent DCKF^3 input, respectively, being the corresponding block diagram depicted in Fig. 2.26.

Analysing this filter, the transfer functions that relate the signals from each sensor with

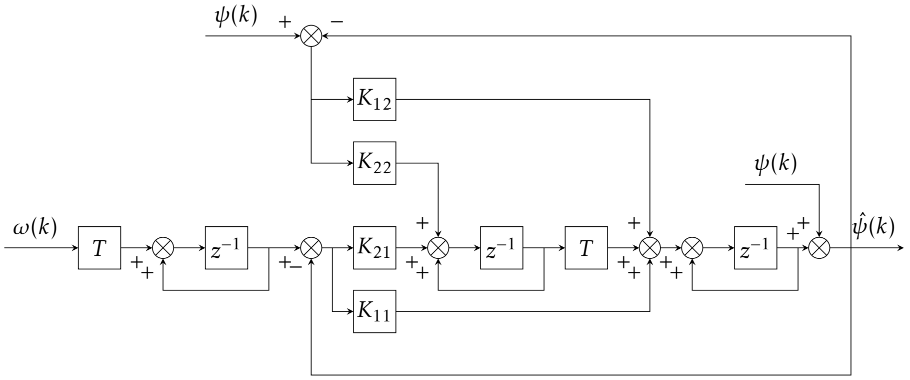

Figure 2.26: DCKF^3 block diagram (LPF: ω, BPF:ψ, HPF:ψ).

the filter output are as follows:

$$F_1(z) = \frac{\hat{\psi}(z)}{\psi(z)} = \frac{z^{-2} - 2z^{-1} + 1}{den(z)} \tag{2.66}$$

$$F_2(z) = \frac{\hat{\psi}(z)}{\psi(z)} = \frac{(-K_{12} + TK_{22})z^{-2} + K_{12}z^{-1}}{den(z)} \tag{2.67}$$

$$F_3(z) = \frac{\hat{\psi}(z)}{\omega(z)} = \frac{(-TK11 + T^2K_{21})z^{-3} + TK_{11}z^{-2}}{(1 - z^{-1})den(z)} \tag{2.68}$$

where

$$den(z) = (1 - K_{12} - K_{11} + TK_{21} + TK_{22})z^{-2} + (K_{11} + K_{12} - 2)z^{-1} + 1. \tag{2.69}$$

The global transfer function is given by:

$$\hat{\psi}(z) = F_1(z)\psi(z) + F_2(z)\psi(z) + F_3(z)\omega(s) \tag{2.70}$$

Establishing $T_1(z) = F_1(z)$, $T_2(z) = F_2(z)$ and $T_3(z) = F_3(z)\frac{1-z^{-1}}{Tz^{-1}}$, the complementary property in the discrete-time is once again satisfied, i.e.

$$T_1(z) + T_2(z) + T_3(z) = 1. \tag{2.71}$$

Thus, the complementary property is also preserved for the discrete-time filters DCKF^3.

2.5.2 Optimal discrete-time Kalman filter design - DKF^3

With the purpose of implementing the KF^3 to estimate the mobile robot attitude in a digital processor, a Discrete Kalman Filter with three input signals (DKF^3) must be implemented.

The attitude estimator detailed in this section will be implemented in the self-localization system used in the navigation experiments presented in chapter 5. The design of this estimator is developed considering the sensors physical measurements that will be merged, assuming that the signals are constant between two sampling times (ZOH assumption).

Lets consider the KF^3 continuous model given by equations (2.52) – (2.53), parameterized with the state-space matrix (2.51). The step invariant discrete-time linear model, for a sampling time T, is given by:

$$\hat{\mathbf{x}}(k+1) = \mathbf{F}\hat{\mathbf{x}}(k) + \mathbf{B}\omega(k) + \mathbf{K}(\mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}(k)) \tag{2.72}$$

$$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) \tag{2.73}$$

where $\omega(k)$ is the sampled angle rate and the system matrices are:

$$\mathbf{F}_{DKF^3} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{B}_{DKF^3} = \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix}, \mathbf{H}_{DKF^3} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{2.74}$$

corresponding to the block diagram depicted in Fig. 2.27. The discrete Kalman gain **K** keeps a full matrix, with a structure similar to that represented in equation (2.54), now computed based on the discrete Kalman methodology, through the equation (2.12).

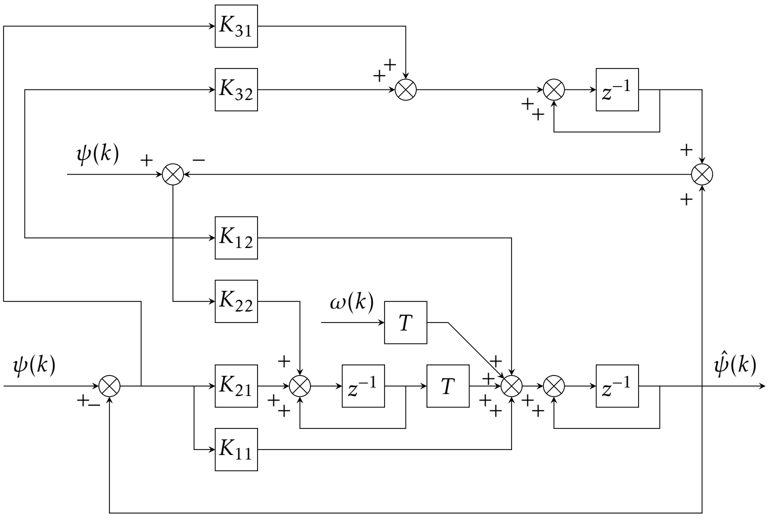

Figure 2.27: DKF^3 block diagram (LPF: ψ, BPF:ψ, HPF:ω).

Similarly to the previous discrete-time estimator (DCKF^3), the DKF^3 is implemented in the same real application, to estimate the mobile robot attitude, merging signals provided by rate gyro, compass and odometry sensors. Thus, to achieve a DKF^3 structure that receives $\begin{bmatrix} \omega(k) & \psi(k) & \psi(k) \end{bmatrix}^T$ as inputs, the same approach than the presented in section 2.5.1 must be implemented to obtain an equivalent structure.

Notice that analysing equations (2.72)–(2.74), it can be observed that the DKF^3 model was obtained from an estimator that receives $\begin{bmatrix} \psi(k) & \psi(k) & \omega(k) \end{bmatrix}^T$ signals in the LPF, BPF and HPF inputs, respectively. Hence, an equivalent structure has been designed to cope with the connection of the rate gyro, compass and odometry sensors, respectively, in the correspondent DKF^3 input. The corresponding block diagram is depicted in Fig. 2.28.

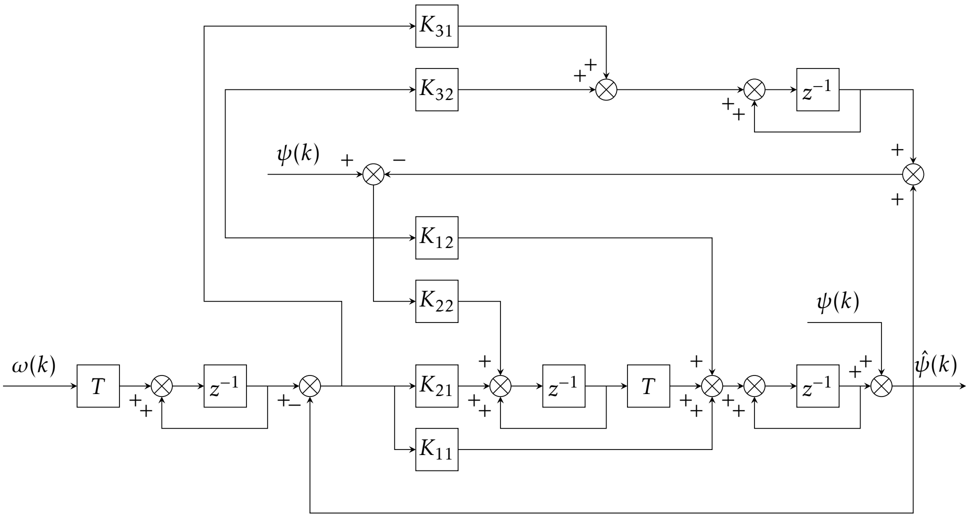

Figure 2.28: DKF^3 block diagram (LPF: ω, BPF:ψ, HPF:ψ).

2.6 Experimental validation

Aiming to validate the proposed estimators in a real application, some experiences were performed to estimate the mobile robot attitude, using a robotic platform with a differential drive configuration. Such a platform is equipped with two encoders coupled to the motors, a digital compass located on the extension arm (robot rear part) to avoid the magnetic interference from the motors, and a rate gyro over the platform (Fig. 2.29). These sensors are connected connected by the Universal Serial Bus (USB) to a laptop, placed on top of the platform, that estimate the mobile robot attitude in real-time.

To test the mobile robot attitude estimation, the DCKF^3 (section 2.5.1) and the DKF^3 (section 2.5.2) were tested considering a trajectory combining both straight lines (constant

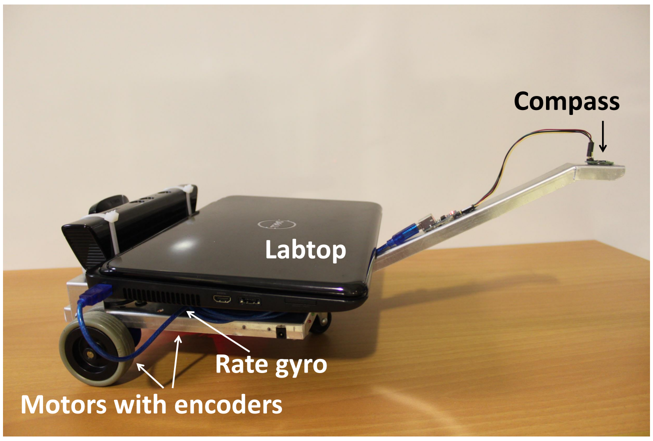

Figure 2.29: Mobile robot platform.

attitude) with semi-circumferences (linear attitude changing), allowing to assess the localization system operation under different experimental conditions. The tests were performed with a 0.1 m/s robot velocity and 5 Hz of sampling frequency. During the robot motion, the real mobile robot trajectory was measured allowing the comparison of the estimated position by the dead-reckoning with the real one (ground truth test) and the corresponding error is analysed. To allow for a ground truth analysis, singular points of the real path were marked at the end of each trajectory. The ground truth path was obtained by interpolation of the marked singular points.

Looking to Table 2.6, it is possible to observe that DKF^3 estimator provides error with less uncertainty than DCKF^3. Figure 2.30 shows the attitude and the angular velocity estimated by the DKF^3, where is possible to see that the estimated variables (position and angular velocity) are very close to the ground truth. Analysing the results depicted in Fig. 2.31, it can be observed that the dead-reckoning results are close to the ground truth trajectory.

Table 2.6: DCKF^3 and DKF^3 estimation error variance - experimental results.

quantity	**DCKF^3**	**DKF^3**
$\hat{\psi}$ $[rad]$	1.9×10^{-2}	1.6×10^{-2}
$\hat{\omega}$ $[rad/s]$	5.3×10^{-3}	3.2×10^{-3}

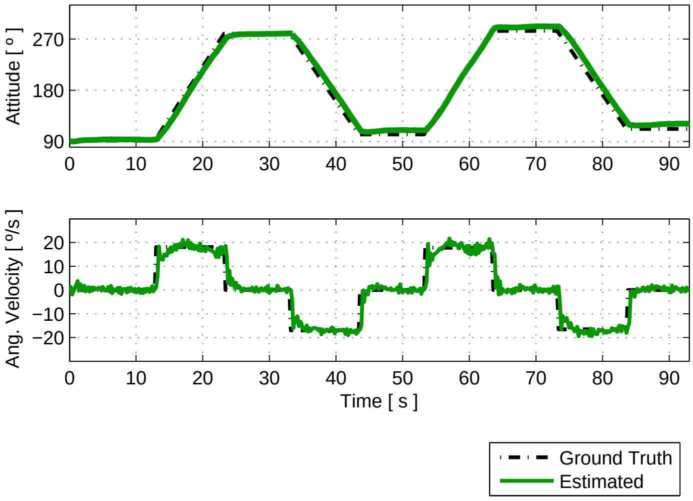

Figure 2.30: Attitude and angular velocity estimated with DKF^3 - experimental results.

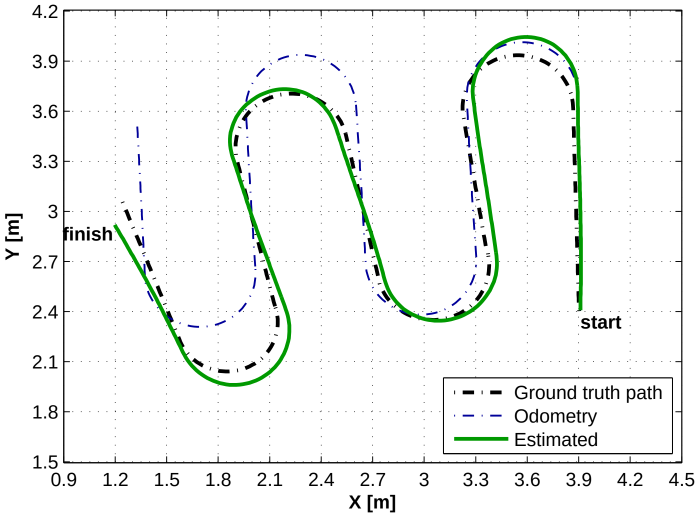

Figure 2.31: Dead-reckoning estimated with DKF^3 - experimental results.

2.7 Conclusions

This chapter presented the design of an optimal attitude and angular slippage estimator for mobile robots. This estimator aims the development of a self-localization system that will be detailed in chapter 3. As will be shown with experimental results, the integration of this estimator in a self-localization system will aid it in the global pose estimation, using only on-board sensors, and avoiding the use of the usual model linearization.

In addition, the design of several alternatives to attitude estimators were developed, extending the by now classic sensor fusion CF design. Usually, CF has been used in sensor fusion design to merge two signals, using a LPF and a HPF, with complementary frequency bands. In this chapter, an approach to design estimators based on complementary filtering, denoted as CF^{m}, to merge the signals provided by several sensors was presented and analysed. The CF^{m} estimator considers a set of m sensors signals merged in different, yet complementary, frequency bands. The estimator output has a unitary magnitude gain over whole frequency spectrum achieving, thus, the complementarity propriety. The proposed approach, besides expanding the inputs number of the sensor fusion, presents the possibility to combine different sensors measurements by the frequency bands. This means that the proposed approach allows to increase the possible combinations according to the number and characteristics of signals used in sensor fusion based on CF.

Aiming the design of a mobile robot attitude estimator, to merge the signals provided by three sensors, the CF^3 was presented and analysed. This estimator corresponds to a particularization of the CF^{m}, when three inputs were considered. Although the CF^3 parameters can be easy tuned according with the desired frequency bands, its estimations corresponds to a non-convex estimation problem. The most important consequence is that optimality is lost in general, i.e., tuning the gains K_1 and K_2 does not render the CF^3 an optimal estimator.

Then, the problem at hand was reformulated and designed as an optimal linear Kalman filtering (called as CKF^3) to perform the same sensor fusion. The optimality was achieved resorting to an optimal Kalman gain with the same dimension than the CF^3 gain matrix.

Finally, an optimal estimator to merge the signals provided by three sensors, based in Kalman filtering, denoted as KF^3, was designed to estimate the mobile robot attitude. The KF^3 is based on the sensory system global model, and is developed through the physical variables of each sensor implemented in the particular estimation process. Thus, a KF^3 to estimate the attitude, merging the signals provided by a digital compass, odometry and rate gyro, was presented and analysed. This optimal estimator will be implemented in the navigation system detailed in chapter 5.

With the purpose of implementing the optimal estimators (CKF^3 and KF^3) in a digital processor, to estimate the mobile robot attitude, the equivalent discrete-time CF estimators were designed, denoted by DCKF^3 and DKF^3, respectively. The mobile robot attitude estimator merges the signals provided by three sensors (compass, encoders and rate gyro) by a digital processor. All discrete estimators were developed considering the sensors physical measurements that will be merged.

Moreover, all designed estimators are shown to preserve the complementary property, i.e., the sum of the three transfer functions of the respective sensors add up to one. Although it is possible to design an infinitude of estimators based on the approach proposed in this chapter, only several of these new filters were presented and analysed, taking into account the estimator design for mobile robot attitude estimation.

3

PCA-based 2D localization system for mobile robots using ceiling vision

3.1 Introduction

The problem of localization without resorting to external sensors and with bounded error estimates has been a major challenge to the scientific community in the area of mobile robotics; see Thrun et al. (2005) and the references therein. The inputs to the localization system are, usually, the measurements provided by the sensor package installed on-board, like compasses, accelerometers, cameras, time of flight cameras, encoders, ..., and the robot has to autonomously use these on-board sensors to be able to look at the environment and rapidly answer the questions: where am I? What am I facing?

The use of vision systems for robot localization is very common due to the ability to obtain information about the environment (Corke, 2011). Many vision systems compute the robot pose (position and attitude) from features of the environment (Maohai et al., 2013; Scaramuzza et al., 2009) or detecting landmarks (Han et al., 2013; Huang et al., 2012; Yuan et al., 2016).

Although many robots use cameras to look around itself to get its global pose in the environment, others use a single camera looking upward (Jeong and Lee, 2005; Jo et al., 2012). The use of vision from the ceiling has the advantage that images can be considered without scaling and are less affected by occlusions due to obstacles moving around. Moreover, ceilings are usually static, increasing the mapping database reliability. However, the computational complexity of such algorithms in order to obtain features is not negligible, thus the implementation in real-time systems still demands the search for other approaches of reduced complexity.

Since such feature based techniques are computationally heavy, the use of PCA in mobile robots for self-localization has been explored (Maeda et al., 1997; Artač et al., 2002; Payá et al., 2007), but using front or omnidirectional cameras. Oliveira (2007) proposes the use of PCA as a positioning system in a terrain reference navigation of UV, using signals extracted by a sonar point downward. However, the results were only obtained in simulation.

Beyond the problems of image processing for self-localization, another challenge is to deal with the fusion of the PCA-based positioning system with the odometry data that is given by the robot kinematics. Mobile robot kinematics (e.g. differential drive) are, in general, non linear. This fact prevents the direct use of a linear KF, that provides optimal estimates with stable error dynamics. To tackle this problem, many localization systems use the EKF to estimate the robot pose merging the odometry with specific features extract from the environment structure, around the robot (Ganganath and Leung, 2012; Nguyen et al., 2016; Yuan et al., 2016), or on the ceiling (Jo et al., 2012; Hwang and Song, 2011; Rusdinar et al., 2012). However, even though it can give a reasonable performance, the EKF is an non-optimal estimator and it may diverge in consequence of wrong linearization or sensor noise (Bailey et al., 2006).

To avoid problems mentioned above, the design of position estimators for UVs, using the estimates provided from an attitude estimator was proposed and simulated by Oliveira (2007); Batista et al. (2010a).

This chapter proposes and experimentally validates an on-board self-localization system composed by a PCA-based positioning system using ceiling vision and two KF to estimate the global position and attitude of a mobile robot with differential drive kinematics. Based on Oliveira (2007), where the data of a sonar pointed downward to the ground are used to compute an UV position, through the PCA algorithm, this approach considers a mobile robot using a video camera looking upward, which ceiling data are used to obtain the robot position. Thus, the proposed self-localization system estimates the robot attitude with the optimal KF detailed in section 2.2. The choice of a LPV model allows the use of a time-varying KF to estimate the position, guaranteing stability. The self-localization system was implemented in a mobile robot, using only on-board sensors, and experimentally validated in an indoor environment (Carreira et al., 2012a,b,c).

This chapter is organized as follows:

- The PCA technique is introduced in section 3.2, leading to the nonlinear positioning system.
- A proof-of-the-concept of PCA-based localization system is presented in section 3.3, where motivation's experimental results were performed in 1D self-localization.
- The 2D linear position estimator, based on a LPV model for differential drive mobile robots, is proposed and detailed in section 3.4.
- A large set of experiments are reported in section 3.5, that experimentally validate and allow to analyse the performance of the proposed PCA-based 2D positioning system and the overall architecture able to solve the localization problem for indoor autonomous mobile robots.
- Finally, in section 3.6 some conclusions about this chapter are drawn.

3.2 PCA-based nonlinear positioning system

This section presents the fundamentals of the proposed positioning system that will integrate the self-localization system. In the proposed methodology, the mobile robot position finds its global position in the environment resorts to PCA, usually denoted as Karhunen-Loève (KL) transform.

The basic idea of this technique is based on the conversion of a set of observable signals, possibly correlated, in a new set of uncorrelated variables, for dimensionality reduction. Thus, each captured signal (images in the proposed approach) from the environment is converted into a correspondent eigenvector, that represents a good approximation to the original, but with lower dimension. This allows to achieve a small and extremely compact representation of the dataset, when compared with the initially acquired. Therefore, the comparison of new signals with the dataset is performed using the fewer data, becoming faster to compute. In addition, the PCA is computed using only linear computation, allowing to achieve an optimal approximation of the stochastic signals, minimizing the mean square error.

Based on Oliveira (2007), the proposed PCA-based positioning system provides the mobile robot position on an indoor environment from the ceiling images captured by a camera installed on board, pointed upward (Carreira et al., 2012a,b,c). Essentially, the mobile robot positioning is based on the pattern comparison problem, through the PCA, between the current image and others previously acquired on the mapping step, and stored in the database.

3.2.1 PCA-based signals compression

This section address the construction of the PCA database, that will be implemented to create the environment map. Consider a set of M stochastic signals $\mathbf{x}_i \in \mathbb{R}^N,\ i = 1,\ldots,M$, each corresponding to the stacked version of an image acquired with the camera installed on-board the mobile robot and represented as a column vector with mean:

$$\mathbf{m}_x = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i \tag{3.1}$$

The purpose of the KL transform is to find an orthogonal basis to decompose a stochastic signal $\mathbf{x}$, from the same original space, to be computed as:

$$\mathbf{x} = \mathbf{U}\mathbf{v} + \mathbf{m}_x \tag{3.2}$$

where vector $\mathbf{v} \in \mathbb{R}^N$ is the projection of $\mathbf{x}$ in the basis, i.e., around the mean value of the ensemble data, given as follows:

$$\mathbf{v} = \mathbf{U}^T(\mathbf{x} - \mathbf{m}_x) \tag{3.3}$$

Matrix $\mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_2 \ldots \mathbf{u}_N\ ]$ should be composed by the N orthogonal column vectors of the basis, verifying the eigenvalue problem:

$$\mathbf{R}_{xx}\mathbf{u}_j = \lambda_j\mathbf{u}_j,\ j = 1,\ldots,N, \tag{3.4}$$

where $\mathbf{R}_{xx}$ is the covariance matrix and λ_j is the covariance associated to the j^{th} component, computed from the set of M signals using:

$$\mathbf{R}_{xx} = \frac{1}{M-1}\sum_{i=1}^{M}(\mathbf{x}_i - \mathbf{m}_x)(\mathbf{x}_i - \mathbf{m}_x)^T. \tag{3.5}$$

Assuming that the eigenvalues are ordered, i.e. $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$, the choice of the first n principal components leads to an approximation to the stochastic signals given by the ratio of the variance associated with the first n and the total (N) components as follows:

$$r_c = \frac{\sum_{j=1}^{n}\lambda_j}{\sum_{j=1}^{N}\lambda_j} \tag{3.6}$$

This step is the key to the large dimensional reduction and, thus, a computational complexity reduction. Usually, the PCA can achieve high rations for r_c, with the selection of few components $(n \ll N)$.

Analysing the PCA algorithm it is possible to conclude that:

i) the PCA is an optimal (in terms of mean squared error) algorithm and compresses a high dimensional dataset into a lower dimensional database, through linear transformations;

ii) the model parameters can be computed directly from the data;

iii) the low dimension of the obtained model, allows the signals processing by computationally inexpensive operations, $\sim \mathcal{O}(nN)$.

These advantages suit specially the problem of mobile robot navigation, using low-cost systems installed on-board, and requiring reduced memory and computational resources.

Let's assume a scenario in the area of indoor mobile robotics (e.g. industrial or similar environments), where a navigation system has to be developed and installed on-board a mobile robots. Based on Oliveira (2007), the steps to implement a PCA-based positioning system using ceiling data will be outlined next:

1. The environment that will be mapped should be partitioned in mosaics with fixed dimensions. This allow the achievement of a PCA-based positioning system with constant resolution;

2. A set of M images with dimension N_x by N_y are captured from the ceiling. The number of signals M to be considered depends on the mission scenario dimension, on images overlapping and the positioning system resolution;

3. The two-dimensional data captured from the images is reorganized in a vector form, e.g. stacking the columns, a set of M stochastic signals $\mathbf{x}_i \in \mathbb{R}^N$, $N = N_x N_y$. If a sub-sample and/or image crop is present, N is the dimension of the extracted pixels from each image to the database. A circular crop removes non-significant parts to compare rotated images (Heras et al., 1997). In this case, when the PCA-based positioning system is running to compute the robot position in real-time, the same approach (sub-sample and/or image crop) must be implemented;

4. The KL transform is computed, using the equations (3.4) and (3.5);

5. The eigenvalues must be ordered and the number n of the principal components should be selected, according with the required level of approximation.

The following data should be recorded for later use:

- The data ensemble mean $\mathbf{m}_x$;
- The matrix transformation with n eigenvectors

$$\mathbf{U}_n = [\mathbf{u}_1 \ \dots \ \mathbf{u}_n]; \tag{3.7}$$

- The projection on the selected basis of all the mosaics, computed using

$$\mathbf{v}_i = \mathbf{U}_n^T(\mathbf{x}_i - \mathbf{m}_x),\ i = 1,\dots,M; \tag{3.8}$$

- The coordinates of the center of the mosaics

$$(x_i, y_i),\ i = 1,\dots,M. \tag{3.9}$$

3.2.2 PCA-based positioning system

Based on Oliveira (2007), during the mission, at the time instants $t_k = Lk$ (where L is a positive integer), the acquired image k will constitute the input signal $\mathbf{x}_k$ to the PCA-based positioning system. The following tasks should be performed:

1. Compute the projection of the signal $\mathbf{x}_k$ into the basis, using

$$\mathbf{v}_k = \mathbf{U}_n^T(\mathbf{x}_k - \mathbf{m}_x); \tag{3.10}$$

2. Search the mosaic that verifies

$$r_{PCA} = \min_i \ \|\mathbf{v}_k - \mathbf{v}_i\|_2; \tag{3.11}$$

3. Given the mosaic i which is closest to the present input, its center coordinates (x_i, y_i) will be selected as the $(x,y)_{PCA}(k)$ measurements.

The relation $\mathbf{f}$ between r_{PCA} allows to obtain the positioning system error covariance $\mathbf{R}$ (observation noise), that will be used in the 2D position estimator design:

$$\mathbf{R} = \mathbf{f}\, r_{\mathrm{PCA}} \tag{3.12}$$

where $\mathbf{f}$ is a parameter chosen according to the environment. It is important to remark that the robot should travel on the environment at the same distance to the ceiling (locally) as when the images were collected. Notice that the ceiling does not need to be planar or horizontal nor at the same height globally. Thus, no scaling corrections are needed. The PCA-based positioning system using ceiling vision can be straightforwardly extended to incorporate data from other sensors installed on-board mobile robots, such as information from ToF cameras or structured-light 3D scanners (e.g. Microsoft Kinect) looking upward, as it will be show in chapter 4.

3.3 Motivation: PCA-based 1D localization system for mobile robots

Aiming to achieve motivation results, the proposed PCA-based positioning system was experimentally validated in a self-localization system along a straight-line (1D localization). The experiments resorts to a mobile robotic platform with the differential drive kinematics. This platform has a laptop that controls the motors through a closed loop motor controller and has a webcam pointing upwards to the ceiling, both connected by USB. The estimated localization was obtained from a KF that merges the position obtained from the PCA-based positioning system with the odometry (Carreira et al., 2012c).

3.3.1 1D Model

The mobile robot kinematic model that describes the movement in a straight line (1D) is,

$$\dot{x} = u + b + \mu_1 \tag{3.13}$$

$$\dot{b} = 0 + \mu_2 \tag{3.14}$$

considering the following assumptions:

- the slippage velocity is constant or slowly varying (i.e. $\dot{b} = 0$);
- the noise in the actuation (motors are in closed loop) and the slippage velocity are assumed as zero-mean uncorrelated white Gaussian noise, $\mu_i \sim N(0, \sigma_i^2)$.

Expressing the model dynamics in a state-space representation with $\mathbf{x} = \begin{bmatrix} x & b \end{bmatrix}^T$,

$$\dot{\mathbf{x}} = \overbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}^{\mathbf{F}} \mathbf{x} + \overbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}^{\mathbf{B}} u + \overbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}^{\mathbf{G}} \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \tag{3.15}$$

$$z = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{H}} \mathbf{x} + \gamma \tag{3.16}$$

The output of this system z is the measurement given by the PCA-based positioning system described in the previous section. Since the position estimator is processed in a digital processor, the discrete model is obtained assuming that the vehicle velocity u is constant (ZOH assumption) between two consecutive processing times, resulting:

$$\mathbf{x}(k+1) = \overbrace{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}^{\mathbf{F}} \mathbf{x}(k) + \overbrace{\begin{bmatrix} T \\ 0 \end{bmatrix}}^{\mathbf{B}} u(k) + \overbrace{\begin{bmatrix} T & T^2/2 \\ 0 & T \end{bmatrix}}^{\mathbf{G}} \mu(k) \tag{3.17}$$

$$z(k) = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{H}} \mathbf{x}(k) + \gamma(k) \tag{3.18}$$

The design of a linear time-invariant Kalman filter for the model described above follows the classical approach referred in the previous chapter (see equations (2.11)-(2.12)) and detailed by Brown and Hwang (1997).

3.3.2 Experimental results

The mobile robot self-localization methodology proposed in this work is tested for the aforementioned mobile robot travelling along a 3 m length straight line. Ceiling images are captured with a constant distance and referenced, allowing for the creation of the PCA eigenspace (the image database referred in section 3.2.2) to capture the principal components of the environment. To create the eigenspace, gray scale images with 320 by 240 pixels are sub-sampled (25 : 1) and transformed into vectors, $\mathbf{x}_i \in \mathbb{R}^N$, $i = 1,\ldots,M$, where M stands for the number of images and N stands for the number of pixels of each image. Notice that since this is a 1D experiment only one coordinate is necessary, along the direction of movement.

The covariances to be used in the KF design were considered as constant along the experiments, being $\mathbf{Q}$ and $\mathbf{R}$ the covariance error in the actuation and the pose estimator, respectively. The value of $\mathbf{Q} = 4.1 \times 10^{-6}$ m^2 was obtained measuring the covariance error of the robot motion along one predefined path. The value of $\mathbf{R} = 6.8 \times 10^{-3}$ m^2 was obtained measuring the covariance error of the pose estimator (position given by the PCA-based positioning system) when the robot moves along the path with images in the eigenspace.

To study the PCA-based positioning system performance, 31 ceiling images (with a distance of 0.1 m), were captured with the mobile robot travelling with a constant velocity of 0.125 m/s along the straight line, as mentioned above. The images have been subsampled with a step of 5 pixels in width and height to reduce the amount of processing data (25 : 1). Analysing the eigenvalues and selecting components that explain the variability of the images in an excess of 80%,m results on an eigenspace (image database) of 4 eigenvectors.

3.3.2.1 Monte Carlo performance tests

To analyse the mobile robot self-localization methodology proposed, a Monte Carlo test composed of 10 experiments as described above has been repeated. Images were captured at 20 Hz and the PCA-based positioning system was acquired. Figure 3.1 gives the localization results obtained in one of those experiments. The results show that the PCA algorithm provides a good approximation to the real robot localization. However, some position errors are observed in instants 6 s, 13 s, 18 s and 22 s (approximately). This happen because the PCA algorithm sometimes finds most similar images in the eigenspace in distant locations. It is important to remark that the results from the KF smooth out the position errors present in the PCA-based positioning system. Figure 3.2 shows the estimated errors for 5 experiments.

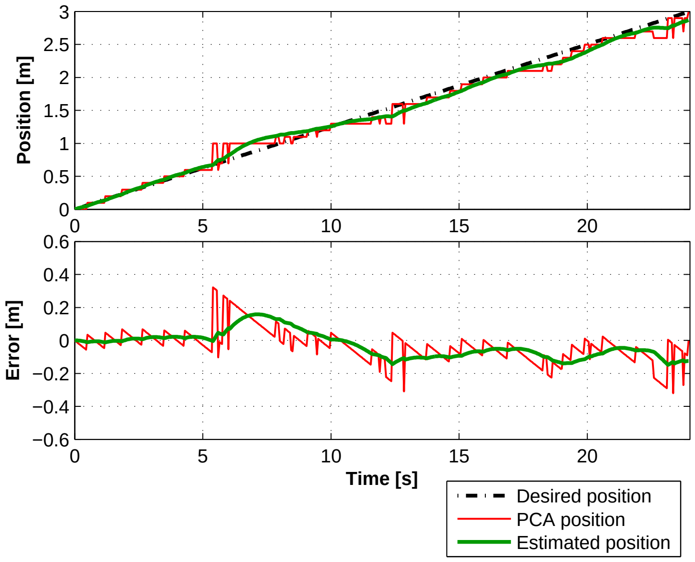

Figure 3.1: Results of PCA-based positioning system and localization estimates from Kalman filter.

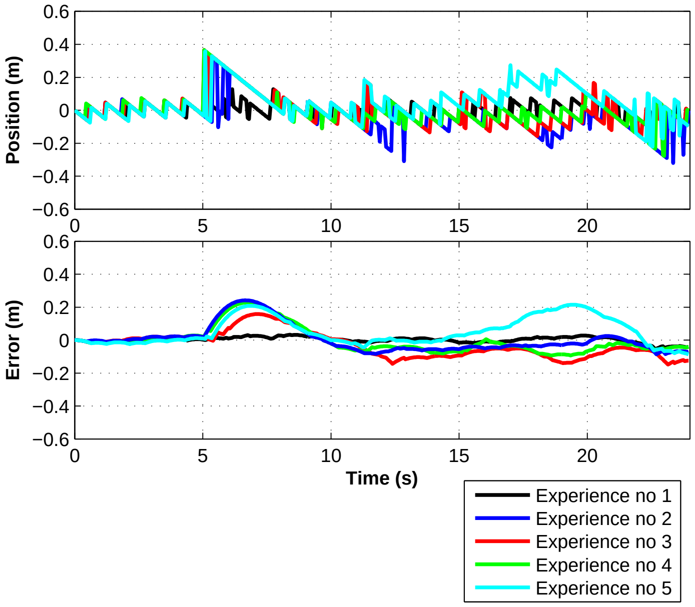

Figure 3.2: Localization errors of tests along a straight line.

3.3.2.2 Stability validation

A second test is performed to analyse the positioning system global stability when the initial position coordinates do not match the robot real initial position. Thus, it is possible to check that the estimator is able to correct the initial position error, as predicted by the stability properties of the KF. In this case, the robot was placed 1 m ahead of the usual initial position. The eigenspace was again created with a distance between acquire images of 0.1 m (same 31 images as in the previous set of tests) and the results presented in Fig. 3.3 show that the positioning system needs less than 1.5 s to provide an accurate estimate of the mobile robot localization.

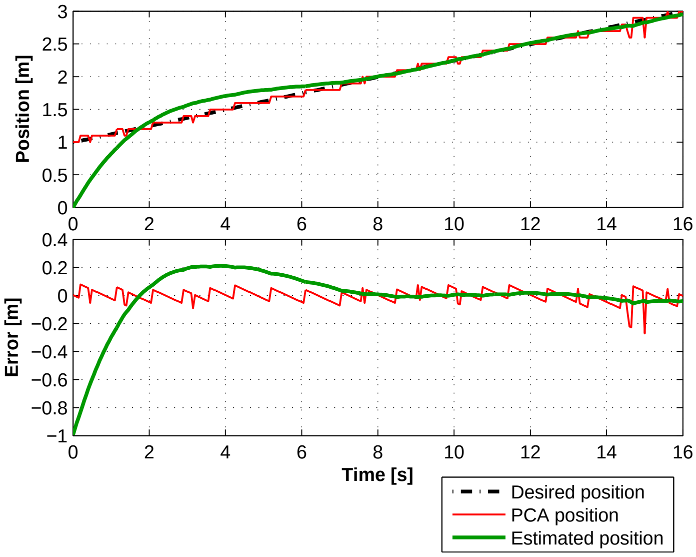

Figure 3.3: Results of positioning system when the robot starts 1 m ahead of the KF initial position.

3.3.2.3 Real-time slippage estimation

Another experiment is realized to analyse the localization system performance. Thus, a set of tests have been performed considering that the mobile robot moves with a constant, artificially imposed, wheel slippage. Two tests are reported considering that the mobile robot travels with a slippage in the wheels, that leads to a constant velocity below 0.1 m/s and 0.2 m/s, respectively in Fig. 3.4 and 3.6, relative to the commanded velocity.

The estimation errors are depicted respectively in Fig. 3.5 and 3.7. Results show that

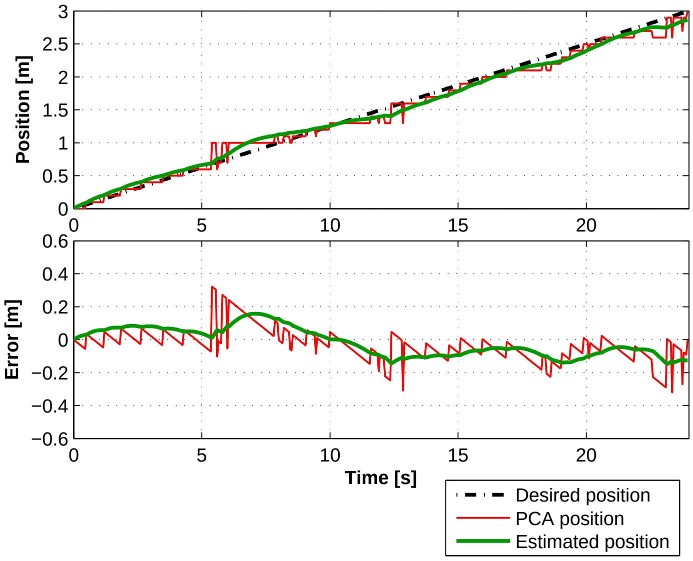

Figure 3.4: Results of the positioning system when the robot moves with a slip velocity of 0.1 m/s.

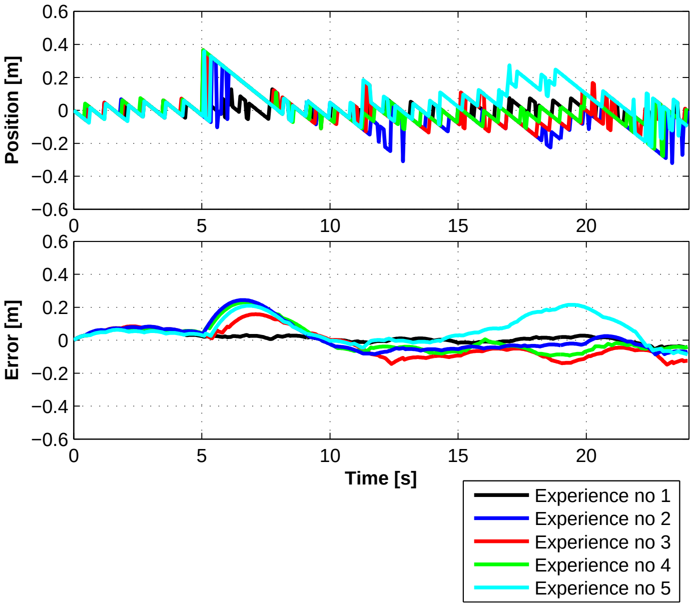

Figure 3.5: Error of positioning system when the robot moves with a slip velocity of 0.1 m/s.

the localization system is able to accurately estimate the mobile robot real position in all situations. The KF estimates present initial higher errors for higher values of slippage (above 0.2 m/s). After a transient of about 5 s (see figure 3.8), the localization system is able to estimate and correct the wheels slippage in real-time and the results obtained in the remaining of the experiments have similar performance as the ones obtained in the experiments without slippage.

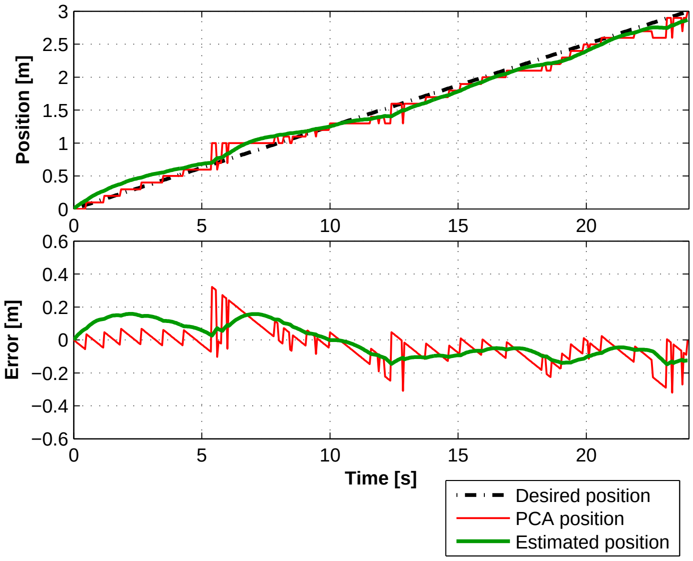

Figure 3.6: Results of the positioning system when the robot moves with a slip velocity of 0.2 m/s.

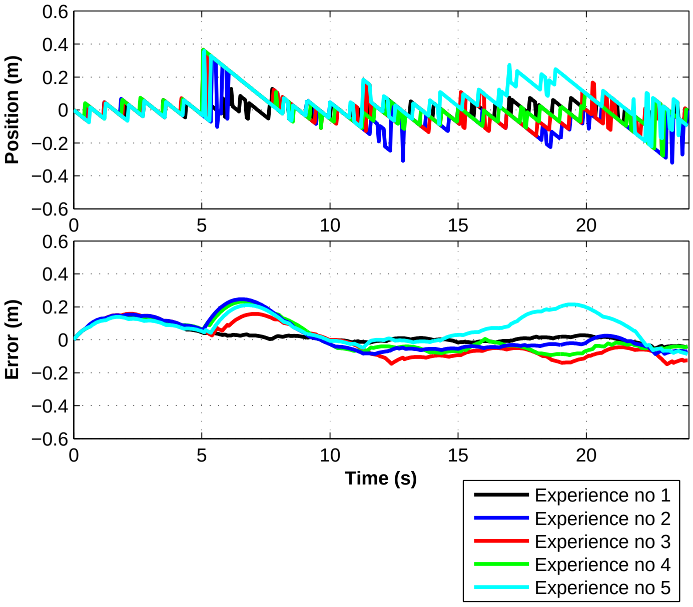

Figure 3.7: Error of positioning system when the robot moves with a slip velocity of 0.2 m/s.

3.3.2.4 Preliminary PCA performance analysis

The PCA-based positioning system design is based on the database construction (eigenspace), from a set of images captured along the path. However, a trade-off between

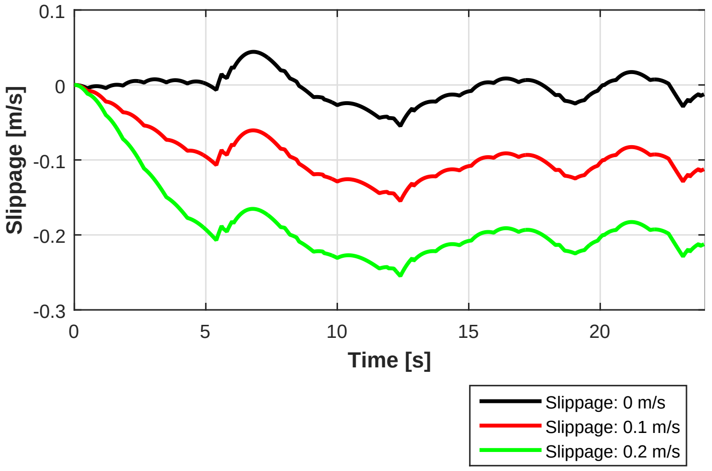

Figure 3.8: Results of bias in KF for different wheels slippery velocity.

the number of images in the database and the accuracy of the positioning system must be found. A trade-off will always be found relating the number of images in the database (eigenspace size) and the accuracy of the proposed positioning system. A preliminary study on the impact of changing these parameters will be reported in this section. The results from a set of tests where the image acquisition step varies in the interval [0.05 ; 0.4] m, i.e., using between 61 and 8 images, respectively, were performed creating different eigenspaces. Hence, the mobile robot positioning system performance has been tested considering an increase between the eigenspace points used (Table 3.1).

Table 3.1: PCA-based positioning system and localization system with different image acquisition steps.

PCA grid [m]	**Sample time [s]**	**No. images stored in PCA**	**σ^2 [m^2]**	
			PCA	**PCA with KF**
0.05	0.4	61	5.45×10^{-3}	3.80×10^{-3}
0.1	0.8	31	6.63×10^{-3}	4.36×10^{-3}
0.2	1.6	16	1.06×10^{-2}	5.25×10^{-3}
0.3	2.4	11	1.36×10^{-2}	3.41×10^{-3}
0.4	3.2	8	6.43×10^{-2}	3.84×10^{-2}

Results show that the PCA-based positioning system with KF was able to identify the correct mobile robot position based on ceiling captured images, even when the distance between knowledge points is increased, reducing the number of images in the eigenspace (Fig. 3.9). For a distance between frames up to 0.3 m, results show that the position error is small, not exceeding 0.15 m. For longer distances between frames, e.g. 0.4 m, the position estimate accuracy presents more degradation. However, even in this case, the error is below 0.4 m, which allows to conclude that the error is less than the distance associated with the image acquisition intervals.

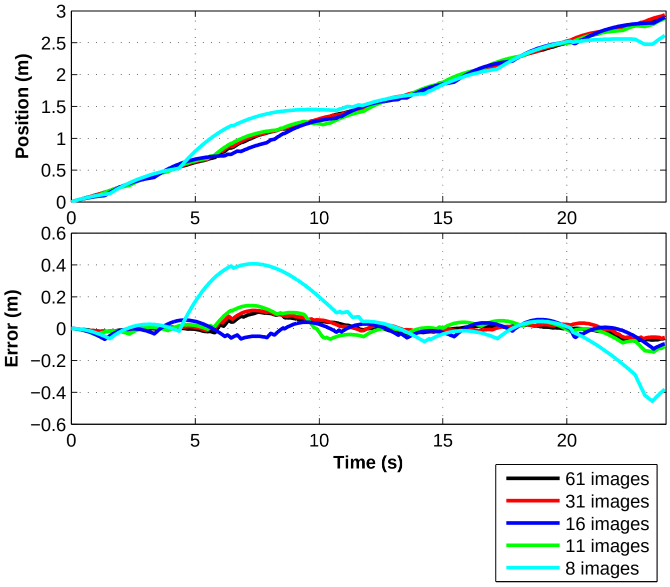

Figure 3.9: Results of PCA together with a KF.

3.4 Sub-optimal 2D position estimator

This section presents the design of a sub-optimal 2D position estimator, based on the PCA nonlinear positioning system just introduced, applied to a mobile robot with differential drive kinematics (Carreira et al., 2012a,b). To this purpose, a LPV model is derived for the differential drive robot kinematics. The LPV design aims the definition of a new model, parameterized with a different variable than the states, with linear dynamics but non-stationary (Shamma, 2012). This approach aims to avoid the use of nonlinear models that lead to the design of EKF with well known stability drawbacks.

3.4.1 LPV model for differential drive robots

The classical differential drive robot kinematics model is given by,

$$\dot{x} = u\cos\psi \tag{3.19}$$

$$\dot{y} = u\sin\psi \tag{3.20}$$

$$\dot{\psi} = \omega \tag{3.21}$$

where u is the common mode input velocity. The deterministic continuous-time model defined by equations (3.19)–(3.21) is nonlinear, which is a strong limitation for the design of a linear estimator, as a KF. However, the model can be rewritten, increasing the state space and choosing carefully new state variables, so that the non-linear system becomes a LPV model. To achieve that structure, differentiating the equations (3.19)–(3.20), using the information in equation (3.21), and considering that digital control will be used with a ZOH present ($\dot{u} = 0$), the follow equations are obtained:

$$\ddot{x} = -u\,\omega\sin\psi = -\omega\dot{y} \tag{3.22}$$

$$\ddot{y} = u\,\omega\cos\psi = \omega\dot{x} \tag{3.23}$$

Choosing as state vector $\mathbf{x} = \begin{bmatrix} x & \dot{x} & y & \dot{y} \end{bmatrix}^T$, the LPV model is obtained as follow:

$$\dot{\mathbf{x}} = \overbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\omega \\ 0 & 0 & 0 & 1 \\ 0 & \omega & 0 & 0 \end{bmatrix}}^{\mathbf{F}(\omega)} \mathbf{x} \tag{3.24}$$

$$\mathbf{y} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{H}} \mathbf{x} \tag{3.25}$$

3.4.2 Discretization

Considering that the LPV model of the differential drive robot is to be implemented in a digital processor with a zero-order hold present, and following the common assumption that ω is constant between sampling instants (ZOH assumption), a stochastic discrete-time version of equations (3.24)–(3.25), is given by:

$$\mathbf{x}(k+1) = \overbrace{\begin{bmatrix} 1 & \frac{\sin(\omega T)}{\omega} & 0 & -\frac{1-\cos(\omega T)}{\omega} \\ 0 & \cos(\omega T) & 0 & -\sin(\omega T) \\ 0 & \frac{1-\cos(\omega T)}{\omega} & 1 & \frac{\sin(\omega T)}{\omega} \\ 0 & \sin(\omega T) & 0 & \cos(\omega T) \end{bmatrix}}^{\mathbf{F}(\omega(k))} \mathbf{x}(k) + \overbrace{\begin{bmatrix} T & \frac{1-\cos(\omega T)}{\omega^2} & 0 & -\frac{\omega T-\sin(\omega T)}{\omega^2} \\ 0 & \frac{\sin(\omega T)}{\omega} & 0 & -\frac{1-\cos(\omega T)}{\omega} \\ 0 & \frac{\omega T-\sin(\omega T)}{\omega^2} & T & \frac{1-\cos(\omega T)}{\omega^2} \\ 0 & \frac{1-\cos(\omega T)}{\omega} & 0 & \frac{\sin(\omega T)}{\omega} \end{bmatrix}}^{\mathbf{G}(\omega(k))} \mu_l(k) \tag{3.26}$$

$$\mathbf{z}(k) = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{H}} \mathbf{x}(k) + \gamma_l(k) \tag{3.27}$$

where μ_l is the corresponding discrete process noise and γ_l is the noise present in the measurements given by the PCA-based positioning system, assumed approximately Gaussian. The explicit time dependence $\omega(k)$ on matrices **F** and **G** is omitted, for better legibility. In the case where $\omega(k) = 0$, i.e. along straight lines, the matrices that describe the model degenerate into:

$$\mathbf{F}(\omega(k)) = \lim_{\omega(k)\to 0} \mathbf{F}(\omega(k)) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.28}$$

$$\mathbf{G}(\omega(k)) = \lim_{\omega(k)\to 0} \mathbf{G}(\omega(k)) = \begin{bmatrix} T & \frac{T^2}{2} & 0 & 0 \\ 0 & T & 0 & 0 \\ 0 & 0 & T & \frac{T^2}{2} \\ 0 & 0 & 0 & T \end{bmatrix} \tag{3.29}$$

The kinematic model describes trajectories with constant angular rate and velocity (Gaspar et al., 2011; Li and Jilkov, 2003; Song et al., 2012). In this LPV model, the parameter variable, $\omega(k)$, is the instantaneous angular rate estimated by an optimal attitude KF from on-board sensors (equation 2.15). Thus, it describes the kinematic of trimming (equilibrium) trajectories for each instantaneous angular rate of the differential drive mobile robot and can be applied in dead-reckoning localization (section 2.6) or merged with a global position system in a time-varying KF, as detailed next.

3.4.3 Observability

To verify that the proposed estimator can lead to an estimator for the state at hand, an observability study is carried out. Assuming that constant angular rate, $\bar{\omega}$, is considered, i.e. trimming trajectories, the LPV model presented in equations (3.26)–(3.27) allows the use of the classical test that results on the computation of the observability matrix:

$$\mathcal{O}_p = \begin{bmatrix} \mathbf{H} \\ \mathbf{H}\mathbf{F}(\bar{\omega}) \\ \mathbf{H}\mathbf{F}(\bar{\omega})^2 \\ \mathbf{H}\mathbf{F}(\bar{\omega})^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\bar{\omega} \\ 0 & \bar{\omega} & 0 & 0 \\ 0 & -\bar{\omega}^2 & 0 & 0 \\ 0 & 0 & 0 & -\bar{\omega}^2 \end{bmatrix} \tag{3.30}$$

which verifies $\text{rank}(\mathcal{O}_p) = 4$, allows to conclude that the model is completely observable.

3.4.4 Position Sub-optimal estimation

The position of the differential drive robot can be estimated resorting to a linear time-varying KF, parameterized by the angular velocity estimate $\hat{\omega}(k)$ and using positioning measurements $(x_{\text{PCA}}(k), y_{\text{PCA}}(k))$, provided by the PCA-based nonlinear system, as depicted in the architecture diagram shown in Fig. 3.10.

Given the observability of the LPV model and assuming a zero-mean uncorrelated Gaussian noise for the errors in process noise $\mu_1 \sim N(0, \sigma_\mu^2)$ and for the PCA-based position system disturbances $\gamma_1 \sim N(0, \sigma_\gamma^2)$, the KF position estimator is implemented by the following equation:

$$\hat{\mathbf{x}}(k+1) = \mathbf{F}(\omega(k))\hat{\mathbf{x}}(k) + \mathbf{K}(k)(\begin{bmatrix} x_{\text{PCA}}(k) & y_{\text{PCA}}(k) \end{bmatrix}^T - \mathbf{H}\hat{\mathbf{x}}(k)) \tag{3.31}$$

Note that the proposed position estimator is sub-optimal due to two main reasons: i) the system dynamics are parameterized by $\hat{\omega}$, that is corrupted by sensor noise leading to the presence of multiplicative noise disturbances, and ii) the PCA-based nonlinear positioning system provides measurements that are not Gaussian, as its performance depends on the amount of discriminating information on the images and the grid size. Moreover, the Gaussian assumption is validated with experiments, as depicted in Fig. 3.15 and discussed later in the experimental section.

Applying the usual approach of KF design, for a process noise with covariance $\mathbf{Q_l}(\mu_l)$, the covariance of the predicted estimation error for the LPV model presented in equations (3.26)–(3.27) is given by:

$$\mathbf{P}(k) = \mathbf{F}(\omega(k))\mathbf{P}(k)\mathbf{F}(\omega(k))^T + \mathbf{G}(\omega(k))\mathbf{Q_l}\mathbf{G}(\omega(k))^T \tag{3.32}$$

After the measurement of a new position from the PCA-based positioning system, the error covariance is updated resorting to:

$$\mathbf{P}(k) = \mathbf{P}(k) - \mathbf{P}(k)\mathbf{H}^T(\mathbf{H}\mathbf{P}(k)\mathbf{H}^T + \mathbf{R_l}(k))^{-1}\mathbf{H}\mathbf{P}(k) \tag{3.33}$$

where $\mathbf{R_l}(k)$ is the covariance of the observed uncertainty, for time instant k, as described in equation (3.12), that now could be given as follows:

$$\mathbf{R_l}(k) = \mathbf{R_{xy}} \cdot \mathbf{f}\, r_{\mathrm{PCA}}(k) \tag{3.34}$$

Considering the uncertainty of the estimator $\mathbf{P}(k)$ and in the PCA-based positioning system $\mathbf{R_l}(k)$, the Kalman gain that is applied in the global position estimator for instant k is obtained as follows:

$$\mathbf{K}(k) = \mathbf{P}(k)\mathbf{H}^T\mathbf{R_l}(k)^{-1} \tag{3.35}$$

The Kalman gain $\mathbf{K}$ is a dynamic gain proportional to the estimation error uncertainty and inversely proportional to measurement uncertainty in the PCA-based positioning system. This 2D position estimator provides estimation with stable error dynamics, for any initial conditions in the state vector.

While the attitude estimator can estimate the angular slippage, through the state s_a (see section 2.2), if an extra state would be considered to describe the linear slippage s_l, being implemented in this LPV model, this new quantity is not observable. However, resorting to the algebraic estimate of the total common mode velocity $\hat{u}$, the linear slippage can be obtained from the follow equation:

$$\hat{s}_l = u - \hat{u} = u - \sqrt{(\hat{x})^2 + (\hat{y})^2} \tag{3.36}$$

3.5 Experimental results

The proposed on-board self-localization system architecture (Fig. 3.10) is composed by the optimal attitude estimator (section 2.2), the PCA-based positioning system (section 3.2.2), and the 2D position estimator (section 3.4.4).

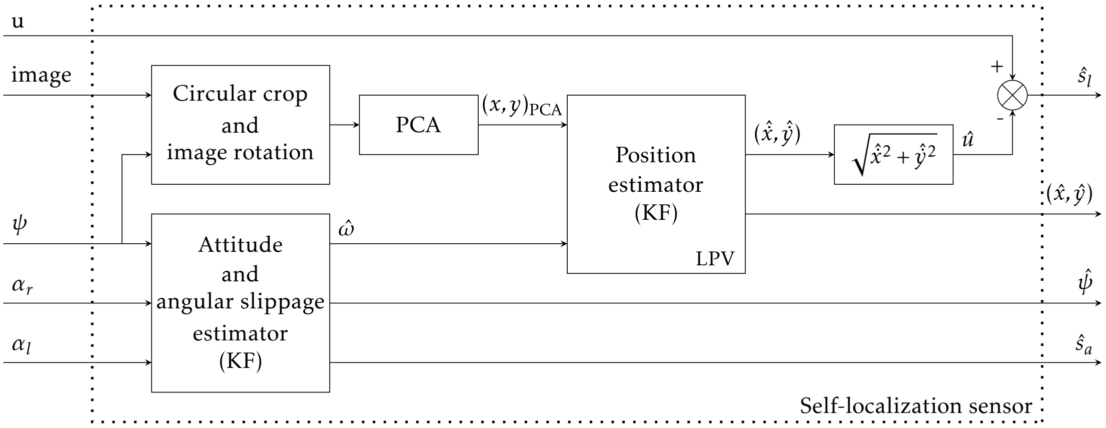

Figure 3.10: Architecture of the localization system.

The following notation is used in Fig. 3.10:

- u - commanded linear velocity;
- ψ - attitude angle given by the compass;
- α_r - angle given by the encoder of the right wheel;
- α_l - angle given by the encoder of the left wheel;
- $(x,y)_{PCA}$ - coordinates given by the PCA-based positioning system;
- $(\hat{x},\hat{y})$ - estimated robot coordinates in the world referential;
- $\hat{\omega}$ - estimated angular velocity;
- $\hat{u}$ - estimated linear velocity;
- $\hat{s}_a$ - estimated angular slippage.
- $\hat{s}_l$ - estimated linear slippage.

The experimental validation of the proposed self-localization system is performed resorting to a differential drive mobile robotic platform equipped with two encoders coupled to the motors, a webcam pointing upwards to the ceiling and a compass, located in a rear-extension arm to avoid the magnetic influence of the motors (Fig. 3.11). See the experiment in Carreira et al. (2012a) for details.

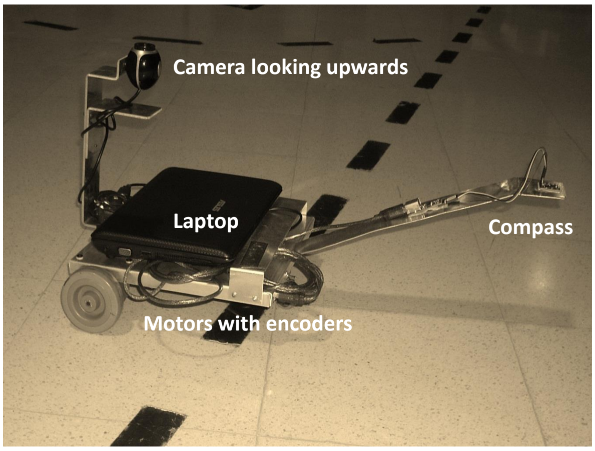

Figure 3.11: Mobile platform with webcam and compass.

3.5.1 Environment

To create the ceiling image database, snapshots were captured at pre-specified grid locations, with the robot in the same attitude, as shown in Fig. 3.12. A PCA eigenspace is computed covering a 5 m × 4.5 m area, with a 0.3 m grid (in x and y axis). To that purpose, the gray

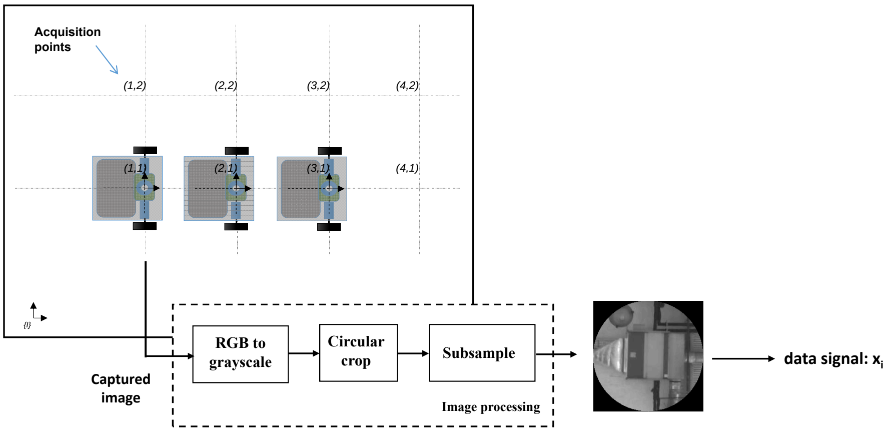

Figure 3.12: Grid map to create a PCA eigenspace.

scale images are cropped with a circular mask, as depicted in Fig. 3.12 to create a rotation invariant shape . The pixels representing the images inside the circle are then arranged into a vector. In order to reduce the amount of processing data, with negligible information loss, the images are subsampled with a 3 pixels step in width and height (data ratio reduction of 9 : 1). An image database of 13 eigenvectors is obtained selecting the components that explain the images variability in an excess of 85%.

Considering the covariances $\mathbf{Q} = \mathbf{Q}(k)$ and $\mathbf{R} = \mathbf{R}(k)$ used in the KF design, which are respectively the error covariance in the actuation system and in the sensors, being the measurement tests performed with a sampled frequency of 5 Hz. The error covariance in the differential actuation is represented by $\mathbf{Q}_\omega$, while the variable $\mathbf{Q_l}$ is the error covariance in the common-mode actuation. For the measurement uncertainty, $\mathbf{R}_\psi$ is the error covariance in the attitude sensor, while $\mathbf{R_{xy}}$ is the error covariance in the positioning system. Thus, to parameterize the attitude estimator, the value $\mathbf{Q}_\omega = 2.44 \times 10^{-5}$ (rad/s)^2 was obtained measuring the error covariance in the differential actuation along predefined circular trajectories. The digital compass error covariance, measured using circular trajectories, is $\mathbf{R}_\psi = 9.4 \times 10^{-3}$ rad^2. For the position estimator, the error covariance in the linear actuation $\mathbf{Q_l} = 1.6 \times 10^{-5}$ m^2, was obtained performing tests based on linear trajectories. The error covariance of the PCA-based positioning system has been calculated with the position error in a predefined trajectory and is given by $\mathbf{R_{xy}} = diag(\begin{bmatrix} 1.32 \times 10^{-2} & 1.32 \times 10^{-2} \end{bmatrix})$ m^2.

3.5.2 Estimation on a lawn-mower trajectory

To test the mobile robot localization system, the proposed estimators were tested in a lawn-mower trajectory. These trajectories have the advantage of combining both straight lines with semi-circumferences allowing assessment the localization system operation under different experimental conditions. Figure 3.13 depicts the results for a lawn-mower trajectory where the robot starts at $x_0 = 3.9$ m, $y_0 = 2.4$ m, $u_0 = \dot{y}_0 = 0.1$ m/s, $\dot{x}_0 = 0$ m/s and initial attitude $\psi_0 = 90$ °. In the first experiment reported, the state of the KF were initialized with the same values. To allow for a ground truth analysis, singular points of the real path were marked at the end of each type of trajectory. The ground truth path was obtained by interpolation of the marked singular points.

Analysing the results of the ground truth test (see Fig. 3.13) it is possible to conclude that the localization system provides an accurate estimate of the real trajectory. It is possible to observe that the 1-sigma uncertainty of the position state estimates (see circles over the estimated trajectory) includes the ground truth path but not the odometry of the robot, as depicted in the zoom area, validating the filter consistency. Figure 3.14 shows that

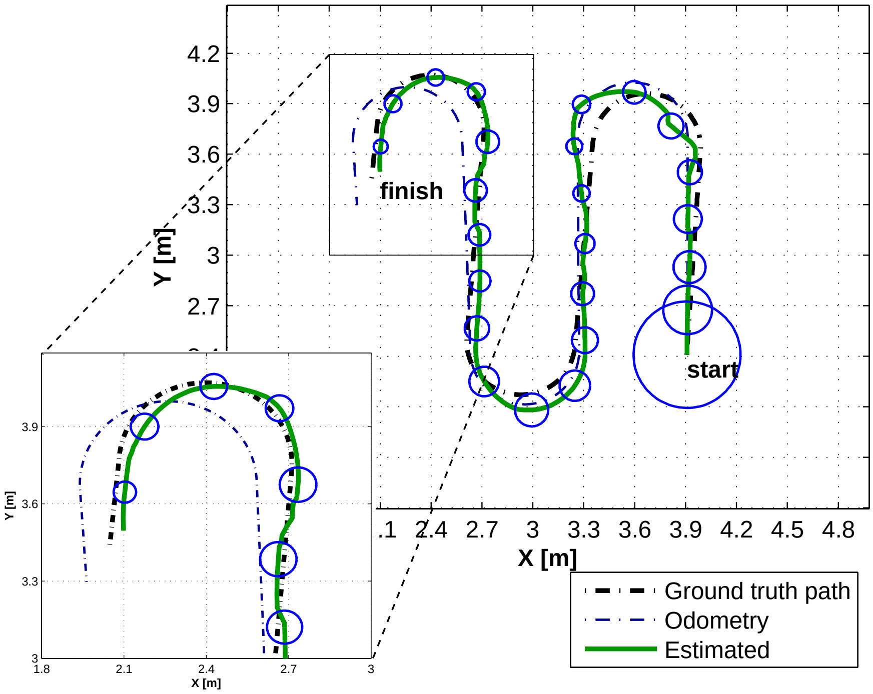

Figure 3.13: Results for a lawn-mower trajectory: 2D trajectory.

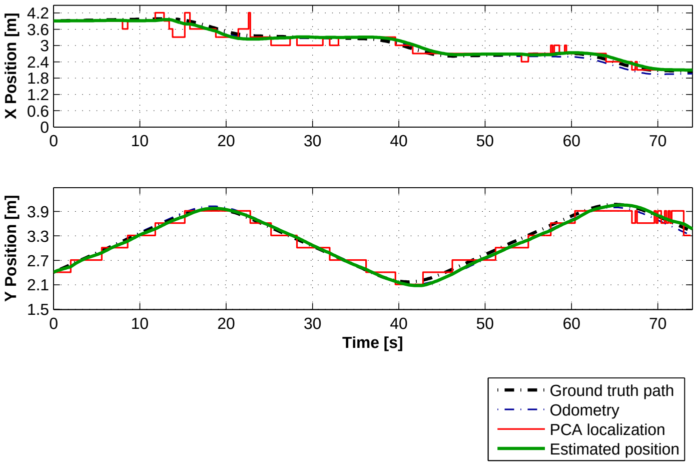

Figure 3.14: Results for a lawn-mower trajectory: X and Y position over time.

the PCA-based nonlinear positioning system provides a position measurement with some error, due to the distance associated to the selected grid positions. However, although these variations in the position provided by the PCA, the estimator is able to recover and smooth the estimates (see Fig. 3.13 and 3.14).

It is also worth measuring the statistical distribution of the position estimation errors to validate the zero mean error (estimator biasness), Gaussian distribution, and co-variance (filter consistency) assumptions. Figure 3.15 shows the histogram of the position errors for the coordinates x and y, respectively. The approximation to a zero mean Gaussian distribution is not exact; nevertheless the non-zero mean is explained by the fact that the trajectory is not random and due to the finite resolution of the PCA-based positioning system (0.3 m). The values computed are $\bar{x} = 1.07 \times 10^{-2}$ m and $\bar{y} = -2.95 \times 10^{-2}$ m, thus very close to zero.

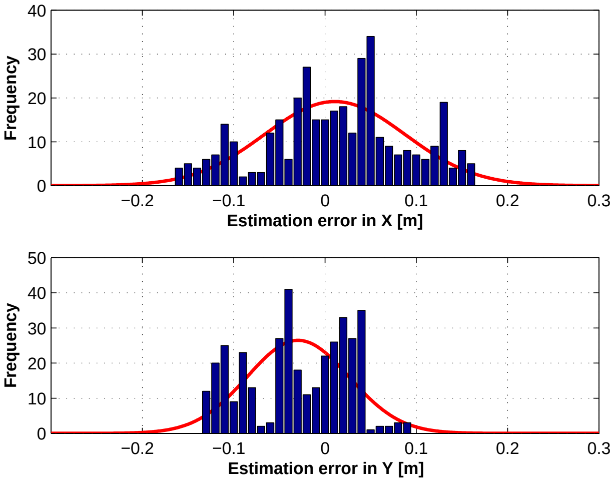

Figure 3.15: Histogram of the estimated position error.

Figures 3.16 to 3.17 depict, respectively, the stabilization of the Kalman gains and the estimators' covariance along time. It can be observed that all estimators converge, in a few sampling times.

Note that the variance obtained from the Ricatti equations for the uncertainty propagation is of similar order to the values computed from the histogram, i.e. in the order of 0.1 m, thus validating the filter performance.

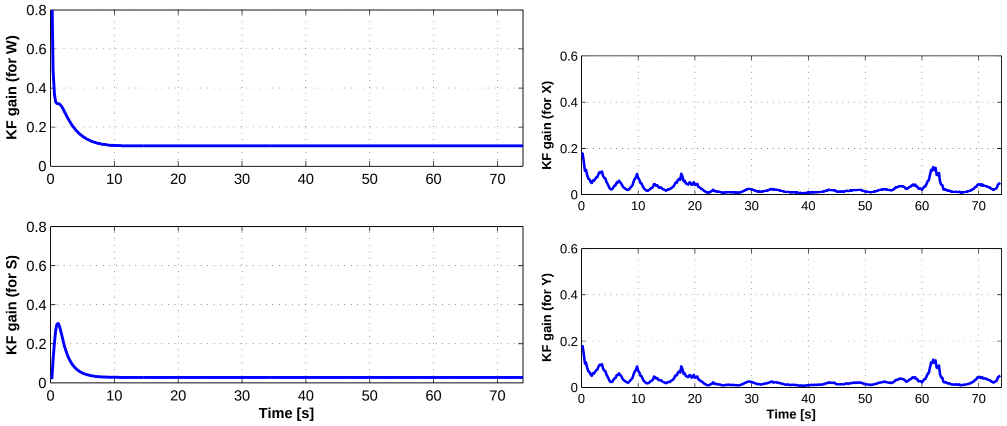

Figure 3.16: Evolution of KF gains during lawn-mower trajectory: attitude estimator (left) and position estimator (right).

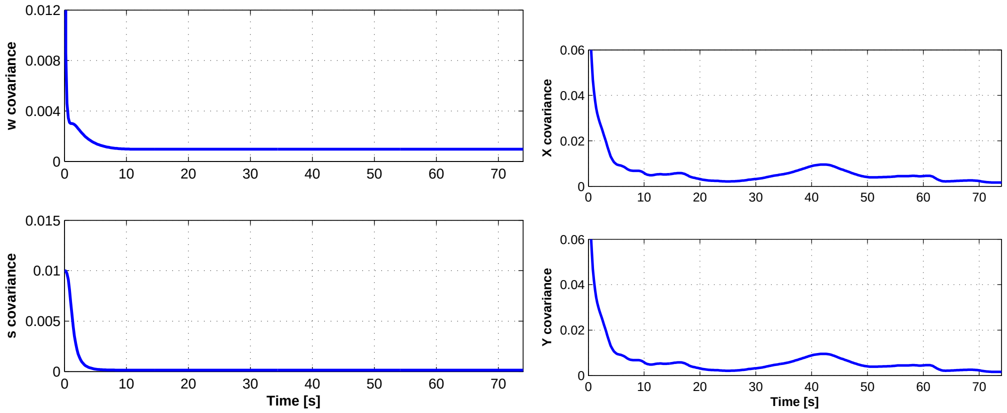

Figure 3.17: Evolution of KF error covariances during lawn-mower trajectory: attitude estimator (left) and position estimator (right).

As it can be observed in the presented results the localization system is always stable, performing a good estimation of the localization, in spite of the low precision system used to measure the position looking up to the ceiling. An odometry-based localization system would necessarily diverge over time. The sensor fusion in the KF allows achieving an accurate sensor with a performance much higher than the performance of each individual sensor. In the previous tests it can be observed that during the curves the system degrades, which can be explained by the inevitable errors presented in the information provided by the compass and the consequent disturbance in the image rotation that will be compared with the images in the PCA eigenspace.

3.5.3 Global stability of position estimator

The global stability of the position estimator, when incorrectly initialized, is studied in this section. Thus, some re-localizations were performed considering that each initial estimated position starts from a different initial conditions, as is presented in Table 3.2.

Table 3.2: Initial conditions of position stability validation in a lawn-mower trajectory.

	x position [m]	y position [m]
Robot position	3.9	2.4
Re-localization 1	4.4	2.4
Re-localization 2	4.4	1.9
Re-localization 3	2.9	1.6

The results are depicted in Fig. 3.18 and Fig. 3.19, allowing to conclude that, in spite of the large initial estimated position error, the system converges to the correct localization. Analysing Fig. 3.19, it is possible to conclude that the position estimator stabilizes in approximately 4 seconds.

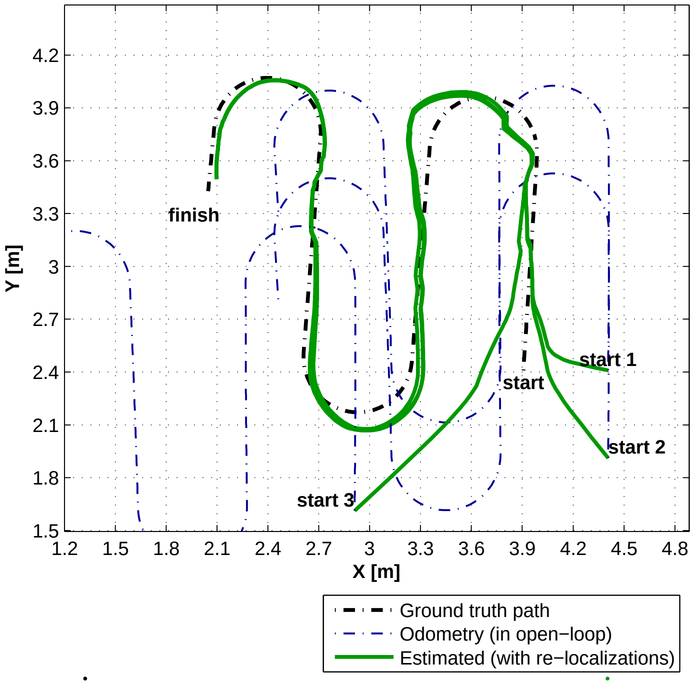

Figure 3.18: Results of position stability tests for a lawn-mower trajectory: 2D trajectories.

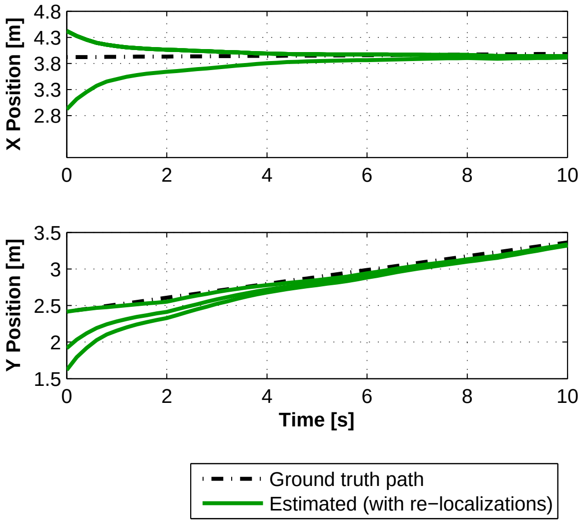

Figure 3.19: Results of position stability tests for a lawn-mower trajectory: First 10 s of trajectory over time.

3.5.4 Global stability of attitude estimator

Another important stability challenge is to assure that the estimator is able to correct the trajectory when the mobile robot starts with an incorrect attitude estimate. This challenge is more severe than the initial incorrect position estimate, because the robot odometry guides it to a much different trajectory.

Considering the initial conditions summarized in Table 3.3, the results presented in Fig. 3.20 show that the attitude KF converges in less than 0.5 s. As the proposed model for the differential drive mobile robot is a LPV model, function of the estimated angular velocity, this small stabilization time leads also to a small position estimate stabilization time, in the order of 3 s. Moreover, Fig. 3.21 shows that, as seems obvious, if only odometry were used, the mobile robot would be completely lost after starting in a wrong initial direction. A zoom of the initial path shows that the attitude estimator converges successfully even considering

Table 3.3: Initial conditions of attitude stability validation in a lawn-mower trajectory.

	attitude [°]
Robot position	90
Re-localization 1	10
Re-localization 2	180
Re-localization 3	270

the worst scenario, i.e. the opposite direction.

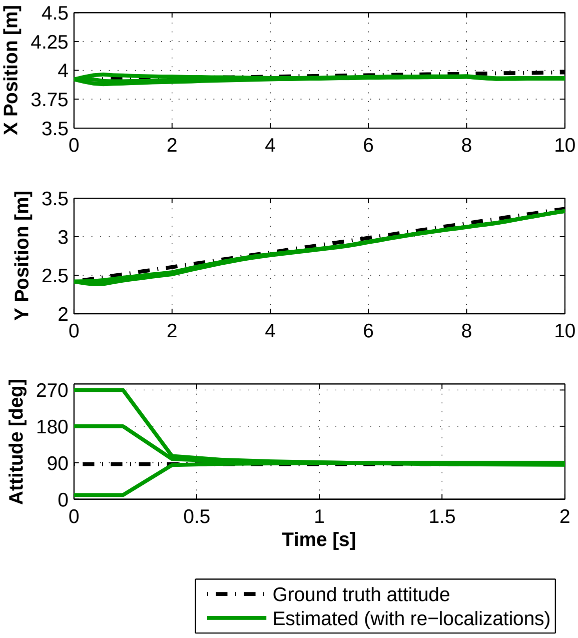

Figure 3.20: Results of attitude stability tests for a lawn-mower trajectory: Estimates convergence.

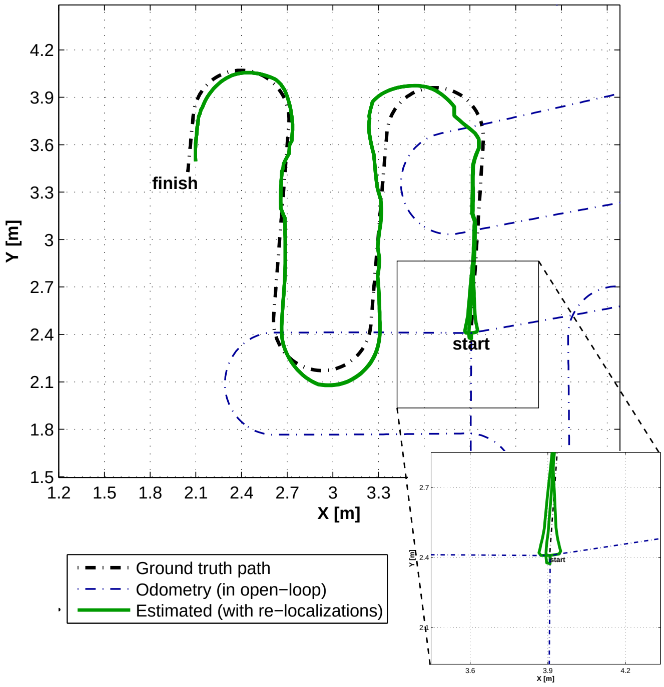

Figure 3.21: Results of attitude stability tests for a lawn-mower trajectory: 2D trajectories.

3.5.5 Global stability with wrong initial position and attitude

A more robustness validation experiment was performed, combining the two tests described previously, i.e, the simultaneous attitude and position stability. For this purpose, three incorrect initial conditions for both position and attitude have been considered (see Table 3.4 for details).

Table 3.4: Initial conditions of position and attitude stability validation in a lawn-mower trajectory.

	x position [m]	y position [m]	attitude [°]
Robot position	3.9	2.4	90
Re-localization 1	4.4	2.4	0
Re-localization 2	4.4	1.9	315
Re-localization 3	2.9	1.6	225

The results, depicted in Fig. 3.22, show that considering the incorrect initial conditions, the robot would follow, in open-loop, the same lawn-mower trajectory, but in a different direction. Considering these conditions, the position estimator could diverge from the robot real position. Instead, the results are always stable and exhibit fast convergence.

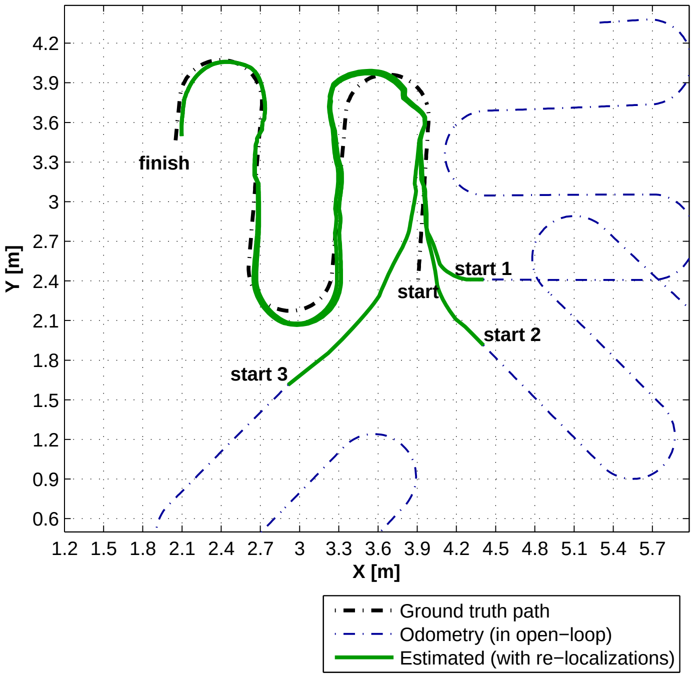

Figure 3.22: Results of stability tests considering a wrong initial position and attitude (map 2D).

3.5.6 Common and differential modes slippage estimation

Finally, another important issue to be considered is the localization in the presence of slippage, both on attitude and on linear position. Slippage is caused by systematic errors which greatly degrade the performance of odometry self-localization, due to the uncertainty in the dimensions of the robot, incorrect calibration, usage degradation, etc. The explicit angular slippage estimation (differential mode) is possible due the characteristics of the designed model (see section 2.2). The linear slippage estimation (common mode) is computed algebraically, as previously detailed in section 3.4.4.

To analyse the angular slippage estimation, new experimental tests were performed, considering an imposed angular slippage of $1°/s$. The results are presented in Fig. 3.23, illustrating that self-localization method proposed is able to cope with this real life phenomena. Figure 3.24 shows the estimated angular slippage $\hat{s}_a$ and its cumulative average $\overline{\hat{s}_a}$. The signal $\hat{s}_a$ exhibits some large variations around the correct angular slippage, probably due to the system noise. Nevertheless, the plot of the signal accumulated average $\overline{\hat{s}_a}$ shows that the estimated angular slippage mean value is close to the imposed angular slippage of $1°/s$, thus validating the estimator $\hat{s}_a$.

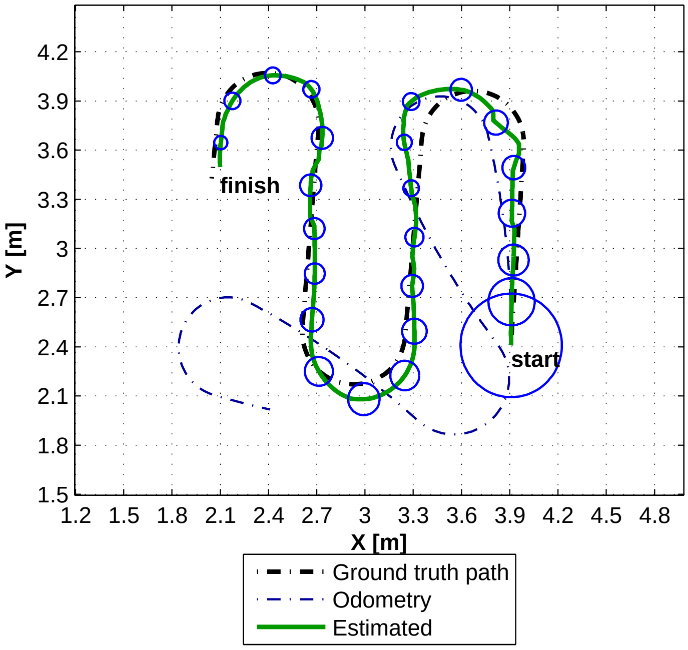

Figure 3.23: Results of angular slippage estimation: 2D trajectories.

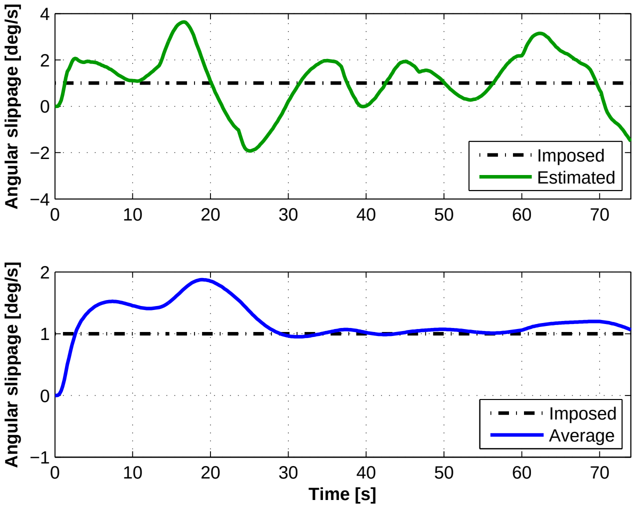

Figure 3.24: Results of angular slippage estimation: differential mode slippage convergence.

To analyse the common mode algebraic slippage computation an experiment was carried out. Thus, an initial velocity $\hat{\dot{y}}(0) = 0.2$ m/s was defined, whereas, in the real robot, a command of $u = 0.1\ m/s$ is set. The linear slippage was computed from the estimated velocities $\hat{\dot{x}}$ and $\hat{\dot{y}}$. The initial evolution of X and Y coordinates can be seen in Fig. 3.25. The incorrect value of $\hat{\dot{y}}(0)$ initially affected the Y position, but it converged to the ground truth path. The computed estimation of the linear velocity $\hat{u}$ and linear slippage $\hat{s}_l$ are depicted in Fig. 3.26. Both $\hat{u}$ and $\hat{s}_l$ converged to the correct value of 0.1 m/s.

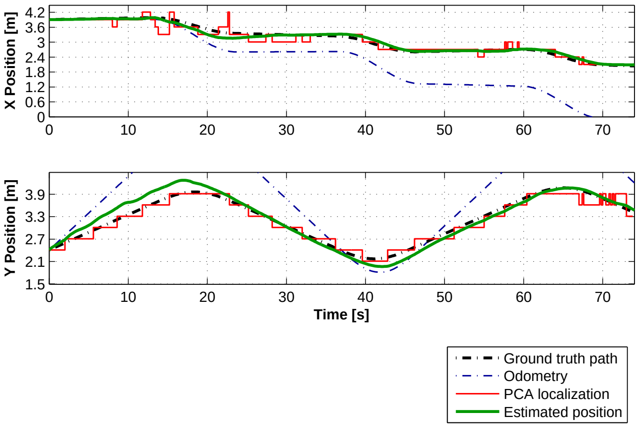

Figure 3.25: Results of common mode slippage position estimation: XY over time.

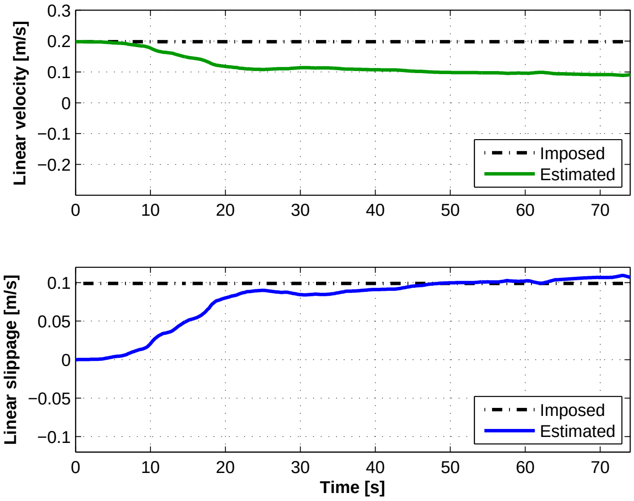

Figure 3.26: Results of common mode slippage position estimation: common mode slippage convergence.

3.6 Conclusions

In this chapter, a new localization system for mobile robots, operating in unstructured environments, was presented and analysed. With the developed approach, the robot is able to self-locate using only on-board sensors, without the need of any external landmarks. The system resorts to PCA of images acquired by a video camera installed on-board, looking upwards to the ceiling. This solution has the advantage of avoiding the need of extracting features from the images.

While many existing localization systems use EKF to merge the environment perception with the mobile robot model, in this system the mobile robot pose is estimated using only linear KFs, avoiding the need of model linearizations. As motivation, the proposed localization system for mobile robots to operate in unstructured environments was experimentally validated along a straight line (1D localization). Several tests were performed namely: i) Monte Carlo performance study; ii) global stability validation; iii) real-time slippage estimation; iv) PCA performance analysis. All tests were successful and allow to conclude that the robot motion fusion with the global position provided by a PCA-based positioning system with ceiling vision, achieves accurate and stable estimates. This motivate the proposed approach to be implemented in the design of a 2D self-localization system for mobile robots with differential drive kinematics.

For the 2D localization, this chapter proposes a self-localization system with an architecture composed by three modules: a PCA-based positioning system; a linear KF to estimate the attitude, the angular velocity and the angular slippage; and a linear time-varying KF to estimate the position. The use of a LPV model for mobile robots with differential drive kinematic allows to estimate the 2D position through a linear KF, guaranteeing a global stability.

With the purpose of testing the performance of the developed self-localization system, several experimental tests were carried out using lawn-mower trajectories: i) self-localization tests with ground truth validation and Monte Carlo performance study; ii) global stability validation for position and attitude estimation; iii) angular velocity estimation and linear slippage algebraic computation. Results show that the self-localization system has fast convergence and guarantees global stability, even when the estimator starts from incorrect positions and attitudes.

4

Extended PCA-based localization using corrupted depth images

4.1 Introduction

The chapter 3 presented a self-localization system for a mobile robot working in indoors unstructured environments. The implemented system resorts to a vision system with one camera installed on-board the robot, capturing images from the ceiling. The proposed self-localization system was experimentally validated, presenting accurate results and with globally stable estimates.

The PCA shows to be an efficient algorithm, converting the images database with an high compression ratio, when compared with the amount of captured data. Moreover, in opposite to many localization systems that needs to extract specific features or landmarks to structure the environment, the use of PCA allowed the creation of a self-localization system for mobile robots, without the need of specific feature extracting.

In spite of vision being one of the most popular sensors in mobile robotics, witch data can be used to solve the localization problem, this system still has a general limitation related to different environmental lighting conditions that decreases the localization systems robustness. This limitation was also felt during the course of the presented experiments, due to the high sensitivity of the PCA methodology to lighting conditions.

To avoid the problem mentioned above, some localization systems are based on ToF sensors. The use of ToF sensors allows the acquisition of depth information about the environment and presents a more robust system, able to cope with different light conditions. Moreover, the ToF cameras allow the capture of depth images, where the sensor is able to receive a grid with depth information from all field of view. However, it is expensive to implement this type of cameras in many mobile robotic platforms.

In the last decade, the companies PrimeSense and Microsoft developed a device primarily for video games, denominated as Kinect, that combines a RGB and a depth camera. Since then, the Kinect device has been used in several robotic applications (Huang et al., 2011; Theodoridis et al., 2013; Barbosa et al., 2015; Yuan et al., 2016; Nguyen et al., 2016).

A very common problem in depth sensors, including the Kinect depth sensor, is the existence of missing data in signals, caused by IR beams that are not well reflected, not returning to the depth sensor receiver. When depth images are used to extract features, only the available information is tacked into account. However, when all data is consider to map the environment or to localize the mobile robot, the existence of missing data may disturb the results. Oliveira and Gomes (2010) address the problem of the signals reconstruction using PCA and suggest that the mean substitution is a simple technique to recover missing data.

This chapter extends the PCA-based positioning system proposed in chapter 3 for use of depth images corrupted with missing data. This PCA extension is integrated in the self-localization system with the attitude estimator detailed in section 2.2 and the 2D position estimator presented in section 3.4. The achieved algorithm improves the self-localization system robustness, even in the case of varying illumination condition, as happen with the robot operation normal along time (Carreira et al., 2013, 2015a).

Therefore, this chapter is organized as follows:

- Section 4.2 presents the mobile robot platform and the motivation for the use of Kinect in the proposed localization system.
- In section 4.3 the PCA for signals with missing data is detailed.
- For performance analysis purposes, section 4.4 presents experimental depth image

reconstruction results, using the PCA algorithm for signals corrupted with different ratios of missing data.

- Motivation experiments considering the mobile robot localization along a straight line are presented in section 4.5.
- In section 4.6, an architecture for mobile robot self-localization, composed by the extended PCA-based positioning system and two KF is introduced, and the experimental results for localization with missing data, estimation stability, and localization in repeatability scenarios are presented.
- Finally, section 4.7 presents some conclusions.

4.2 The mobile robot

The experimental validation of the positioning system proposed in this chapter is performed resorting to a mobile robotic platform with differential drive kinematics. A Microsoft Kinect is installed on the platform, pointing upwards to the ceiling, together with a digital compass located on the extension arm (robot rear part), with a similar approach to the presented in section 3.5, as depicted in Fig. 4.1.

The Kinect includes a RGB camera with a Video Graphics Array (VGA) resolution (640 × 480 pixels) and a 2D depth sensor (640 × 480 pixels). The use of this sensor for mobile robots localization could combine the capture of a RGB image and a depth map about the environment, obtaining RGB-D images, as shown in Fig. 4.2. This image depicts the ceiling

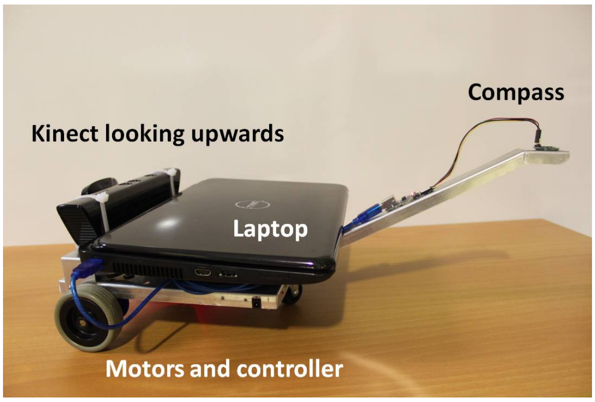

Figure 4.1: Mobile platform equipped with kinect sensor and compass.

view captured by the Kinect installed on-board the mobile robot. Note that it is possible to observe both the 3D shape of the existing technical installations in the ceiling and its color.

The robot moves indoors, in buildings whose ceiling has some information (e.g. building-related systems such as HVAC, electrical and security systems, etc.). It is possible to use the signals captured by a Kinect looking upward (RGB image, depth map or both) by an algorithm that can provide mobile robot global position in the environment.

Due to limitations found in image-based mobile robot localization approaches, regarding lighting changes, and aiming the development of an efficient self-localization solution that can work in places with uncontrolled lighting changes, only the Kinect depth signal is used, resorting to an adaptation to the method proposed in chapter 3 to the problem at hand.

However, as it is possible to observe in Fig. 4.2, due to the geometry and properties of some objects, several IR beams are not well reflected and, thus, can not be detected by the depth sensor receiver. In the case of Kinect, such a problem results in the existence of points with null distance (0 mm) inside the data array with the depth values (distances to various points in the plane), that may lead to erroneous results in the localization system. In this chapter an extension of a PCA-based positioning system will be presented, aiming to cope with lighting changes common to usual vision systems, and experimentally validated.

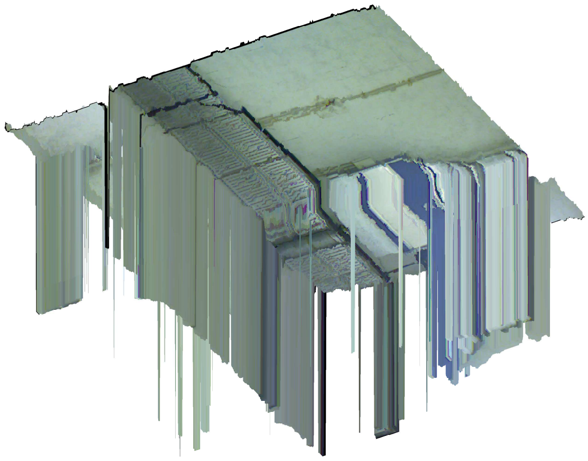

Figure 4.2: RGB-D image of the ceiling view obtained by the kinect installed on-board the mobile robot.

4.3 PCA for signals with missing data

In common PCA-based approaches, the eigenspace of acquired data is characterized by the corresponding mean, equation (3.1), and covariance, equation (3.5). However, the existence of missing data in signal $\mathbf{x}_i$ corrupts the PCA mean value computation, creating an orthogonal space with erroneous data.

Following the approach proposed by Oliveira and Gomes (2010), in the case where missing data occurs, the covariance matrix is computed using only the non-corrupted data. Thus, a vector $\mathbf{l}$ with length N consisting of boolean values is used to mark the real and missed data of a signal $\mathbf{x}_i$. Then, considering the j^{th} component of an acquired signal $\mathbf{x}_i$, the index $\mathbf{l}_i(j)$ is set to 1 if the signal $\mathbf{x}_i(j)$ is available and it is set to 0 if there is a missing data.

Hence, to avoid the negative impact of the sensor signals missing data in the PCA-based positioning system performance, an extension to this methodology is proposed in this chapter. Thus, instead of considering all values of the M stochastic signals to compute the mean ensemble, $\mathbf{m}_x$, and the covariance, $\mathbf{R}_{xx}$, only the correct data is used to compute the orthogonal space, and the values corresponding to missing data are neglected. Thus, based on the auxiliary vector, $\mathbf{l}$, the follow auxiliary counters are defined:

$$\mathbf{c} = \sum_{i=1}^{M} \mathbf{l}_i \tag{4.1}$$

$$\mathbf{C} = \sum_{i=1}^{M} \mathbf{l}_i \mathbf{l}_i^{\,T} \tag{4.2}$$

Considering the set with M signals, the mean ensemble for the j^{th} component is given by:

$$\mathbf{m}_x(j) = \frac{1}{c(j)} \sum_{i=1}^{M} \mathbf{l}_i(j)\mathbf{x}_i(j), \; j = 1, \ldots, N \tag{4.3}$$

and the covariance element $\mathbf{R}_{xx}(j,k)$, $\{j,k\} = 1, \ldots, N$ is computed as follows:

$$\mathbf{R}_{xx}(j,k) = \frac{1}{C(j,k)-1} \sum_{i=1}^{M} \mathbf{l}_i(j)\mathbf{l}_i(k)\mathbf{y}_i(j)\mathbf{y}_i(k) \tag{4.4}$$

where $\mathbf{y}_i = \mathbf{x}_i - \mathbf{m}_x$.

Considering the new mean ensemble and covariance of the PCA database computed without corrupted data, equation (4.3) and equation (4.4), respectively, the decomposition into the orthogonal space follows the PCA algorithm classical approach (see section 3.2.1 for more details), i.e. $\mathbf{v} = \mathbf{U}^T(\mathbf{x} - \mathbf{m}_x)$. The matrix $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \ldots \ \mathbf{u}_N \ ]$ should be composed by the N orthogonal column vectors of the basis, verifying the eigenvalue problem:

$$\mathbf{R}_{xx}\mathbf{u}_j = \lambda_j \mathbf{u}_j, \ j = 1, ..., N, \tag{4.5}$$

Assuming that the eigenvalues are ordered, i.e. $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$, the choice of the first $n \ll N$ principal components leads to stochastic signals approximation given by the ratio on the covariances associated with the components, i.e. $\sum_n \lambda_n / \sum_N \lambda_N$.

4.4 PCA-based depth image reconstruction

As mentioned in the section 4.2, the self-localization system proposed in this chapter uses depth images corrupted with missing data, captured from the ceiling. Thus, to validate the concept of the image reconstruction resorting to a PCA eigenspace, a test of reconstructing a depth image corrupted with missing data is performed.

In order to create the PCA eigenspace, a set of 125 depth images is captured along a grid map with a distance of 0.3 m (in x and y axis) in an area of $5m \times 4.5m$. Considering that the Kinect depth sensor has a resolution of 640 by 480 points, and with the purpose of reducing the amount of data stored in the PCA eigenspace, the depth images are cropped to a depth image with 120 by 160 points, extracting the central area of the images, and transforming them into vectors $\mathbf{x}_i \in \mathbb{R}^{19200}$, $i = 1, \ldots, 125$.

With the purpose to test the depth image reconstruction using the proposed extension of the PCA algorithm, a new depth image is captured from the ceiling, corrupted with about 15 % of missing data ratio. The depth image is shown on left side of Fig. 4.3, where the black pixels correspond to the missing data. Applying the proposed algorithm, the depth image reconstruction is performed, decomposing the captured corrupted signals into the orthogonal space, obtaining the corresponding reconstructed image, according to the following steps:

1. Identify the non-corrupted data in the vector $\mathbf{l}$;

2. Substitute the corrupted data $\mathbf{x}_i(j)$ by the corresponding mean $\mathbf{m}_x(j)$;
3. Decompose the signal after the mean substitution into the orthogonal eigenspace $\mathbf{v}$;
4. Compute the reconstructed depth image recovering the signals from the eigenspace computed before.

Analysing the depth images in Fig. 4.3, it is possible to conclude that the PCA is able to construct a similar depth image (right), removing all corrupt pixels present in the original captured image (left). It is also possible to observe that pixels with non corrupted data are not equal to the captured depth image. However, in this case, the PCA algorithm is performing three important tasks at same time: i) compression of the acquisition database; ii) removing the corrupted data of the depth image; iii) and computing an eigenspace to compare the reconstructed depth image with the PCA database, without explicit features extraction.

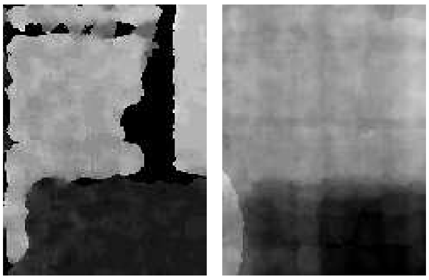

Figure 4.3: Captured depth image with corrupted data (left) and reconstructed through PCA (right).

To analyse the robustness of the corrupted depth image reconstruction technique proposed, a new test was performed, removing depth information from the acquired data. The depth image reconstruction using PCA was simulated, adding from 20 % to 80 % random missing data. Figure 4.4 shows the captured depth images with the added missing data on the top, and the reconstructed images at the bottom. Analysing the results it is perceptible that the reconstruction of the depth image, by the proposed PCA algorithm, provides good results until a missing data ratio of about 60 %. More experimental results showing how the amount of missing date can affect the localization system accuracy will be presented next. However, observing the corrupted depth images, it is possible to see that for higher ratios of missing data, there is too much degradation to reconstruct the corrupted image.

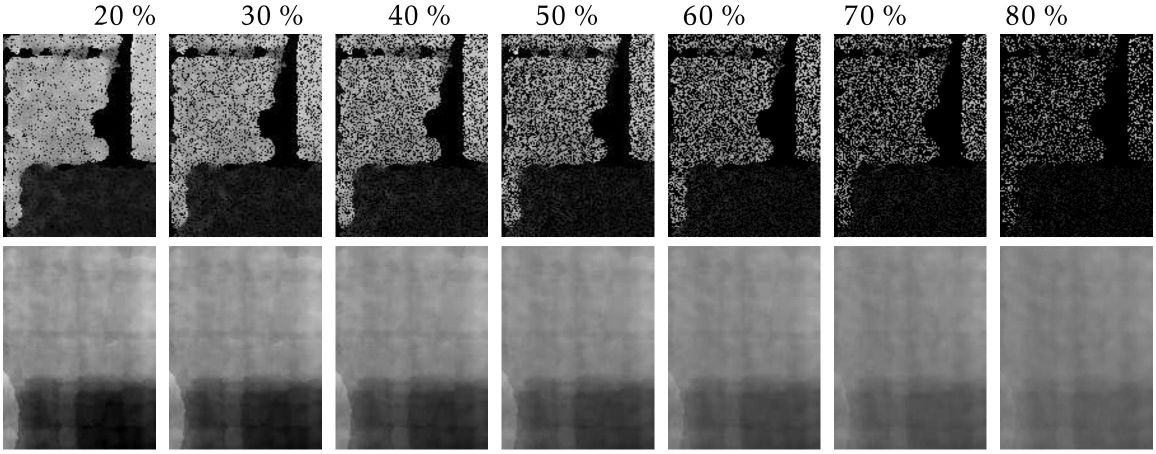

Figure 4.4: Captured depth image with added corrupted data (top) and reconstructed through PCA (bottom).

4.5 Motivation: experimental results along a straight line (1D localization)

As concept validation, the proposed self-localization method was initially developed considering a straight line (1D localization), using the model detailed in section 3.3.1. Thus, to create the PCA eigenspace, a set of 31 depth images has been captured along a straight line with 3 m of length (sampling ratio of 0.1 m). Considering that the Kinect depth sensor has a resolution of 640 by 480 points, and with the purpose of reducing the amount of data stored in the PCA eigenspace, the depth images are compressed with a ratio of 100 : 1, and transformed into vectors $\mathbf{x}_i \in \mathbb{R}^{3072}$, $i = 1, \dots, 31$.

The mobile robot follows along a straight line with constant velocity and the position estimation has been obtained using the model detailed in section 3.3.1. The position estimates are based on data obtained from the on-board sensors and the commands to the actuators, assuming constant values between sampling times (ZOH assumption).

4.5.1 Monte Carlo performance tests

To analyse the mobile robot self-localization methodology proposed in this section, a Monte Carlo test composed of 10 experiments, as described above, have been repeated. Images were captured with a frequency of 5 Hz to be processed by the PCA-based positioning system; Fig. 4.5 gives the localization results obtained in one of those experiments. The results show that the PCA algorithm provides a reasonable approximation to the real robot localization. However, due to the existence of missing data, the position obtained by the PCA algorithm often gives incorrect results.

Analysing Fig. 4.5, it is possible to see that the obtained position often reaches errors greater than 0.1 m (distance at which the images are acquired to the eigenspace). Figure 4.6 shows results of three tests, where it is possible to see the existence of large perturbations in the results of the PCA-based positioning system. Even the fusion of this results with the odometry, through the KF detailed in section 3.3, is not always able to estimate the robot position with accuracy. These wrong estimates happen because the PCA-based positioning

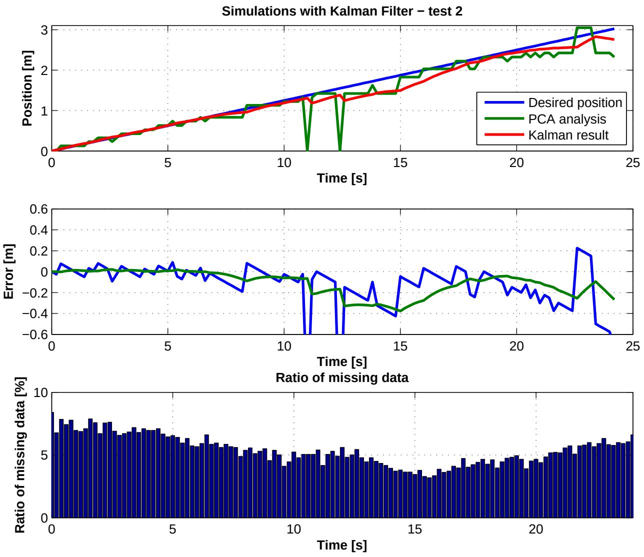

Figure 4.5: Results of PCA-based positioning system and localization estimates from KF.

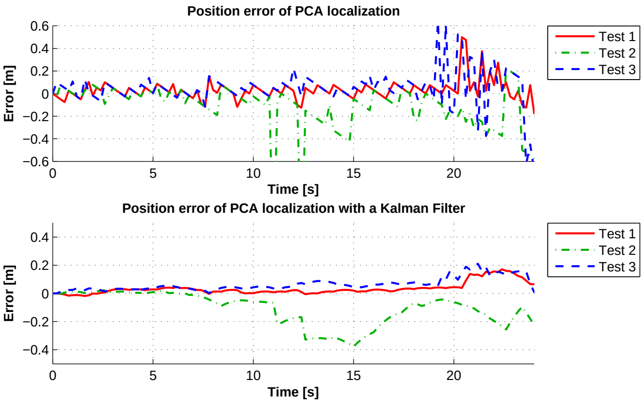

Figure 4.6: Localization errors of tests along a straight line.

system often provides results with large error. As it is possible to see, the position error obtained by the PCA-based positioning system often exceeds 0.1 m.

4.5.2 Monte Carlo performance tests with missing data correction

Following the methodology proposed in section 4.3, successful tests have been made to check the enhanced performance of the localization system in presence of missing data. Thus, to validate this extension to the PCA-based approach, the same acquired depth data has been considered.

Comparing Fig. 4.7 with Fig. 4.5, it is possible to observe that the proposed method is able to eliminate the existing missing data and provide a position value with better accuracy. Analysing the results presented in Fig. 4.8, it is possible to see that the results now obtained present position errors smaller than 0.1 m. Once the proposed method is able to compute the robot position with a better accuracy, its fusion with the odometry, through a KF, always provides smoothly results and near to the real robot localization.

Finally, analysing the histogram of the position error obtained by the PCA-based position system after the 10 performed tests (see Fig. 4.9), it is possible to see that the error of both methods is approximately Gaussian with a mean error close to zero.

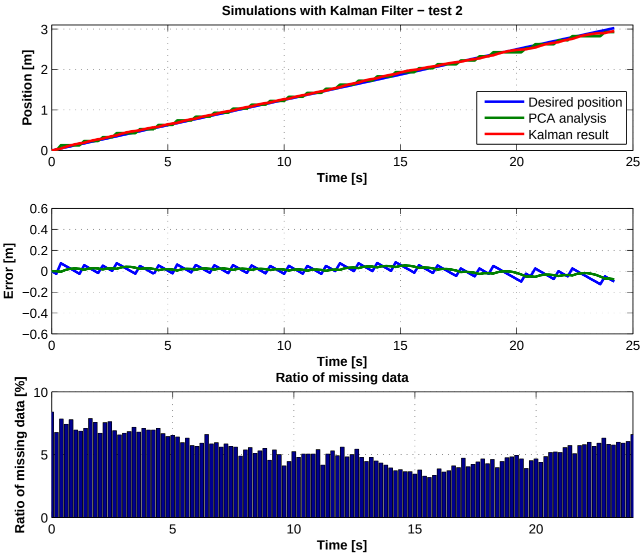

Figure 4.7: Results of PCA-based positioning system and localization estimates from Kalman filter - new method.

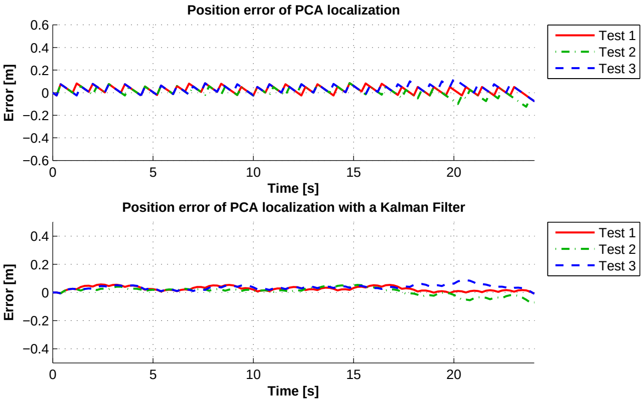

Figure 4.8: Localization errors of tests along a straight line - new method.

Considering that the data to create the PCA eigenspace are acquired with 0.1 m of distance, it is possible to observe that all estimated position errors are less than the sampling distance of PCA eigenspace, while that considering signals with missing data, only about 68 % of results (1 standard deviation) are inside of this distance.

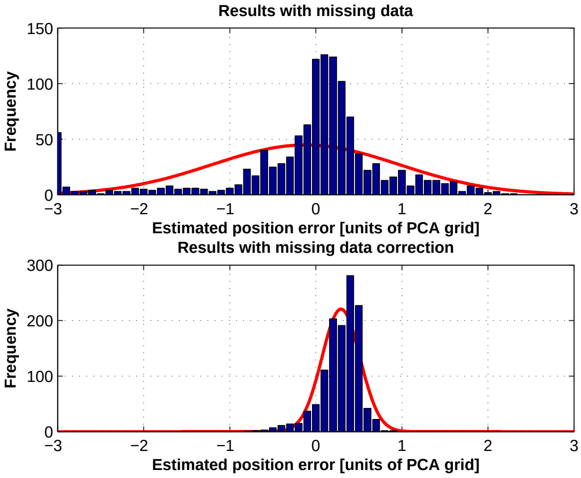

Figure 4.9: Distribution of the estimated position error for both methods, considering a PCA grid with 0.1 m

4.6 PCA-based 2D localization using depth images

In order to solve the problem of 2D mobile robot localization, a new PCA eigenspace has been created with a set of captured depth images along a grid map with a distance of 0.3 m (in x and y axis) in an area of 5 m × 4.5 m. This is a manual process to ensure that all images are captured in the same direction and the depth sensor is in the right grid position (Fig. 4.10). The captured depth images are cropped with a circular mask allowing the rotation and comparison of captured depth images when the robot is in the same position, but with different attitude, during a mission. In order to compress the amount of data, the depth images are sampled with a compression ratio of 100 : 1 and converted into a vector that will be added to the PCA eigenspace. In section 3.5, a similar approach using a RGB camera was followed, but the method revealed to be sensitive to lighting changes.

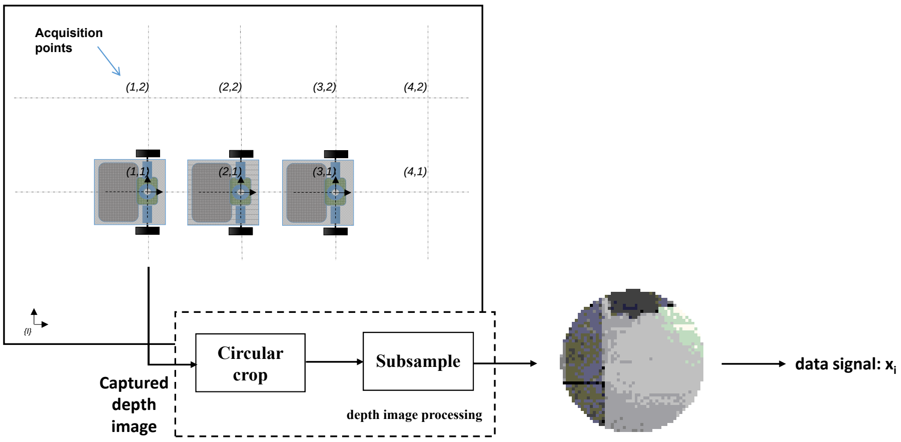

Figure 4.10: Grid map and depth image processing to create a PCA eigenspace.

During the mission, the signal $\mathbf{x}$ is decomposed into the orthogonal space considering only the non-corrupted data. Thus, before the projection of the depth image into the orthogonal space, the mean substitution should be followed, i.e, all j^{th} component of the signal $\mathbf{x}_i$ with corrupted data should be replaced by the corresponding mean value $\mathbf{m}_x(j)$. This method removes the effect of the corrupted data in its decomposition into the orthogonal space, $\mathbf{v} = \mathbf{U}^T(\mathbf{x} - \mathbf{m}_x)$.

The robot position $[x\,y]^T{}_{PCA}$ was obtained by finding a given neighborhood δ, the mosaic whose eigenvector is nearest to the acquired signal decomposed into the orthogonal space:

$$\forall_i \|[x\,y]^T{}_{PCA} - [x_i\,y_i]^T\|_2 < \delta,\ r_{PCA} = \min_i \|\mathbf{v} - \mathbf{v}_i\|_2; \tag{4.6}$$

Given the mosaic i that verifies this condition, its center coordinates $[x_i\,y_i]^T$ are selected as the robot position, obtained by the PCA-based positioning system.

During an experiment, the data captured from the sensors (Kinect, compass and encoders) has been used in a self-localization system based in two KF and the extended PCA-based positioning system, as detailed in Fig. 4.11. The implemented architecture allows the estimation of the robot attitude and position, as well as the angular motion velocity and the robot angular slippage, using only the signals obtained by the on-board sensors. To achieve these estimations, the self-localization system integrates the optimal attitude estimator, detailed in section 2.2, and the 2D position estimator, presented in section 3.4.4.

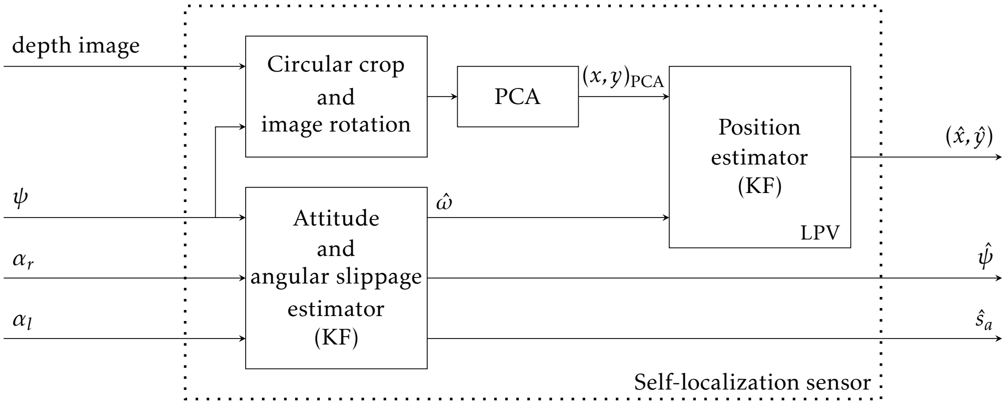

Figure 4.11: Architecture of the self-localization system using depth images.

The following notation is used in Fig. 4.11:

- ψ - attitude angle given by the compass;
- α_r - angle given by the encoder of the right wheel;
- α_l - angle given by the encoder of the left wheel;
- $(x,y)_{PCA}$ - coordinates given by the PCA-based positioning system;
- $(\hat{x},\hat{y})$ - estimated robot coordinates in the world referential;
- $\hat{\omega}$ - estimated angular velocity;
- $\hat{s}_a$ - estimated angular slippage.

Detailing the architecture of the self-localization system presented in Fig. 4.11, the KF depicted on the left of the figure implements the attitude optimal estimator model that is responsible to estimate the mobile robot attitude and the angular slippage (see section 2.2 for details). Once all acquired depth images for the PCA database are taken with the same orientation and compressed with a circular crop (Fig. 4.10), then, during a mission, the acquired depth images must be rotated to zero degrees of attitude, using the compass angle, and compressed with the same circular crop. The position estimator (on the right of the figure) implements the same LPV model detailed in the previous chapter (see section 3.4.4), fusing it with the position obtained by the extended PCA algorithm presented above.

4.6.1 Results for 2D localization with classical PCA algorithm

To observe the effect of corrupted data in the mobile robot self-localization, the depth images captured along the grid, after being subsampled and cropped with a circular mask, as presented before, are compressed using the classical PCA algorithm. The selection of the best eigenvectors is performed through eigenvalues that exceed 85 % of the total of the eigenvalues, creating an eigenspace with 60 eigenvectors.

To test the mobile robot self-localization performance of the proposed approach in a environment, several tests have been performed with the classical lawnmower type trajectory, combining both straight lines and curves, with a 0.1 m/s robot velocity and 5 Hz of sampling frequency. During the robot motion, the real mobile robot trajectory has been measured, allowing the comparison of the estimated position with the real one (ground truth test).

Analysing Fig. 4.12 it is possible to see that, due the existence of missing data, the

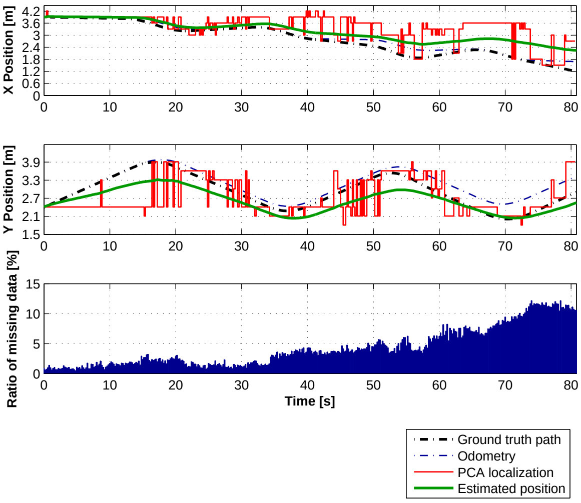

Figure 4.12: Estimated position along time without corrupted data correction.

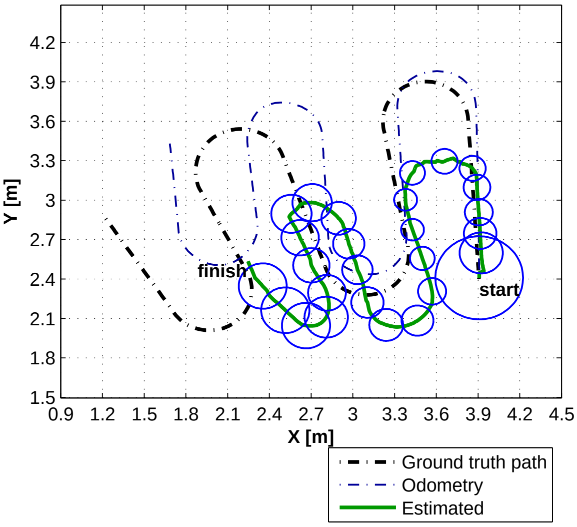

Figure 4.13: Map with estimated position considering a ground truth path, without corrupted data correction.

position obtained by the PCA algorithm often gives incorrect values, causing erroneous estimations in the self-localization system. Figure 4.13 shows the results of the PCA-based positioning system fused with the LPV model, in 2D localization. Analysing Fig. 4.12 and Fig. 4.13, it is possible to conclude that the erroneous position obtained by the PCA, which is caused by the corrupted data, leads the estimator to a far localization than the described by the mobile robot (ground truth path).

4.6.2 Results for 2D localization with depth images reconstruction

Following the PCA algorithm extension proposed in section 4.3, a new PCA eigenspace was created, computing the mean ensemble and the signals covariance, considering only the non-corrupted data. Thus, choosing the best eigenvectors, which eigenvalues exceed 85 % of the total, an eigenspace with 30 eigenvectors was created.

As it is possible to see in Fig. 4.14, the position results obtained by the extended PCA algorithm are very close to the ground truth trajectory. Therefore, fusing the kinematic model of the robot with the position obtained by extended PCA algorithm in the KF allows estimating position values with a very good accuracy.

Figure 4.15 shows the position estimated with the ground truth trajectory and the position obtained by the odometry. Comparing the results of the odometry with the estimated position it is possible to see an angular slippage in motion, that is increasing the difference between the estimated attitude and the one obtained by the odometry along time. This angular slippage is caused by systematic errors, such as uncertainties in the dimensions of the wheels, eccentric shaft problems, misalignment of the shafts, etc. It is possible to observe that in the initial part of the trajectory the estimator obtains a result close to the odometry. However, the localization system can approximate the estimated

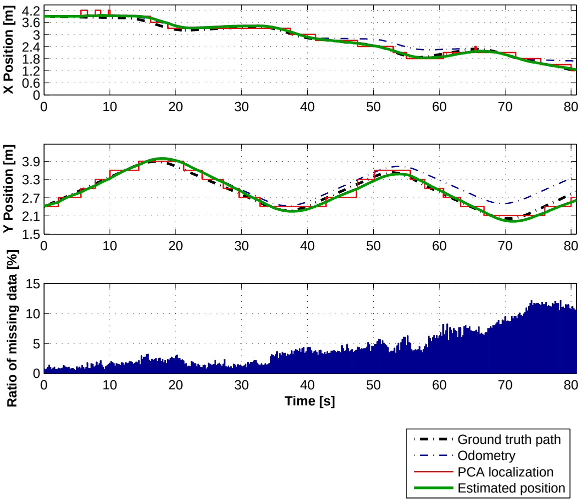

Figure 4.14: Estimated position along time.

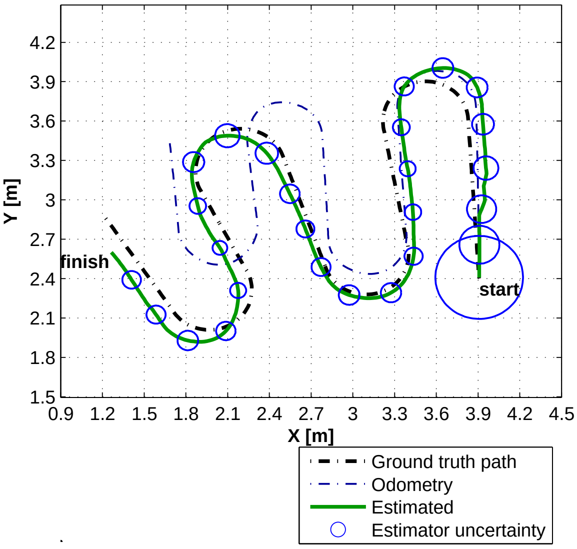

Figure 4.15: Map with estimated position considering a ground truth path.

position with the ground truth trajectory.

The Kalman gains stabilization along time are shown in Fig. 4.16, where it is possible to see that they converge in few sampling instants.

Analysing the results of the attitude estimator in Fig. 4.17, it is possible to observe that the estimated attitude is very close to the ground truth, allowing to conclude that this KF provides results with good accuracy. Furthermore, analysing Fig. 4.18, it is possible to observe the existence of an angular slippage of −0.5 rad/s (positive for slippage in clockwise direction), that is detected at 40 s by the attitude estimator. Looking at Fig. 4.14 after 40 s

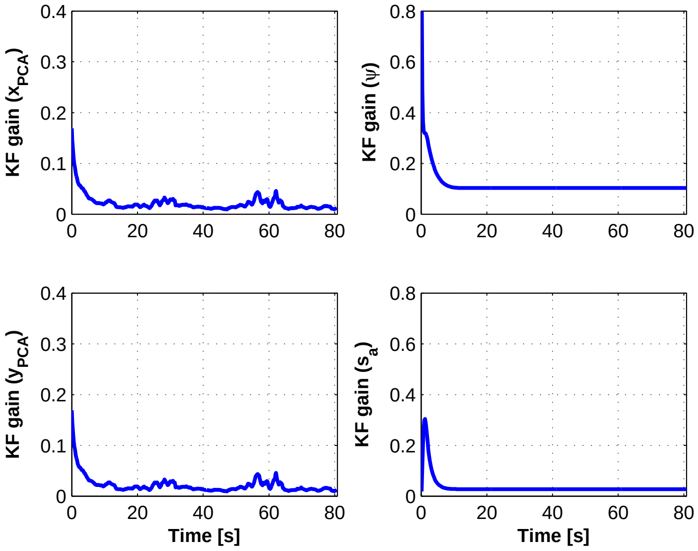

Figure 4.16: Evolution of the Kalman filter gains: position estimator (left) and attitude estimator (right).

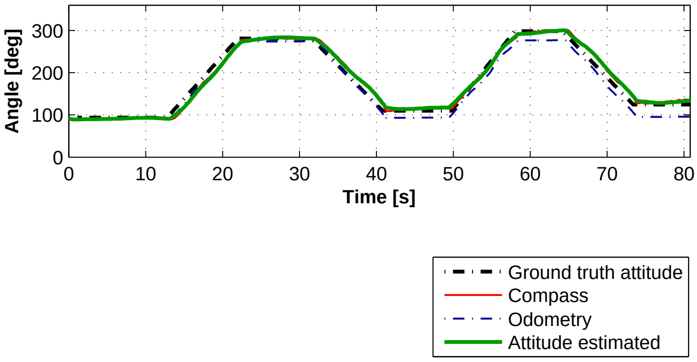

Figure 4.17: Evolution of the attitude estimated considering a ground truth path.

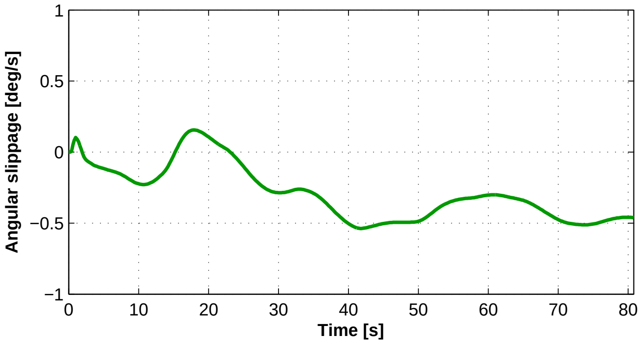

Figure 4.18: Angular slippage estimated.

(instant which is detected angular slippage), the results of the position estimator are closer to the ground truth path than the odometry.

Finally, analysing the histograms of Fig. 4.19, it is possible to conclude that the statistical distribution of the estimated position errors is approximately Gaussian with a mean close to zero. Moreover, comparing the variation of the distribution with the distance of the grid map acquired to create the PCA eigenspace (0.3 m), it is possible to see that the proposed self-localization system is able to estimate the position with an error less than the distance between the acquired depth images.

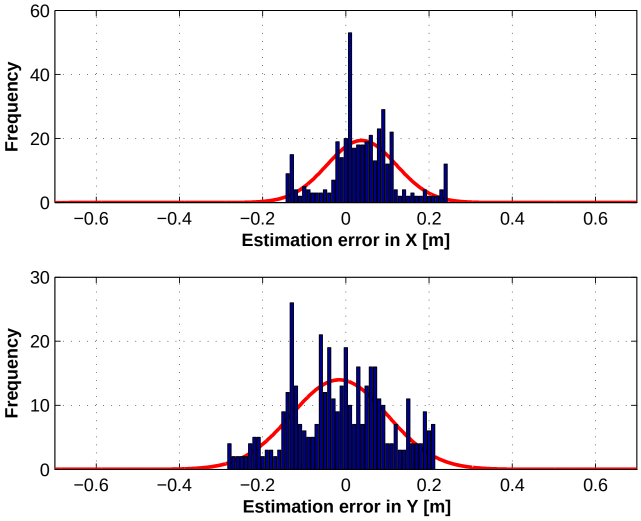

Figure 4.19: Distribution of the estimated position error for both axis.

4.6.3 Results for 2D localization with imposed corrupted data

The results presented in the previous section show the performance of the proposed algorithm in mobile robot localization considering depth images corrupted with missing data, not exceeding 13 % of the signal data (Fig. 4.14). The presence of the corrupted data in depth images is due to the geometry and properties of some objects that disturb several waves, causing the presence of the missing data. Therefore, as it is possible to observe in Fig. 4.14, the presence of corrupted data in the depth images is completely random along the trajectory.

Thus, in order to analyse the robustness of the proposed PCA-based positioning system in the presence of missing data, new self-localization tests have been performed simulating depth images with a missing data ratio between 20 % and 90 %. For this test the same data captured from the sensors used in the previous section is considered, but random corrupted data on the captured depth images was added. The amount and position of the missing data present in the captured depth images was analysed, guaranteeing that the simulated corrupted data are randomly added in the non-corrupted data, keeping the same ratio in all depth images along the experiment. Figure 4.20 shows the original captured depth image in the initial position and the transformed data with the missing data added randomly, respectively with a ratio of 40 %, 60 % and 80 % of the signal length.

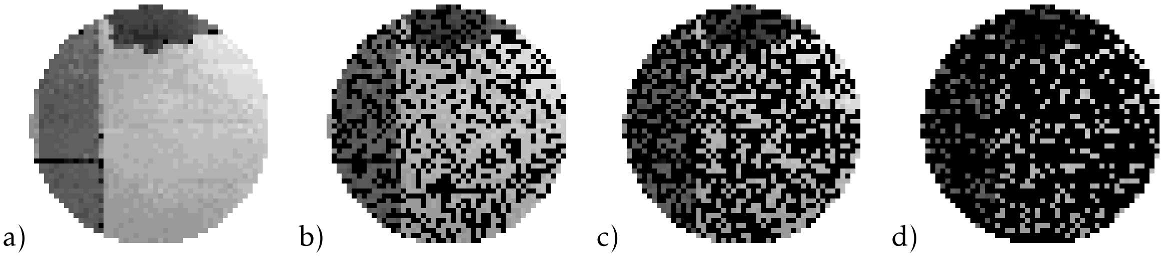

Figure 4.20: First captured depth image: original (a), and with added imposed corrupted data ratio to a ratio of 40 % (b), 60 % (c) and 80 % (d).

Analysing the results, Fig. 4.21–4.22 show that the proposed method is able to find a depth image close to the captured one along the travel, even with a ratio of 40 % corrupted data in all signals. Thus, the fusion of the extended PCA-based positioning system with the KF allows the estimation with accuracy, presenting similar results to the obtained with the original data, where the ratio of missing data is less than 15 %, allowing the estimation with

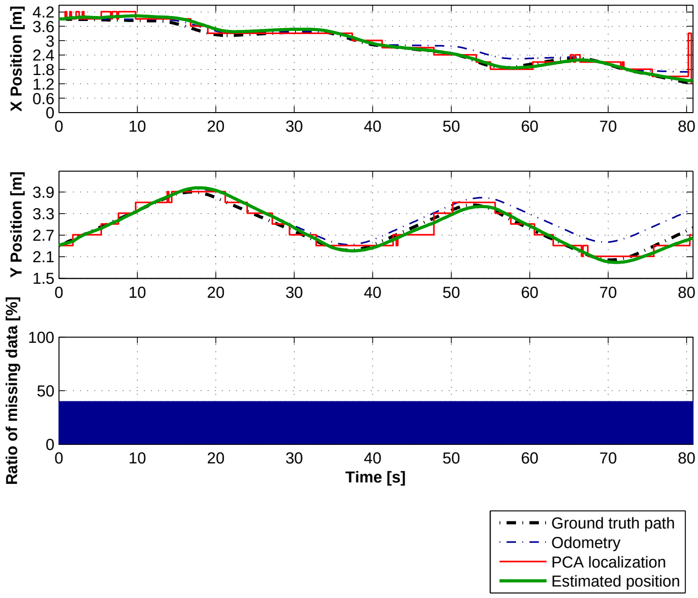

Figure 4.21: Estimated position along time with 40 % of imposed corrupted data.

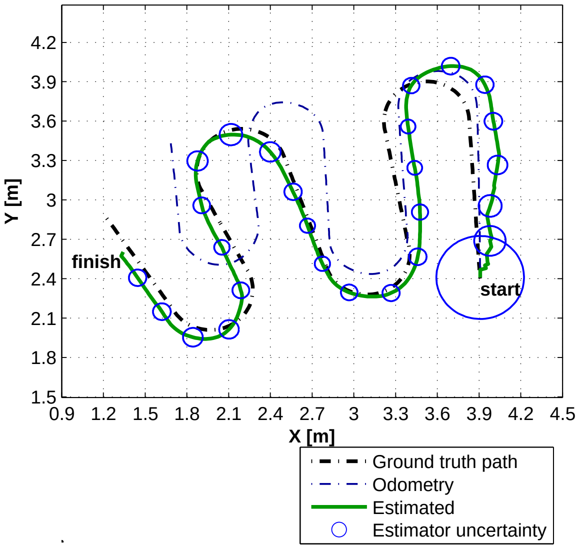

Figure 4.22: Map with estimated position considering a ground truth path, with 40 % of imposed corrupted data.

accuracy, after the stabilization of the angular slippage.

Analysing the results using depth images with 60 % of missing data ratio, presented in Fig. 4.23, it is possible to observe that the proposed algorithm is able to find the correct eigenvector in most of the acquired depth images. However, Fig. 4.23 also shows instants of time in which, given the large amount of data corrupted (see Fig. 4.20), the algorithm found principal components similar to another captured image in a far location. This localization error, constant for several sampling times, led the KF to provide biased estimates, as it can be seen in Fig. 4.24.

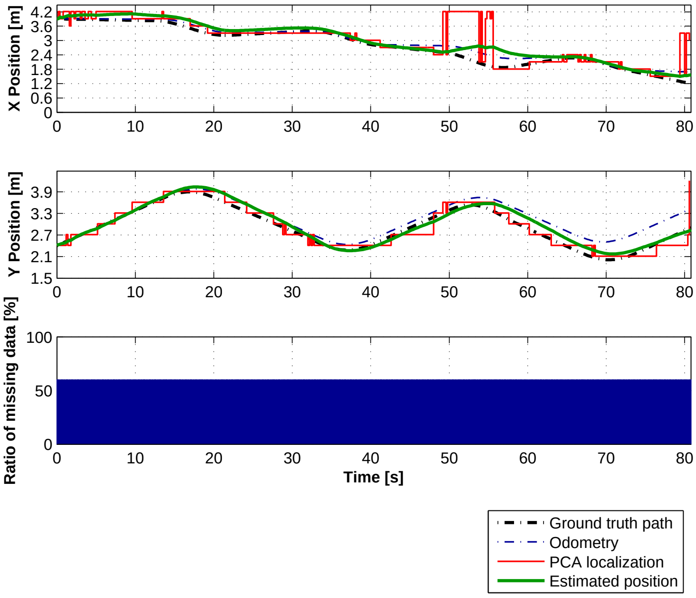

Figure 4.23: Estimated position along time with 60 % of imposed corrupted data.

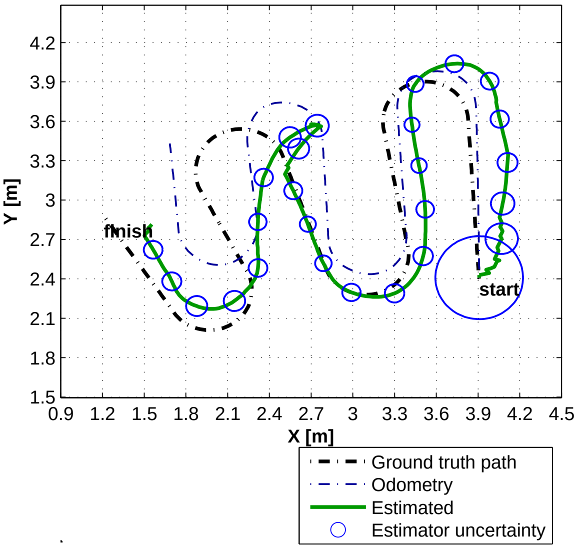

Figure 4.24: Map with estimated position considering a ground truth path, with 60 % of imposed corrupted data.

Finally, Fig. 4.25 shows the estimation uncertainty of the PCA-based positioning system. As it is possible to observe, the measurement accuracy is related to the ratio of corrupt data, but its relation is non-linear. The estimation uncertainty is approximately equal in x and y coordinates to low ratio of corrupt data, while for high ratios, the accuracy is different for both axis. Moreover, analysing the results presented in Figs. 4.21–4.24 and the estimation uncertainty in Fig. 4.25, it is possible to conclude that the extended PCA-based positioning system is able to estimate an accurate position, considering depth images with corrupted data until about 50 % of ratio.

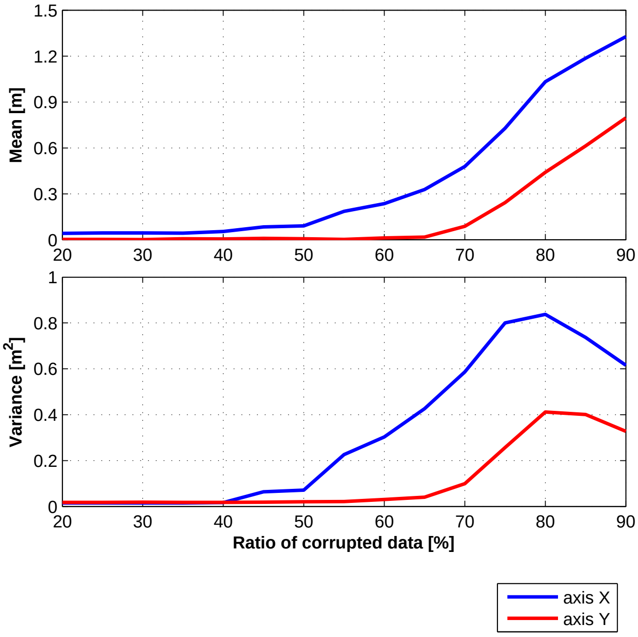

Figure 4.25: Uncertainty of the PCA-based self-localization system with imposed corrupted data correction.

4.6.4 Global stability with wrong initial position and attitude

As shown in the previous chapter, the self-localization system stability is an important characteristic since it ensures that the estimator always converges to the real value for any initial condition. Considering that the proposed self-localization system is composed by two KF, the system global convergence is only achieved if each one is globally stable. For this purpose, three incorrect initial conditions for both position and attitude have been considered, as it is detailed in Table 4.1.

The results depicted in Fig. 4.26 show that, considering the incorrect initial conditions the robot would follow, in open-loop, the same lawnmower trajectory, but in a different direction. Considering these conditions, the position computed from odometry could diverge from the robot real position. However, the two KF can stabilize the position close to the real. Figure 4.27 shows that the KF results are always stable and exhibit fast convergence to the ground truth.

Table 4.1: Initial conditions of position and attitude stability validation in a lawn-mower trajectory.

	x_0 [m]	y_0 [m]	ψ_0 [°]
Robot position	3.9	2.4	90
Re-localization 1	4.4	2.4	0
Re-localization 2	4.4	1.9	315
Re-localization 3	2.9	1.6	225

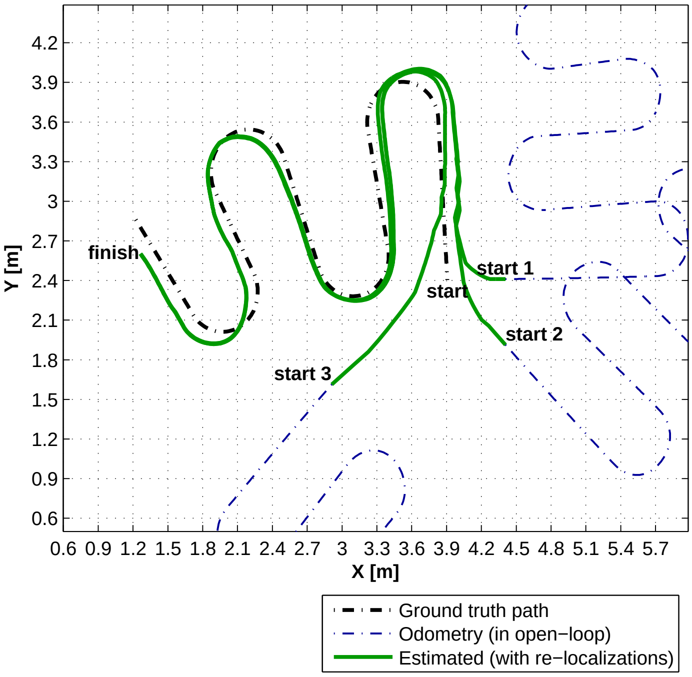

Figure 4.26: Results of stability tests considering a wrong initial position and attitude estimates (map 2D).

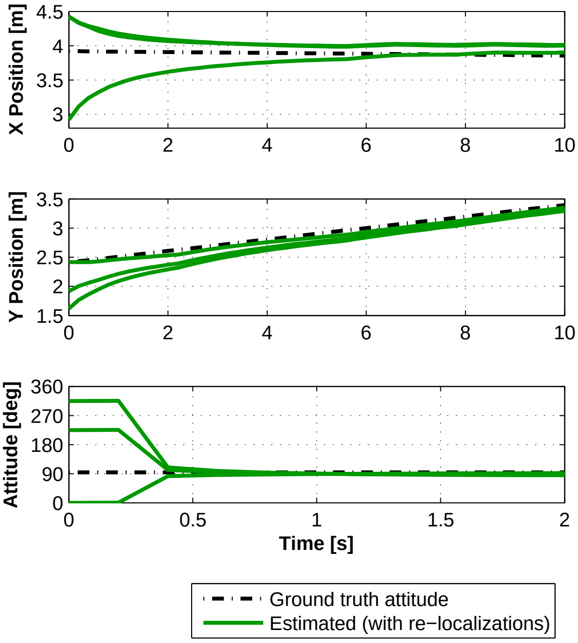

Figure 4.27: Results of stability tests considering a wrong initial position and attitude (estimation along time).

4.6.5 2D localization results for a longer trajectory

After the validation of the proposed PCA-based self-localization system considering depth images corrupted with missing data, another test has been performed, expanding the motion of the robot to a large area and longer trajectories. Thus, a set of 1115 snapshots with depth images being captured in an area of 18.9 m × 9.6 m, along the grid map with a distance of 0.3 m, keeping the robot with the same zero attitude (see Fig. 4.10). The captured depth images are cropped with the same circular mask and sampled with a compression ratio of 100 : 1 and converted into a vector that will be added to the PCA eigenspace. Therefore, analysing the corresponding PCA eigenvalues and selecting the same number of eigenvectors (30) of the previous experience, the selected components explain the images variability in an excess of 93 %. This corresponds to a reduction of 99.9 % in the memory resources when compared with the capacity needed to store the captured database, and 98.2 % when compared with the size after the sub-sample.

Since the physical structure of the environment when the test is performed is composed by equal equipments (pipes, HVAC units, light fixtures), distributed repetitively along the ceiling, the increase of the mapping area to 18.9 m × 9.6 m led the acquisition of similar images at different locations (Fig. 4.28). In consequence of the existence of repetitive scenarios, when the PCA-based positioning system is comparing the eigenvector with all eigenspace, it is easily possible to find depth images, similar to the image captured during a mission, in a far localization. This causes large uncertainty in the position estimated by the PCA algorithm and, in consequence, in all self-localization system.

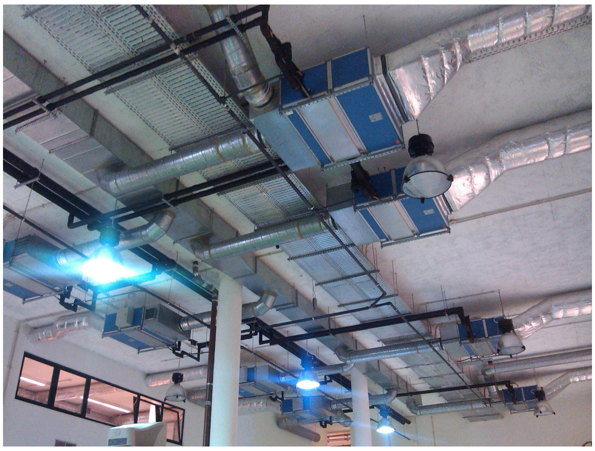

Figure 4.28: Ceiling view of the environment with repetitive elements.

To solve the problem of localization in scenarios with repetitive elements, the searching considering a neighbourhood δ around the last mobile robot estimated position, proposed in equation (4.6), has been applied. In the performed test, the neighbourhood radius is variant and defined based on the eigenvectors distance obtained in last position estimation, through the linear relation $\delta = f_\delta \cdot r_{PCA}$. Considering the repeatability characteristics of the environment and the eigenvectors distance, the linear relationship was tunned to $f_\delta = 5 \times 10^{-4}$. This value ensures the searching of the nearest eigenvector in a radius between 1.5 m and 6.5 m around the estimated mobile robot position $(\hat{x}(k), \hat{y}(k))$, as depicted in Fig. 4.29.

To test the mobile robot self-localization performance, considering repetitive scenarios, several tests have been performed along a predefined path with length of 93 m, combining both straight lines and curves, in order to travel two laps inside the large mapped area, following the same method of the last experiments. Thus, the robot was moving with a

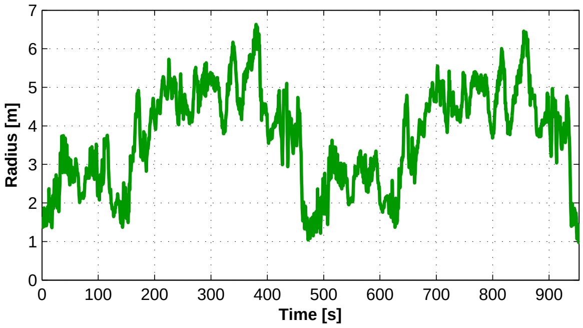

Figure 4.29: Radius of searching neighborhood around the robot along time.

velocity of 0.1 m/s and depth data is acquired with a 2.5 Hz of sampling rate and the real mobile robot trajectory was measured, allowing the comparison of the estimated position with the real one (ground truth test).

Figures 4.30 and 4.31 present the results of the self-localization system, considering the PCA searching in a neighborhood, the position estimated by the KF and the robot real path, measured in the ground. The results presented in Fig. 4.30 show that the extended PCA algorithm, searching in neighborhood around the last robot estimated position, is able to achieve an accurate position of the robot in a large area with repetitive scenario, allowing good performance on the global self-localization system. Note that the precision of the grid

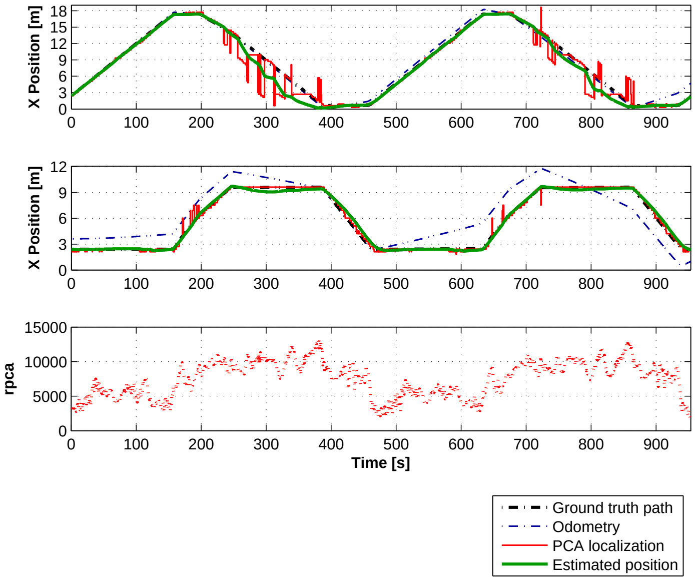

Figure 4.30: Estimated position along time, searching in a neighborhood.

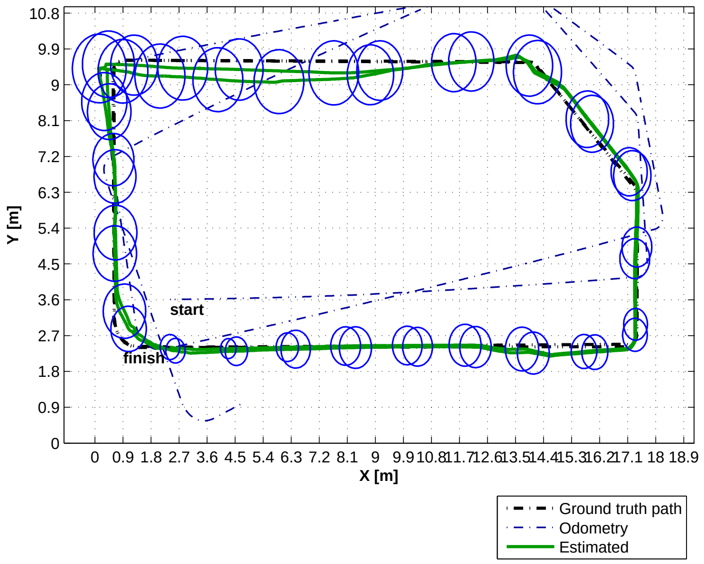

Figure 4.31: Map with estimated position considering a ground truth path, searching in a neighborhood.

with depth images is 0.3 m of distance.

In Fig. 4.31 it is possible to see that the self-localization system is able to converge the estimated position with the ground truth path while the odometry of the robot diverges completely from the real path. The blue circles represents the position uncertainty obtained by the KF. As it is possible to observe in Fig. 4.31, when the robot is in the top area, the uncertainty increases. The performance degradation occurs due to the fact that in this area the ceiling has less information. Nevertheless, Fig. 4.30 and Fig. 4.31 show that the robot is able to estimate its own position.

4.7 Conclusions

This chapter presented and analysed the use of ceiling depth images in a PCA-based localization system, aiming the robustness improvement with different lighting conditions. The existence of missing data in depth images is sometimes inevitable and it can induce a positioning system to an erroneous localization. Thus, an extension of a PCA methodology aiming to avoid the negative impact of missing data in signals has been developed and experimentally validated, allowing the follow advantages: i) compression of the acquired database; ii) corrupted data remotion in the depth image; iii) creation of new eigenvectors to compare the reconstructed depth image with the PCA database.

The implemented enhanced PCA-based localization system has been applied in a mobile robot with a Kinect sensor installed on-board, looking upwards to the ceiling, where the depth sensor often provides signals with missing data, caused by IR beams that not were reflected.

All tests were successfully performed, allowing to conclude that the proposed approach is useful in a number of mobile robotics applications where the existence of missing data is inevitable and causes a localization systems performance degradation. The robustness test allows to conclude that the proposed algorithm is able to estimate an accurate position, considering depth images with corrupt data ratio up to 50 %. The method has been tested in a environment with equal equipment installed repetitively in the ceiling at different positions, causing repeatable scenarios in the acquired depth images during the robot missions. Searching for the closest vector in a neighbourhood demonstrates the ability of PCA to be implemented in a system localization with environment repeatability. Moreover, the proposed method allows to validate the application of the Kinect depth sensor in a mobile robot localization system based on an extension of a classical PCA algorithm to operate in unstructured environments.

The integration of the PCA-based positioning system with linear KF allows to obtain a localization system globally stable, under the Gaussian approach. The method was successfully validated in a self-localization system, using only on-board sensors and estimating the position with a global stable error dynamics.

5

Mobile robot navigation using a PCA-based self-localization system

5.1 Introduction

Navigation is a crucial and an inherent task to the execution of missions for which mobile robots are designed. When a map-based navigation is considered, the mobile robot must move itself like a "goal oriented machine that can sense, plan and act" (Corke, 2011, p. 92). The previous chapters presented approaches on how a mobile robot can self-localize in an indoor environment, using only its own sense, i.e., the data extracted from a sensors package installed on-board. The navigation system detailed in this chapter resorts from the knowledge about the ceiling mapping, covering the entire environment free-space along a grid, to perform the mobile robot planning and acting tasks.

The use of roadmaps is common to plan the mobile robot path, being the Generalized Voronoi Diagram (GVD) one classical approach to create routes, into the free space, defined by a set of points into the free space that maximize the distance to the obstacles. When the obstacles and free space are represented using an occupancy grid, it is common to obtain the GVD based on a grid (Barraquand and Latombe, 1991; Yang and Hong, 2007; Kuderer et al., 2014; Garrido and Moreno, 2015). The path is planned using minimization algorithms, such as the Dijkstra algorithm. This approach reduces the planning complexity to a few possible routes and, simultaneously, ensures that one safety path to the destination is found.

In addition to planning, the position control along the path is needed to ensure that the robot follows it until reaches the goal position (Aicardi et al., 1995; Park and Kuipers, 2011; Bascetta et al., 2016). Navigation Functions (NF) are a type of Artificial Potential Fields (APF) that has the advantage to plan and control the mobile robot along an environment, ensuring an unique minimum located in the goal. This approach was implemented to control the pose of one non-holonomic mobile robot (Tanner and Kyriakopoulos, 2000; Tanner et al., 2001), multi-robots teams (Loizou and Kyriakopoulos, 2003; Dimarogonas et al., 2006; Widyotriatmo and Hong, 2011) or UAV teams (Chen et al., 2007).

In this chapter, the navigation system using the PCA-based self-localization system described in previous chapters is detailed and experimentally validated. The developed system configures the environment from the ceiling mapping as an occupancy grid. Then, implementing the approach proposed by Barraquand and Latombe (1991), the GVD corresponding to the real environment free space is build. Based in Tanner and Kyriakopoulos (2000) and Loizu et al. (2004), a motion controller resorting to a Dipolar Navigation Function (DNF) was defined to drive the robot toward temporary goals, along a waypoints' set. Then, the robot moves toward the goal along a safety path, while its pose is estimated by the PCA-based self-localization system.

Thus, this chapter is organized as follows:

- The proposed navigation system architecture, that integrates the PCA-based self-localization system, is presented in section 5.2, being also detailed the main steps for its implementation.
- The section 5.3 address the path planning steps to achieve the waypoints' set toward a goal. Thus, this section details the GVD construction corresponding to the ceiling mapping and the implemented approach to select these waypoints.
- The local controller based on a DNF is detailed in section 5.4, aiming to control the robot motion toward each waypoint.
- The section 5.5 illustrates the experimental results obtained with the navigation of a mobile robot in an indoor real environment, using the PCA-based self-localization system for ceiling depth images.
- Finally, in section 5.6 some conclusions about this chapter are drawn.

5.2 Navigation architecture

This section presents the proposed navigation system architecture for indoor mobile robot, using the self-localization system addressed in this chapter. This system allows the navigation from a point to another, using only on-board sensors and the knowledge about the environment acquired during the ceiling image mapping for the PCA database. The proposed navigation system for indoor mobile robots is depicted in Fig. 5.1.

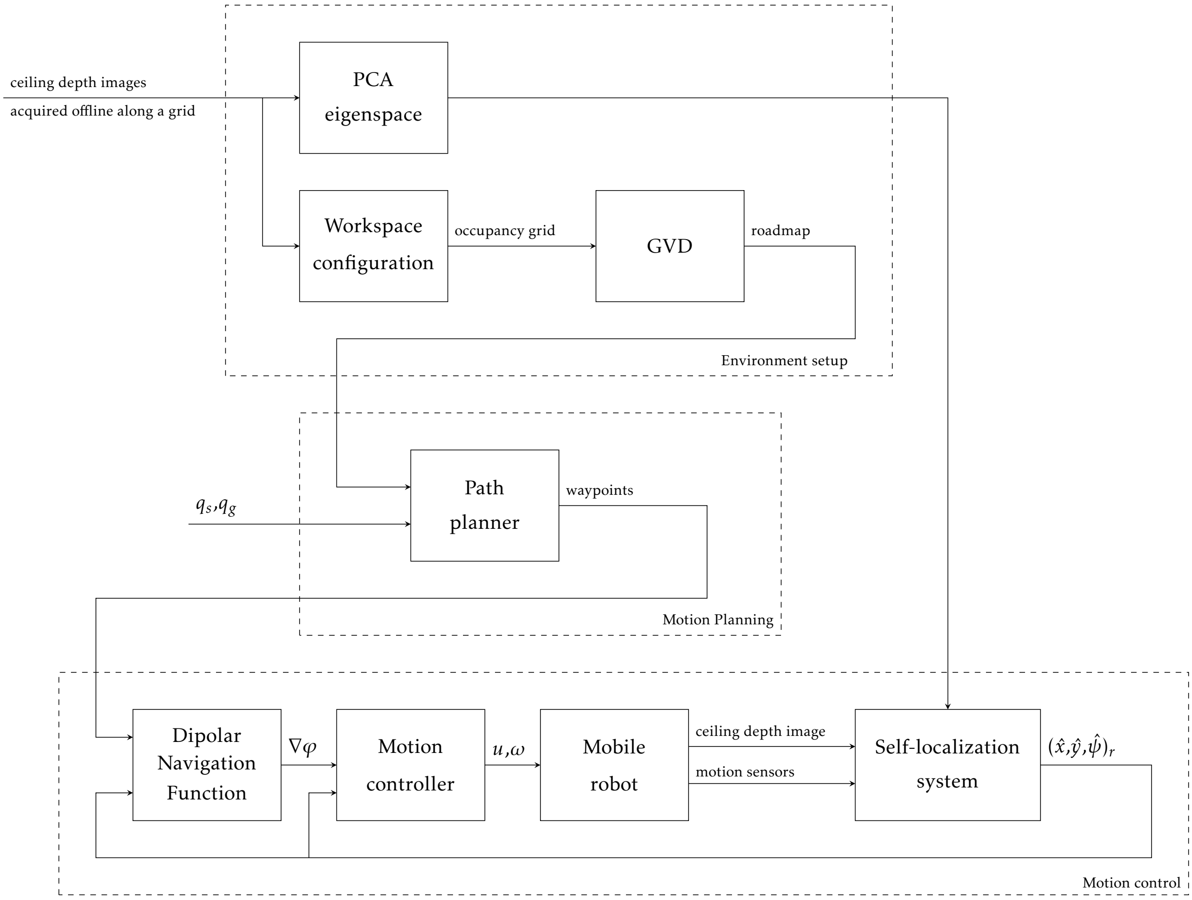

Figure 5.1: Navigation system architecture.

The developed navigation system is designed for indoor mobile robots navigation, using the PCA-based self-localization system detailed in chapters 3 and 4. In this self-localization systems, a set of ceiling depth images are acquired from the environment along a grid, covering all environment free-space (see Fig. 4.10). The acquired ceiling depth images are converted to an eigenspace, through the PCA algorithm (section 4.3), to create a compressed database. The images acquisition process and the PCA compression is only performed once, allowing the robot to know its environment, in which it will move.

Taking into account that the implemented self-localization system requires, initially, an off-line mapping, the navigation system is designed to take advantage of the knowledge acquired during this process. Thus, after the eigenspace computation, the robot workspace is defined through an occupancy grid according with the PCA databse. To configure the robot workspace, the occupancy grid was marked as free or occupied, as the ceiling image was captured or not in the corresponding grid position.

The occupancy grid definition allows to compute a roadmap into the robot free space, represented by a GVD. Similar to the eigenspace construction, the GVD-based roadmap is computed only once. Later, the roadmap is used to plan all paths that the robot will perform within its workspace. The configuration space, performed in this off-line process, requires the definition of a safety free space limit. Thus, all free cells that are in the occupancy grid external border are also set as occupied. This process creates a virtual obstacle around the workspace, usefull to create the GVD.

Aiming to perform a mission, before the robot starts its motion towards a goal, a path planner computes the shortest route on the roadmap between these two positions. The reference path will be defined by a set of waypoints, corresponding to the GVD vertices, found along the shortest route.

To start the mission, the first waypoint is considered as a temporary goal that the robot will have to reach. To move the robot toward this goal, a DNF is implemented to generate a smooth path through the potential field gradient of a local world. In the implemented approach, the temporary goal is the attractive field, while the boundary of a circular world around it is the repulsive field, that repels the robot to the center. The DNF gradient is computed in a closed loop, being the mobile robot pose given by the estimates results obtained from the PCA-based self-localization system.

Finally, to generate the mobile robot motion, a controller defines the velocity that is sent to each wheel so that the robot follows the DNF negative gradient. When the robot gets closer to the temporary goal, it is changed to the next one. This process repeats until the mobile robot reaches the final goal.

5.3 Planning using a GVD-based roadmap

The proposed navigation system uses a roadmap to define the path that takes the robot from a point to another. Roadmaps are usually implemented by robots in a similar way as highways are used along a journey. Thus, to move from a point to another using an highway network, the driver plans the path to take an highway near to the start place. Then he plans the path along the highway until reach near to the destination place. Finally, he plans the path from the highway departure point to his destination (Choset et al., 2005, p. 108).

5.3.1 Generalized Voronoi diagram

The roadmap implemented in the navigation system was defined by a GVD, which is a method to design a roadmap that maximizes the distance between the robot path and the obstacles (Choset et al., 2005, pp. 118-119). The roadmap represented by a GVD is composed by a set of points, denoted as Voronoi edges, equidistant to the two most nearby obstacles. When more than two edges intersect at the same point, this correspond to a meet point (also called as vertex), equidistant to three or more obstacles. Thus, in a GVD-based roadmap, the edges are equivalent to the roads and the vertices to crossroads. In the implemented approach, a set of GVD vertices positions are selected in the planning step, defining the waypoints that will be used as temporary goals.

The use of a GVD has the following advantages: i) it reduces the free space size represented by an occupancy grid to a few set of points connected between them, decreasing significantly the complexity in the planning step; ii) it ensures the definition of a path between the start and goal positions, avoiding the obstacles represented on the map; iii) it defines a clearance path, since the most of planning is done along points, which position maximizes the distance to obstacles.

To construct a roadmap it is assumed that the robot is moving inside an workspace $\mathcal{W}$ occupied with obstacles. Lets consider an i^{th} obstacle, $\mathcal{WO}_i$, inside the workspace. This representation in a configuration space is defined by a set of positions, $\mathbf{q}$, that are intercepted by the robot, $R(\mathbf{q})$. Thus, the corresponding obstacle space is defined as follows:

$$\mathcal{Q}_i = \mathbf{q} \in \mathcal{Q} | R(\mathbf{q}) \bigcap \mathcal{WO}_i \neq 0 \tag{5.1}$$

and the free space is defined as the set of all points that are not intercepted by any obstacle, as follows:

$$\mathcal{Q}_{free} = \mathcal{Q} \backslash \left(\bigcup_i \mathcal{Q}_i \right) \tag{5.2}$$

The GVD is a roadmap that splits the free space in regions which positions, $\mathbf{q}$, are closer to an obstacle, i, than all others, a. These regions, denoted as Voronoi regions, are defined as:

$$\mathcal{F}_i = \mathbf{q} \in \mathcal{Q}_{free} | d_i(\mathbf{q}) \leq d_a(\mathbf{q}), \forall a \neq i \tag{5.3}$$

where $d_i(\mathbf{q})$ is the closer distance between a configuration, $\mathbf{q}$, and an obstacle $\mathcal{Q}_i$, defined as the closer obstacle, and $d_a(\mathbf{q})$ the distances to all the others. The distance to the closer obstacle $\mathcal{QO}_i$ is defined by the follow equation:

$$d_i(\mathbf{q}) = \min_{\mathbf{c} \in \mathcal{QO}_i} d(\mathbf{q}, \mathbf{c}) \tag{5.4}$$

Being $\mathcal{QO}_i$ and $\mathcal{QO}_j$ the space occupied by two obstacles, the GVD is the set of points equidistant to both:

$$S_{ij} = x \in \mathcal{Q} | d(i) = d(j) \tag{5.5}$$

The definition of the two-equidistant points S_{ij} restricts the Voronoi region boundary points:

$$F_{ij} = \mathbf{q} \in S_{ij} | d_i(\mathbf{q}) = d_j(\mathbf{q}) \leq d_a(\mathbf{q}), \forall a \tag{5.6}$$

The GVD is finally defined by the union of the two-equidistant faces:

$$\mathrm{GVD} = \bigcup_i \bigcup_j \mathcal{F}_{ij} \tag{5.7}$$

5.3.2 GVD-based roadmap definition from the wave front algorithm

As detailed in section 5.2, the mapping for the PCA-based positioning system is performed by the capture of ceiling depth images, covering the entire free-space along a grid. This information is used to configure the workspace as an occupancy grid. Thus, free cells correspond to grid positions where ceiling depth images were captured, and occupied cells where it was impossible for the robot to capture them. This occupancy grid is composed by a binary information, where zero (0) is used to mark free cells and one (1) for occupied cells. Note that, as described above, the free-space cells corresponding to the workspace boundary are also marked as occupied to create, this way, a "virtual wall" around it.

The GVD construction for discrete grid configuration spaces is computed by a distance grid, using the brushfire algorithm (Choset et al., 2005, pp. 128-129) to find cells common to several obstacles. Barraquand and Latombe (1991) propose the application of a wave front expansion procedure to define the distance grid and, in parallel, the points that belong to the GVD.

Thus, lets consider the occupancy grid described above, where the occupied cells are labeled with one (1) and the free cells are labeled with zero (0). According to this approach, the grid distance is computed by the application of a wave front propagation from the obstacles and the external workspace border, along the free-space. Thus, all boundary points are labeled with a distance equal to zero ($d(\mathbf{q}) = 0$). Then, all neighbors cells to the boundary that are inside $\mathcal{Q}_{free}$ are labeled to one ($d(\mathbf{q}) = 1$); next, all cells neighboring to those that have $d(\mathbf{q}) = 1$ and are inside $\mathcal{Q}_{free}$ are labeled with two ($d(\mathbf{q}) = 2$); etc. The wave front is, then, propagated until all free cell are labeled with the corresponding distance to the closer obstacle.

In parallel, the GVD is defined by the set of points where the "waves" propagated by the obstacles and environment boundary meet. The points where the wave front collides have the same distance to two different obstacles and, then, these are the points on the GVD (Choset et al., 2005, pp. 128-129). When the wave front collisions results in multiple neighbors, the points selected to the GVD are defined through a thinning algorithm (Haralick and Shapiro, 1991, pp. 170-171).

5.3.3 Path connectivity using a GVD-based roadmap

Path planning is an essential task to move from a point to another. Usually, when a driver moves toward a destination place, he plans the pathway before the motion start. The path is usually select by choosing the shortest distance between two places. In the case where an highway system is available, the driver searches for points close to his current localization to take the highway and the desired destination to deliver it. Then, he searches for the shortest route, using only the highway system, which links these two points.

The mobile robots path planning using a roadmap has a very similar approach to traveling along an highway. Thus, before the mobile robot starts the motion from the current position to another, it needs to plan its own path.

Lets consider a mobile robot located in a position $\mathbf{q_s} \in \mathcal{Q}_{free}$ that will move towards a destination position $\mathbf{q_g} \in \mathcal{Q}_{free}$. The path planning consists in the definition of the $\mathbf{q} \in \mathcal{Q}_{free}$ points set that allows the connection between $\mathbf{q_s}$ and $\mathbf{q_g}$. This connection is performed by the follow roadmaps properties (Choset et al., 2005, p. 109):

- Accessibility: there is a path from $\mathbf{q_s} \in \mathcal{Q}_{free}$ to some $\mathbf{q_s'} \in \mathrm{RM}$;
- Departability: there is a path from $\mathbf{q_g'} \in \mathrm{RM}$ to some $\mathbf{q_g} \in \mathcal{Q}_{free}$;
- Connectivity: there is a path in RM between $\mathbf{q_s'}$ and $\mathbf{q_g'}$.

Lets analyse the case where the path is defined between two points coincident in the roadmap, i.e., $\mathbf{q_s} = \mathbf{q_s'} \in \mathrm{RM}$ and $\mathbf{q_g} = \mathbf{q_g'} \in \mathrm{RM}$. Since the GVD is defined by a set of points that define a roadmap inside the free space ($\mathrm{RM}(\mathbf{q}) \in \mathcal{Q}_{free} \forall \mathbf{q}$), there is a path that connects $\mathbf{q_s}$ and $\mathbf{q_g}$, defined by a set of $\mathbf{q} \in \mathcal{Q}_{free} \forall \mathbf{q}$.

However, when $\mathbf{q_s} \notin \mathrm{RM}$ and $\mathbf{q_g} \notin \mathrm{RM}$, which is the most common, the connection between $\mathbf{q_s}$ and $\mathbf{q_g}$ requires their connection to the roadmap. This connection is performed by the accessibility and departability proprieties (Choset et al., 2005, pp. 121-123).

Since $\mathbf{q_s} \in \mathcal{Q}_{free}$ and $\mathbf{q_s} \notin \mathrm{RM}$, the start point is inside a Voronoi region $\mathcal{F}_i \in \mathcal{Q}_{free}$ of an obstacle $\mathcal{QO}_i$. Being the Voronoi diagram a set of points, S_{ij}, that are equidistant to two neighbour obstacles $\mathcal{QO}_i$ and $\mathcal{QO}_j$, then, all points between $\mathbf{q_s}$ and the shortest point $\mathbf{q} \in \mathrm{RM}$ are inside $\mathcal{F}_i$. This means that moving $\mathbf{q}$ away from the closest obstacle can be achieved as $\mathbf{q_s'}$ and the accessibility can be ensured.

A similar approach, but in reverse direction, can be done to ensure the departability propriety. Thus, $\mathbf{q_g'}$ can be achieved by moving $\mathbf{q_g}$ in the direction to the GVD until reach a point $\mathbf{q} \in \mathrm{RM}$. All points between $\mathbf{q_g'}$ and $\mathbf{q_g}$ are inside the Voronoi region $\mathcal{F}_i \in \mathcal{Q}_{free}$ of the obstacle $\mathcal{QO}_i$, the closest to $\mathbf{q_g}$. Thus, it is possible to connect $\mathbf{q_g'}$ and $\mathbf{q_g}$ with a set of points $\mathbf{q} \in \mathcal{Q}_{free}$, ensuring the departability propriety.

5.3.4 Path planner using the Dijkstra algorithm

In the proposed navigation system architecture (Fig. 5.1), the path planning to the destination is a task executed before the robot starts the corresponding motion. The path is planned by the search for one that, using the roadmap the bulk of planning, minimizes the distance from $\mathbf{q_s}$ to $\mathbf{q_g}$. In the implemented approach, the shorter path is computed by the Dijkstra algorithm (Dijkstra, 1959). The Dijkstra algorithm is a classical approach to search for the short path between two points, being also used in GVD-based roadmaps (Wein et al., 2005; Bhattacharya and Gavrilova, 2008; Dong et al., 2010).

The graph used by the Dijkstra algorithm is defined by all points connection that compose the GVD, being the weights established by the distances between them. The connection of points distributed along a grid induces a graph where the nodes corresponds to cells and the edges is the connections between them (Choset et al., 2005, p. 523). The graph is completed by adding two nodes corresponding to $\mathbf{q_s}$ and $\mathbf{q_g}$. Each of these nodes are connected to another in the GVD, corresponding to $\mathbf{q'_s}$ and $\mathbf{q'_g}$, respectively. Thus, the node corresponding to $\mathbf{q'_s}$ is found by the euclidean distance minimization to $\mathbf{q_s}$. The definition of the node corresponding to $\mathbf{q'_s}$ is performed in a similar way. The weights assigned to these added edges are set to the corresponding euclidean distance.

The shorter path is composed by a set of edges connecting the start and goal nodes, wherein the sum of their lengths is the minimum. The Dijkstra algorithm begins by analysing the length of all the edges connected to the start node, and sets it as precedent to the next node, whose edge is the shorter. Then, the same methodology is applied to nodes at the next level, i.e., from the nodes that are connected to the start node. The edges lengths connected to each node are analysed and each one is defined as precedent to a next node, that is connected by the shortest edge. Once all precedence of the second level nodes are defined. The method is repeated for the next levels until the goal node is reached and its precedence is set.

At that time, once all precedences are defined from the goal until the start nodes, the shortest path, using of GVD, is defined by these nodes and the edges that connect them. Finally, the GVD vertices positions along the planned path are extracted, defining the set of waypoints that the robot must follow, and the goal position is added.

5.4 Local planning and control using navigation functions

In addition to the path planning problem, a mobile robot navigation system must be able to safely move it until reaches the destination. Since the path is composed by a set of waypoints extracted from a roadmap that maximizes distance to the existing obstacles in the environment, the robot pose control along these points ensures a collision free motion, when the workspace has no changes. Despite the experiences presented in this work does not yet consider the existence of other obstacles in the environment, this section presents an approach that will allow, in the future, to add this issue to the navigation system. Thus, in addition to the global path planning, a local planner is designed to control the robot motion, while it is moving to temporary goals along the path.

The APF is a classical obstacle avoiding algorithm, proposed by Khatib (1986), usually implemented in mobile robot navigation. This approach is based on the idea of magnetic fields applied to a (magnetic) body in a space. Thus, the trajectory is a point moving over an artificial potential force field, caused by attractive and repulsive fields. The attractive field is produced by the target (local goal) and the repulsive fields are caused by other obstacles that may exist round the robot.

NF is a type of APF, developed by Rimon and Koditschek (1992), to plan paths through potential field gradients, ensuring only one minimum located on the goal. This approach considers that the robot moves attracted by a field in the target, while a boundary of workspace and obstacles with circular shape act as repulsive fields (Choset et al., 2005, pp. 93-96). Thus, the NF is defined by a combination of three functions:

$$\varphi \triangleq \hat{\varphi} \circ \sigma \circ \sigma_d \tag{5.8}$$

where each one adds special characteristics to the NF. Thus:

- $\hat{\varphi} = \frac{\gamma}{\beta}$ is the APF function that combines the attractive (γ) and repulsive (β) functions;
- $\sigma(x) \triangleq \frac{x}{x+1}$ is a function that constrains the values of the potential function from the range $[0, \infty[$ to $[0, 1]$;
- $\sigma_d \triangleq x^{1/k}$ is a function that ensures only one minimum in the NF, located in the goal.

Thus, consider a mobile robot with radius r_r, located in a position $\mathbf{q} = [\ x \quad y\ ]^T$ and moving towards a desired position $\mathbf{q_d} = [\ x_d \quad y_d\ ]^T$, inside a bounded workspace [1]. A map

$\varphi : \digamma \rightarrow [0,1]$ is a NF if φ is:

- smooth on $\digamma$, i.e., at least the second partial derivative exist;
- polar and with a unique minimum at $\mathbf{q_d}$;
- admissible on $\digamma$, i.e., is uniformly maximal on the boundary of $\digamma$;
- a Morse function, i.e., the critical points are non-degenerative and a any small perturbation can avoid local minimums.

The NF can be described as follows:

$$\varphi = \frac{\gamma_d}{\left(\gamma_d{}^k + \beta\right)^{\frac{1}{k}}} \tag{5.9}$$

where:

- $\gamma_d = \|\mathbf{q} - \mathbf{q_d}\|^2$, is a function that represent the attractive potential field, computed from the distance towards the goal;
- β is a function that represent the repulsive potential field, defined by the environment boundary and the obstacles;
- k is an integer positive tuning coefficient.

In the classical NF, it is assumed that the robot is moving in a circular world, centered in $\mathbf{q_0} = \begin{bmatrix} x_0 & y_0 \end{bmatrix}^T$, with radius r_0, being the corresponding repulsive function given as follows:

$$\beta_0 = -\|\mathbf{q} - \mathbf{q_0}\|^2 + \rho_0^2 \tag{5.10}$$

where $\rho_0 = r_0 - r_r$.

Considering that the local world has a set of M circular obstacles with radius r_i, $i = 1, 2, \ldots, M$, located in the position $\mathbf{q_i} = \begin{bmatrix} x_i & y_i \end{bmatrix}^T$, the repulsive function corresponding to each obstacle is given as follows:

$$\beta_i = \|\mathbf{q} - \mathbf{q_i}\|^2 - \rho_i^2 \tag{5.11}$$

where $\rho_i = r_i + r_r$.

[1] The configuration and gradient vectors are arbitrarily referred as row or column. In line with the remaining systems of this work and because it is the convention most commonly used in the robotics community they are defined as column vectors. (Choset et al., 2005, p. 483)

Finally, the global repulsion function, β, is computed by combining all the repulsive functions as follows:

$$\beta = \prod_{i=0}^{M} \beta_i = \beta_0 \prod_{i=1}^{M} \beta_i \tag{5.12}$$

The mobile robot motion control using APF consider that there is a point $\mathbf{q}$ moving along the negative gradient of the potential function, i.e., $\dot{\mathbf{q}} = -\nabla\varphi$, with $\nabla\varphi = \begin{bmatrix} \frac{\partial\varphi}{\partial x} & \frac{\partial\varphi}{\partial x} \end{bmatrix}^T$. The NF gradient represented by the equation (5.9) is given by:

$$\nabla\varphi = \frac{\nabla\gamma_d \left({\gamma_d}^k + \beta\right)^{\frac{1}{k}} - \gamma_d \nabla\left(({\gamma_d}^k + \beta)^{\frac{1}{k}}\right)}{\left({\gamma_d}^k + \beta\right)^{\frac{2}{k}}} \tag{5.13}$$

where:

$$\nabla\gamma_d = 2(\mathbf{q} - \mathbf{q_d}) \tag{5.14}$$

$$\nabla\left(({\gamma_d}^k + \beta)^{\frac{1}{k}}\right) = \frac{1}{k}\left(({\gamma_d}^k + \beta)^{\frac{1}{k}-1}\right)\left(k\nabla(\gamma_d){\gamma_d}^{k-1} + \nabla\beta\right) \tag{5.15}$$

The repulsion field gradient is given as following:

$$\nabla\beta = \sum_{i=0}^{M}\left(\nabla\beta_i \prod_{j=0, j\neq i}^{M} \beta_j\right) \tag{5.16}$$

where:

$$\nabla\beta_i = \begin{cases} -2(\mathbf{q} - \mathbf{q_i}) & ,\text{if } i = 0 \\ 2(\mathbf{q} - \mathbf{q_i}) & ,\text{if } i > 0 \end{cases} \tag{5.17}$$

5.4.1 Dipolar navigation functions with non-holonomic constrains

The control using a classical NF is made by moving the mobile robot through its negative gradient with an holonomic motion, i.e., without consider any kinematic constraints. However, when a mobile robot with differential drive kinematics is controlled by a NF, the corresponding non-holonomic constrains must be taken into account.

Lets consider a mobile robot with differential drive kinematics, located at a position $\mathbf{q}$ with an attitude ψ, represented by the configuration state $\mathbf{p} = [\ \mathbf{q}^T \quad \psi\ ]^T$. The corresponding kinematic model is described by the follow equation:

$$\dot{\mathbf{p}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & 0 \\ \sin(\psi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \tag{5.18}$$

Aiming this technique implementation in mobile robots with differential drive kinematics, Tanner and Kyriakopoulos (2000) and Loizu et al. (2004) and Loizou and Kyriakopoulos (2008) propose the use of NV with dipolar potential fields to provide suitable trajectories for those constraints. Thus, to create a DNF for the navigation of one differential drive mobile robot towards a desired pose $\mathbf{p_d} = [\ \mathbf{q_d}^T \quad \psi_d\ ]^T$, the equation (5.9) must be modified as follows:

$$\varphi = \frac{\gamma_d}{\left(\gamma_d{}^k + H_{nh}\beta\right)^{\frac{1}{k}}} \tag{5.19}$$

where the attractive function is redefined to $\gamma_d = ||\mathbf{p} - \mathbf{p_d}||^2$ and H_{nh} assume a pseudo-obstacle form that drives all trajectories to the desired pose, being defined by the follow equation:

$$H_{nh} = \varepsilon_{nh} + \eta_{nh} \tag{5.20}$$

with ε_{nh} a positive coefficient and,

$$\eta_{nh} = \left\|(\mathbf{q} - \mathbf{q_d})^{\mathbf{T}}\mathbf{n_d}\right\|^2 \tag{5.21}$$

$$\mathbf{n_d} = \left[\ \cos(\psi_d) \quad \sin(\psi_d)\ \right]^T \tag{5.22}$$

Attending to this modification, the DNF gradient represented by the equation (5.19) is given as follows:

$$\nabla\varphi = \frac{\nabla\gamma_d\left(\gamma_d{}^k + H_{nh}\beta\right)^{\frac{1}{k}} - \gamma_d\nabla\left((\gamma_d{}^k + H_{nh}\beta)^{\frac{1}{k}}\right)}{\left(\gamma_d{}^k + H_{nh}\beta\right)^{\frac{2}{k}}} \tag{5.23}$$

where

$$\nabla\left((\gamma_d{}^k + H_{nh}\beta)^{\frac{1}{k}}\right) = \frac{1}{k}\left((\gamma_d{}^k + H_{nh}\beta)^{\frac{1}{k}-1}\right)\left(k\nabla(\gamma_d)\gamma_d{}^{k-1} + H_{nh}\nabla\beta + \beta\nabla H_{nh}\right) \tag{5.24}$$

$$\nabla\gamma_d = 2(\mathbf{p} - \mathbf{p_d}) \tag{5.25}$$

$$\nabla H_{nh} = \begin{bmatrix} \frac{\partial H_{nh}}{\partial x} & \frac{\partial H_{nh}}{\partial y} & \frac{\partial H_{nh}}{\partial \psi} \end{bmatrix}^T = \begin{bmatrix} (2(x - x_d)\cos^2(\psi_d)) \\ (2(y - y_d)\sin^2(\psi_d)) \\ -\sin(2\psi_d)\left((x - x_d)^2 - (y - y_d)^2\right) \end{bmatrix} \tag{5.26}$$

Once computed the partial derivative of all function included in DNF, its gradient is defined by the corresponding partial derivatives in the cartesian axis, as follows:

$$\nabla\varphi = \begin{bmatrix} \frac{\partial\varphi}{\partial x} & \frac{\partial\varphi}{\partial x} & \frac{\partial\varphi}{\partial\psi} \end{bmatrix}^T \tag{5.27}$$

5.4.2 Optimal NF parameter computation to avoid local minimum

As described above, the NF allows the definition of a potential field with no local minimum. According with the APF methodology, the motion follows the NF negative gradient. To ensure the absence of local minimum, the attractive gradient $\nabla\gamma_d$ must dominate over the repulsive gradient $\nabla\beta$ for all $\mathbf{q}$ inside the workspace.

Koditschek and Rimon (1990) analysed that increasing the parameter k, it is possible to eliminate critical points inside the workspace. Following this approach, there is a minimum value for k, from which the NF negative gradient tends to γ_d, from any $\mathbf{q}$.

Thus, lets consider the NF gradiente depicted in equation (5.13). This function has critical points for $\nabla\varphi = 0$, which can be obtained as follow:

$$\nabla\gamma_d\left({\gamma_d}^k + \beta\right)^{\frac{1}{k}} = \gamma_d\nabla\left(({\gamma_d}^k + \beta)^{\frac{1}{k}}\right) \tag{5.28}$$

Simplifying the equation (5.28), the critical condition is simplified as follow:

$$k\beta\nabla\gamma_d = \gamma_d\nabla\beta \tag{5.29}$$

Taking the magnitude for both sides, it is possible to obtain the follow equation:

$$2k\beta = \sqrt{\gamma_d}||\nabla\beta|| \tag{5.30}$$

Thus, the sufficient condition for the equation (5.30) ensures no critical points is given as follow:

$$k > \frac{\sqrt{\gamma_d}||\nabla\beta||}{2\beta} \tag{5.31}$$

for all $\mathbf{q} \in \mathcal{Q}_{free}$.

According with Loizou and Kyriakopoulos (2003), when non-holonomic constrains are consider, the repulsive function is replaced by $S = H_{nh}\beta$, leading to a sufficient condition to avoid critical points given as follow:

$$k > \frac{\sqrt{\gamma_d}\|\nabla S\|}{2S} \tag{5.32}$$

for all $\mathbf{q} \in \mathcal{Q}_{free}$.

5.4.3 Non-holonomic control using dipolar navigation functions

Aiming to drive a mobile robot with non-holonomic constraints using a DNF, a motion controller to take into account those restrictions must be applied.

To analyse the control stability, lets consider a Lyapunov function candidate defined by the DNF detailed above:

$$V(\mathbf{p}) = \varphi(\mathbf{p}) \tag{5.33}$$

the corresponding Lyapunov function derivative is given as follow:

$$\dot{V}(\mathbf{p}) = \frac{\partial V}{\partial t} + \nabla V(\mathbf{p})\dot{\mathbf{p}} = \nabla V(\mathbf{p})\dot{\mathbf{p}} \tag{5.34}$$

When the non-holonomic constraints are considered, replacing the equation (5.18) in equation (5.34), the Lyapunov function derivative become as follows:

$$\dot{V}(\mathbf{p}) = u\left(\frac{\partial\varphi}{\partial x}\cos\psi + \frac{\partial\varphi}{\partial y}\sin\psi\right) + \omega\frac{\partial\varphi}{\partial\psi} \tag{5.35}$$

Loizu et al. (2004) propose the following control laws to stabilize a mobile robot with differential drive kinematics constraints using a DNF:

$$u = -K_u \mathrm{sgn}\left(\frac{\partial\varphi}{\partial x}\cos(\psi) + \frac{\partial\varphi}{\partial y}\sin(\psi)\right)\left(\left(\frac{\partial\varphi}{\partial x}^2 + \frac{\partial\varphi}{\partial y}^2\right) + \|\mathbf{q} - \mathbf{q_d}\|^2\right) \tag{5.36}$$

$$\omega = \begin{cases} K_\omega(\psi_{nh} - \psi) & , \triangle < 0 \\ -K_\omega \frac{\partial\varphi}{\partial\psi} & , \triangle \geq 0 \end{cases} \tag{5.37}$$

where K_u and K_ω are positive constants and:

$$\triangle = K_\omega \frac{\partial\varphi}{\partial\psi}(\psi_{nh} - \psi) - K_u\left(\left(\frac{\partial\varphi}{\partial x}^2 + \frac{\partial\varphi}{\partial y}^2\right) + \|\mathbf{q} - \mathbf{q_d}\|^2\right)\left|\frac{\partial\varphi}{\partial x}\cos(\psi) + \frac{\partial\varphi}{\partial y}\sin(\psi)\right| \tag{5.38}$$

$$\psi_{nh} = \mathrm{atan2}\left(\frac{\partial\varphi}{\partial y}\mathrm{sgn}\left((\mathbf{q} - \mathbf{q_d})^T\mathbf{n_d}\right), \frac{\partial\varphi}{\partial x}\mathrm{sgn}\left((\mathbf{q} - \mathbf{q_d})^T\mathbf{n_d}\right)\right) \tag{5.39}$$

To analyse the control laws stability, the equations (5.36)–(5.37) are replaced in the equation (5.35), according with the level of $\triangle$.

Thus, when $\triangle < 0$, the equation (5.35) becomes as follow:

$$\begin{aligned}\dot{V}(\mathbf{p}) =& -K_u \operatorname{sgn}\left(\frac{\partial\varphi}{\partial x}\cos(\psi)+\frac{\partial\varphi}{\partial y}\sin(\psi)\right)\left(\frac{\partial\varphi}{\partial x}\cos(\psi)+\frac{\partial\varphi}{\partial y}\sin(\psi)\right)\\ &\left(\left(\frac{\partial\varphi}{\partial x}^2+\frac{\partial\varphi}{\partial y}^2\right)+\|\mathbf{q}-\mathbf{q_d}\|^2\right)+K_\omega\frac{\partial\varphi}{\partial\psi}(\psi_{nh}-\psi)\\ =& -K_u\left|\frac{\partial\varphi}{\partial x}\cos(\psi)+\frac{\partial\varphi}{\partial y}\sin(\psi)\right|\left(\left(\frac{\partial\varphi}{\partial x}^2+\frac{\partial\varphi}{\partial y}^2\right)+\|\mathbf{q}-\mathbf{q_d}\|^2\right)+K_\omega\frac{\partial\varphi}{\partial\psi}(\psi_{nh}-\psi)=\\ =&\triangle<0\end{aligned}\tag{5.40}$$

On other hand, when $\triangle \geq 0$, the equation (5.35) becomes as follow:

$$\begin{aligned}\dot{V}(\mathbf{p}) =& -K_u \operatorname{sgn}\left(\frac{\partial\varphi}{\partial x}\cos(\psi)+\frac{\partial\varphi}{\partial y}\sin(\psi)\right)\left(\frac{\partial\varphi}{\partial x}\cos(\psi)+\frac{\partial\varphi}{\partial y}\sin(\psi)\right)\\ &\left(\left(\frac{\partial\varphi}{\partial x}^2+\frac{\partial\varphi}{\partial y}^2\right)+\|\mathbf{q}-\mathbf{q_d}\|^2\right)-K_\omega\frac{\partial\varphi}{\partial\psi}^2=\\ =& -K_u\left|\frac{\partial\varphi}{\partial x}\cos(\psi)+\frac{\partial\varphi}{\partial y}\sin(\psi)\right|\left(\left(\frac{\partial\varphi}{\partial x}^2+\frac{\partial\varphi}{\partial y}^2\right)+\|\mathbf{q}-\mathbf{q_d}\|^2\right)-K_\omega\frac{\partial\varphi}{\partial\psi}^2\leq 0\end{aligned}\tag{5.41}$$

Analysing the equations (5.40)–(5.41) it is possible to conclude that the control laws application, equations (5.36)– (5.37), leads to $\dot{V}(\mathbf{p}) \leq 0$, attracting the mobile robot to the desired pose.

5.5 Experimental results

5.5.1 Environment roadmap

To validate the proposed navigation system some experiments were performed. Thus, set of 1571 depth images from the ceiling was captured along a grid with 0.3 m, placing the robot in different positions (keeping the same attitude), covering all free space, and the eigenspace was created (see section 4.6 for more details). As detailed in section 5.2, an occupancy grid was built using the information about the existence of depth images captured from the ceiling. The cells where images were captured are defined as free space and the remaining as occupied. Aiming to improve the GVD accurancy, the occupancy grid was resized with a scale 1 : 3. The GVD roadmap was computed using the wave front algorithm, (section 5.3.2), and the GVD vertices (points intercepted by more than two GVD edges) were defined as the set of possible waypoints. These points are stored to be used by the DNF-based controller, during the robot motion, as the local circular worlds center and the corresponding goal positions.

Figure 5.2 shows the final distance grid and the corresponding GVD-based roadmap. In this figure, the grayscale represents the distance to the obstacles (black area), while the blue lines and circles represents the edges, computed with the front wave algorithm, and the vertices (the set of possible waypoints), respectively. The GVD-based roadmap is depicted in a metric map in Fig. 5.3.

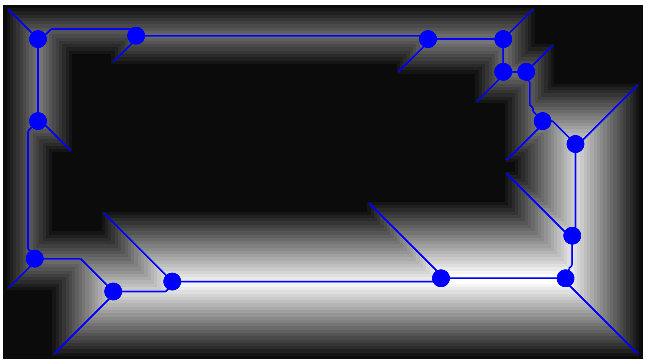

Figure 5.2: GVD-based roadmap over a distance grid to the obstacles (black: occupied space; blue lines: edges; blue points: vertices; grayscale: distance to obstacles, increasing with brightness).

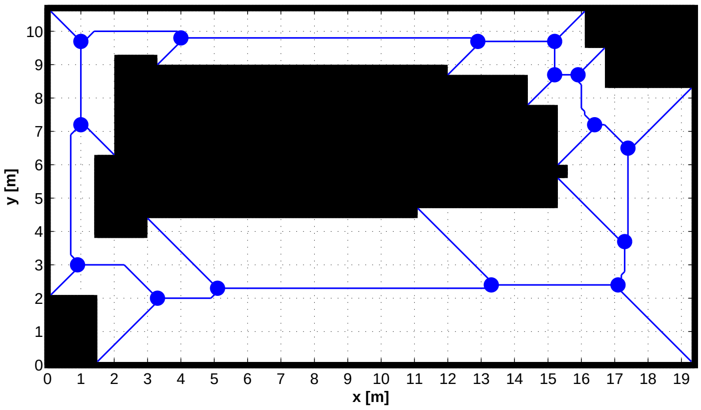

Figure 5.3: Occupancy grid with GVD-based roadmap (black: occupied space; blank: free space; blue lines: edges; blue points: vertices (possible waypoints).

Having defined the roadmap, the next step is to plan a path for the robot motion. Thus, its actual position ($\mathbf{q_s}$) and the destination ($\mathbf{q_g}$) are connected to the corresponding closest points in the roadmap, and the shortest path on the GVD is computed through the Diskstra's algorithm (section 5.3.4). Finally, the waypoints' set that will be used by the DNF-based controller as temporary goals is selected according with the planned path and the destination point ($\mathbf{q_g}$) is added.

Attending of plan execution, the robot movement results from a DNF defined on a local workspace, centered on the waypoint, and using it as temporary goal position. The DNF is designed to drive the mobile robot toward its center, with a desired attitude in the direction to the next waypoint. Thus, the robot is moved attracted by the temporary goal while is repelled by the circular world boundary around. This potential field guides the robot to follow a path close to the GVD-based roadmap.

When the robot gets close enough to the temporary goal, the goal is changed to the next waypoint and a new DNF is defined to drive it in direction to the next one. This process is repeated until the mobile robot goes through all waypoints and reaches its destination ($\mathbf{q_g}$).

5.5.2 Self-localization system integration

To validate the proposed navigation system in a real environment, some experiments were performed in an industrial-like indoor environment, containing several systems, such as, electrical and HVAC systems, lights, technical gutters, among others (see Fig. 4.28). To carried out the experiments, a mobile robot platform was equipped with a Kinect device, looking upward to the ceiling, a rate gyro, a compass and encoders coupled to the motors (Fig. 5.4).

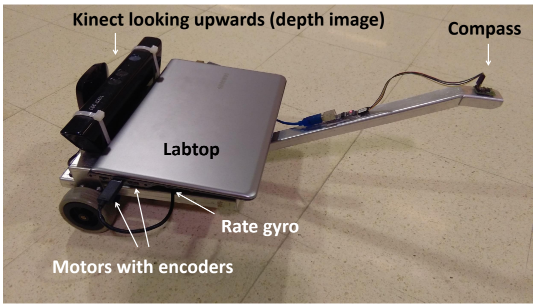

Figure 5.4: Mobile platform equipped with kinect device and motion sensors.

The PCA-based self-localization system is integrated with the DNF-based controller in a closed loop, as depicted in Fig. 5.1. The experiments was performed with the PCA-based positioning system using depth images (see section 4.3 for more details). Thus, during the motion, the depth image captured from the ceiling is rotated and decomposed into the PCA eigenspace to obtain the robot position. Since the robot is moving in an environment with repetitive elements, the searching by similar images in the PCA eigenspace is performed considering a neighborhood radius, $\delta = 3$ m, around the last robot estimated position (see 4.6 for details). The results obtained from the extended PCA algorithm for corrupted depth images are merged with the LPV, through the time-varying KF designed to estimate the mobile robot global position (section 3.4.4). The robot attitude is estimated by the KF designed to merge the signals provided by three sensors: rate gyro, compass and encoders' odometry (section 2.5.2).

To move the mobile robot toward the goals (temporary or final), the position and attitude estimated by the self-localization system are used in a DNF-based controller, that drives it through an APF. Thus, the DNF gradients are computed for the estimated pose, $\hat{\mathbf{p}} = [\ \hat{x} \quad \hat{y} \quad \hat{\psi}\ ]^T$, and the new mobile robot linear and angular velocity are obtained from the corresponding control laws defined in the equations (5.36)– (5.37). Attending that the LPV model is designed for trajectories with constant velocity, a states replacement are made allowing the position estimation with different velocities. Thus, the linear velocity sent to the mobile robot is projected onto the cartesian axes, considering the estimated attitude and the corresponding states used in the LPV model are replaced by these new value, i.e., $\dot{x} = u\cos(\hat{\psi})$ and $\dot{y} = u\sin(\hat{\psi})$.

5.5.3 Path following via waypoints

In this experiment, the robot departs from the position $\mathbf{q_s} = \begin{bmatrix} 6.0 & 3.3 \end{bmatrix}^T$ m and attitude $\psi_s = 0$ ° toward a goal $\mathbf{q_g} = \begin{bmatrix} 12.0 & 9.6 \end{bmatrix}^T$ m and attitude $\psi_g = 180$ °. To carry out this motion, the waypoint set defined in the planning step is detailed in Table 5.1.

The experiment is performed considering that the mobile robot is commanded by the controller with the gains tunned to $K_u = 0.3$ and $K_\omega = 0.5$. It was considered that, along the motion, the temporary goal (waypoint) changes to the next when the corresponding

Table 5.1: Goals defined in the path planning step.

	x position [m]	**y position [m]**
Temporary goal no 1	13.3	2.4
Temporary goal no 2	17.1	2.4
Temporary goal no 3	17.3	3.7
Temporary goal no 4	17.4	6.5
Temporary goal no 5	16.4	7.2
Temporary goal no 6	15.9	8.7
Temporary goal no 7	15.2	8.7
Temporary goal no 8	15.2	9.7
Temporary goal no 9	12.9	9.7
Final goal ($\mathbf{q_g}$)	12.0	9.6

distance error is less than 1 m. The robot motion stops its mission when the distance to the goal position is less than 0.1 m. During the motion, the linear and angular velocities are saturated with $u_{max} = \pm 0.15$ m/s and $\omega_{max} = \pm 1$ rad/s, respectively.

To observe the real path described by the robot while it was moving, some points were marked on the floor. After the experiment, these position points were measured in the cartesian axes to obtain the ground truth.

Figure 5.5 shows the robot trajectory obtained in this experiment. As it is possible to observe, the estimated trajectory is close to the ground truth points, that represents the real mobile robot trajectory. Moreover, it is also possible to observe that the traveled path is made toward the waypoints and, each time that the robot gets close to this temporary goal, it changes the direction to the next. Analysing the Fig. 5.5, it is possible to conclude that this approach succeeds in driving the robot closely to the path of largest clearance with a smooth trajectory.

The results provided by the PCA-based positioning system along the experiment are depicted in Fig. 5.6. Analysing these results, it is possible to observe that, sometimes, the PCA-based positioning system provides positions measurements with some error. This happens due to the distance associated with the database grid because the PCA algorithm sometimes finds similar images in distant locations. Although the variations in the positioning obtained from the PCA algorithm, the KF is able to decrease these errors and smooth the robot estimated positions, that are provided by the self-localization system.

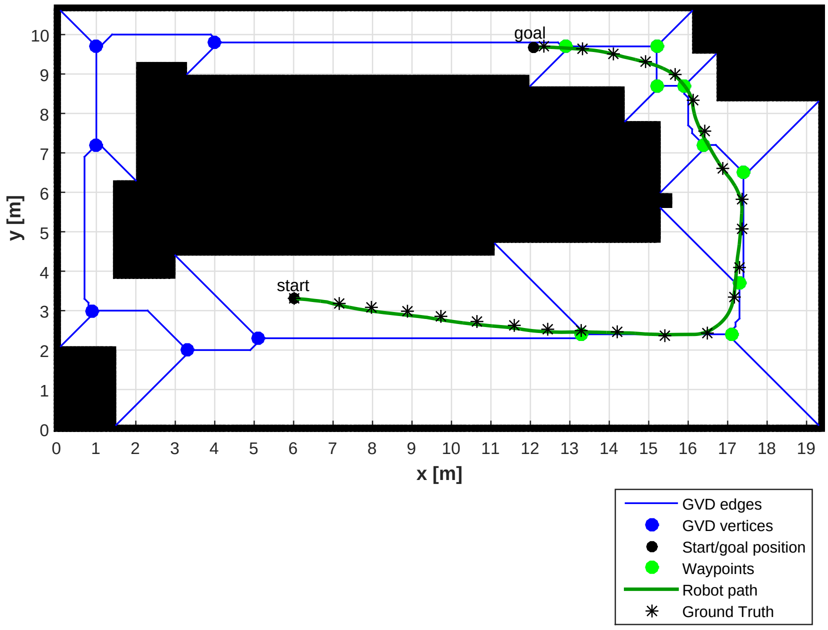

Figure 5.5: Map with path following via waypoints - experimental results.

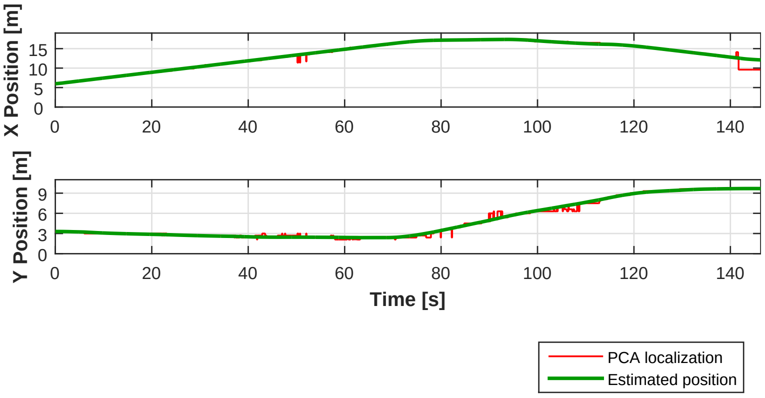

Figure 5.6: Mobile robot position along time - experimental results.

5.5.4 Path following via waypoints with wrong initial position

Another experiment was performed to move the mobile robot toward the same goal position, $\mathbf{q_g} = \begin{bmatrix} 12.0 & 9.6 \end{bmatrix}^T$ m, but considering that the self-localization estimator is initialize with a wrong position. Thus, despite the real mobile robot departs from the position $\mathbf{q_s} = \begin{bmatrix} 6.0 & 3.3 \end{bmatrix}^T$ m and attitude $\psi_s = 0$ °, the self-localization is initialized with the position states $\hat{x}(0) = 8.0$ m and $\hat{y}(0) = 1.5$ m.

Analysing the results depicted in Fig. 5.7 – 5.8, it is possible to observe that, despite the wrong estimated position, the self-localization system converge to the real trajectory. This means that, once stabilized the position estimation, the robot may be correctly driven to its destination. In addition, it is possible to observe in Fig. 5.8 that the self-localization system stabilize in approximately 5 seconds.

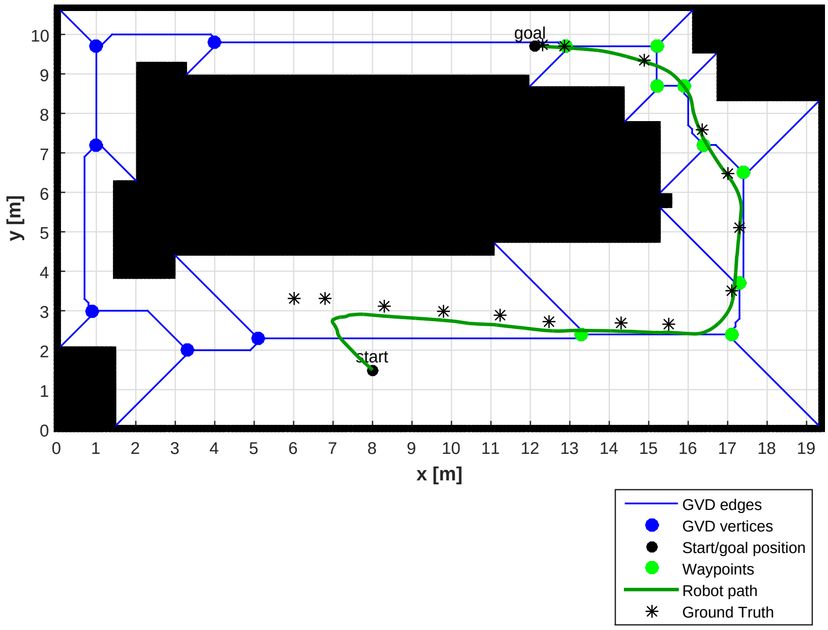

Figure 5.7: Map with wrong initial position - experimental results.

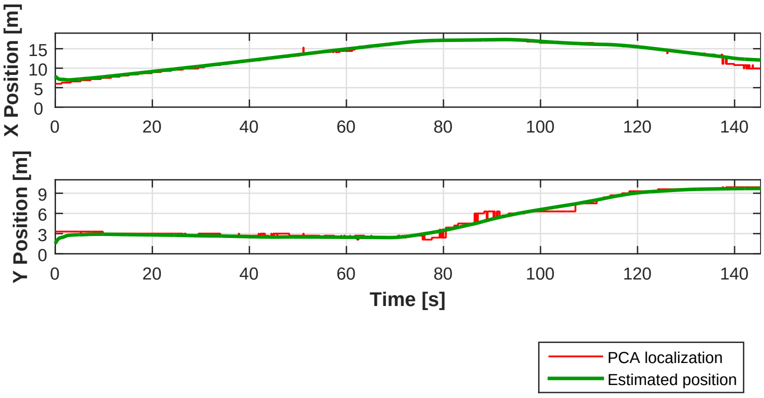

Figure 5.8: Mobile robot position along time with wrong initial condition - experimental results.

5.5.5 Path following via waypoints with one intermediate goal

Another experiment was performed, considering a navigation toward a destination, but adding one intermediate goal along the path that most be visited. In this experiment, the mobile robot departs from the position $\mathbf{q_s} = \begin{bmatrix} 6.0 & 3.3 \end{bmatrix}^T$ m and attitude $\psi_s = 0$ ° toward a goal $\mathbf{q_g} = \begin{bmatrix}1.0 & 9.7\end{bmatrix}^T$ m and attitude $\psi_g = 180$ °, passing by one intermediate goal in the positions $\mathbf{q_{g_1}} = \begin{bmatrix}12.9 & 9.7\end{bmatrix}^T$ m.

Since the path is composed by one intermediate ($\mathbf{q_{g_1}}$) and a final goal ($\mathbf{q_g}$), the planning step is performed two times along the experiment. Thus, a first planning is computed to define the waypoints to drive the mobile robot to $\mathbf{q_{g_1}}$. Then, the DNF-based controller moves the robot toward this intermediate goal, along the first waypoints' set. When it is close to the first intermediate goal, the second planning is computed from the current estimated position to the final goal $\mathbf{q_g}$. Table 5.2 shows the set of temporary goals, selected in both path planning steps, aiming to carrying out this mission.

Table 5.2: Goals defined in the path planning step with one intermediate goals.

	x position [m]	**y position [m]**
Temporary goal no 1	13.3	2.4
Temporary goal no 2	17.1	2.4
Temporary goal no 3	17.3	3.7
Temporary goal no 4	17.4	6.5
Temporary goal no 5	16.4	7.2
Temporary goal no 6	15.9	8.7
Temporary goal no 7	15.2	8.7
Temporary goal no 8	15.2	9.7
Temporary goal no 9 ($\mathbf{q_{g_1}}$)	12.9	9.7
Temporary goal no 10	4.0	9.8
Final goal ($\mathbf{q_g}$)	1.0	9.7

In this experiment the controller gains was tunned to the same values, $K_u = 0.3$ and $K_\omega = 0.5$. It was considered that, along the motion, the temporary goal (waypoint) changes to the next when the corresponding distance error is less than 1 m, and robot motion stops when the distance to the final goal position is less than 0.3 m.

Figure 5.9 shows the mobile robot trajectory described in this experiment. Analysing these results, it is possible to observe that the estimated trajectory is also close to the ground truth. Looking to the top of the figure, it is possible to see a slight difference between the ground truth points and the estimated trajectory. This happens because the robot moves under a poor ceiling, i.e., with very few information. Figure 5.10 shows the positions provided by the PCA algorithm along time. Analysing these results it is possible to see that, at the end of the trajectory (after about 140 *s*), the positions provided by the PCA algorithm have large variations. This shows what can happen when the robot navigates under a ceiling with very few information, and even those have repeatability. However, the KF can smooth these variations, estimating a mobile robot position close to the ground truth.

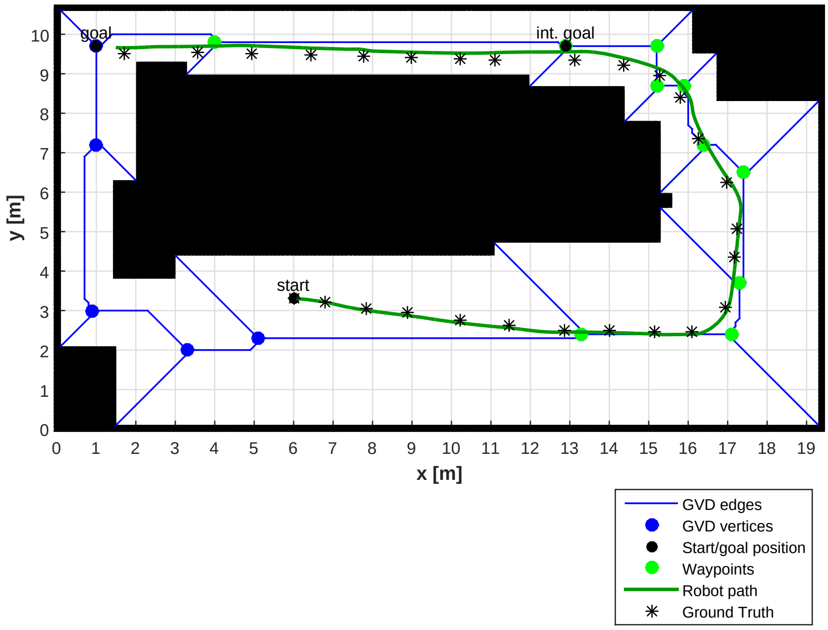

Figure 5.9: Map with path following via waypoints with one intermediate goal - experimental results.

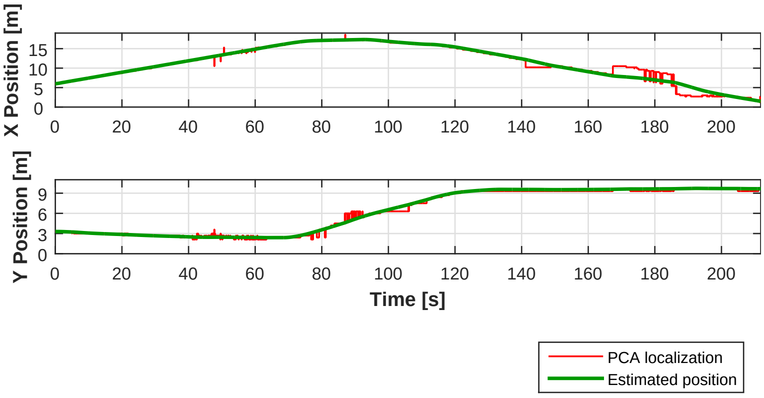

Figure 5.10: Estimated position along time with one intermediate goal - experimental results.

5.5.6 Path following via waypoints in a closed loop

A last experimental test was performed considering the robot motion along a large trajectory, in a closed loop. In this experiment, the robot starts from the initial position $\mathbf{q_s} = \begin{bmatrix} 1.8 & 10.0 \end{bmatrix}^T$ m and attitude $\psi_s = 180$ ° toward to a final goal $\mathbf{q_g} = \begin{bmatrix} 1.0 & 9.7 \end{bmatrix}^T$ m and with the same attitude ($\psi_g = 180$ °). Instead these positions are close, to carry out this mission, two intermediate goals are defined in the positions $\mathbf{q_{g_1}} = \begin{bmatrix} 6.0 & 2.4 \end{bmatrix}^T$ m and $\mathbf{q_{g_2}} = \begin{bmatrix} 12.9 & 9.7 \end{bmatrix}^T$ m, respectively. These intermediate goals will do the robot describe a large lap, in a loop closed.

Since the path is composed by two intermediate goal ($\mathbf{q_{g_1}}$ and $\mathbf{q_{g_2}}$) and a final goal ($\mathbf{q_g}$), the planning step is performed three times along the experiment. Thus, a first planning is computed to define the waypoints to drive the mobile robot to $\mathbf{q_{g_1}}$ and the DNF-based controller moves the robot toward this intermediate goal. When it is close to the first intermediate goal, another planning is computed from the mobile robot current pose to the second intermediate goal $\mathbf{q_{g_2}}$. The process repeats until the mobile robot reaches the final position $\mathbf{q_g}$. Table 5.3 shows the set of temporary goals, defined in all planning steps, in order to carry out this mission.

Table 5.3: Goals defined in the path planning step to describe a large trajectory.

	x position [m]	y position [m]
Temporary goal no 1	1.0	9.7
Temporary goal no 2	1.0	7.2
Temporary goal no 3	0.9	3.0
Temporary goal no 4	3.3	2.0
Temporary goal no 5	5.1	2.3
Temporary goal no 6 ($\mathbf{q_{g_1}}$)	6.0	2.4
Temporary goal no 7	13.3	2.4
Temporary goal no 8	17.1	2.4
Temporary goal no 9	17.3	3.7
Temporary goal no 10	17.4	6.5
Temporary goal no 11	16.4	7.2
Temporary goal no 12	15.2	8.7
Temporary goal no 13	15.2	9.7
Temporary goal no 14 ($\mathbf{q_{g_2}}$)	12.9	9.7
Temporary goal no 15	4.0	9.8
Final goal ($\mathbf{q_g}$)	1.0	9.7

Figure 5.11 shows the mobile robot trajectory described along the experiment. As it is possible to observe, despite the start and the final goal position are close, the mobile robot describes a long trajectory, passing by the two intermediate goals. Along the experiment the robot position estimated by the self-localization system is close to the ground truth points and the robot reaches the final goal, after describing a large trajectory with about 45 *m*.

Figure 5.12 shows the robot position computed by the PCA algorithm and the estimated by the self-localization system along time. Analysing these results it is possible to observe that the PCA-based positioning system provide accurate results where the ceiling has rich information (up to about 220 s), while where the ceiling has very few information (after about 220 s), the obtained position results presents with greater error. However, even in this case the KF is able to smooth the trajectory, using the information provided from the motion sensors, improving thus the estimation results.

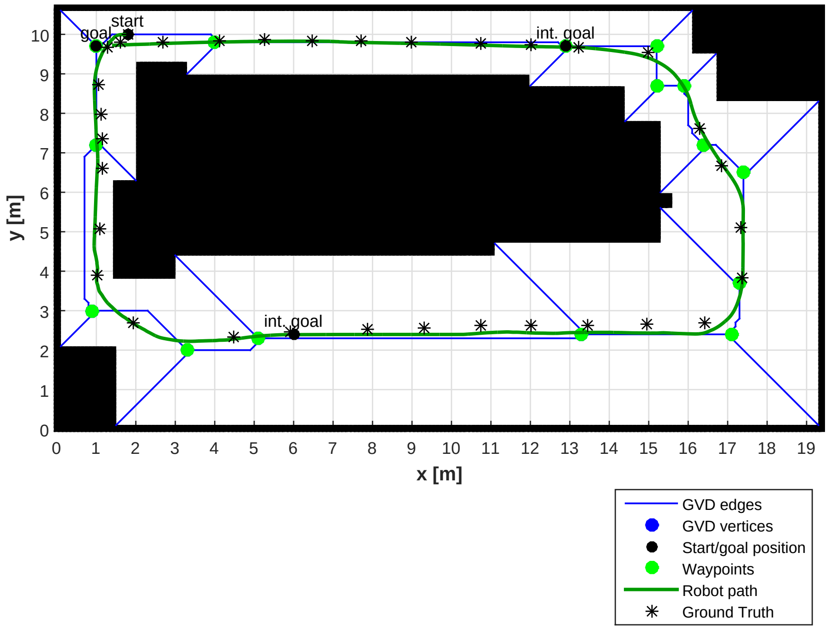

Figure 5.11: Map with path following along a large trajectory with two intermediate goals - experimental results.

Analysing the results obtained with this experiment, it is possible to conclude that the integration of the PCA-based self-localization system in a navigation system allows the mobile robot motion along a large trajectory in a closed loop, providing estimates close to the ground truth.

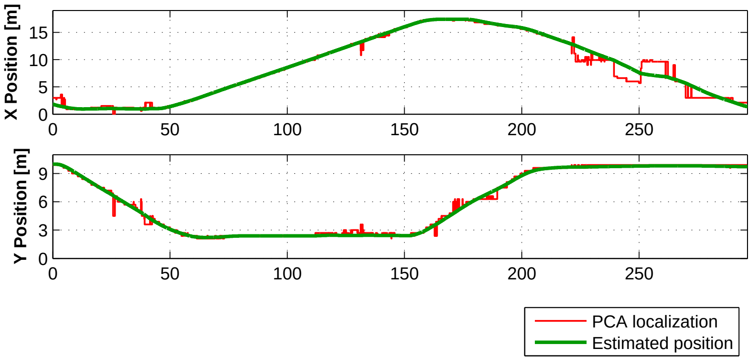

Figure 5.12: Estimated position along time in a large trajectory - experimental results.

Finally, this experiment was repeated so that the robot describes two laps in a closed loop, i.e., a path with approximately 90 m. Thus, the same sequence of planning and path following control was performed three times, aiming to describe the first lap. Then, the same planning and path following control steps was repeated, from the current robot estimated pose, in order to describe the second lap.

Figure 5.13 shows the map with the estimated trajectory described by the robot along this experiment, and the Fig. 5.14 depicts the corresponding position along time. Analysing these results it is possible to observe that the robot describes similar trajectories in both laps. Moreover, looking to Fig. 5.14, it is possible to see that in the results provided by the PCA-based positioning system up to about 300 *s* are similar to those achieved after this time. This mean that, when the robot returns to the same place, the PCA algorithm provides similar position results.

In addition, it is also possible to observe that the estimates in the top of Fig. 5.13 is less accurate than presented in the rest of the trajectory. As preciously mentioned, these estimates are obtained in an area where ceiling has very few information and even those with scenery repeatability, leading to the PCA algorithm to provide less precise positions. These results correspond to the instants 220 – 300 s and after 520 *s* (approximately) depicted in Fig. 5.14. However, as can be observed in Fig. 5.13-5.14, even that the position provided by the PCA algorithm may be less accurate in areas where the ceiling has very few information, when the robot returns to an area with a rich ceiling, the PCA-based positioning system returns to provide accurate results, improving the estimates obtained by the self-localization

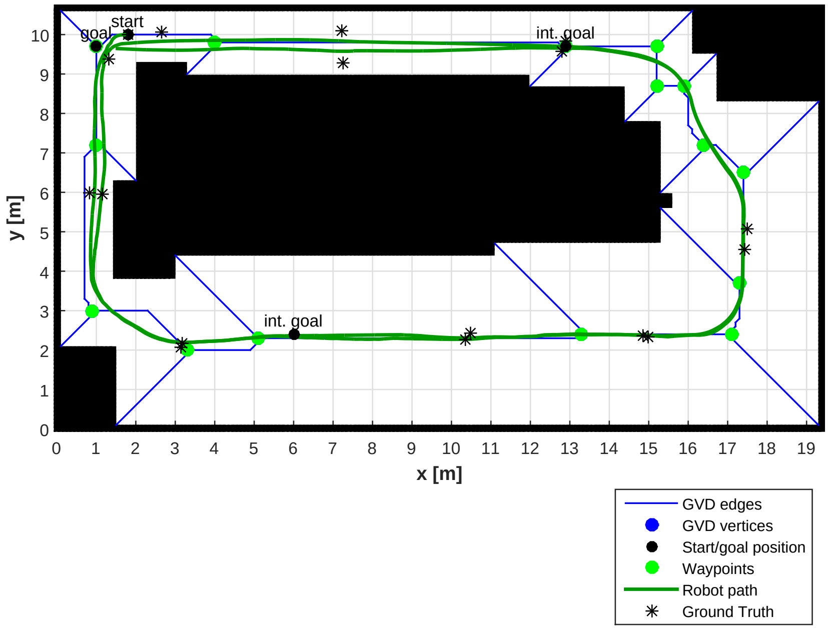

Figure 5.13: Map with path following along a large trajectory with two laps in a closed loop - experimental results.

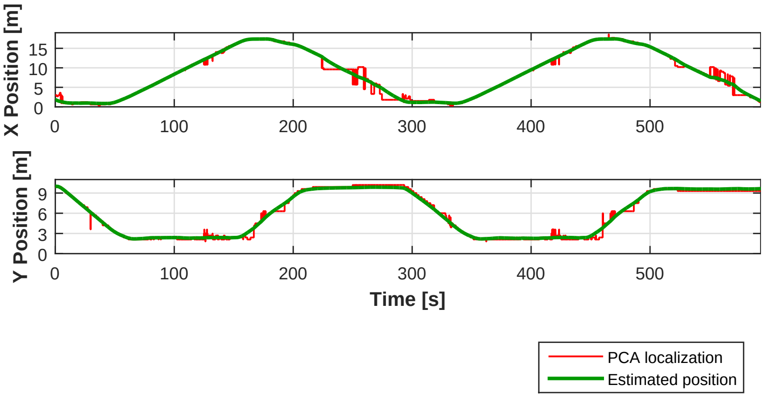

Figure 5.14: Estimated position along time in a large trajectory with two laps in closed loop - experimental results.

system. This means that, the positioning system provides similar results when the robot returns to the same place and captures depth images that can be distinguished from the rest.

Finally, it should be noted that, after a navigation with approximately 90 m, in which the final trajectory is traveled in an area with few ceiling information, when the robot stops close to the final goal (at the experiment finish) its estimated position has an error of e_x = 0.16 m and e_y = 0.27 m, relating to the real measured. This means that the robot reaches the destination with an error smaller than the ceiling mapping grid size, i.e., smaller than the position resolution provided by the PCA algorithm (0.3 m). This is achieved even finishing each lap with bad conditions, such as, capture ceiling depth images with few information and repeatability.

5.6 Conclusions

This chapter presented a navigation system for mobile robots using a PCA-based self-localization system. In the proposed approach, a roadmap based on GVD, a global path planner and a DNF was combined to define a set of waypoints and control the mobile robot motion toward to destinations positions.

Thus, the information about the captured images for the PCA database was used to define an occupancy grid that represents the workspace. Such grid was useful to build a roadmap based on a GVD, from which a set of waypoints to guide the robot can be obtained. The possible waypoints corresponds to the GVD vertices, i.e., points on the GVD that are intercepted by more than two edges. A path planner using Dijkstra algorithm was implemented to search for the shortest path between the robot position and its destination, using the GVD-based roadmap. Then, a set of waypoints was selected, aiming to be used as temporary goals. The DNF implementation along the selected waypoints in the path planning step allows to control the motion of a mobile robot with differential drive kinematics close to the GVD-based roadmap.

In order to analyse the proposed PCA-based navigation system some experiments were performed in a real environment, using a mobile robot motion equipped with on-board sensors. The experimental validation was performed considering different conditions: i) navigation via waypoints between two locations in the environment; ii) navigation with wrong initial estimates; iii) navigation along a large mission with one intermediate goal that must to be visited along the mission; iv) navigation along a large trajectory, in a closed loop, with two intermediate goals.

The experimental results showed that the mobile robot reaches the destination position. Moreover, during the motion, the self-localization system provides estimates close to the real trajectory described by the mobile robot. Even in case where the estimator starts from a wrong position, the self-localization system stabilizes in few seconds, allowing a good navigation toward its destination, close to the ground truth points.

Moreover, experimental results shows that the mobile robot can navigate along large trajectories, localizing itself only with the information provided from motion sensors installed on-board and the comparison of ceiling depth images with a database previous collected, through the PCA algorithm. This approach allows the implementation of a navigation system where the robot can know that it returned to the same place, without the need to recognize specific features from the environment that was previous viewed.

6

Mobile robots implementations using PCA-based localization

6.1 Introduction

In the chapters 3 and 4 a self-localization system for mobile robots, resorting to a PCA-based positioning system was detailed and experimentally validated using video or depth images, respectively. As described, the position provided by this algorithm is unique in the environment, being obtained by the corresponding position to the most similar image stored in the database. This chapter presents some implementations addressing a PCA-based localization system for mobile robots, using depth images extracted from the ceiling.

Thus, the enhanced PCA-based self-localization system using ceiling depth images is integrated in a closed loop control system to stabilize the mobile robot pose. In the implemented approach the mobile robot is commanded by the pose controller proposed by Aicardi et al. (1994), while its global pose is estimated by the self-localization system detailed in section 4.6.

In addition, in this chapter an alternative approach to implement the PCA-based localization for mobile robots in unstructured environments, using depth images captured from the ceiling, and resorting only to on-board sensors is presented. The method resorts to a PCA-based positioning system, filtered in a Bayesian probabilistic grid and combined with the attitude estimator (section 2.2) and the 2D position estimator (section 3.4.4) to estimate the mobile robot pose. In the implemented system, the environment information is captured only with on-board sensors installed in a differential drive robot. To avoid the negative impact caused by the corrupted data existing in the 2D depth sensor, the PCA extension to reconstruct depth images proposed in section 4.3 is applied in this approach (Rodrigues et al., 2013b).

Thus, this chapter is organized as follows:

- Section 6.2 details the integration of the PCA-based self-localization system using ceiling depth images (chapter 4) in a closed loop control system to stabilize the mobile robot pose. The implemented system is validated with experiments in a real indoor environment.
- Section 6.3 presents a self-localization system that merges the PCA-based positioning system with a ML, through a probabilistic grid-map. The PCA-based ML system is experimentally validated using depth images captured from the ceiling and the results are shown in this section;
- Finally, section 6.4 presents some conclusions about the work described in this chapter.

6.2 Mobile robot pose control using a PCA-based self-localization system

Control is a crucial task for a robot to perform missions. A common problem in mobile robots is the motion control towards a certain static pose. In this section, the PCA-based self-localization system is integrated in a closed loop to control the mobile robot pose. Thus, in the proposed approach, the controller proposed by Aicardi et al. (1994) were applied to stabilize the mobile robot pose, while its global localization is estimated by the self-localization system described in section 4.6. The results is experimentally validated by the mobile robot pose stabilization in a real environment.

6.2.1 Kinematic error

Lets consider a differential drive mobile robot located in a global position $\mathbf{x_r} = \begin{bmatrix} x_r & y_r & \psi_r \end{bmatrix}^T$, with non null distance to a goal position. The robot is moving in the inertial world $X_I Y_I$ with linear and angular velocity, u and ω respectively. The kinematic model is defined by the follow equation:

$$\dot{\mathbf{x}}_\mathbf{r} = \begin{bmatrix} \cos(\psi_r) & 0 \\ \sin(\psi_r) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \tag{6.1}$$

The design of the pose controller should allow to move the mobile robot to a position $\mathbf{x_g} = \begin{bmatrix} x_g & y_g \end{bmatrix}^T$ and an attitude aligned with the goal frame, being the corresponding error depicted in Fig. 6.1.

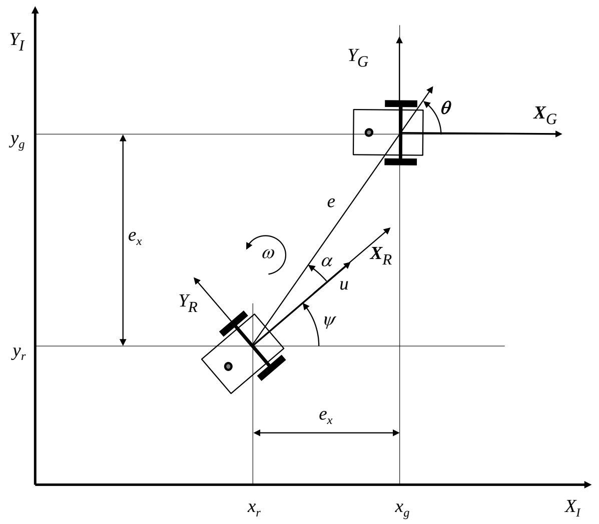

Figure 6.1: Pose error definition.

The following notation is used in Fig. 6.1:

- x_r and y_r are the actual mobile robot position;
- ψ_r is the actual mobile robot attitude;
- x_g and y_g are the mobile robot goal position;
- e_x and e_y are the error in cartesian coordinates, defined by $e_x = x_g - x_r$ and $e_y = y_g - y_r$, respectively;

- e is the distance error between actual and goal position;
- α is the angle between the robot frame and the distance vector;
- θ is the angle of the goal vector with respect with inertial frame.

Analyzing the Fig. 6.1, it is possible to define the mobile robot pose error in polar coordinates:

$$\begin{cases} e = \sqrt{{e_x}^2 + {e_y}^2} \\ \alpha = -\psi + atan2(e_y, e_x) & e > 0 \\ \theta = \psi + \alpha \end{cases} \tag{6.2}$$

leading to the kinematic model, defined for a non null distance error ($e > 0$):

$$\begin{cases} \dot{e} = -u\cos(\alpha) \\ \dot{\alpha} = -\omega + u\frac{\sin(\alpha)}{\alpha} & e > 0 \\ \dot{\theta} = +u\frac{\sin(\alpha)}{\alpha} \end{cases} \tag{6.3}$$

6.2.2 Controller for mobile robot pose stabilization

The mobile robot stabilization to any desired pose consists in the definition of a control law applied to the equation (6.3), that asymptotically drive the mobile robot to a null limiting point $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$, guaranteeing an asymptotic system stabilization towards the goal.

Thus, applying the approach proposed by Aicardi et al. (1994), a candidate Lyapunov function composed by two terms V_1 and V_2, that are related with the distance error vector, e, and the "alignment error vector", $\begin{bmatrix} \alpha & \theta \end{bmatrix}^T$, respectively, is given as follows:

$$V = V_1 + V_2 = \underbrace{\frac{1}{2}\lambda e^2}_{V_1} + \underbrace{\frac{1}{2}(\alpha^2 + h\theta^2)}_{V_2}, \qquad \lambda, h > 0 \tag{6.4}$$

The pose stabilization is guaranteed if the candidate Lyapunov function, equation (6.4), is always positive ($V > 0$) and its derivative, with the applied control function is negative ($\dot{V} < 0$), for all positive parameters λ and h. Thus, deriving the equation (6.4), it is obtain the follows:

$$\dot{V} = \dot{V}_1 + \dot{V}_2 = \lambda e\dot{e} + (\alpha\dot{\alpha} + h\theta\dot{\theta}) \tag{6.5}$$

Replacing the equation (6.3), in the equation (6.4) it becomes as follows:

$$\dot{V} = \underbrace{-\lambda e u \cos(\alpha)}_{\dot{V}_1} + \underbrace{\alpha\left[-\omega + u\frac{\sin(\alpha)}{\alpha}\frac{(\alpha + h\theta)}{e}\right]}_{\dot{V}_2} \tag{6.6}$$

Aicardi et al. (1994) propose the follow control law for the linear velocity:

$$u = (\gamma\cos(\alpha))e, \qquad \gamma > 0 \tag{6.7}$$

where γ is a positive parameter. The application of this control law in the term $\dot{V}_1$, becomes as follow:

$$\dot{V}_1 = -\lambda(\gamma\cos^2(\alpha))e^2 \leq 0 \tag{6.8}$$

This means that the application of the control law, equation (6.7) ensures the first term of equation (6.6) is non-increasing along time and lower bounded by zero and allows an asymptotic convergence to a non-negative finite limit, validating the first term of the Lyapunov function.

Repeating the same process to the second term and replacing the equation (6.7) in the equation (6.6), it becomes:

$$\dot{V}_2 = \alpha\left[-\omega + \gamma\frac{\cos(\alpha)\sin(\alpha)}{\alpha}(\alpha + h\theta)\right] \tag{6.9}$$

Aicardi et al. (1994) propose the follow control law for the angular velocity:

$$\omega = k\alpha + \gamma\frac{\cos(\alpha)\sin(\alpha)}{\alpha}(\alpha + h\theta), \qquad k > 0 \tag{6.10}$$

which ensures the asymptotic convergence of the second term:

$$\dot{V}_2 = k\alpha^2 \leq 0 \tag{6.11}$$

Thus, the addition of the equations (6.8) and (6.11) leads to the derivative of the original Lyapunov function, presented in equation (6.4), as follow:

$$\dot{V} = \dot{V}_1 + \dot{V}_2 = -\lambda(\gamma\cos^2(\alpha))e^2 - k\alpha^2 \leq 0 \tag{6.12}$$

that is a negative semi-definite function, validating the Lyapunov function and ensuring an asymptotic convergence to the limit point. According with the analyze detailed by Aicardi et al. (1994), the controller can stabilize the mobile robot pose through a smooth path when the controller parameters γ, k and h are tuned as follows:

$$h > 1, \quad 2\gamma < k < (h+1)\gamma \tag{6.13}$$

6.2.3 Pose control system architecture

Aiming to implement a mobile robot pose stabilizer controller, the PCA-based self-localization system is integrated in a closed loop control system, as depicted in Fig. 6.2.

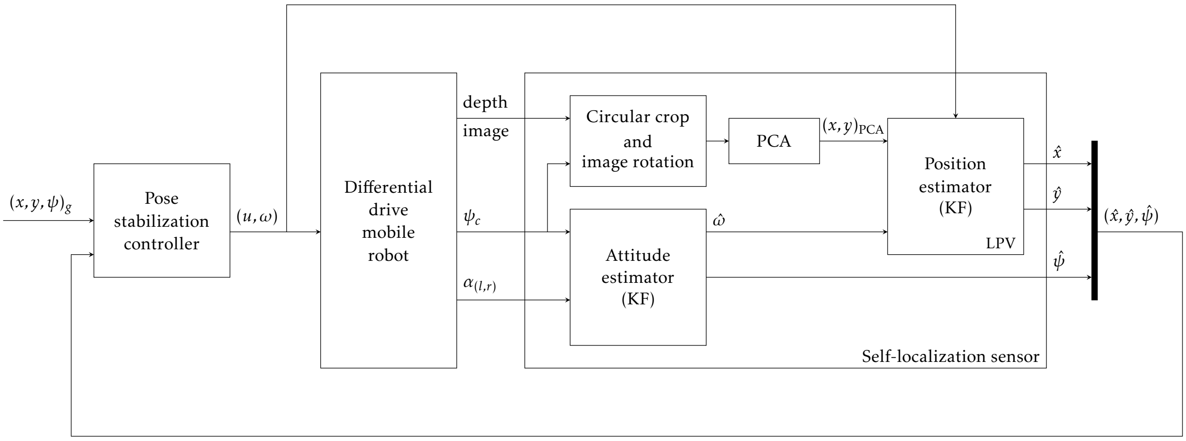

Figure 6.2: Pose control system architecture using the PCA-based self-localization system.

The architecture presented in this section implements the PCA-based positioning system using depth images (see details in section 4.3). Thus, during the motion, the depth image captured from the ceiling is rotated and decomposed into the PCA eigenspace to obtain the robot position. The results obtained from the PCA algorithm, $(x,y)_{\text{PCA}}$, are merged with the LPV, through the KF detailed in section 3.4.4, to estimate the global robot position. The robot attitude is estimated by the KF detailed in section 2.2, that merges the signals provided by the digital compass, ψ_c, with the odometry, $\alpha_{(l,r)}$. As detailed in chapters 3 and 4, this self-localization system estimates the mobile robot global pose $(\hat{x},\hat{y},\hat{\psi})$.

To stabilize the mobile robot in the desired pose, a closed loop controller defines the action to be sent to the motors, in order to reduce the error between the pose estimated by the PCA-based self-localization system and the desired one. Thus, the distance error,

e and the “alignment error”, $\begin{bmatrix} \alpha & \theta \end{bmatrix}^T$, are computed using the equation (6.2). The new mobile robot linear and angular velocity are obtained from the corresponding control laws, i.e., from the equations (6.7) and (6.10), respectively. Considering that the LPV model is designed for trajectories with constant velocity, a similar approach to that of section 5.5.2 is implemented. Thus, the linear velocity sent to the mobile robot is projected onto the cartesian axes, considering the estimated attitude, and the corresponding states used in the LPV model are replaced by these new values, i.e., $\dot{x} = u\cos(\hat{\psi})$ and $\dot{y} = u\sin(\hat{\psi})$.

6.2.4 Experimental results

To analyze the pose controller performance using the PCA-based self-localization system, some experiments are performed using a mobile robot with a differential drive kinematics equipped with on-board sensors (see section 4.2 for more details).

To create the PCA database, a set of 110 depth images is captured from the ceiling, along a grid map with a distance of $0.3m$ (in x and y axis). The PCA eigenspace is then computed, implementing the approach detailed in section 4.6. The selection of the best eigenvectors is performed through eigenvalues that exceed 85 % of the total of the eigenvalues, creating an eigenspace with 52 eigenvectors.

All experiences are performed aiming to control the mobile robot pose in $\mathbf{x_g} = \begin{bmatrix} 1.5 & 1.5 & 0 \end{bmatrix}^T$, when it starts from a set of different initial poses, presented in Table 6.1. The controller gains are tuned to $\gamma = 0.5$, $k = 1.2$ and $h = 4$, and it is assumed that the mobile robot arrives to the goal pose when the following condition is reached: $e \leq 0.05$ m and $|\theta| \leq 5$ °. During the motion, the linear and angular velocities of the robot are saturated with $u_{max} = 0.15$ m/s and $\omega_{max} = 1.5$ rad/s.

Analyzing Fig. 6.3, it is possible to observe that the mobile robot describes a path around the stabilization point and stops with the desired pose. The described trajectories are different, according to the initial relative pose. Although it needed to perform a maneuver when the robot starts in front of the goal position, it always arrives with positive linear velocity, i.e., moving in forward direction. Figure 6.5 shows the mobile robot attitude along time for all experimental tests. Analyzing these results, it is possible to see that, while the controller is attracting the mobile robot to the stabilization point, its attitude is

Table 6.1: Initial position and attitude of the mobile robot in posture stabilization tests.

	x position [m]	y position [m]	attitude [°]
Test no 1	2.4	1.5	0
Test no 2	2.1	2.1	0
Test no 3	1.5	2.4	0
Test no 4	0.9	2.1	0
Test no 5	0.6	1.5	0
Test no 6	0.9	0.9	0
Test no 7	1.5	0.6	0
Test no 8	2.1	0.9	0

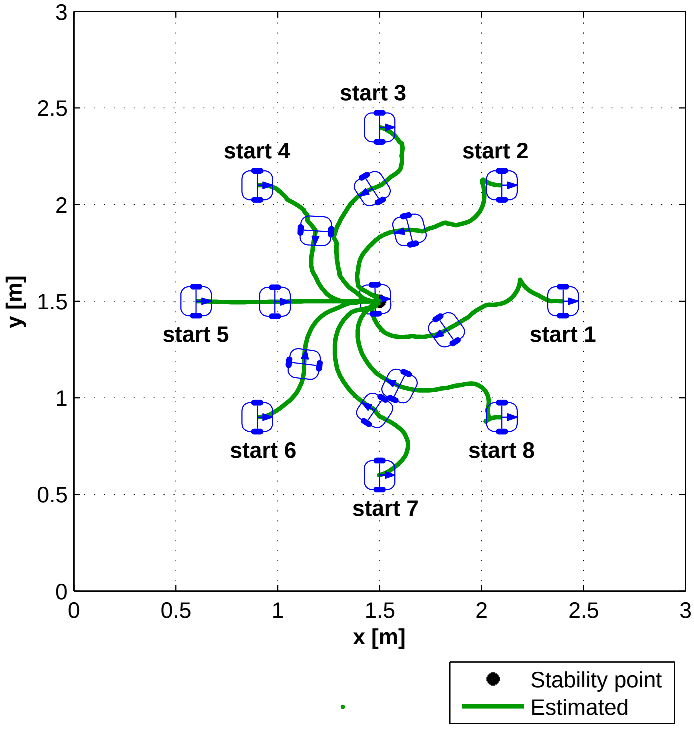

Figure 6.3: Mobile robot path for different initial poses[1] - experimental results.

also corrected to the desired goal.

Finally, these results show that the PCA-based self-localization system can be integrated with a mobile robot pose controller, stabilizing the robot in the desired pose. Notice that the

[1] The robots drawings are obtained from “CAS robot navigation toolbox” Arras (2004)

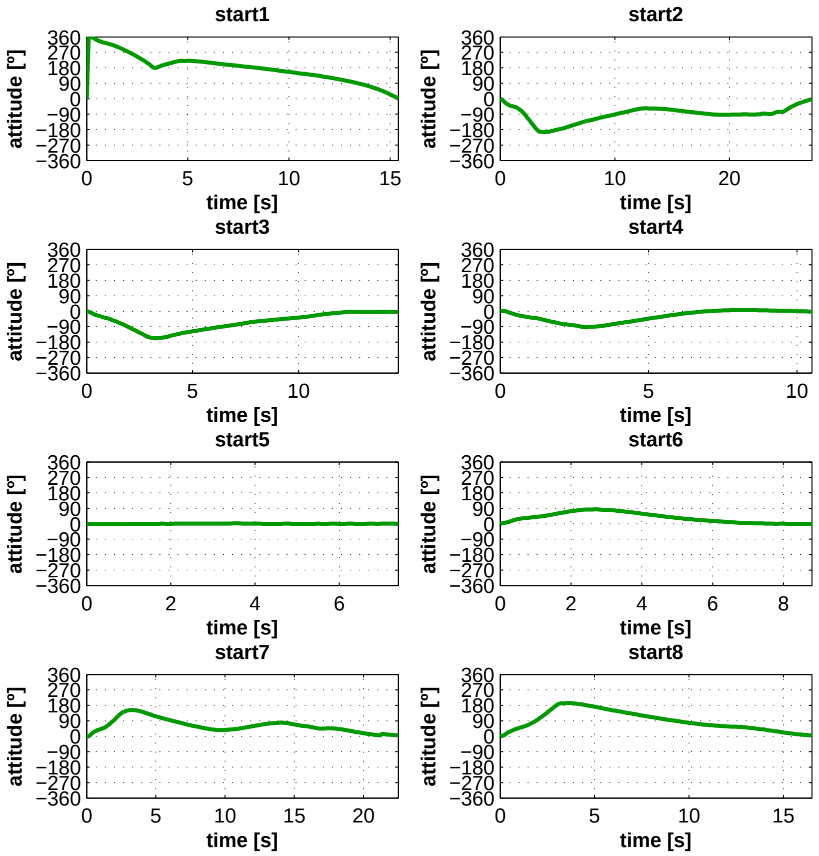

Figure 6.4: Mobile robot attitude along time for different initial poses - experimental results.

Figure 6.5: Mobile robot attitude along time for different initial poses - experimental results.

accuracy obtained in the final pose stabilization is greatly reduced when compared with the PCA-based positioning system grid resolution (0.3 m). This is achieved because the position obtained from the PCA-based system is merged with the LPV model, before it is used by the controller. Moreover, the pose controller only uses the results provided by the two linear KF, to estimate the position and the attitude respectively, that are globally stable.

6.3 PCA-based baeysian grid method for mobile robots positioning

The approach detailed in this section implements the PCA-based positioning system, resorting to ML methodology. The ML algorithm, that is an application of a Bayes filter, is often used in robot navigation, where the mobile robot position is achieved through a probabilistic grid-map (Thrun et al., 2005; Choset et al., 2005). This approach implements a multi-modal probabilistic function to represent the probability about the robot position, allowing the tracking of multiple possible localizations, using a discretization of the environment in a probabilistic grid-map.

The experimental validation of the positioning system proposed in this section has been performed, resorting to the mobile robot platform described in section 4.2, i.e., a differential drive mobile robot, with a Microsoft Kinect installed on the platform, pointing upwards to the ceiling, and a digital compass, located on the extension arm (robot rear part), as depicted in Fig. 4.1.

6.3.1 PCA-based grid-map construction

The PCA-based positioning system with ML detailed in this section also considers that the mobile robot moves in an indoor environment, under a ceiling with some informations (e.g. building-related systems such as HVAC, electrical and security systems, etc.). The depth information provided from ceiling images is used to build the environment grid-map and, during the motion, to implement the Bayesian filter. Note that, as detailed in section 4.2, due to geometry and properties of some objects, several IR beams are not well reflected and, thus, can not be detected by the depth sensor receiver. This results in the existence of points with null distance (0 mm) inside the data array with the depth values (see Fig. 4.2).

To avoid the erroneous results provided by the computation with corrupted depth signals, the environment grid-map was build using the PCA for signals corrupted with missing data, as detailed in section 4.3.

Thus, to build the environment grid-map, consider a set of $\mathbf{x}_i \in \mathbb{R}^N,\ i = 1,\ldots,M$ acquired by a Kinect depth sensor installed on-board the mobile robot, into two dimensional space. If the environment map has an obstacle-free grid with the size, $M = M_x M_y$, where M_x and M_y are the number of mosaics (ceiling depth images) captured in x and y axis, respectively. The environment grid-map is composed by the corresponding main eigenvectors, computed as described in section 4.3. Thus, each cell of the grid-map stores the depth image eigenvectors $\mathbf{v_i}$, and the global coordinates, $[x_i\ y_i]^T$, where it was captured.

In the approach presented in the section 4.6.5, during the mission, the signal $\mathbf{x}$ has been decomposed into the orthogonal space $\mathbf{v}$, and the position provided by the PCA-based positioning system, $[x_i\ y_i]^T_{PCA}$, is given by the mosaic i that corresponds to the eigenvector $\mathbf{v}_i$ most similar to $\mathbf{v}$, through the equation (4.6).

However, in the approach presented in this section, the position provided by the PCA-based positioning system has been obtained, integrating the eigenvectors distance with the predict motion, through a Bayesian probabilistic grid.

6.3.2 Bayesian positioning system with PCA-based probabilistic observations

The ML provides the robot localization through a grid, that represents the probability that the robot knows about its own position. This algorithm applies the Markov assumption, where knowledge of the previous state and current inputs is enough to predict the probability of the current state. Thus, ML can estimate the states $\mathbf{x}$ in the instant k based only in the instant $k-1$. The ML is a multi-modal algorithm that allows the tracking of multiple possible positions, disambiguating the possible repetitive information, and compute the right mobile robot position.

During an experiment, when the robot is moving, the probability about the robot states is continuously updated with the robot motion prediction and consequent observation, following the Markov algorithm in discrete instant k, where $\mathbf{x}(k-1)$ is the state vector in the previous iteration, $u(k)$ is the commanded linear velocity, $\mathbf{z}(k)$ is the observation in the current iteration and $\mathbf{m}$ is the environment grip-map (see Fig. 6.6).

```
Markov localization ($\mathbf{x}(k-1)$,$u(k)$,$\mathbf{z}(k)$,$\mathbf{m}$)
for i = 1 to M do
  $\bar{P}(\mathbf{x}_i(k)) = \sum_{i=i}^{N} p(\mathbf{x}_i(k)|\mathbf{x}_i(k-1), u(k))P(\mathbf{x}_i(k-1))$
  $P(\mathbf{x}_i(k)) = p(\mathbf{z}(k)|\mathbf{x}_i(k), \mathbf{m})\bar{P}(\mathbf{x}_i(k))$
return $P(\mathbf{x}(k))$
```

Figure 6.6: Markov localization algorithm.

The step corresponding to the correction of the measurement (or update) is performed considering that the observation, $\mathbf{z}(k)$, is obtained by the signal captured from the Kinect sensor and processed by the missing data correction algorithm. The final probability of the cell i in the instant k is given by the following equation:

$$P(\mathbf{x}_i(k)) = p(\mathbf{z}(k)|\mathbf{x}_i(k), \mathbf{m})\bar{P}(\mathbf{x}_i(k)), i = 1, \ldots, M \tag{6.14}$$

where $\bar{P}(\mathbf{x}_i(k))$ is the corresponding probability computed by the prediction step.

During a motion, the acquired depth image is rearranged to the signal $\mathbf{x}$ and, after apply the mean substitution method in the corrupted data, i.e, all j^{th} component of the signal $\mathbf{x}_i$ with corrupted data is replaced by the corresponding mean value $\mathbf{m}_x(j)$, the signal is decomposed into the orthogonal space. Thus, the signal $\mathbf{x}$ is projected into the orthogonal space, $\mathbf{v} = \mathbf{U}^T(\mathbf{x} - \mathbf{m}_x)$, by the same enhanced PCA algorithm for mobile robot localization using depth images corrupted with missing data, described in section 4.6.

The probability of the robot being in any position of the map is obtained by the distance between the captured image and the corresponding image into the PCA eigenspace:

$$p(\mathbf{z}(k)|\mathbf{x}_i(k), \mathbf{m}) = (1 - \eta\|\mathbf{v} - \mathbf{v}_i\|_2), i = 1, \ldots, M \tag{6.15}$$

where η is a normalization factor, which ensures $\sum p(\mathbf{z}(k)|\mathbf{x}_i(k), \mathbf{m}) = 1$, $\mathbf{v}$ is the eigenvector of the captured image, $\mathbf{v}_i$ is the eigenvector of the i image into the eigenspace and $p(\mathbf{z}(k)|\mathbf{x}_i(k), \mathbf{m})$ is the probability to observe $\mathbf{z}(k)$ in the state $\mathbf{x}_i(k)$, considering the map $\mathbf{m}$.

Finally, the robot position given by the PCA-based positioning system with ML is obtained finding the mosaic that has the maximum probability calculated through the equation (6.14):

$$[x\,y]^T = \max_{\mathbf{x}_i(k)}(P(\mathbf{x}_i(k))), i = 1, \ldots, M \tag{6.16}$$

6.3.3 Experimental results

6.3.3.1 Experimental setup

In order to perform an experience in a real environment, a set of ceiling depth images was captured along a grid, in an area of 18.9 × 9.6 m. The depth images were cropped, sampled with a compression ratio of 100 : 1 and converted into a vector to be added to the PCA database. Then, the PCA eigenscape was created by component selection to explain image variability to a minimum of 85 % (see section 4.6 for more details).

The PCA-based positioning system with ML is integrated in a self-localization system with the attitude estimator (section 2.2) and the 2D position estimator (section 3.4). The system architecture is depicted in Fig. 6.7, being used the following notation:

- u - commanded linear velocity;
- ψ_c - attitude angle given by the compass;
- θ_r - angle given by the encoder of the right wheel;
- θ_l - angle given by the encoder of the left wheel;
- $(x,y)_{ML-PCA}$ - position coordinate given by the PCA-based positioning system with ML;
- $(\hat{x},\hat{y})$ - estimated robot coordinates in the world referential;
- $\hat{\psi}$ - estimated attitude;
- $\hat{\omega}$ - estimated angular velocity.

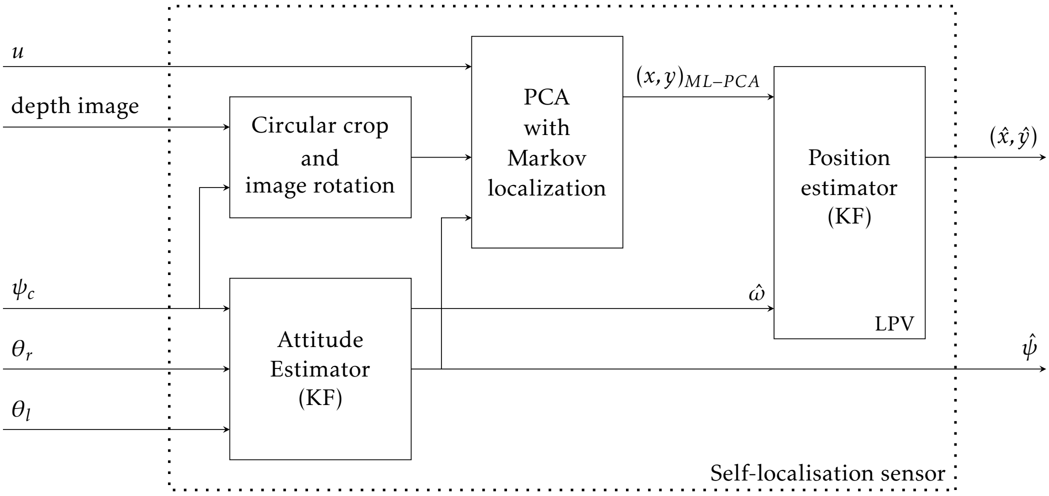

Figure 6.7: Architecture of the self-localization system based on PCA with ML

Detailing the architecture of the self-localization system presented in Fig. 6.7, the KF depicted on the left of the figure implements the optimal attitude estimator, presented in section 2.2, to estimate the attitude and the angular velocity of the mobile robot. Once all depth images acquired for the PCA database are taken with the same orientation and compressed with a circular crop (Fig. 4.10), during a mission, the acquired depth images must be rotated to zero degrees of attitude, using the compass angle, and compressed with the same circular crop. The 2D position estimator (on the right of the figure) implements the LPV model for differential drive robots as a function of the estimated angular velocity in a KF (section 3.4), fusing it with the position obtained by the PCA with ML algorithm, detailed in section 6.3.2.

Resorting to this architecture, it is possible to estimate the mobile robot attitude with global stable error dynamics and to improve the position estimated by the PCA-based positioning system with ML.

6.3.3.2 Bayesian motion prediction

The method detailed in Fig. 6.7 applies the ML algorithm to compute the position of the robot in a probabilistic grid, considering the position of the depth images captured to create the PCA database. Thus, the state vector of the Bayesian filter is the position obtained by the PCA with the ML algorithm: $\mathbf{x}_i = [x_{ML-PCA} \;\; y_{ML-PCA}]^T$. As presented in section 6.3.2, the filtering step of the Bayesian filter is computed through a probabilistic grid based on the PCA algorithm applied to the captured depth image. With the propose of computing automatically the probability of the state transition in the prediction step, and assuming that the robot moves in the direction given by the attitude estimator, the following method has been applied.

Considering a grid with M_x and M_y cells in x and y axis, respectively, a probability map with $M_x \times M_y$ is created and initialized with the same probability in all states $P(\mathbf{x}(0)) = \frac{1}{M}$, where $M = M_x \times M_y$.

During a mission, following the ML algorithm, the robot position probability map is updated in a prediction step, considering the probability of the robot reaching a new state $\mathbf{x}_i(k)$ and the commanded linear velocity $u(k)$ along the direction $\psi(k)$. This probability is obtained by the sum of all possible ways for the robot to reach the mentioned state and is given by the following equation:

$$\bar{P}(\mathbf{x}_i(k)) = \sum_{i=1}^{M} p(\mathbf{x}_i(k)|\mathbf{x}_i(k-1), u(k))P(\mathbf{x}_i(k-1)), i = 1, \ldots, M \tag{6.17}$$

Once that the states are stored in a grid, the probability map computation in the prediction step is performed considering the geometry of the grid and the velocity and attitude of the robot, as shown in Fig. 6.8.

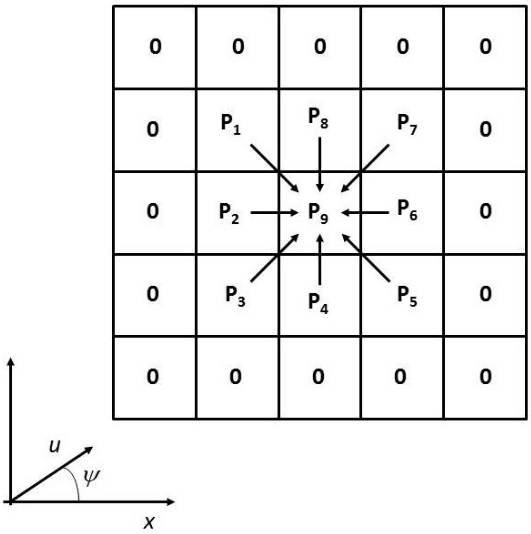

Figure 6.8: Probabilities in prediction step.

Analyzing the probability state transition of Fig. 6.8 and considering that the robot is moving with a velocity u in a direction with an attitude ψ, the probability states transition is given as described above. The predicted probability for a robot being in $\mathbf{x}_i(k)$ is computed as follow:

$$\bar{P}(\mathbf{x}_i(k)) = \eta \sum_{l=1}^{9} p_l P(\mathbf{x}_l(k-1)) \tag{6.18}$$

where η is a normalizer, which ensures $\sum \bar{P}(\mathbf{x}_i(k)) = 1$ and p_l the probability for a robot moving from the cell l. Thus, considering that the robot is moving with an attitude ψ, the possible way that is proposed in this section to compute the state transition probability from each 8 neighbour cells ($l = 1 \ldots 8$) is given as follows:

$$p_l = \begin{cases} (\cos(\psi + \operatorname{atan2}(y_l - y_9, x_9 - x_l))^q & if p_l > 0 \\ 0 & if p_l <= 0 \end{cases} \tag{6.19}$$

$$p_l = \eta \cdot p_l(1 - p_9) \tag{6.20}$$

with η being a normalization factor, which ensure $\sum p_l = 1$, and p_9 being the probability that the robot is kept in the same state.

Once $p_l < 1, l = 1 \ldots 8$, the exponent q is a tuning parameter that allows to increase the probability between the states in the same direction of the robot attitude, reducing the probability between the states whose transition is caused by angular motion uncertainty (diagonal direction in state transition).

Finally, considering that the robot is moving with a velocity u in a direction with an attitude ψ, on a probabilistic grid with size d, and commanded by a digital processor with a sampling time T, the new predict states probability map $\mathbf{x}_i(k)$ is obtained as follows:

$$\bar{P}(\mathbf{x}_i(k)) = \quad \eta(\textstyle\sum_{l=1}^{8}(p_l \frac{u \cdot T}{d} P(\mathbf{x}_l(k-1))) + p_9(1 - \frac{u \cdot T}{d}) \cdot P(\mathbf{x}_9(k-1))) \tag{6.21}$$

6.3.3.3 Results for 2D localization

To test the mobile robot self-localization performance of the proposed approach, an experiment was performed in an environment with repeatability, along a predefined path with about 93 m, combining both straight lines and curves and travelling two laps inside the mapped area. During the experiment, the robot was moving with a velocity of 0.1 m/s and the data acquisition was performed with a sampling frequency of 2.5 Hz.

Figures 6.9–6.12 show the probabilistic grid performed by ML with a filtering step based on the PCA algorithm, where the “hot colors” (dark red) represent high localization probability and “cold colors” (blue) the low localization probability. The blue areas in the center, bottom and top of Fig. 6.9 represent unmapped areas which do not exist in the PCA eigenspace and, so, have null probability.

Figure 6.12 shows that the bayesian algorithm is able to quickly disambiguate possible repeatability of the scenery and to find the right position. This figure shows that after 30

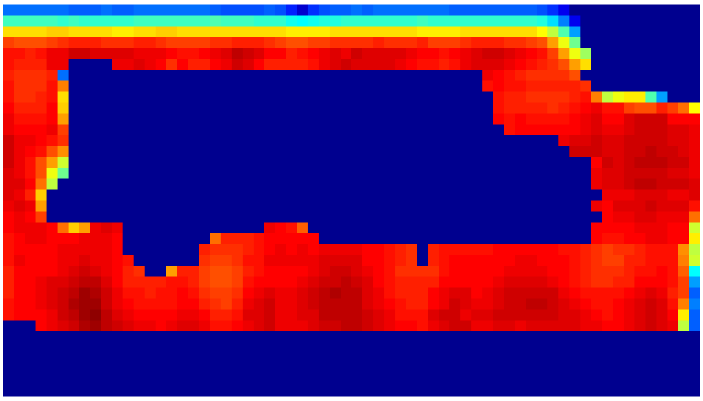

Figure 6.9: Localization probability grid after 1 observation.

samplings (12 seconds), the positioning system provides an high probability about the right position of the robot, which can be validated with the results presented in Fig. 6.13.

Figures 6.13 and 6.14 present the results of the self-localization system, comparing the position obtained by the PCA-based positioning system with ML, the global estimated position by the KF and the real path of the robot measured in the ground. The results of Fig. 6.13 show that the PCA-based positioning system with ML is able to achieve an accurate position of the robot, allowing good performance on the global self-position sensor. Note that the grid with depth images has an accuracy of 0.3 m, whereas most of the position estimated is close to the ground truth, with less error than the grid dimension.

Figure 6.14 shows that the self-localization system estimates the path correctly while the path estimated by odometry diverges completely from the ground truth path. The blue circles represent the position uncertainty obtained by the KF. As it is possible to observe, the uncertainty increases when the robot is in the top of the Fig 6.14. This happened because in this area the ceiling has less information. Nevertheless, Fig 6.13 and Fig 6.14 show that the robot is still able to estimate its position, even when uncertainty increases.

Analyzing the results of the attitude estimator in Fig. 6.15, it is possible to observe that the estimated attitude is very close to the ground truth, allowing to conclude that this KF provides results with good accuracy.

Finally, analyzing the distribution of the estimated position error in Fig. 6.16, can be concluded that it is approximately Gaussian with a mean close to zero. The distribution is

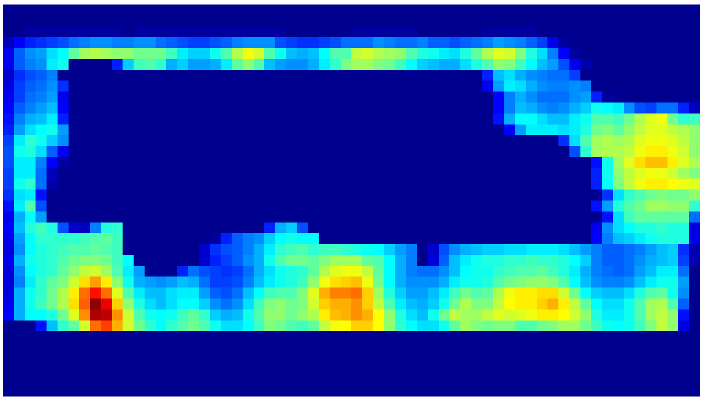

Figure 6.10: Localization probability grid after 10 observations.

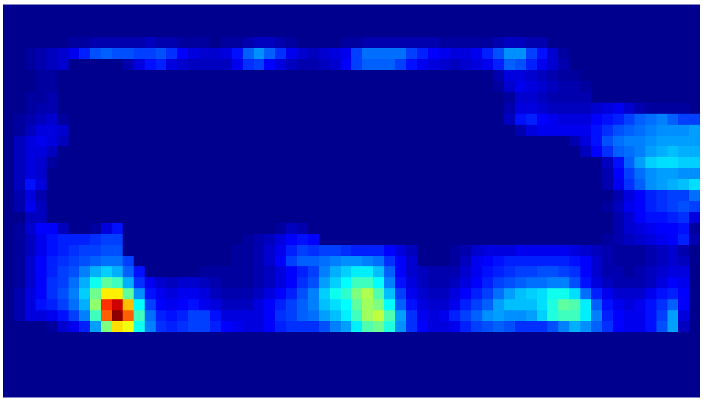

Figure 6.11: Localization probability grid after 20 observations.

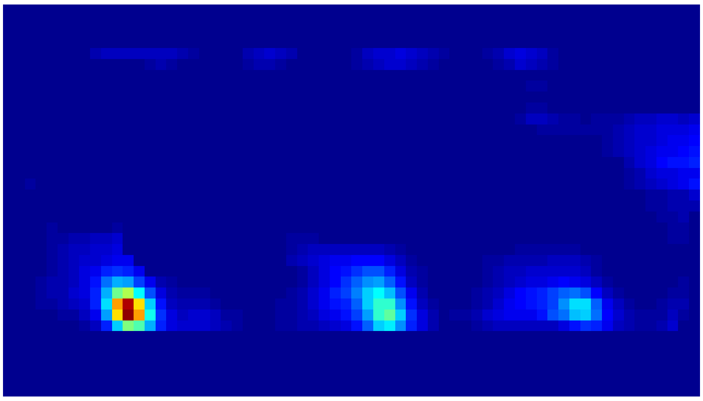

Figure 6.12: Localization probability grid after 30 observations.

non-zero mean Gaussian because the trajectory is not random and due to the finite resolution of the PCA probabilistic grid (0.3 m). However, the mean error in both axis is very close to zero being $\bar{e_x} = 4 \times 10^{-3}$ m and $\bar{e_y} = 4 \times 10^{-3}$ m, thus giving an positive indication for the validation of the Gaussian error assumption of the KF design.

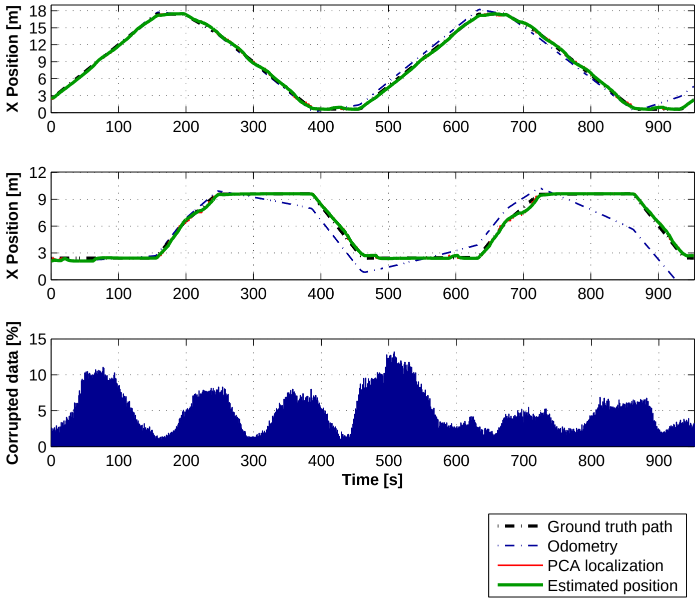

Figure 6.13: Estimated position along time.

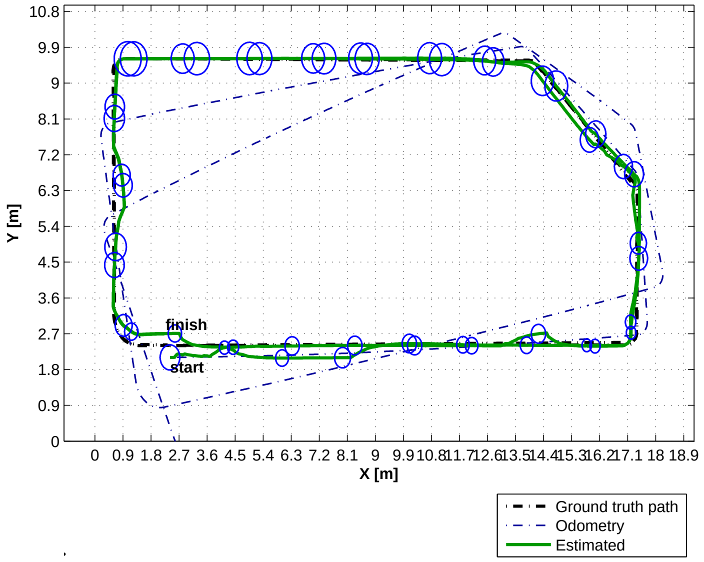

Figure 6.14: Map with estimated position considering a ground truth path.

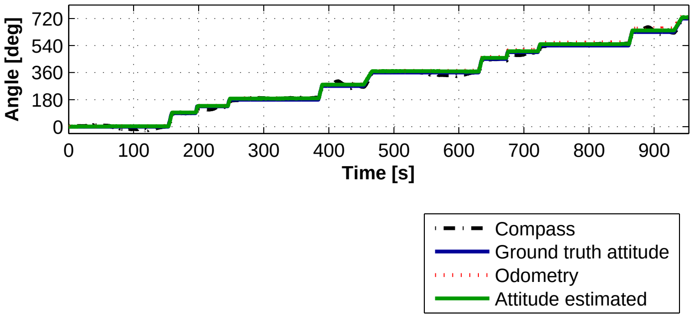

Figure 6.15: Estimated attitude along time.

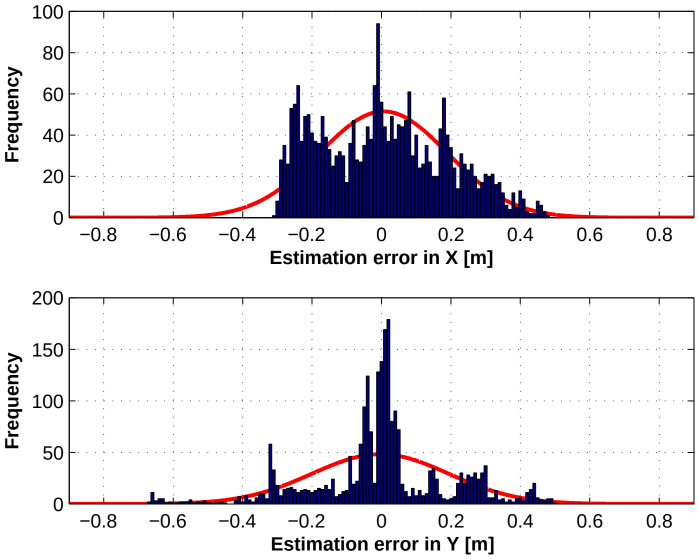

Figure 6.16: Distribution of the estimated position error.

6.4 Conclusions

This chapter presented some implementations using a PCA-based positioning systems. The first implementation combines the PCA-based self-localization system integration, using ceiling depth images with a closed loop to control the mobile robot pose. Thus, the implemented approach used the raw data of depth images captured from the ceiling and the signals of a compass and the encoders' odometry to estimate the mobile robot global pose. The pose error was computed from the estimations provided from the PCA-based self-localization system. The experimental results showed that the developed estimator can be integrated in a closed loop, stabilizing the mobile robot in the desired pose when it starts from different initial conditions.

In addition, this chapter also presented an alternative approach to implement a PCA-based positioning system for mobile robot global self-localization in unstructured environments, i.e., that does not need any beacons or other artefacts structuring the environment. The proposed method resorted to a PCA-based positioning system, filtered in a Bayesian probabilistic grid and combined with two linear KF to estimate the mobile robot global pose.

Considering that the existence of scenarios with repetitive elements or very similar in different points of the environment is usual and can induce the positioning systems to a erroneous localization of mobile robots, the Bayesian method presented in this chapter presents a possible solution to this problem.

The integration of the Bayesian PCA-based positioning system with a linear KF allows to obtain a suboptimal localization of the mobile robot. The sub-optimality is due to the non-linear PCA-based positioning system behaviour. The obtained results show that the PCA-based positioning system with ML implementation in the self-localization with linear KF allows to achieve a mobile robot estimated position with an error distribution approximately Gaussian.

7

Conclusions and future work

7.1 Conclusions

This work addressed the indoor mobile robot localization, using only sensors installed on-board. It was considered that the robots navigate under static ceilings with rich information, e.g. HVAC, electrical, security systems, etc. Notice that ceilings with these characteristics can be found in industrial environments or similar. Along this work, it was implemented and experimentally validated several algorithms that can be combined in an indoor self-localization system architecture, designed for differential drive mobile robots.

To define the mobile robot global position, the implemented system used a video or depth camera looking upward to the ceiling, acquiring grayscale or depth images, respectively. In opposition to the most common approaches, that requires complex algorithms to extract specific features or detect landmarks, the implemented approach uses only the raw data extracted from images, achieving the global pose without the need to structure the environment.

Aiming to achieve the robot global position in the environment, the self-localization system resorts to the PCA of the ceiling images raw data, namely the KL transform. The position is found by the eigenvectors comparison of the acquired image with an eigenspace, representing the environment map. To obtain the mobile robot position, using the proposed PCA-based positioning system, it is assumed that: i) the information about the environment map is stored on-board the robot in a compact database; ii) the ceilings are static and contain enough information so that the robot can capture different images along its motion.

Thus, to validate the proposed PCA-based positioning system, a manual mapping was performed, acquiring a set of ceiling images, in known positions, covering the area where the robot will move. The PCA algorithm application allows to store information about the environment with high compression ratio. This greatly reduces the amount of data stored on-board the mobile robot, when compared with the data dimension associated to the captured from all mapping images. Consequently, this allowed to implement an indoor mobile robot positioning system with low complexity and avoiding the need to structure the environment, e.g., corners, lines or landmarks.

In addition to the problems of image processing for self-localization, another challenge is to deal with the fusion of the PCA-based positioning system with the odometry data given by the robot kinematics. Thus, to improve the self-localization robustness, the results obtained from the PCA-based positioning system were merged with the robot kinematics using only linear KF.

The combination of a PCA-based positioning system with a KF was experimentally validated along a straight line (1D localization). This approach aimed to prove the system concept before it has been expanded to a 2D localization. To perform these experiences, a differential drive mobile robot was equipped with a video camera installed on-board, looking upwards to the ceiling. The PCA eigenspace was created using a set of ceiling images captured along a straight line. Several tests were performed to analyse the results given by the PCA-based positioning system, and after their fusion with the odometry by a KF: i) Monte Carlo performance study, ii) global stability validation, iii) real-time slippage estimation, and iv) PCA performance assessment.

All tests were successfully performed, being possible to conclude that the use of the PCA algorithm in ceiling images can be useful to achieve the robot position with reasonable precision. In addition, it was possible to see that the PCA-based positioning system combination with a linear KF improves the estimates accuracy, even if there is a wrong initial estimate or slippage. This allowed to conclude that the proposed approach can be applied to mobile robot self-localization systems that operate in indoor environments, under ceilings with rich information.

The purposed approach was extended to 2D localization and experimentally validated. The self-localization problem of mobile robot with differential drive kinematics, using images captured from the ceiling, led to two issues: i) ceiling images may appear rotated when compared to those captured during the mapping; ii) differential drive kinematic is non-linear, preventing the access to optimal estimates as provided by a KF.

The first issue was addressed by applying a circular crop in the image center. This allowed to rotate the images to the same rotation used in the mapping, ensuring that data can be compared. To solve the second issue, the position obtained from the PCA-based positioning system was merged by a KF, through the implementation of a LPV model for differential drive robots. This solution led to the self-localization system design with an architecture conceptually composed by three modules:

i) an optimal attitude estimator, using the data obtained from a digital compass and motion sensors installed on-board the robot;

ii) a PCA-based nonlinear positioning system that provides measurements on the absolute 2D position, using (grayscale or depth) images acquired from the ceiling;

iii) a sub-optimal 2D position estimator, based on a LPV model for differential drive robots, parameterized by the estimated instantaneous angular rate.

With the purpose of testing the performance of the developed self-localization system, several experimental tests were carried out: i) self-localization tests with ground truth validation and Monte Carlo performance study; ii) global stability validation for position and attitude estimation, and iii) angular slippage estimation and linear slippage algebraic computation. Results show that the self-localization system has fast convergence and guarantees global stability, due the KF properties, even when the estimator starts from incorrect positions and attitudes.

Thus, the experimental results shown that the proposed system implementation can solve the problem of mobile robots self-localization, using visual images captured from the ceiling, without the need to extract specific characteristics from the environment, and providing globally stable estimates. To achieve these estimates, the self-localization system requires that the robot have installed a camera looking to the ceiling, to obtain the global position from the PCA algorithm, and a sensor that provides the robot global attitude in the environment, such as a compass.

The use of images captured by a video camera installed on-board was enough to validate the global stability of the propose self-localization system, despite the general limitation related to different environment lighting conditions that often degraded the localization systems robustness. This limitation was felt sometimes, causing a degradation in results provided by the PCA-based positioning system.

To overcome this limitation, another prototype was build, but with a Microsoft Kinect device looking upward to the ceiling, instead of the video camera. This allowed to update the developed self-localization system to use depth images captured from the ceiling. Depth images have the advantage of being robust to different lighting conditions. However, due to the geometry and properties of same objects, several waves are not reflected, causing the existence of missing data in depth images. Since the PCA algorithm consider all captured data to map the environment or to localize the mobile robot, the presence of missing data can disturb the results, induced the positioning system to erroneous localizations.

Thus, to avoid the negative impact of missing data in signals, an extension to reconstruct corrupted signals was implemented in the PCA-based positioning system, aiming to use depth images captured from the ceiling. To validate the concept of the image reconstruction resorting to a PCA eigenspace, a test to reconstruct a depth image corrupted with missing data was performed. The results show that the proposed PCA algorithm can reconstruct depth images, providing good results until a missing data ratio of about 50 %.

This new approach was also experimentally validated to achieve the mobile robot position along a straight line (1D localization) and in the 2D self-localization system. The enhanced PCA-based positioning system implementation into the self-localization systems was similar to the previous algorithm using gray images. The performed tests considered was the follows: i) assessment of the depth images reconstruction robustness; ii) Monte Carlo tests; iii) localization with imposed corrupted data.

All tests were successfully performed, allowing to conclude that the proposed approach is able to deal with the existence of missing data in depth images, avoiding the localization systems performance degradation. Thus, the implemented algorithm to deal with corrupted depth images allows: i) to compress the acquired database; ii) to remove missing data existing in the depth image; iii) to create eigenvectors to compare the reconstructed depth image with the PCA database. Moreover, the robustness test allows to conclude that the proposed algorithm is able to estimate an accurate position, considering depth images with corrupt data ratio up to about 50 %.

The self-localization system using depth images was tested considering a larger environment. When the mapping area increased, the existence of equal equipment installed repetitively in the ceiling at different places, became a reality in the considered environment. This caused the existence of repeatable scenarios, where similar ceiling images may be captured in different positions. When this happens, the search for the robot's position through the closest depth image can generate too many erroneous values. To solve this issue, the PCA-based positioning system was implemented with a searching for the closest vector in a neighbourhood around the robot estimated position. This solution demonstrated the ability of PCA to be implemented in a system localization with environment repeatability. However, since this search is not done in the entire database, but only in a restricted area, this approach requires that the position estimator is initialized close to the right position and within the neighbourhood radius.

Aiming to make the robot move autonomously in the environment, the PCA-based self-localization system was integrated in a navigation system with a path planner and a motion controller. The implemented system uses the information about the capture ceiling images for the PCA database to define an occupancy grid that represents the workspace. Such grid was useful to build a roadmap based on a GVD, from which a set of waypoints can be obtained, corresponding to the GVD vertices. During the motion, an APF based on DNF planes a local path to drive the robot toward temporary or final goals, according to the waypoints' set. The DNF implementation allowed to adapt the local planning and the motion control to the mobile robot with differential drive kinematics.

The PCA-based self-localization system was integrated in the proposed navigation system. Some experiments were performed in a real environment, using a mobile robot equipped with on-board sensors (Kinect looking upward, a rate gyro, a compass and the encoders), and considering different conditions: i) navigation via waypoints between two locations in the environment; ii) navigation starting from a wrong position estimate; iii) navigation along a large mission with one intermediate point that must to be visited along the mission; iv) navigation along a large trajectory, describing laps in a closed loop.

The experimental results showed that the mobile robot describes trajectories close to the real trajectory, reaching its destination close to the estimated position. Moreover, the self-localization can stabilizes in few seconds when is initialized with wrong estimates, allowing a good navigation toward its destination. The navigation tests along large trajectories demonstrate that the PCA algorithm provides less accurate positions when the robot captures depth images in ceilings with very few information. However, when this happen, the estimates are improved by the 2D position estimator. In spite of this, experimental results show that, when the robot returns to a place where depth ceiling images can be easily distinguished, the positions provided by the PCA algorithm becomes again to be more accurate and, consequently, the self-localization system estimates also improves. Finally, experimental results also allowed to observe that the PCA algorithm provides similar position results when the robot returns to the same place.

With the purpose of extending the attitude estimator to implement the sensor fusion with more motion sensors, this work introduced a new class of complementary filters extending the classic sensor fusion CF design. The estimators design in the proposed approach aims to merge the signals provided from a set of sensors working in different, yet complementary, frequency bands. The estimator output has a unitary magnitude gain over whole frequency spectrum, achieving, thus, the complementarity propriety. The straightforward division in frequency bands led to a sub-optimal solution when more than two bands are used. Then, the problem was reformulated as an optimal linear KF. Finally, an optimal estimator to merge the signals provided by three sensors, based in Kalman filtering, was designed to estimate the mobile robot attitude. Moreover, all solutions are shown to preserve the complementary property, i.e. the sum of all transfer functions add up to one.

Some attitude estimator, presenting different approaches and inputs signals, were implemented in a simulation model, to merge the signals provided from three sensors: compass, odometry and rate gyro. Once after the warm up time the rate gyros bias keeps constant and can be compensated, the best solution was achieved with the linear KF implementation, using the rate gyro signals in the LPF input. This overcomes the magnetic interferences in the compass signals. Some discrete-time attitude estimators were combined with the LPV model in dead-reckoning localization, achieving accurate results.

Note that, although the approach presented in this work was implemented to estimate the mobile robot attitude, this method can easily be adapted to estimate variables in other processes, resorting to sensor fusion based on CF.

Moreover, some example of implementations using a PCA-based self-localization systems are presented. Thus, the PCA-based self-localization system was integrated with in a closed loop control system. The implemented approach allowed to stabilize the mobile robot in a desired pose. The experimental results show that the global pose estimated by the proposed PCA-based self-localization system can be integrated with a closed loop controller, stabilizing the mobile robot in the desired pose when it starts from different initial conditions.

Finally, an alternative approach to implement a PCA-based 2D positioning system using a probabilistic position was detailed. In the implemented approach, the position is obtained by the integration of a ML with the PCA. Thus, a Bayesian algorithm propagates the robot positions along a grid, being the states updated by comparing the eigenvectors of the captured image with the corresponding cell. Despite of this approach being more complex than the standard PCA-based positioning system, the results show that the integration of the PCA with ML can disambiguate the robot position in case of repetitive scenarios. As it happened with the PCA-based positioning systems detailed above, the results provided by this approach can also be merged with the odometry by the linear KF, increasing the estimated position accuracy.

As a conclusion, this work detailed some approaches to design mobile robot localizations systems, using only the on-board sensors. The proposed approaches resorts to a mobile robot using a positioning system based on ceiling (grayscale or depth) images. The implemented systems use raw data extracted directly from the sensors, in algorithms with linear computation. The use of PCA-based positioning system and two KF, decoupling the attitude and the position estimation, allows to achieve accurate and stables estimates, without the need to extract any specific features from the environment structures. In addition, the environment database was compressed with a great ratio, reducing the required resources to store the information and to compute the mobile robot pose.

7.2 Future work

In the future, the PCA-based positioning system will be expanded to use RGB images and depth, simultaneously. In this case two different approaches can be explored: i) the implementation of one PCA-based positioning system considering the two signals concatenated; ii) the use of the two positioning systems merged in a KF. Thus, it is possible to analyze the systems performance with the implementation of different approaches.

The proposed navigation system will be integrated in a multi-agents system, allowing the multi-robot navigation in the same environment. To perform this task, the knowledge about the environment, i.e., the eigenspace with ceiling information, must be shared by all robots. The sharing of all robots' global pose will allow to control their motion using multi-robot navigation functions. Thus, it will be possible to achieve multi-robot navigation, avoiding collision, using only information provided from ceiling images and motion sensors, without the need of features extraction.

Later, to implement the self-localization system in a mobile robot that performs SLAM tasks, the proposed PCA-based 2D positioning system will be updated to create an incremental PCA database. The integration of this improvement in the multi-agent system will allow teams of robots to explore the environment, sharing the knowledge and taking advantage of the PCA algorithm.

Finally, the integration of people detection algorithms in this navigation system will allow to implement tasks of interaction between robots and humans. When a mobile robot navigates among humans, it is important to consider the safety (physical and psychic) distances between them. This approach will improve the humans comfort when they are in the presence of vehicles in movement.

References

Aicardi, M., Casalino, G., Balestrino, A., Bicchi, A., 1994. Closed loop smooth steering of unicycle-like vehicles, in: Proceedings of the 33rd IEEE Conference on Decision and Control, IEEE, Lake Buena Vista, Florida. pp. 2455–2458. DOI: 10.1109/CDC.1994.411509.

Aicardi, M., Casalino, G., Bicchi, A., Balestrino, A., 1995. Closed loop steering of unicycle like vehicles via lyapunov techniques. IEEE Robotics & Automation Magazine 2, 27–35. DOI: 10.1109/100.388294.

Almansa-Valverde, S., Castillo, J.C., Fernández-Caballero, A., 2012. Mobile robot map building from time-of-flight camera. Expert Systems with Applications 39, 8835 – 8843. DOI: 10.1016/j.eswa.2012.02.006.

Arkin, R., 1987. Motor schema based navigation for a mobile robot: An approach to programming by behavior, in: Proceedings of ICRA 1987, the IEEE International Conference on Robotics and Automation, IEEE, Raleigh, USA. pp. 264–271. DOI: 10.1109/ROBOT.1987.1088037.

Arras, K.O., 2004. Cas robot navigation toolbox. URL: http://www2.informatik.uni-freiburg.de/~arras/resources.html.

Arslan, O., Koditschek, D.E., 2016. Exact robot navigation using power diagrams, in: Proceedings of ICRA 2016, the IEEE International Conference on Robotics and Automation (accepted), Stockholm, Sweden. pp. 1–8. DOI: 10.1109/ICRA.2016.7487090.

Artač, M., Jogan, M., Leonardis, A., 2002. Mobile robot localization using an incremental eigenspace model, in: Proceedings of ICRA 2002, the IEEE International Conference on

Robotics and Automation, IEEE, Washington, DC, USA. pp. 1025–1030. DOI: 10.1109/ROBOT.2002.1013490.

Arunkumar, G., Sukumar, S., Vachhani, L., 2014. Lyapunov function based steering law for generalized voronoi diagram (gvd) construction by a mobile robot, in: Proceedings of ACODS 2014, the 3rd International Conference on Advances in Control and Optimization of Dynamical Systems, Kanpur, India. pp. 103 – 108. DOI: 10.3182/20140313-3-IN-3024.00137.

Auger, F., Hilairet, M., Guerrero, J., Monmasson, E., Orlowska-Kowalska, T., Katsura, S., 2013. Industrial applications of the kalman filter: A review. IEEE Transactions on Industrial Electronics 60, 5458–5471. DOI: 10.1109/TIE.2012.2236994.

Bacca, B., Salvi, J., Cufí, X., 2011. Appearance-based mapping and localization for mobile robots using a feature stability histogram. Robotics and Autonomous Systems 59, 840–857. DOI: 10.1016/j.robot.2011.06.008.

Baerveldt, A.J., Klang, R., 1997. A low-cost and low-weight attitude estimation system for an autonomous helicopter, in: Proceedings of INES'97, the 1997 IEEE International Conference on Intelligent Engineering Systems, Budapest, Hungarian. pp. 391–395. DOI: 10.1109/INES.1997.632450.

Bailey, T., Durrant-Whyte, H., 2006. Simultaneous localization and mapping (slam): part ii. IEEE Robotics Automation Magazine 13, 108 – 117. DOI: 10.1109/MRA.2006.1678144.

Bailey, T., Nieto, J., Guivant, J., Stevens, M., Nebot, E., 2006. Consistency of the ekf-slam algorithm, in: Proceedings of ICIRS 2006, the IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Beijing, China. pp. 3562–3568. DOI: 10.1109/IROS.2006.281644.

Barbosa, J., Cardeira, C., Oliveira, P., Batista, P., Silvestre, C., 2015. Design and validation of an rgb-d based localization system-integration in a docking system. Journal of Intelligent & Robotic Systems 80, 423–440. DOI: 10.1007/s10846-015-0181-7.

Barraquand, J., Latombe, J.C., 1991. Robot motion planning: A distributed representation approach. The International Journal of Robotics Research 10, 628–649. DOI: 10.1177/027836499101000604.

Bascetta, L., Cucci, D.A., Matteucci, M., 2016. Kinematic trajectory tracking controller for an all-terrain ackermann steering vehicle, in: Proceedings of IAV2016, the 9th IFAC Symposium on Intelligent Autonomous Vehicles, Elsevier, Leipzig, Germany. pp. 13–18. DOI: 10.1016/j.ifacol.2016.07.600.

Batista, P., Silvestre, C., Oliveira, P., 2010a. Optimal position and velocity navigation filters for autonomous vehicles. Automatica 46, 767–774. DOI: 10.1016/j.automatica.2010.02.004.

Batista, P., Silvestre, C., Oliveira, P., Cardeira, B., 2010b. Low-cost attitude and heading reference system: Filter design and experimental evaluation, in: Proceedings of ICRA 2010, the IEEE International Conference on Robotics and Automation, Anchorage, Alaska, USA. pp. 2624–2629. DOI: 10.1109/ROBOT.2010.5509537.

Bay, H., Tuytelaars, T., Van Gool, L., 2006. Surf: Speeded up robust features, in: Proceedings of ECCV 2006, the 9th European Conference on Computer Vision, Graz, Austria. pp. 401–417. DOI: 10.1007/11744023_32.

Bhattacharya, P., Gavrilova, M.L., 2008. Roadmap-based path planning-using the voronoi diagram for a clearance-based shortest path. IEEE Robotics & Automation Magazine 15, 58–66. DOI: 10.1109/MRA.2008.921540.

Bing, H., Gang, L., Jiang, G., Hong, W., Nan, N., Yan, L., 2011. A route planning method based on improved artificial potential field algorithm, in: Proceedings of ICCSN 2011, the 3rd IEEE International Conference on Communication Software and Networks, IEEE, Xi'an, China. pp. 550–554. DOI: 10.1109/ICCSN.2011.6014330.

Biswas, J., Veloso, M., 2012. Depth camera based indoor mobile robot localization and navigation, in: Proceedings of ICRA 2012, the IEEE International Conference on Robotics and Automation, IEEE, Saint Paul, USA. pp. 1697–1702. DOI: 10.1109/ICRA.2012.6224766.

Bogh, S., Schou, C., Rühr, T., Kogan, Y., Dömel, A., Brucker, M., Eberst, C., Tornese, R., Sprunk, C., Tipaldi, G.D., 2014. Integration and assessment of multiple mobile manipulators in a real-world industrial production facility, in: Proceedings of ISR/Robotik 2014, the 41st International Symposium on Robotics, VDE, Munich, Germany. pp. 1–8. URL: http://ieeexplore.ieee.org/abstract/document/6840145/.

Bonin-Font, F., Ortiz, A., Oliver, G., 2008. Visual navigation for mobile robots: A survey. Journal of intelligent and robotic systems 53, 263–296. DOI: 10.1007/s10846-008-9235-4.

Brown, R., 1973. Integrated navigation systems and kalman filtering: a perspective. Navigation 19, 355–362. URL: http://www.ion.org/publications/abstract.cfm?articleID=100992.

Brown, R., Hwang, P., 1997. Introduction to Random Signals and Applied Kalman Filtering. 3rd ed.

Budiharto, W., Santoso, A., Purwanto, D., Jazidie, A., 2011. Multiple moving obstacles avoidance of service robot using stereo vision. TELKOMNIKA (Telecommunication Computing Electronics and Control) 9, 433–444. DOI: 10.12928/telkomnika.v9i3.733.

Burgard, W., Cremers, A.B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., Thrun, S., 1999. Experiences with an interactive museum tour-guide robot. Artificial intelligence 114, 3–55. DOI: 10.1016/S0004-3702(99)00070-3.

Calusdian, J., Yun, X., Bachmann, E., 2011. Adaptive-gain complementary filter of inertial and magnetic data for orientation estimation, in: Proceedings of ICRA 2011, the IEEE International Conference on Robotics and Automation, pp. 1916–1922. DOI: 10.1109/ICRA.2011.5979957.

Cao, M., Hashimoto, H., 2013. Specific person recognition and tracking of mobile robot with kinect 3d sensor, in: Proceedings of IECON 2013, the 39th Annual Conference of the IEEE Industrial Electronics Society, IEEE, Vienna, Austria. pp. 8323–8328.

Carreira, F., Calado, J., Cardeira, C., Oliveira, P., 2013. Enhanced pca-based localization using depth maps with missing data, in: Proceedings of Robotica 2013, the 13th

International Conference on Autonomous Robot Systems and Competitions, IEEE, Lisboa, Portugal. pp. 56–63. DOI: 978-989-97531-2-9.

Carreira, F., Calado, J., Cardeira, C., Oliveira, P., 2015a. Enhanced pca-based localization using depth maps with missing data. Journal of Intelligent & Robotic Systems 77, 341–360. DOI: 10.1007/s10846-013-0013-6.

Carreira, F., Calado, J.M.F., Cardeira, C., 2011. A mobile robot navigation planning in a human populated environment, in: Proceedings of Robotica 2011, the 11th International Conference on Autonomous Robot Systems and Competitions, Lisboa, Portugal. pp. 15–20. URL: http://www.dem.ist.utl.pt/~cardeira/papers/Robotica2011Proceedings_Fcarreira.pdf.

Carreira, F., Calado, J.M.F., Cardeira, C., Oliveira, P., 2015b. Complementary filter design with three frequency bands: Robot attitude estimation, in: Proceedings of ICARSC 2015, the 15th IEEE International Conference on Autonomous Robot Systems and Competitions, IEEE, Vila Real, Portugal. pp. 168–173. DOI: 10.1109/ICARSC.2015.33.

Carreira, F., Canas, T., Silva, A., Cardeira, C., 2006. i-merc: A mobile robot to deliver meals inside health services, in: Proceedings of RAM 2006, the IEEE Conference on Robotics, Automation and Mechatronics, pp. 1–8. DOI: 10.1109/RAMECH.2006.252651.

Carreira, F., Christo, C., Valério, D., Ramalho, M., Cardeira, C., Calado, J.M.F., Oliveira, P., 2012a. 2d pca-based localization for mobile robots in unstructured environments, in: Proceedings of IROS 2012, the IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Vilamoura, Portugal. pp. 3767–3868. DOI: 10.1109/IROS.2012.6386272.

Carreira, F., Christo, C., Valério, D., Ramalho, M., Cardeira, C., Calado, J.M.F., Oliveira, P., 2012b. Experimental Validation of a PCA-Based Localization System for Mobile Robots in Unstructured Environments. Technical Report. IDMEC/CSI. URL: http://www1.dem.ist.utl.pt/cardeira/Mesh/Ca12b.pdf.

Carreira, F., Christo, C., Valério, D., Ramalho, M., Cardeira, C., Calado, J.M.F., Oliveira, P., 2012c. Experimental validation of a PCA-based localization system for mobile robots in unstructured environments, in: Proceedings of Robotica 2012, the 12th International Conference on Autonomous Robot Systems and Competitions, Guimarães, Portugal. pp. 69–74.

Cavallo, A., Cirillo, A., Cirillo, P., De Maria, G., Falco, P., Natale, C., Pirozzi, S., 2014. Experimental comparison of sensor fusion algorithms for attitude estimation, in: Proceedings of IFAC 2014, the 19th World Conference of the International Federation of Automatic Control, Cape Town, South Africa. pp. 7585–7591. DOI: 10.3182/20140824-6-za-1003.01173.

Chen, J., Dawson, D.M., Salah, M., Burg, T., 2007. Cooperative control of multiple vehicles with limited sensing. International Journal of Adaptive Control and Signal Processing 21, 115. DOI: 10.1002/acs.921.

Choi, B.S., Lee, J.W., Lee, J.J., Park, K.T., 2011. A hierarchical algorithm for indoor mobile robot localization using rfid sensor fusion. IEEE Transactions on Industrial Electronics 58, 2226–2235. DOI: 10.1109/TIE.2011.2109330.

Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G.A., Burgard, W., Kavraki, L.E., Thrun, S., 2005. Principles of Robot Motion: Theory, Algorithms, and Implementations. MIT Press, Cambridge, MA.

Colon, E., De Cubber, G., Ping, H., Habumuremyi, J., Sahli, H., Baudoin, Y., 2007. Integrated robotic systems for humanitarian demining. International Journal of Advanced Robotic Systems 4, 219–228. DOI: 10.5772/5694.

Conner, D.C., Rizzi, A.A., Choset, H., 2003. Composition of local potential functions for global robot control and navigation, in: Proceddings of IROS 2003, the IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Las Vegas, USA. pp. 3546–3551. DOI: 10.1109/IROS.2003.1249705.

Corke, P.I., 2011. Robotics, Vision & Control: Fundamental Algorithms in Matlab. Springer.

Correa, D.S.O., Sciotti, D.F., Prado, M.G., Sales, D.O., Wolf, D.F., Osório, F.S., 2012. Mobile robots navigation in indoor environments using kinect sensor, in: Proceedings of CBSEC

2012, the 2nd Brazilian Conference on Critical Embedded Systems, IEEE, Campinas, Brazil. pp. 36–41. DOI: 10.1109/CBSEC.2012.18.

Craig, K., 2009. Sensor fusion - it's hot! : Complementary filtering to meet demanding performance requirements. Mechatronics in Design , 16.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection, in: Proceedings of CVPR 2005, the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, San Diego, California, USA. pp. 886–893. DOI: 10.1109/CVPR.2005.177.

Dellaert, F., Burgard, W., Fox, D., Thrun, S., 1999a. Using the condensation algorithm for robust, vision-based mobile robot localization, in: Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, Fort Collins, Colorado, USA. pp. 1–7. DOI: 10.1109/CVPR.1999.784976.

Dellaert, F., Fox, D., Burgard, W., Thrun, S., 1999b. Monte carlo localization for mobile robots, in: Proceedings ICRA 1999, the IEEE International Conference on Robotics and Automation, Detroit, USA. pp. 1322–1328 vol.2. DOI: 10.1109/ROBOT.1999.772544.

DeSouza, G.N., Kak, A.C., 2002. Vision for mobile robot navigation: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 237–267. DOI: 10.1109/34.982903.

Dias, F., Schafer, H., Natal, L., Cardeira, C., 2015. Mobile robot localisation for indoor environments based on ceiling pattern recognition, in: Proceedings of ICARSC 2015, the IEEE International Conference on Autonomous Robot Systems and Competitions, IEEE, Vila Real, Portugal. pp. 65–70. DOI: 10.1109/ICARSC.2015.32.

Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numerische mathematik 1, 269–271. DOI: 10.1007/BF01386390.

Dimarogonas, D.V., Loizou, S.G., Kyriakopoulos, K.J., Zavlanos, M.M., 2006. A feedback stabilization and collision avoidance scheme for multiple independent non-point agents. Automatica 42, 229–243. DOI: 10.1016/j.automatica.2005.09.019.

Do, H.N., Jadaliha, M., Choi, J., Lim, C.Y., 2015. Feature selection for position estimation using an omnidirectional camera. Image and Vision Computing 39, 1–9. DOI: 10.1016/j.imavis.2015.04.002.

Dong, H., Li, W., Zhu, J., Duan, S., 2010. The path planning for mobile robot based on voronoi diagram, in: Proceedings of ICINIS 2010, the 3rd International Conference on Intelligent Networks and Intelligent Systems, IEEE, Shenyang, China. pp. 446–449. DOI: 10.1109/ICINIS.2010.105.

Durrant-Whyte, H., Bailey, T., 2006. Simultaneous localization and mapping: part i. IEEE Robotics Automation Magazine 13, 99 –110. DOI: 10.1109/MRA.2006.1638022.

Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., Burgard, W., 2012. An evaluation of the rgb-d slam system, in: Proceedings of ICRA 2012, the IEEE International Conference on Robotics and Automation, IEEE, Saint Paul, Minnesota, USA. pp. 1691–1696. DOI: 10.1109/ICRA.2012.6225199.

Endres, F., Hess, J., Sturm, J., Cremers, D., Burgard, W., 2014. 3-d mapping with an rgb-d camera, in: IEEE Transactions on Robotics, IEEE. pp. 177–187. DOI: 10.1109/TRO.2013.2279412.

Euston, M., Coote, P., Mahony, R., Kim, J., Hamel, T., 2008. A complementary filter for attitude estimation of a fixed-wing uav, in: Proceedings of IROS 2008, the IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Nice. pp. 340–345. DOI: 10.1109/IROS.2008.4650766.

Evers, V., Menezes, N., Merino, L., Gavrila, D., Nabais, F., Pantic, M., Alvito, P., 2014. The development and real-world application of frog, the fun robotic outdoor guide, in: Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing, ACM, Baltimore, Maryland, USA. pp. 281–284. DOI: 10.1145/2556420.2557638.

Faragher, R., 2012. Understanding the basis of the kalman filter via a simple and intuitive derivation. IEEE Signal processing magazine 29, 128–132. DOI: 10.1109/MSP.2012.2203621.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24, 381–395. DOI: 10.1145/358669.358692.

Fox, D., Burgard, W., Dellaert, F., Thrun, S., 1999. Monte carlo localization: Efficient position estimation for mobile robots, in: Proceedings of AAAI'99, the 16th National Conference on Artificial Intelligence, American Association for Artificial Intelligence, Menlo Park, CA, USA. pp. 343–349. URL: http://dl.acm.org/citation.cfm?id=315149.315322.

Ganganath, N., Leung, H., 2012. Mobile robot localization using odometry and kinect sensor, in: Proceedings of ESPA 2012, the 1st IEEE International Conference on Emerging Signal Processing Applications, IEEE, Las Vegas, USA. pp. 91–94. DOI: 10.1109/ESPA.2012.6152453.

Garrido, S., Moreno, L., 2015. Mobile robot path planning using voronoi diagram and fast marching. Robotics, Automation, and Control in Industrial and Service Settings , 92–108DOI: 10.4018/978-1-4666-8693-9.

Gaspar, T., Oliveira, P., Silvestre, C., 2011. Uav-based marine mammals positioning and tracking system, in: Proceedings of ICMA 2011, the IEEE International Conference on Mechatronics and Automation, IEEE, Beijing, China. pp. 1050–1055. DOI: 10.1109/ICMA.2011.5985805.

Ge, S.S., Cui, Y.J., 2002. Dynamic motion planning for mobile robots using potential field method. Autonomous Robots 13, 207–222. DOI: 10.1023/A:1020564024509.

Gil, A., Mozos, O., Ballesta, M., Reinoso, O., 2010. A comparative evaluation of interest point detectors and local descriptors for visual slam. Machine Vision and Applications 21, 905–920. DOI: 10.1007/s00138-009-0195-x.

Gonçalves, D., Arsenio, A., 2015. Human-driven multi-robot design process for social interactions with children on complex environments, in: Proceedings of ICARA 2015, the International Conference on Automation, Robotics and Applications, IEEE, Queenstown, New Zealand. pp. 70–76. DOI: 10.1109/ICARA.2015.7081127.

Gutmann, J.S., Fox, D., 2002. An experimental comparison of localization methods continued, in: Proceedings of IROS 2002, the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland. pp. 454–459. DOI: 10.1109/IRDS.2002.1041432.

Habib, M., 2008. Development of robot and navigation techniques for humanitarian demining, in: Proceedings of INDIN 2008, the 6th IEEE International Conference on Industrial Informatics, Daejeon, South Korea. pp. 418–423. DOI: 10.1109/INDIN.2008.4618135.

Han, S.B., Kim, J.H., Myung, H., 2013. Landmark-based particle localization algorithm for mobile robots with a fish-eye vision system. IEEE/ASME Transactions on Mechatronics 18, 1745–1756. DOI: 10.1109/TMECH.2012.2213263.

Haralick, R.M., Shapiro, L.G., 1991. Computer and Robot Vision. volume 1. 1st ed., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Hart, P.E., Nilsson, N.J., Raphael, B., 1968. A formal basis for the heuristic determination of minimum cost paths. IEEE transactions on Systems Science and Cybernetics 4, 100–107. DOI: 10.1109/TSSC.1968.300136.

Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D., 2014. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments, in: Experimental robotics, Springer. pp. 477–491. DOI: 10.1007/978-3-642-28572-1_33.

Heras, D.B., Cabaleiro, J.C., Blanco, V., Costas, P., Rivera, F.F., 1997. Principal component analysis on vector computers, in: Palma, J.M.L.M., Dongarra, J. (Eds.), Proceding of VECPAR'96, the International Conference on Vector and Parallel Processing, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 416–428. DOI: 10.1007/3-540-62828-2_133.

Higgins, W., 1975. A comparison of complementary and kalman filtering. IEEE Transactions on Aerospace and Electronic Systems 11, 321–325. DOI: 10.1109/TAES.1975.308081.

Huang, A.S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., Roy, N., 2011. Visual odometry and mapping for autonomous flight using an rgb-d camera, in: Proceedings of ISRR 2011, the 15th International Symposium on Robotics Research, Flagstaff, Arizona, USA. pp. 1–16. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.352.6446.

Huang, W., Tsai, C., Lin, H., 2012. Mobile robot localization using ceiling landmarks and images captured from an rgb-d camera, in: Proceedings of AIM 2012, the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, IEEE, Kachsiung, Taiwan. pp. 855–860. DOI: 10.1109/AIM.2012.6265979.

Hwang, S.Y., Song, J.B., 2011. Monocular vision-based slam in indoor environment using corner, lamp, and door features from upward-looking camera. IEEE Transactions on Industrial Electronics 58, 4804–4812. DOI: 10.1109/TIE.2011.2109333.

Iizuka, S., Nakamura, T., Suzuki, S., 2014. Robot navigation in dynamic environment using navigation function apf with slam, in: Proceedings of Mecatronics 2014, the 10th France-Japan/8th Europe-Asia Congress on Mecatronics, IEEE, Tokyo, Japan. pp. 89–92. DOI: 10.1109/MECATRONICS.2014.7018605.

Ilias, B., Nagarajan, R., Murugappan, M., Helmy, K., Awang Omar, A.S., Abdul Rahman, M.A., 2014. Hospital nurse following robot: hardware development and sensor integration. International Journal of Medical Engineering and Informatics 6, 1–13. DOI: 10.1504/IJMEI.2014.058521.

Jafari, O., Mitzel, D., Leibe, B., 2014. Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras, in: Proceedings of ICRA 2014, the IEEE International Conference on Robotics and Automation, Hong Kong. pp. 5636–5643. DOI: 10.1109/ICRA.2014.6907688.

Jeong, W., Lee, K.M., 2005. CV-SLAM: a new ceiling vision-based SLAM technique, in: Proceedings of IROS 2005, the IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Alberta, Canada. pp. 3195–3200. DOI: 10.1109/IROS.2005.1545443.

Jo, S., Choi, H., Kim, E., 2012. Ceiling vision based slam approach using sensor fusion of sonar sensor and monocular camera, in: Proceedings of ICCAS 2012, the 12th International Conference on Control, Automation and Systems, IEEE, Kuala Lumpur, Malaysia. pp. 1461–1464.

Jogan, M., Artač, M., Skočaj, D., Leonardis, A., 2003. A framework for robust and incremental self-localization of a mobile robot, in: Proceedings of ICVS'03, the 3rd International Conference on Computer Vision Systems, Springer-Verlag, Graz, Austria. pp. 460–469. URL: http://dl.acm.org/citation.cfm?id=1765473.1765524, DOI: 10.1007/3-540-36592-3_44.

Jogan, M., Leonardis, A., Wildenauer, H., Bischof, H., 2002. Mobile robot localization under varying illumination, in: Proceedings of ICPR 2002, the 16th International Conference onPattern Recognition, IEEE, Quebec City, Quebec, Canada. pp. 741–744. DOI: 10.1109/ICPR.2002.1048409.

Jolliffe, I., 2002. Principal Component Analysis. Springer-Verlag. DOI: 10.1007/b98835.

Kalman, R., 1960. A new approach to linear filtering and prediction problems. Journal of Basic Engineering 82, 35–45. DOI: 10.1115/1.3662552.

Kalman, R.E., Bucy, R.S., 1961. New results in linear filtering and prediction theory. Journal of Fluids Engineering 83, 95–108. DOI: 10.1115/1.3658902.

Karlsson, R., Gustafsson, F., 2003. Particle filter for underwater terrain navigation, in: Proceedings of the 2003 IEEE Workshop on Statistical Signal Processing, St. Louis, MO, USA. pp. 526–529. DOI: 10.1109/SSP.2003.1289507.

Kelly, A., Nagy, B., Stager, D., Unnikrishnan, R., 2007. Field and service applications - an infrastructure-free automated guided vehicle based on computer vision - an effort to make an industrial robot vehicle that can operate without supporting infrastructure. IEEE Robotics Automation Magazine 14, 24–34. DOI: 10.1109/MRA.2007.901317.

Khatib, O., 1986. Real-time obstacle avoidance for manipulators and mobile robots. The international journal of robotics research 5, 90–98. DOI: 10.1177/027836498600500106.

Kim, J., Zhang, F., Egerstedt, M., 2010. A provably complete exploration strategy by constructing voronoi diagrams. Autonomous Robots 29, 367–380. DOI: 10.1007/s10514-010-9200-5.

Koditschek, D.E., Rimon, E., 1990. Robot navigation functions on manifolds with boundary. Advances in Applied Mathematics 11, 412–442. DOI: 10.1016/0196-8858(90)90017-S.

Koenig, S., Likhachev, M., 2002. D* lite, in: Proceedings of the AAAI'02, the 18th National Conference on Artificial Intelligence, AAAI. pp. 476–483. URL: http://dl.acm.org/citation.cfm?id=777167.

Kubelka, V., Reinstein, M., 2012. Complementary filtering approach to orientation estimation using inertial sensors only, in: Proceedings of ICRA 2012, the IEEE International Conference on Robotics and Automation, St. Paul, MN. pp. 599–605. DOI: 10.1109/ICRA.2012.6224564.

Kuderer, M., Sprunk, C., Kretzschmar, H., Burgard, W., 2014. Online generation of homotopically distinct navigation paths, in: Proceedings of ICRA 2014, the IEEE International Conference on Robotics and Automation, IEEE, Hong Kong. pp. 6462–6467. DOI: 10.1109/ICRA.2014.6907813.

Leonard, J.J., Durrant-Whyte, H.F., 1991. Simultaneous map building and localization for an autonomous mobile robot, in: Proceedings of IROS'91, the IEEE/RSJ International Workshop on Intelligent Robots and Systems, IEEE, Osaka, Japan. pp. 1442–1447. DOI: 10.1109/IROS.1991.174711.

Levinson, J., Thrun, S., 2010. Robust vehicle localization in urban environments using probabilistic maps, in: Proceedings of ICRA 2010, the IEEE International Conference on Robotics and Automation, IEEE, Anchorage, Alaska, USA. pp. 4372–4378. DOI: 10.1109/ROBOT.2010.5509700.

Li, G., Yamashita, A., Asama, H., Tamura, Y., 2012a. An efficient improved artificial potential field based regression search method for robot path planning, in: Proceedings of ICMA 2012, the IEEE International Conference on Mechatronics and Automation, IEEE, Chengdu, China. pp. 1227–1232. DOI: 10.1109/ICMA.2012.6283526.

Li, T., Su, Y.T., Liu, S.H., Hu, J.J., Chen, C.C., 2012b. Dynamic balance control for biped robot walking using sensor fusion, kalman filter, and fuzzy logic. IEEE Transactions on Industrial Electronics 59, 4394–4408. DOI: 10.1109/TIE.2011.2175671.

Li, X.R., Jilkov, V.P., 2003. Survey of maneuvering target tracking. part i. dynamic models. IEEE Transactions on aerospace and electronic systems 39, 1333–1364. DOI: 10.1109/TAES.2003.1261132.

Loevsky, I., Shimshoni, I., 2010. Reliable and efficient landmark-based localization for mobile robots. Robotics and Autonomous Systems 58, 520–528. DOI: 10.1016/j.robot.2010.01.006.

Loizou, S.G., Kyriakopoulos, K.J., 2003. Closed loop navigation for multiple non-holonomic vehicles, in: Proceedings of ICRA 2003, the IEEE International Conference on Robotics and Automation, IEEE, Taipei, Taiwan. pp. 4240–4245. DOI: 10.1109/ROBOT.2003.1242255.

Loizou, S.G., Kyriakopoulos, K.J., 2008. Navigation of multiple kinematically constrained robots. IEEE Transactions on Robotics 24, 221–231. DOI: 10.1109/TRO.2007.912092.

Loizu, S., Dimarogonas, D.V., Kyriakopoulos, K.J., 2004. Decentralized feedback stabilization of multiple nonholonomic agents, in: Proceedings of ICRA'04, the IEEE International Conference on Robotics and Automation, IEEE, Barcelona, Spain. pp. 3012–3017. DOI: 10.1109/ROBOT.2004.1307519.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94.

Lu, X., Li, J., Liu, W., 2014. A study of complementary filter algorithm for four-rotor helicopters attitude control system. Sensors & Transducers Journal 174, 298–306. URL: http://www.sensorsportal.com/HTML/DIGEST/P_2217.htm.

Maeda, S., Kuno, Y., Shirai, Y., 1997. Active navigation vision based on eigenspace analysis, in: Proceedings of IROS 1997, the IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Grenoble, France. pp. 1018–1023. DOI: 10.1109/IROS.1997.655133.

Malagón-Borja, L., Fuentes, O., 2009. Object detection using image reconstruction with pca. Image and Vision Computing 27, 2–9. DOI: 10.1016/j.imavis.2007.03.004.

Maohai, L., Han, W., Lining, S., Zesu, C., 2013. Robust omnidirectional mobile robot topological navigation system using omnidirectional vision. Engineering Applications of Artificial Intelligence 26, 1942–1952. DOI: 10.1016/j.engappai.2013.05.010.

Mitsantisuk, C., Ohishi, K., Katsura, S., 2012. Estimation of action/reaction forces for the bilateral control using kalman filter. IEEE Transactions on Industrial Electronics 59, 4383–4393. DOI: 10.1109/TIE.2011.2173092.

Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., 2002. Fastslam: A factored solution to the simultaneous localization and mapping problem, in: Proceedings of the AAAI-02, the 18th National Conference on Artificial Intelligence, AAAI, Edmonton, Alberta, Canada. pp. 593–598. URL: http://dl.acm.org/citation.cfm?id=777092.777184.

Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., 2003. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, in: Proceedings of IJCAI-03, the 18th International Joint Conference on Artificial Intelligence, IJCAI, Acapulco, Mexico. pp. 1151–1156. URL: http://dl.acm.org/citation.cfm?id=1630824.

Moutarlier, P., Chatila, R., 1990. An experimental system for incremental environment modelling by an autonomous mobile robot, in: The First International Symposium on Experimental Robotics I, Springer-Verlag, London, UK. pp. 327–346. DOI: 10.1007/BFb0042528.

Murray, D., Little, J.J., 2000. Using real-time stereo vision for mobile robot navigation. Autonomous Robots 8, 161–171. DOI: 10.1023/A:1008987612352.

Nakazato, K., Touma, Y., Hagiwara, H., Asami, K., Komori, M., 2015. Fpga-based stereo vision system using census transform for autonomous mobile robot, in: Proceedings of ICIEV 2015, the 4th International Conference on Informatics, Electronics & Vision, IEEE, Fukuoka, Japan. pp. 1–4. DOI: 10.1109/ICIEV.2015.7334017.

Nguyen, T.H., Kim, D.H., Lee, C.H., Kim, H.K., Kim, S.B., 2016. Mobile robot localization and path planning in a picking robot system using kinect camera in partially known environment, in: Proceedings of AETA 2016, the International Conference on Advanced Engineering Theory and Applications, Springer, Busan, Korea. pp. 686–701. DOI: 10.1007/978-3-319-50904-4_70.

Nourbakhsh, I., Powers, R., Birchfield, S., 1995. Dervish an office-navigating robot. AI magazine 16, 53. DOI: 10.1609/aimag.v16i2.1133.

Ok, K., Ansari, S., Gallagher, B., Sica, W., Dellaert, F., Stilman, M., 2013. Path planning with uncertainty: Voronoi uncertainty fields, in: Proceedings of ICRA 2013, the IEEE International Conference on Robotics and Automation, IEEE, Karlsruhe, Germany. pp. 4596–4601. DOI: 10.1109/ICRA.2013.6631230.

Oliveira, P., 2007. MMAE terrain reference navigation for underwater vehicles using PCA. International Journal of Control 80, 1008–1017. DOI: 10.1080/00207170701242515.

Oliveira, P., Gomes, L., 2010. Interpolation of signals with missing data using principal component analysis. Multidimensional Systems and Signal Processing 21, 25–43. DOI: 10.1007/s11045-009-0086-3.

Park, J.J., Kuipers, B., 2011. A smooth control law for graceful motion of differential wheeled mobile robots in 2d environment, in: Proceedings of ICRA 2011, the IEEE International Conference on Robotics and Automation, IEEE, Shanghai, China. pp. 4896–4902. DOI: 10.1109/ICRA.2011.5980167.

Pascoal, A., Kaminer, I., Oliveira, P., 2000. Navigation system design using time-varying complementary filters. IEEE Transactions on Aerospace and Electronic Systems 36, 1099–1114. DOI: 10.1109/7.892661.

Payá, L., Reinoso, O., Gil, A., Pedrero, J., Ballesta, M., 2007. Appearance-based multi-robot following routes using incremental pca, in: Knowledge-Based Intelligent Information and Engineering Systems, Springer. pp. 1170–1178. DOI: 10.1007/978-3-540-74827-4_146.

Pham, D., Dahia, K., Musso, C., 2003. A kalman-particle kernel filter and its application to terrain navigation, in: Proceedings of the 6th International Conference of Information Fusion, Queensland, Australia. pp. 1172–1179. DOI: 10.1109/ICIF.2003.177370.

Pradhan, N., Burg, T., Birchfield, S., 2011. Robot crowd navigation using predictive position fields in the potential function framework, in: Proceedings of ACC 2011, the American Control Conference, IEEE, San Francisco, CA, USA. pp. 4628–4633. DOI: 10.1109/ACC.2011.5991384.

Pradhan, N., Burg, T., Birchfield, S., Hasirci, U., 2013. Indoor navigation for mobile robots using predictive fields, in: Proceedings of ACC 2013, the American Control Conference, IEEE, Washington, DC, USA. pp. 3237–3241. DOI: 10.1109/ACC.2013.6580330.

Pustianu, A.I., Serbencu, A., Cernega, D.C., 2011. Mobile robot control using face recognition algorithms, in: Proceedings of ICSTCC 2011, the 15th International Conference on System Theory, Control, and Computing, IEEE, Sinaia, Romania. pp. 1–6. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6085692&tag=1.

Raghavan, A., Ananthapadmanaban, H., Sivamurugan, M., Ravindran, B., 2010. Accurate mobile robot localization in indoor environments using bluetooth, in: Proceedings of ICRA 2010, the IEEE International Conference on Robotics and Automation, Anchorage, Alaska, USA. pp. 4391–4396. DOI: 10.1109/ROBOT.2010.5509232.

Rimon, E., Koditschek, D.E., 1992. Exact robot navigation using artificial potential functions. IEEE Transactions on robotics and automation 8, 501–518. DOI: 10.1109/70.163777.

Rodrigues, J., Cardeira, C., Carreira, F., Calado, J.M.F., Oliveira, 2013a. Experimental validation of a visual odometry system for indoor unstructured environments, in: Proceedings of ICAR 2013, the 16th International Conference on Advanced Robotics, IEEE, Montevideo, Uruguay. pp. 1–6. DOI: 10.1109/ICAR.2013.6766496.

Rodrigues, J., Carreira, F., Calado, J.M.F., Cardeira, C., Oliveira, P., 2013b. A bayesian grid method pca-based for mobile robots localization in unstructured environments, in: Proceedings of ICAR 2013, the 16th International Conference on Advanced Robotics, IEEE, Montevideo, Uruguay. pp. 1–6. DOI: 10.1109/ICAR.2013.6766487.

Rowekamper, J., Sprunk, C., Tipaldi, G., Stachniss, C., Pfaff, P., Burgard, W., Rowekamper, J., Sprunk, C., Tipaldi, G., Stachniss, C., Pfaff, P., Burgard, W., 2012. On the position accuracy of mobile robot localization based on particle filters combined with scan matching, in: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, Vilamoura, Portugal. pp. 3158–3164. DOI: 10.1109/IROS.2012.6385988.

Rusdinar, A., Kim, J., Lee, J., Kim, S., 2012. Implementation of real-time positioning system using extended kalman filter and artificial landmark on ceiling. Journal of Mechanical Science and Technology 26, 949–958. DOI: 10.1007/s12206-011-1251-9.

Samejima, I., Nihei, Y., Hatao, N., Kagami, S., Mizoguchi, H., Takemura, H., Osaki, A., 2015. Building environmental maps of human activity for a mobile service robot at the "miraikan" museum, in: Proceedings of FSR13, the 9th International Conference on Field and Service Robotics, Springer, Brisbane, Australia. pp. 409–422. DOI: 10.1007/978-3-319-07488-7_28.

Scaramuzza, D., Fraundorfer, F., Siegwart, R., 2009. Real-time monocular visual odometry for on-road vehicles with 1-point ransac, in: Proceedings of ICRA 2009, the IEEE International Conference on Robotics and Automation, IEEE, Kobe, Japan. pp. 4293–4299. DOI: 10.1109/ROBOT.2009.5152255.

Shamma, J.S., 2012. An overview of lpv systems, in: Control of linear parameter varying systems with applications. Springer, pp. 3–26. DOI: 10.1007/978-1-4614-1833-7_1.

Shih, C.L., Ku, Y.T., 2016. Image-based mobile robot guidance system by using artificial ceiling landmarks. Journal of Computer and Communications 4, 1. DOI: 10.4236/jcc.2016.411001.

Siegwart, R., Nourbakhsh, I., 2004. Introduction to Autonomous Mobile Robots. A Bradford book, MIT Press. URL: http://mitpress.mit.edu/books/introduction-autonomous-mobile-robots.

Simmons, R., Koenig, S., 1995. Probabilistic robot navigation in partially observable environments, in: Proceedings of IJCAI 95, the14th International Joint Conference on Artificial Intelligence, Montreal, Quebec. pp. 1080–1087. URL: http://dblp.uni-trier.de/db/conf/ijcai/ijcai95.html#SimmonsK95.

Smith, R.C., Cheeseman, P., 1986. On the representation and estimation of spatial uncertainty. The international journal of Robotics Research 5, 56–68. DOI: 10.1177/027836498600500404.

Song, H., Shin, V., Jeon, M., 2012. Mobile node localization using fusion prediction-based interacting multiple model in cricket sensor network. IEEE Transactions on Industrial Electronics 59, 4349–4359. DOI: 10.1109/TIE.2011.2151821.

Sprunk, C., Lau, B., Pfaff, P., Burgard, W., 2016. An accurate and efficient navigation system for omnidirectional robots in industrial environments. Autonomous Robots , 1–21DOI: 10.1007/s10514-016-9557-1.

Steinbauer, G., Bischof, H., 2005. Illumination insensitive robot self-localization using panoramic eigenspaces, in: Proceedings of RoboCup 2004: Robot Soccer World Cup VIII, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 84–96. DOI: 10.1007/978-3-540-32256-6_7.

Stentz, A., 1994. Optimal and efficient path planning for partially-known environments, in: Proceedings of ICRA 1994, the IEEE International Conference on Robotics and Automation, IEEE, San Diego, USA. pp. 3310–3317. DOI: 10.1109/ROBOT.1994.351061.

Stowers, J., Hayes, M., Bainbridge-Smith, A., 2011. Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor, in: Proceedings of ICM 2011, the IEEE International Conference on Mechatronics, IEEE, Istanbul, Turkey. pp. 358–362. DOI: 10.1109/ICMECH.2011.5971311.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D., 2012. A benchmark for the evaluation of rgb-d slam systems, in: Proceedings of IROS 2012, the IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Vilamoura, Portugal. pp. 573–580. DOI: 10.1109/IROS.2012.6385773.

Suh, Y.S., 2006. Attitude estimation by multiple-mode kalman filters. IEEE Transactions on Industrial Electronics 53, 1386–1389. DOI: 10.1109/TIE.2006.878292.

Surmann, H., Nüchter, A., Hertzberg, J., 2003. An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments. Robotics and Autonomous Systems 45, 181–198. DOI: 10.1016/j.robot.2003.09.004.

Tanner, H.G., Kyriakopoulos, K.J., 2000. Nonholonomic motion planning for mobile manipulators, in: Proceedings of ICRA 2000, the IEEE International Conference on

Robotics and Automation, IEEE, San Francisco, CA, USA. pp. 1233–1238. DOI: 10.1109/ROBOT.2000.844767.

Tanner, H.G., Loizou, S.G., Kyriakopoulos, K.J., 2001. Nonholonomic stabilization with collision avoidance for mobile robots, in: Proceedings of IROS 2001, the IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Maui, USA. pp. 1220–1225. DOI: 10.1109/IROS.2001.977149.

Teslić, L., Škrjanc, I., Klančar, G., 2011. Ekf-based localization of a wheeled mobile robot in structured environments. Journal of Intelligent & Robotic Systems 62, 187–203. DOI: 10.1007/s10846-010-9441-8.

Theodoridis, T., Hu, H., McDonald-Maier, K., Gu, D., 2013. Kinect enabled monte carlo localisation for a robotic wheelchair, in: Frontiers of Intelligent Autonomous Systems. Springer. volume 466, pp. 17–27. DOI: 10.1007/978-3-642-35485-4_2.

Thrun, S., Burgard, W., Fox, D., 2005. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press.

Thrun, S., Fox, D., Burgard, W., Dellaert, F., 2001. Robust monte carlo localization for mobile robots. Artificial Intelligence 128, 99–141. DOI: 10.1016/S0004-3702(01)00069-8.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., et al., 2006. Stanley: The robot that won the darpa grand challenge. Journal of field Robotics 23, 661–692. DOI: 10.1002/rob.20147.

Wang, Y., Li, N., Liu, M., 2014. Design and implementation of an ahrs based on mems sensors and complementary filtering. Advances in Mechanical Engineering 3, 1–11. DOI: 10.1155/2014/214726.

Wein, R., Van Den Berg, J.P., Halperin, D., 2005. The visibility-voronoi complex and its applications, in: Proceedings of SCG 2005, the twenty-first annual symposium on Computational geometry, ACM. pp. 63–72. DOI: 10.1145/1064092.1064104.

Weingarten, J.W., Gruener, G., Siegwart, R., 2004. A state-of-the-art 3d sensor for robot navigation, in: Proceedings of IROS 2004, the IEEE/RSJ International Conference on

Intelligent Robots and Systems, IEEE, Sendai, Japan. pp. 2155–2160. DOI: 10.1109/IROS.2004.1389728.

Widyotriatmo, A., Hong, K.S., 2011. Navigation function-based control of multiple wheeled vehicles. IEEE Transactions on Industrial Electronics 58, 1896–1906. DOI: 10.1109/TIE.2010.2051394.

Yang, D.H., Hong, S.K., 2007. A roadmap construction algorithm for mobile robot path planning using skeleton maps. Advanced Robotics 21, 51–63. DOI: 10.1163/156855307779293724.

Yuan, W., Li, Z., Su, C.Y., 2016. Rgb-d sensor-based visual slam for localization and navigation of indoor mobile robot, in: Proceedings of ICARM 2016, International Conference on Advanced Robotics and Mechatronics, IEEE, Macau, China. pp. 82–87. DOI: 10.1109/ICARM.2016.7606899.

Zhang, X., Xian, B., Zhao, B., Zhang, Y., 2015. Autonomous flight control of a nano quadrotor helicopter in a gps-denied environment using on-board vision. IEEE Transactions on Industrial Electronics 62, 6392–6403. DOI: 10.1109/TIE.2015.2420036.

Zhou, B., Li, M., Qian, K., Dai, X., Fang, F., 2015. Long-range outdoor localization of a mobile robot using a binocular camera, in: Proceedings of IECON 2015, the 41st Annual Conference of the IEEE Industrial Electronics Society, IEEE, Yokohama, Japan. pp. 909–914. DOI: 10.1109/IECON.2015.7392215.