Universidad Zaragoza

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

Trabajo Fin de Máster

Máster en Ingeniería de Sistemas e Informática

Curso 2012/2013

Evaluación y comparación de sistemas de planificación de navegación de robots en entornos dinámicos

María Teresa Lorente Cebrián

Director: Luis Montano Gella

Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Diciembre 2013

Evaluación y comparación de sistemas de planificación de navegación de robots en entornos dinámicos

RESUMEN

Este trabajo aborda un análisis comparativo de diferentes técnicas de planificación de movimientos en entornos dinámicos. Se basa en trabajos anteriores, en los que se desarrollaron dos técnicas de planificación de movimientos para un robot que se mueve en un entorno dinámico. Se trata de técnicas de navegación robocéntricas en las que el modelo del entorno dinámico se basa en el espacio de velocidad-tiempo del robot, donde se representan tanto los objetos estáticos como dinámicos. La primera técnica trabaja sobre un espacio de velocidades bidimensional (velocidad lineal-velocidad angular). Explota la idea de identificar la mejor estrategia en función de la situación en la que se encuentra el robot, y que depende de la localización y velocidad relativas entre el robot y los obstáculos. La segunda técnica optimiza una función objetivo en el espacio de velocidad-tiempo para obtener comandos óptimos y trayectorias seguras, ponderando criterios de maximización de velocidad, seguridad (distancia a obstáculos) y suavidad de movimientos. Además, incorpora la técnica desarrollada en el primer trabajo como heurística para mejorar la toma de decisiones, dando lugar a *Strategies-Optimization*.

Para evaluar el rendimiento de la navegación con dichas técnicas se define una serie de métricas, que permiten seleccionar los mejores parámetros de optimización para cada tipo de escenario. Estas métricas evalúan y comparan los comportamientos en diferentes escenarios, lo que permite tener una evaluación completa de todas las técnicas.

Además, en aplicaciones reales los robots tienen que moverse en escenarios tanto de interior como de exterior. Sin embargo, para que los robot construyan un mapa del entorno, se localicen y navegen utilizan diferentes sensores, debido al tipo de información disponible (laser en interior y GPS en exterior) y a la incertidumbre de cada sensor en cada momento (pérdida o reducción de precisión del GPS, pocas características para construir el mapa). Esto provoca discontinuidades en localización o incluso pérdida de ello, lo que debe evitarse. En este trabajo se presenta una técnica de localización unificada para entornos de interior-exterior que permite una transición continua entre una zona de la que se dispone un mapa construido con los sensores láser a bordo del robot y una zona que utiliza el GPS para la localización del robot.

Evaluation and comparison of planning systems for robot navigation in dynamic environments

ABSTRACT

This work addresses a comparative analysis of different techniques for robot motion planning in dynamic environments. It is based on previous works, which developed two motion planning techniques based on the velocity-time space of the environment to map static and moving objects. The first technique works on a bi-dimensional velocity space (linear velocity-angular velocity). It exploits the idea of situation-based strategies identification, which depends on several relative locations and velocities between the robot and the obstacles. The second technique searches for the optimal commands obtained by optimizing an objective function in the velocity-time space, weighing criterion for maximizing velocity, safety (distance to obstacles) and smooth of movements. In addition, it integrates the first technique as heuristic to improve the decision making procedure and thus yielding the *Strategies-Optimization* technique.

Metrics for assessing the performance of the navigation with those techniques in different scenarios are defined, which allows obtain the best optimization parameters and have a complete comparison of the techniques.

Moreover, in real robotic applications the robots have to move within indoor and outdoor scenarios. However, to achieve mapping, localization and navigation tasks, the robots utilize different sensors, due to the type of available information (rangefinder infoor, GPS outdoor) and the sensor uncertainty at each moment (lost or reduced precision of GPS, few features to build a map and to localize). These issues lead to discontinuities in localization or even lost of it, which has to be avoided. This work presents a unified localization technique for indoor-outdoor environments that allows a seamless transition between a mapped zone using laser rangefinder on-board sensors and a GPS based localization zone.

Contents

1 Introduction **2**
- 1.1 Motivation . . . 2
- 1.2 Structure of the work . . . 3

2 Motion planning in dynamic environments **6**
- 2.1 Related work . . . 6
- 2.2 Modeling the environment . . . 7
  - 2.2.1 The *DOVTS* space . . . 7
  - 2.2.2 Dealing with multiple objects in the *DOVS* . . . 10
  - 2.2.3 Static objects in the *DOVS* . . . 10
- 2.3 Decision making strategies . . . 11
- 2.4 *Situation-Strategies* planning technique . . . 12
  - 2.4.1 Decision variables . . . 12
  - 2.4.2 Navigation planning . . . 13
- 2.5 *Cost-Optimization* planning technique . . . 17
  - 2.5.1 *Strategies-Optimization*: integration of *Situation-Strategies* and *Cost-Optimization* . . . 19

3 *Situation-Strategies* evaluation and comparison with *Cost-Optimization* and *Strategies-Optimization* **21**
- 3.1 Metrics for assessment . . . 21
  - 3.1.1 Simulation scenarios . . . 21

4 Seamless localization **25**
- 4.1 Related work . . . 25
- 4.2 Localization framework . . . 26
- 4.3 GPS-based orientation . . . 26
  - 4.3.1 Improvement of the GPS-based orientation method . . . 28
  - 4.3.2 Results . . . 30

5 Conclusions **33**

A Simulations **35**
- A.1 *Cluttered scenario* . . . 35

B Article: *Seamless indoor-outdoor robust localization for robots* **37**

List of figures **51**

List of tables **51**

Bibliography **53**

Chapter 1

Introduction

1.1 Motivation

In order to use robots in real environments, they need to adapt the motion to the obstacles appearing during the mission. A lot of work has been devoted over the years to developing motion planning and reactive navigation techniques. Many of these have focused on static scenarios, and the solutions can be considered to be sufficiently robust. However, in many of the missions in which robots have to operate together with people (i.e. museums, rescue, factories, etc.), robust planning in dynamic environments is compulsory.

[10] and [11] develop new planning techniques which confer the high degree of manoeuvrability needed in highly dynamic environments and that combine decision strategies and optimization techniques for safe navigation, working directly on the command space, in our case the velocity space. This work introduces the metrics defined in [15] to quantitatively asses the performance of the resulting motion, which in turn makes available well-founded criteria for selecting the control parameters of the planner. [15] include the results for [11] and its integration with [10].

One of the objectives of this work is to evaluate the technique developed in [10] with those metrics to obtain a comparison of the different planning techniques. Specifically, the tasks involved were:

- Use the same software to construct the static and moving object maps of the scenario as in [11].
- Evaluate the technique using the metrics in scenarios with more obstacles and moving in non-straight trajectories.
- Analize the results obtained and compare them with respect to the other planning techniques.

The final result of the analysis and the conclusions extend the article [15], sent to the

journal *Autonomous Robots*.

It is common in indoor robotics applications to assume a limited environment (a room, a building floor...) for localization purposes. This limitation is forced by the need of a finite map of the features in the zone to localize the robot. Due to the sparseness of the features needed to get a reliable localization system, the use of maps in outdoor scenarios is uncommon. Instead, outdoor applications usually utilize GPS based localization which avoids any limitation in the environment as it is accessible almost everywhere.

However, there are few systems that provide a continuous localization for both indoor and outdoor scenarios in such a way that the robot is not confined in a limited space. Main difficulties come from the fact that very different sensors (odometry, IMU, rangefinders, GPS...) are needed to get a good estimation. These problems become more obvious during the transitions as measurements are more imprecise.

Another objetive of this work is to contribute a unified framework for a seamless localization during navigation within different type of environments. The tasks related to this issue were:

- Identify the zones of an environment where the localization system has to operate, which have different characteristics.
- Implement a method to localize the robot integrating different sources of information, which weights them depending on the situation of the robot.
- Propose a technique to determine the orientation of the robot based on GPS measurements.
- Evaluate the technique with real experiments

The results are presented in [21] (attached in the appendix) in the *Fist Iberian Conference on Roboics*. In section 4.3.1 an extension of the method to estimate the orientation from GPS measurements is proposed, and the results are written as part of [20], for the *Int. Conference on Robotics and Automation*.

The work carried out is part of the projects *Intelligent Technologies for Autonomous Transportation of Goods Indoors and Outdoors ($TITAM_ie$)* and *Teams of robots for Logistics, Maintenance and Environment monitoring (TELOMAN)*.

1.2 Structure of the work

Following this introduction, chapter 2 deals with planning techniques for robot navigation in dynamic environments. It includes the modeling of the environment of the robot and the general procedure followed to take decisions for navigating. Then, it analyzes two planning techniques and their integration, as well as metrics for their evaluation and comparison. Next,

chapter 4 introduces a method to provide a seamless localization system between indoor and outdoor scenarios. The evaluation of the technique with real experiments is also shown. The conclusions about the work carried out are included in chapter 5.

The appendix section contains the article *Seamless indoor-outdoor robust localization for robots* ([21]).

Finally, a list of figures and tables, and the bibliography consulted are included.

Chapter 2

Motion planning in dynamic environments

2.1 Related work

Traditionally, reactive approaches have been combined with global planning techniques leading to so called iterative motion planning, e.g. ([3], [5], and [12]). These techniques calculate several steps ahead depending on the time available. The planner evaluates different branches in a tree within a horizon and works out a partial trajectory.

Modeling the dynamic environment is one of the major problems which it is necessary to solve. The dynamics of the environment has to be described efficiently so as to reduce the response time while complying with motion decisions in real time. In [1], expanded in [18], the VOS (Velocity Obstacle Space) deterministic model was proposed to model environments with moving obstacles.

The issue of safety for motion planning in dynamic environments was considered in [9], reasoning about the time horizon for it. The motion safety issue has been formally studied by the concept of Inevitable Collision States (ICS) developed in [4]. ICS provides the knowledge to ensure safe robot motion by computing the states in which the robot must not be located. [19] addressed the motion safety issue in the Velocity Obstacles framework using the ICS concept by experimenting with a time horizon appropriate for the velocity obstacle.

[16] proposed a model that considered the continuous velocity space as the control space, in which all velocity commands driving to trajectories in the free space could be chosen, but the trajectories are not explicitly computed. The dynamics of the environment is mapped in this space, so it is implicitly used when the velocity commands are finally selected for every sampling period. This information allows to deal with the safety issue, computing the safest strategies in a time horizon within the field of view of the robot sensors, computing new commands to react. Thus the computational time is reduced, and planning techniques can be applied in real time. The idea of computing motions to the robot passes before the

moving object or the robot slows down until the object passes is exploited as a strategy in the planning techniques developed by [10] and [11], which are addressed in this work.

Specifically, the work carried out in this Master's thesis focuses on the formalization and evaluation of the *Situation-Strategies* technique ([10]), and its comparison with respect to *Cost-Optimization* and *Situation-Optimization* techniques ([11]).

2.2 Modeling the environment

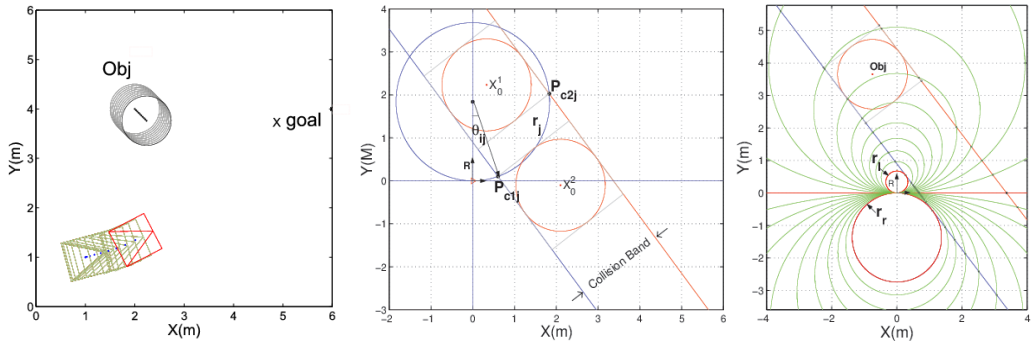
The basis of the planning techniques developed in [10] and [11] is the robot-environment model. This model is completely defined in [14]. It represents the three-dimensional velocity-time space in a reference attached to the robot in a robocentric approach. Intuitively, free commands and commands leading to collision are explicitly represented in this space. It is also possible to simultaneously reason about safety issues because the further evolution of the environment is directly mapped in the model. Some assumptions are made in order to simplify the modeling:

- The non holonomic robot plans long-term *straight and circular* paths, within the planning space-time horizon. But more complex trajectories can be followed when the navigation plan is executed by connecting short circular paths, maintaining a continuous curvature, as the model is re-computed every sampling period.
- Circular, clothoid, anti-clothoid and straight trajectories are achieved to execute the planned ones, complying with the kinodynamic constraints of the robot, and maintaining a continuous curvature.
- The objects move following straight paths with constant velocity. As the model is recomputed every sampling period, the motion of the objects could be considered as a sequence of short straight paths.
- The moving objects in the model are modeled as polygons such as wrapping squares or rectangles. For moving objects it is not necessary to use their exact shape, it is enough to model the space swept by them when they are moving.

Most of these constraints can be relaxed, and do not pose a limitation for the essence of the proposed method. For the sake of clarity [14] apply these restrictions to explain the model building procedure, which is outlined in the next section.

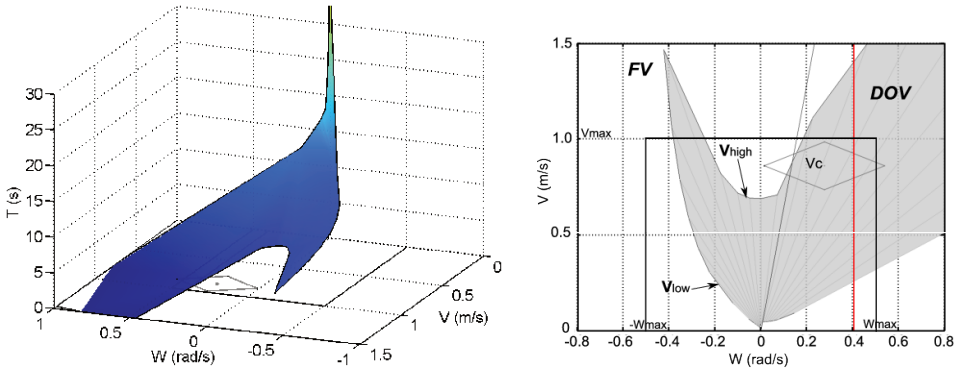
2.2.1 The *DOVTS* space

The modeling of the moving objects in the velocity space represents the maximum and minimum velocities that the robot must have to just collide when the object passes in front of the robot or the obstacle passes after the robot, respectively. They are computed for a set of paths from the current robot location. From this computation, free and prohibited velocities are represented in the velocity-time space. An implicit information about the proximity and



(a) (b) (c)

Figure 2.1: (a) Workspace. (b) Collision band, path r_j and collision points P_{c1j} and P_{c2j} in the robocentric (R) Configuration Space, (c) Multiple paths.



(a) (b)

Figure 2.2: (a) Velocity-time $DOVTS$ space. (b) Projection of $DOVTS$, $DOVS$, on the plane (v, w).

moving objects velocities is mapped. This allows to apply different strategies, depending on that information and the priorities of the mission, time or safety.

In this section the $DOVTS$ (Dynamic Object Velocity-Time) space defined in [14] is introduced. The maximum robot velocity following a path leading to collision with a moving object and the minimum velocity to escape before an object arrives and their associated times are computed. In Fig. 2.1a, the workspace with a robot and an object is represented. In Fig. 2.1b the corresponding Configuration space is shown, in which the robot is a point and the object is enlarged with the robot radius. In this figure the *band of collision* is also represented. This is the area swept by the object when moving in a straight line. The object is depicted in two locations ($\mathbf{x_o}^1$ and $\mathbf{x_o}^2$), which are related to the locations the robot would reach (points P_{c1j} and P_{c2j}) following a circular trajectory r_j. P_{c1j} is the robot position after the object has just passed it at time t_{1j} and P_{c2j} is the position which the robot has just passed before the object reaches it at time t_{2j}.

From the known object location $\mathbf{x_o} = (x_o, y_o, \phi_o)$ in the robot local reference and its velocity $\mathbf{v_o}$, the points of collision ($P_{c1j}(x_{1j}, y_{1j})$ and $P_{c2j}(x_{2j}, y_{2j})$) and the corresponding times t_{1j} and t_{2j} are calculated as in [14]. These calculations are extended to the whole space, considering a range of curvature radii between r_l and r_r (see Fig. 2.1c). $\mathbf{V}_{low}$ and $\mathbf{V}_{high}$ are respectively the set of maximum and minimum velocities computed for all the r_j paths:

$$\begin{aligned} \mathbf{V}_{low} &= \{\mathbf{v}_{1j}\},\ \mathbf{t}_{low} = \{\mathbf{t}_{1j}\}, \forall r_j \\ \mathbf{V}_{high} &= \{\mathbf{v}_{2j}\},\ \mathbf{t}_{high} = \{\mathbf{t}_{2j}\}, \forall r_j \end{aligned} \tag{2.1}$$

Definition 1. *DOVT* dynamic object is defined in the velocity space as,

$$DOVT = \{(v, w, t) | \mathbf{V}_{low} \leq (v, w) \leq \mathbf{V}_{high}, \mathbf{t}_{low} \leq t \leq \mathbf{t}_{high}\}$$

Definition 2. *FVT* is defined in the velocity space as,

$$FVT = \{(v, w, t) \notin DOVT\}$$

Definition 3. *DOVTS* space is defined as,

$$DOVTS = DOVT \cup FVT$$

Figure 2.2a represents *DOVTS*. The intuitive idea is that choosing velocity commands under *DOVT* might lead to collision if this command is applied from now on. In other words, safe velocity commands have to be chosen in the free velocity space outside the *DOVT* surface. Commands under the *DOVT* surface can be selected, but they can only be temporally applied. This time can be estimated from the t co-ordinate, the time to collision being the *distance* to the surface from the current (w, v) coordinates.

In a first approach, in [10], the proposed planning strategies will use the (w, v) projection of *DOVTS*. Figure 2.2b represents this projection, which is named *DOVS* (Dynamic Object Velocity space). Note that the circular paths are transformed in this plane into straight lines ($r = v/w$). In this figure, $\mathbf{v}_1$ and $\mathbf{v}_2$ are the extreme velocities computed for the path r_j.

Definition 4. The *DOV* object is defined as:

$$DOV = \{(v, w) | \mathbf{V}_{low} \leq (v, w) \leq \mathbf{V}_{high}\}$$

The velocities belonging to *DOV* are not safe ones. So, safe velocity commands, *FV*, have to be selected outside of *DOV*, that is,

$$FV = \{(v, w) \notin DOV\}$$

The contour of *DOV* represents the velocity limits defined in equations 2.1. In Fig. 2.2b, *FV* and *DOV* velocities are depicted.

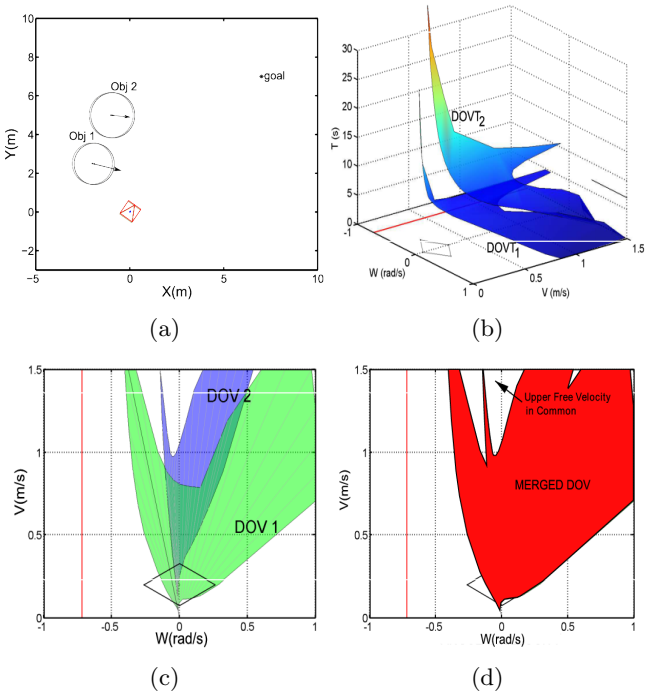


Figure 2.3: (a) A situation with two moving objects near to each other in *WS*. (b) The situation represented in *DOVTS*. (c) Projection of both moving objects in *DOVS space*.(d) The merged *DOV* object.

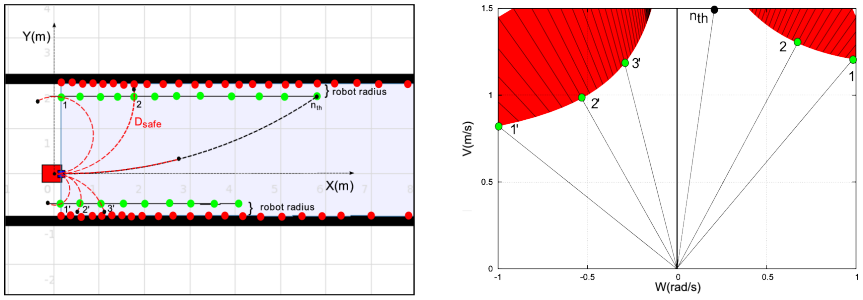
2.2.2 Dealing with multiple objects in the *DOVS*

In the $DOVTS$ space, several moving objects are represented as their corresponding $DOVT$ surfaces. Figure 2.3 shows a situation with two objects. Clearly, the highest surface in Fig. 2.3b corresponds to the farthest or the slowest object with respect to the robot, and the lowest surface to the nearest or the quickest obstacle. Working in this space, the velocity-time room between both surfaces can be utilized for maneuvering among the objects. This implies working directly in the $DOVT$ space. [11] dealt with this issue and it will be outlined in section 2.5. Figure 2.3c shows the projection of both surfaces in the DOV space. When this space is used to plan the trajectories, the DOV objects are merged to obtain one compound DOV object, as represented in Fig. 2.3d. Now reasoning in this space to compute the motion commands in the free velocity space (FV) is made using the merged object,

$$DOV_{merged} = \cup_{i=1}^{m}(DOV_i)$$

2.2.3 Static objects in the *DOVS*

The representation of static objects in $DOVS$ requires a slightly different formulation from that used with the dynamic objects.



(a) (b)

Figure 2.4: (a) Mapping a corridor into *DOVS*. The red points correspond to scan laser points, the green points are the points after extending the walls with the robot radius. (b) The corridor mapped into *DOVS* showing the maximum velocity reachable on each path.

Let D_{safe} be the length that the robot covers on a path to reach zero speed from its current velocity. If the distance to a static object is greater than D_{safe}, then the circular path in *DOVS* is mapped as free, i.e., all these velocities are free of collision. Otherwise, the maximum velocity that the robot should have to stop at before a collision, $\mathbf{v}_{stop}$, is calculated, and it is mapped into *DOVS*. Any higher velocity is mapped as a collision one. Thus, the static and moving objects are represented in the same framework. Figure 2.4 depicts the result of mapping a static corridor-like object into *DOVS*.

2.3 Decision making strategies

This section explains the general strategy of decision making in the planning technique developed in [10]. The problem is similar to a situation in which a pedestrian aims to cross a road while cars are passing. The pedestrian must decide how to do this safely. In our case the robot has to decide how to do the same. In such a situation, the main decision the robot must take is whether to pass before the obstacle arrives or to wait until the obstacle has passed.

The motion planning techniques are inspired from behaviors of humans navigating in pedestrian traffic that tries to take advantage of the complete manoeuvrability of the robot while simultaneously trying to reach the following goal in the shortest time possible. The robot can decide the motion plan for a time horizon, generating a trajectory at the maximum velocity possible in every instant, given the environment perceived by the sensors at every moment. The main criterion of the system is to preserve the safety of the robot, and consequently of the objects around it. Nevertheless, other criteria are simultaneously or hierarchically balanced, such as the rapid motion towards the goal, smooth changes in motions, etc.

Figure 2.5 reflects this idea. Basically there are two main planning strategies: passing before the object (*RobotFront*) and passing after the object (*RobotBehind*). In turn, each

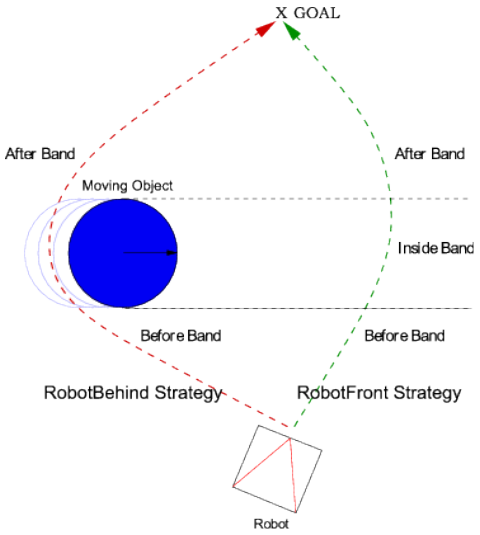


Figure 2.5: Trajectory in green corresponds to when the robot passes before moving obstacle and the trajectory in blue to when the robot passes after moving obstacle.

strategy is decomposed into different sub-strategies depending on whether the robot is before, inside or after the collision band. The execution of each motion strategy depends in turn on several *decision variables*, such as the relative location and the relative velocity between the robot and the objects, and the number of surrounding objects, formally defined in the next section.

2.4 *Situation-Strategies* planning technique

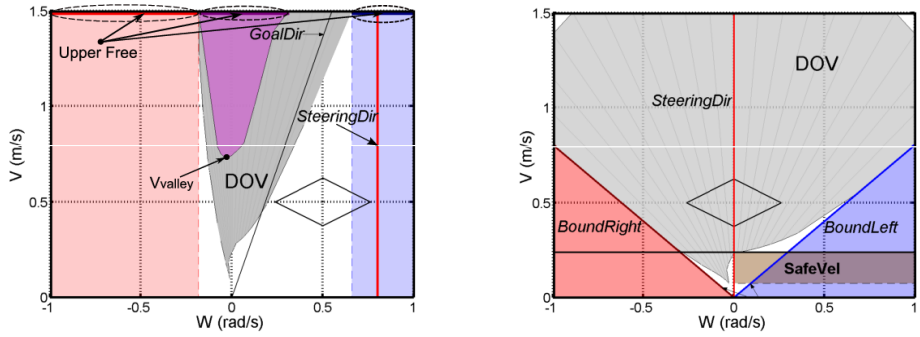
The decision making strategies defined in the previous section, extend to different *situations* that the robot might detect by means of its onboard sensors. The situation identification is made from several *decision variables* computed in $DOVS$ and WS spaces. This section is a formalization of the technique developed in [10].

2.4.1 Decision variables

A set of decision variables is computed to detect the situations. Some of the variables are obtained from the workspace (WS), others from the velocity space ($DOVS$). Table 2.1 summarizes all of them, including a brief explanation about their meaning. The table shows the acronyms, the meaning of the variables and their use in the different situations. The variables are utilized in the navigation strategies to identify the current situation, from which a specific motion action will be processed. Figure 2.6 depicts some of the decision variables mapped on the $DOVS$ space.

Variable	Meaning	Situation
WS		
$RelPos$	relative position before, in or after the collision band	all
$AngDis$	angular distance robot-goal	all
DOVS		
$FV - DOV$	free and non-free (DOV)velocities	all
$\mathbf{V}_{high}$-$\mathbf{V}_{low}$	min.velocities to pass before-max.velocities to pass behind	PassingBefore, SlowingDown
V_{Valley}	$min(\mathbf{V}_{high})$	PassingBefore
$UpperFree$	free angular velocities at maximum linear velocity	PassingBefore, PassingAligned
$LowerFree$	free velocities in lower zone ($\mathbf{v} < \mathbf{V}_{low}$)	SlowingDown
$BehindVel - FrontVel$	velocities to manoeuver with respect to the object	AvoidingObject, PassingBehind
$MaxAngVel$	maximum angular velocity	all
$SafeVel$	velocities to brake before crash	SlowingDown
$BoundRight - BoundLeft$	boundaries of free velocities at both sides	AvoidingObject
$SteeringDir$	mapped angular deviation	all
$GoalDir$	mapped goal direction	all

Table 2.1: Decision variables in WS and DOVS spaces. $\mathbf{V}_{high}$-$\mathbf{V}_{low}$ are computed for all the radii.



(a) (b)

Figure 2.6: Decision variables on *DOVS* space. (a) Situation in which there are *UpperFree* (magenta, blue and pink). (b) Situation in which *UpperFree* is empty, and would lead to collisision (grey).

2.4.2 Navigation planning

The robocentric planner developed in [10] assumes that there is a path planner at a higher level computing the consecutive subgoals to be reached in order to move to the final goal. The planner executes cyclically. It establishes a medium-term local plan in the horizon defined by the field of view of the sensors, but only the first step of this plan is executed for a sampling period. Thus the plan can be modified in the next steps according to de environment dynamics. In each sampling period the relative situation and motion between robot and objects are analyzed by means of the decision variables defined in the previous section and the situations identified.

Figure 2.7 shows the situation tree, which is evaluated for every cycle, establishing the navigation strategy for the situation identified. It instances the decision strategies described in section 2.3. The leaves are all the situations that can be identified. The intuitive idea

for the applied actions is to choose velocity commands in successive steps in the velocity free space of $DOVS$ (FV). Roughly speaking, the criteria to select these commands are to apply a sequence of maximum linear and angular accelerations to reach the maximum linear velocity if it is possible.

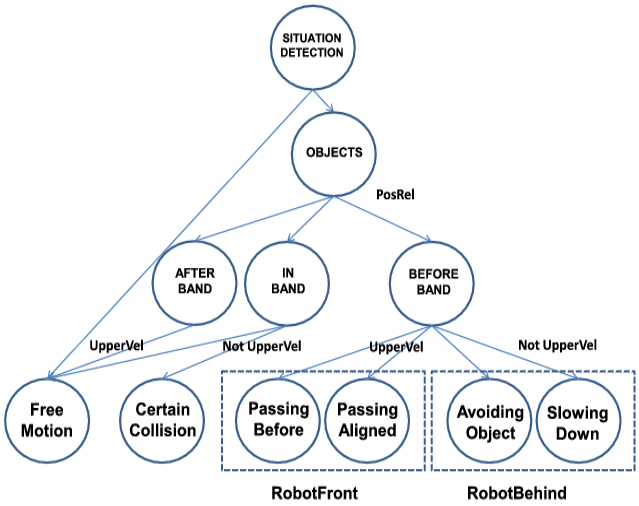


Figure 2.7: Situations Tree. Leaves represent all the situations that can be identified. Each of them has an associated navigation strategy.

If there are no objects in the field of view, the $FreeMotion$ situation is reached. The action applied aligns the robot towards the goal, reaching $GoalDir$ mapped in $DOVS$, by applying a sequence of clothoid (to reduce the angular deviation to the goal) and anticlothoid (to reach the maximum linear velocity) trajectories.

If there are objects in the field of view, the variable $RelPos$ (Table 2.1) expressing the relative situation of the robot with respect to the collision band of the object is analyzed. If the robot has passed the object ($AfterBand$ in the tree), the $FreeMotion$ situation is reached. In the $InBand$ state two situations can be reached. The first is $FreeMotion$ that applies actions as previously explained. Otherwise, a $CertainCollision$ situation appears when the robot is in a state of inevitable collision. This situation can only happen when objects suddenly appear close to the robot.

From the $BeforeBand$ state, one of the two global strategies is selected: $RobotFront$ or $RobotBehind$. In turn, depending on the relative position and velocity of the robot and objects, and on the safety criteria, four *situations* can be reached: $PassingBefore$, and $PassingAligned$ for the first strategy, and $SlowingDown$ and $AvoidingObject$ for the second. The current situation is detected by means of the decision variables. More details of each situation are now given.

a) RobotFront Strategy

The $Robotfront$ strategy shown in Fig. 2.5 is implemented by means of two possible substrategies, depending on the situation detected: $PassingBefore$ or $PassingAligned$. These situations are distinguished by whether there are free velocities in the $UpperFree$ zone or not. The actions to be applied in both situations are explained.

PassingBefore Situation

In this situation the robot passes before the object (see Fig. 2.8). To achieve this, an $UpperFree$ velocity over $\mathbf{V}_{high}$ is selected ($V3$). This velocity is reached through a sequence of a clothoid ($V1 - V2$) to align the robot towards the goal until reaching $V2$ velocity at maximum angular acceleration, and an anti-clothoid ($V2 - V3$) to speed up and reach the $V3$ velocity in the minimum number of steps. The sequence is maintained until the next sampling period.

PassingAligned Situation

In this situation (see Fig. 2.9) a zone free of collision exists on one side of the DOV, which matches the velocities leading the robot in the same direction as the object movement. Safety is the priority in this case, so an extremal control to escape this situation has to be applied. The robot combines clothoid and anti-clothoid trajectories to align and escape.

b) RobotBehind Strategy

This strategy leads the robot to pass after the moving obstacle (see Fig. 2.5). When the $UpperFree = \{\emptyset\}$ velocities are not reachable (see Fig. 2.6b), the only option for the robot is slowing down and choosing *LowerFree* velocities so that the object passes first, using the $BoundRight$ (BR) or $BoundLeft$ (BL) velocities. The manoeuver is made using the *SafeVel* defined in Table 2.1, which is the highest linear velocity value at which the robot can brake to avoid a collision. The trajectories computed cause the robot to go around the object, passing after it and thus avoiding entering the *collision band*.

Two situations can be found, depending on the free velocities that can be reached: *SlowingDown* or *AvoidingObject* (see Fig. 2.7).

SlowingDown Situation

This corresponds to the case in which $UpperVel$ velocities in the $DOVS$ cannot be chosen because they would lead to collision. Since the upper velocities are prohibited (see Fig. 2.10a), a prioritized safety criterion is taken. A sequence of anti-clothoid and clothoid trajectories is applied to escape from the dangerous velocities. When the $UpperFree \neq \{\emptyset\}$, an anti-clothoid at maximum acceleration makes the robot move towards the goal at maxi-

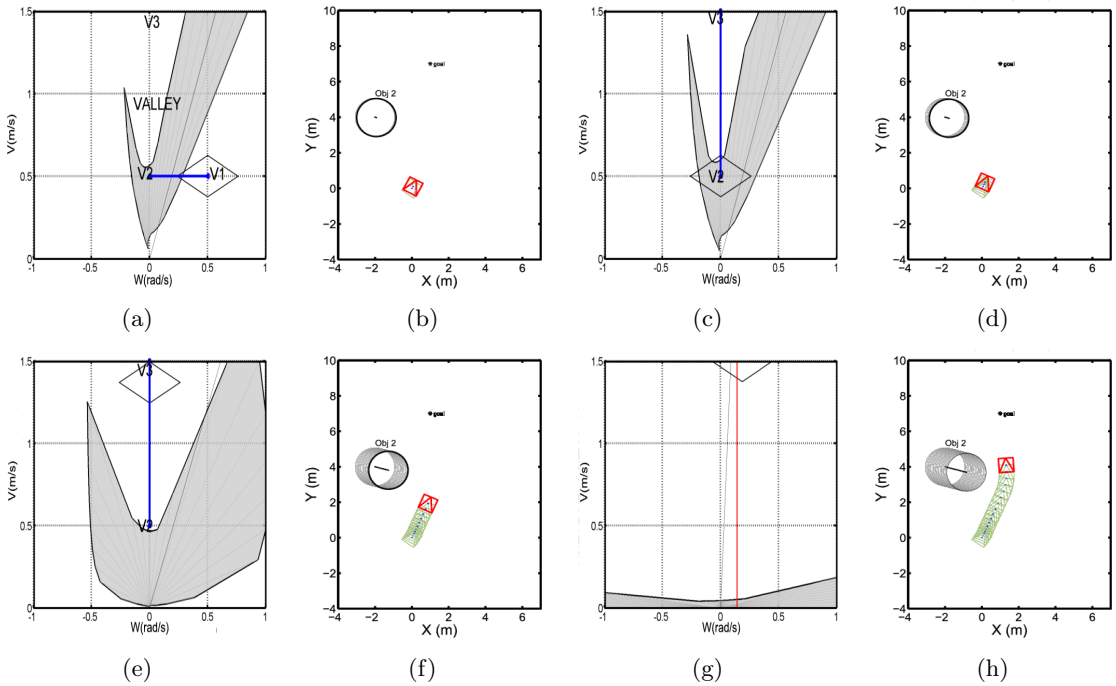


Figure 2.8: Evolution of *PassingBefore* situation. The highest velocities ($V3$) can be chosen for the robot to pass before the moving object.

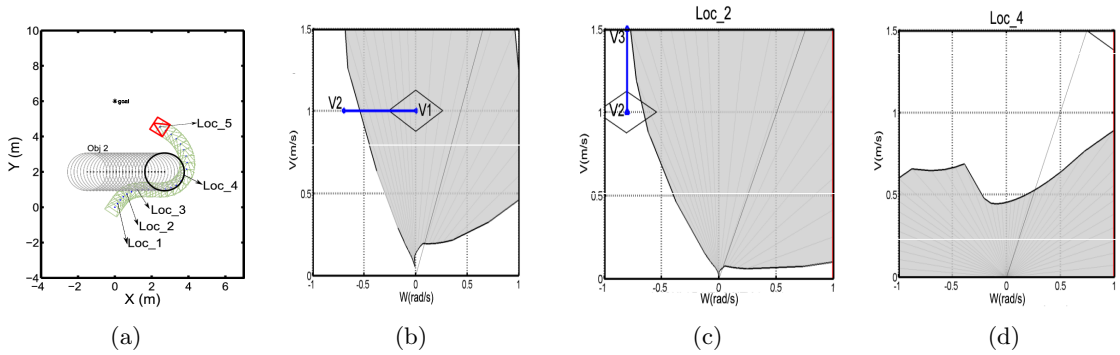


Figure 2.9: Evolution in *PassingAligned* situation. The robot has to align to the direction of motion of the object to avoid collision by passing in front of the object, as can be seen between Loc_3 and Loc_4, where a $FreeMotion$ situation appears.

mum velocities. Figure 2.10 represents this sequence.

AvoidingObject Situation

This represents a situation in which the moving object velocity lies in the zone of DOV, and so there is a danger of collision (see Fig. 2.11). There are two zones in $UpperFree$

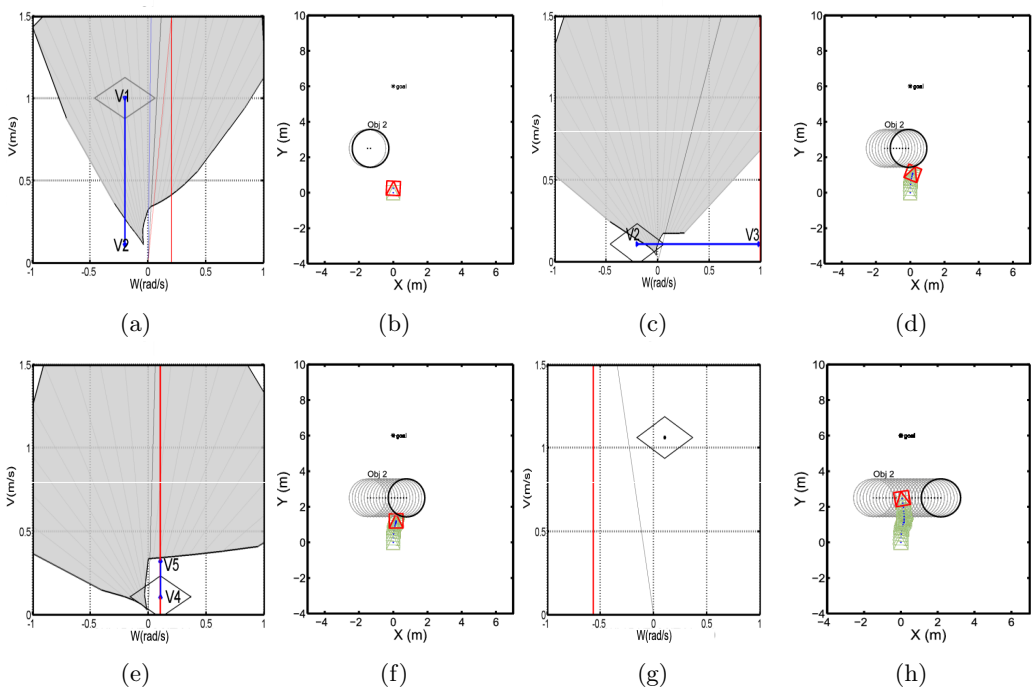


Figure 2.10: Evolution in *SlowingDown* situation. The robot has to slow down to avoid the collision, by combining anti-clothoid trajectories until reaching high and free velocities, permitting the object passes before the robot.

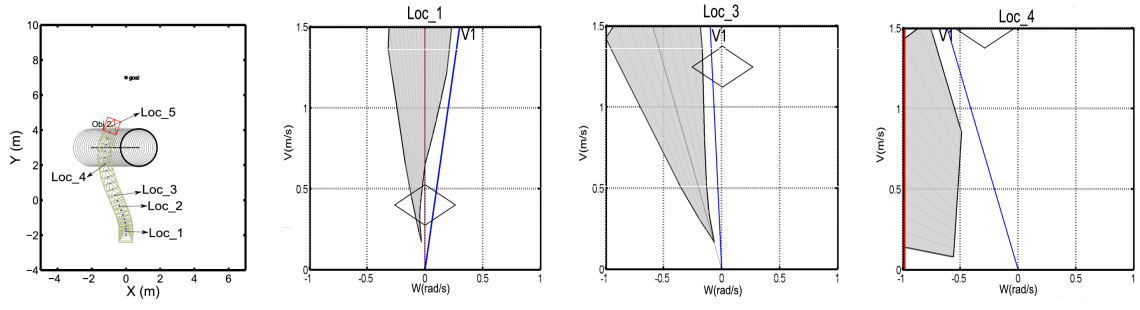
velocities free of collision, representing angular velocities higher than the represented by $BoundRight(BR)$ and $BoundLeft(BL)$. First the angular velocity closest to the $GoalDir$ is chosen to escape from collision, reaching maximum linear velocity $V1$.

Although these strategies are designed to reach the maximum velocities whilst maintaining the robot and environment safety, they do not ensure an optimal control in the sense of minimum time to goal, or under another criterium. The technique developed in [11], introduced in the next section, minimizes a cost function that balances several criteria related to time to goal and safety.

2.5 *Cost-Optimization* planning technique

A main objective of this Master's thesis is to compare *Situation-Strategies* with *Cost-Optimization* and *Strategies-Optimization*. Thus, this section summarizes the basis of the two last techniques but does not give details. To extend the information refer to [11] and [15].

The *Cost-optimization* planning technique ([11]) is based on the optimization of a cost function to compute the optimal motion commands, stating the optimization problem in the $DOVTS$ space. Working on the bidimensional projection ($DOVS$) of the $DOVTS$ as described in section 2.4 restricts the action and movement capacity of the robot, due to the



(a) (b) (c) (d)

Figure 2.11: Evolution of *AvoidingObject* situation. The robot manoeuvres to pass in front of or behind the object, depending on the current relative situation between the goal and the robot. Blue line is $GoalDir$, representing every moment the direction for aligning the goal.

tendency to choose movements in the velocity free space (FV), when some commands under the forbidden obstacle surface ($DOVT$) are still available.

First, the *Cost-Optimization* technique discretizes the $DOVTS$ creating a mesh of rectangular prismatic cells with regular sizes, δ_v, δ_ω, δ_t, balancing both precision and computational cost. The space defined by the forbidden surface is represented as *occupied cells*. Second, an A^*-like search algorithm is used in this space to find the optimal trajectory in terms of velocity sequences. It explores iteratively the free cells or nodes of the mesh, opening several paths from the initial node and generating a spanning tree from the lowest level corresponding to the current time ($t = 0$). In each iteration, the neighbor cells in FVT, corresponding to free velocities, are visited and the one with the lowest cost is expanded. Finally, a cost objective function is defined to guide the search, which balances time to goal, safety in terms of proximity to obstacles, and smooth motions with limited velocity changes:

$$f(c) = g(c) + h(c)$$

where $g(c)$ is the cost of reaching the current cell from the initial cell and $h(c)$ is a heuristic term that estimates the cost of reaching the goal from the current cell. This heuristic cost is defined from three weighted components:

$$h(c) = \alpha_v \cdot h_v + \alpha_d \cdot h_d + \alpha_s \cdot h_s$$

The first term h_v is computed as the minimum number of iterations in v and w to reach the next velocity goal (or the number of cells to traverse in FVT). In Fig. 2.12 the $DOVTS$ for two moving objects is represented. As can be seen, a velocity goal is computed for each moving object. These are used to lead the search sequentially for the optimal trajectories. The second one h_d estimates the time that the robot needs to reach the goal from the current location, without considering obstacles. The third component h_s contributes to safe motions, avoiding obstacles. It is measured as the distance from the corresponding cell to the $DOVT$ surface in the T axis.

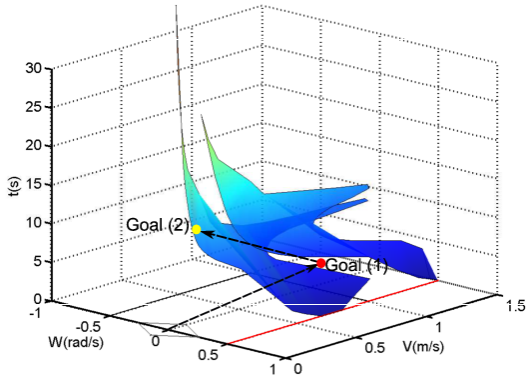


Figure 2.12: Two moving objects in *DOVTS* space. Goal(1) and Goal(2) represent two velocity goals (*GoalVel*) computed in the Free Velocity Spàce of *DOVTS*.

2.5.1 *Strategies-Optimization*: integration of *Situation-Strategies* and *Cost-Optimization*

This section outlines the *Strategies-Optimization* technique, defined in [11] and extended in [15].

In the *Cost-Optimization* approach, the velocity subgoals are chosen only using a proximity to the aligning direction (*GoalDir* variable) criterion. This imposes some constraints on finding the optimal command in the presence of obstacles. The *Situation-Strategies* method allows velocity subgoals to be chosen using more complex criteria than that proposed in *Cost-Optimization*. For this reason, [11] propose an approach to integrate both techniques.

The key point is the velocity cost term h_v in the cost optimization function, which is redefined as:

$$h^s(c) = \alpha_v \cdot h_v^s + \alpha_d \cdot h_d + \alpha_s \cdot h_s$$

where h_v^s is now computed from the $GoalVel$ of the *Situation-Strategies* technique.

[11] evaluates the performance between *Cost-Optimization* and the integration of both *Situation-Strategies* and *Cost-Optimization*. Figure 2.13 shows two simulations in a simple scenario. The first one is carried out without using *Situation-Strategies*, that is, only the *Cost-Optimization* is active. The second one represents the same scenario in which both *Situation-Strategies* and *Cost-Optimization* are active. It can be seen that not using the *Situation-Strategies* yields a longer and more oscillatory trajectory. In the integrated *Strategies-Optimization* the maximum linear velocity is also maintained for longer.

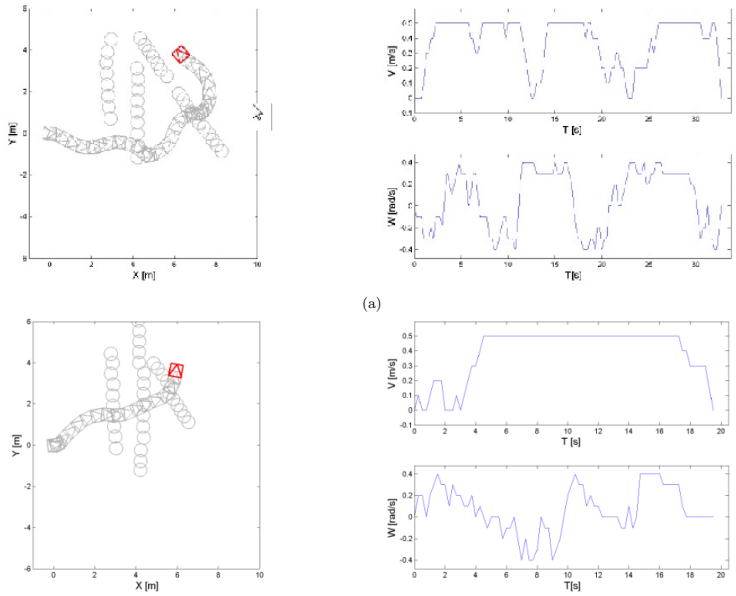


Figure 2.13: Image taken from [11], where simulations in two cases are displayed: (a) Only the *Cost-Optimization* technique is active, $\alpha_v = 0, \alpha_d = 1, \alpha_s = 1$. (b) Both techniques are active, $\alpha_v = 1, \alpha_d = 0.5, \alpha_s = 1$. Notice that in the second case the maximum linear velocity is maintained whilst in the first one the time to goal is higher and the trajectory more oscillatory.

Chapter 3

Situation-Strategies evaluation and comparison with *Cost-Optimization* and *Strategies-Optimization*

3.1 Metrics for assessment

[15] defines a methodology to evaluate the quality of the movement with the *Cost-Optimization* and *Strategies-Optimization* techniques, as a function of the α parameters of the cost function. Table 3.1 shows the metrics used for evaluation, defined to evaluate the performance of the techniques and to determine the most suitable selection of the α parameters for the *Cost-Optimization* and *Strategies-Optimization* methods: the percentage in time to goal improvement, in smoothness of the motion measured as change of the velocity (acceleration), and in the safe distance. Instead of using absolute values for quantifying the criteria, [15] computes relative values with respect to the worst case in each scenario.

The work carried out for the Master's thesis focuses on using the same metrics to evaluate the *Situation-Strategies* approach and compare it with the others techniques.

3.1.1 Simulation scenarios

Two kinds of navigation scenarios with moving objects have been chosen. The first includes few moving objects around the robot (*uncluttered scenario*, between 10 and 12)), and four different goals to achieve. The second contains a high density of moving objects reducing the ability of the robot to navigate (*cluttered scenario*, between 20 and 25), and also four different goals to reach. Both scenarios are non structured and the obstacles moving around reduce the robot capability of navigating. We have also introduced non-straight line trajectories for the objects, to test simultaneously the robustness of the navigation techniques. Figure 3.1 depicts a snapshot of the simulation in one of the scenarios. Appendix A contains several steps during one of the simulation in the cluttered scenario.

Metrics	Expressions
time saving, m_t (%)	$\frac{1}{M} \cdot \sum_{j=1}^{M} 1 - \left[\frac{t_{(j)}}{T_{max}} \right]$
reduction in the change of v, m_a (%)	$\frac{1}{M} \cdot \sum_{j=1}^{M} 1 - \left[\frac{\delta v_{(j)}}{\delta V_{max}} \right]$
safety distance, m_d (%)	$\frac{1}{M} \cdot \sum_{j=1}^{M} 1 - \left[\frac{d_{(j)}}{D_{max}} \right]$
success rate, m_s (m)	$\frac{1}{M} \cdot \sum_{j=1}^{M} s_{(j)}$

Table 3.1: Evaluation Metrics. M is the number of tests for each method and kind of scenario, T_{max}, δV_{max} and D_{max} are the maximum values of time, change of velocity and distance to obstacles in each test, respectively.

Parameters	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}
α_v	0.498	0.333	0.00	0.00	0.048	1.00	0.40	0.20	0.40	0.50	0.25	0.25
α_d	0.005	0.333	1.00	0.20	0.0	0.00	0.20	0.40	0.40	0.1670	0.25	0.50
α_s	0.498	0.333	0.00	0.80	0.952	0.00	0.40	0.40	0.20	0.333	0.50	0.25

Table 3.2: Set of representative samples of combinations of α parameters. The time to goal, smoothness of the motion and the safety are weighted.

The method used to track the moving objects was EKF-based, in which the state vector included the location and the velocities of the tracked objects. The static and moving object maps of the scenario were obtained using the method developed in [13] to model a dynamic environment.

The *Situation-Strategies*, *Cost-Optimization* and *Strategies-Optimization* techniques have been tested and compared. For the two last approaches, a set of representative samples of the set of all combinations of α parameters are used to evaluate the cost function (Table 3.2), ranging from values that try to improve the *time of mission* to others that assess the *safety*. All these scenarios account for 200 test situations, which have been considered representative for extracting well-founded conclusions about the performance of the techniques and how to choose the weighting parameters for navigation.

The three techniques were analyzed using the evaluation parameters defined in 3.1 for each of the twelve optimization parameter α combinations shown in 3.2. For the metrics shown in 3.1, $M = 4$ beacuse 4 different goals have been defined in each kind of scenario.

Figure 3.1.1 shows the results in the uncluttered and in the cluttered scenarios, respectively. The discontinuity in some trend lines and the zero points correspond to combinations with low rates of success (m_s) and are therefore discarded. As it could be expected the re-

sults for *Cost-optimization* and *Strategies-Optimization* outperformed the ones obtained for *Situation-Strategies* in all the metrics and scenarios. Time saving and reduction in change of velocities were better, and regarding safety the use only of the *Situation-Strategies* leads to a more conservative navigation, far from the obstacles.

Roughly speaking, better results are obtained for the three metrics using the integrated *Strategies-Optimization* technique. Also, it can be seen that in uncluttered scenarios the best performance is obtained with parameters that give more weight to the time to velocity goal term (α_v), or that give equal weight to the three criteria, that is combinations $\mathbf{p_1}$ and $\mathbf{p_2}$. Conversely, for cluttered scenarios, it is important to highlight the safety term (α_s), for example parameters $\mathbf{p_4}$ and $\mathbf{p_9}$, although weighting the three terms similarly (like in $\mathbf{p}_2$) also provides a good behavior. $\mathbf{p_3}$ has been discarded because it drives to collision in some scenarios.

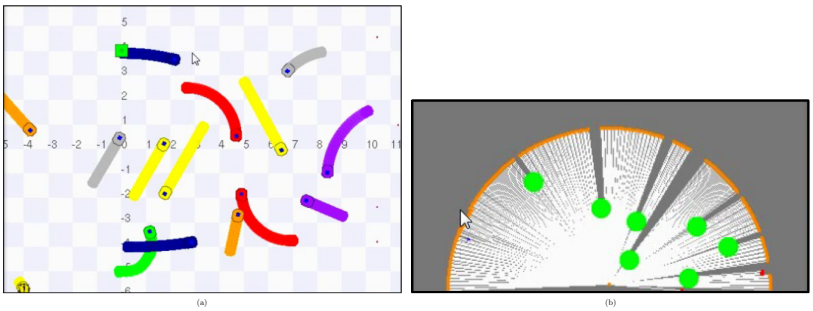
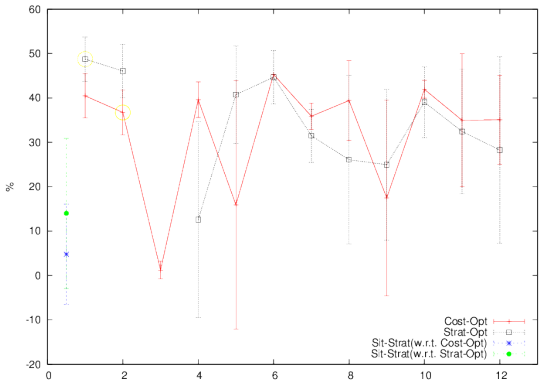
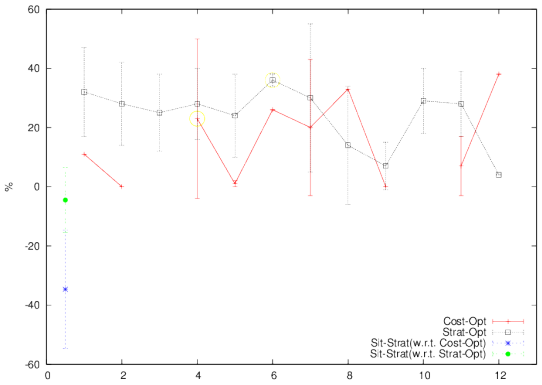


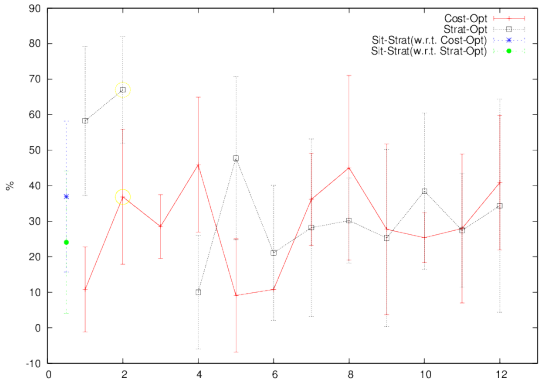
Figure 3.1: (a) A snaphot of the scenario. The controlled robot is the green object coming from the lower space. (b) The objects tracked by the perception system in that instant.



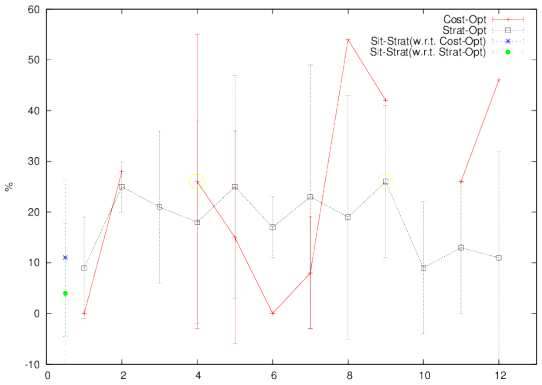
(a) Time saving



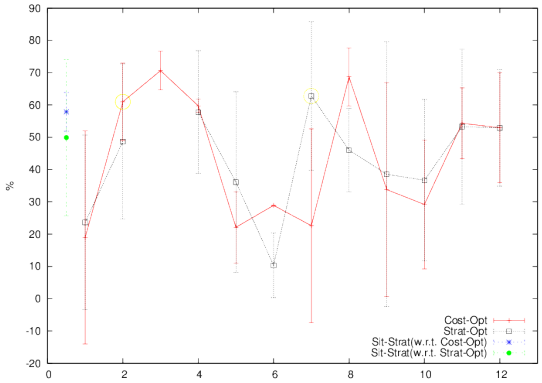
(a) Time saving



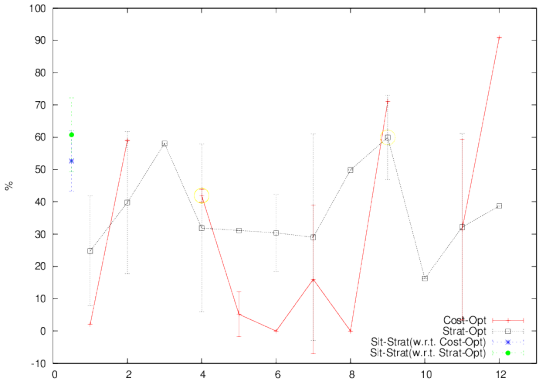
(b) Reduction in the change of v



(b) Reduction in the change of v



(c) Increase in safety factor



(c) Increase in safety factor

Figure 3.2: Values of the metrics for the uncluttered (left) and cluttered (right) scenarios for *Situation-Strategies*, *Cost-Optimization* and *Strategies-Optimization* techniques. Yellow circles represent the maximum value with no collisions.

Chapter 4

Seamless localization

The scenarios we consider are very heterogeneous, composed by mapped and unmapped zones and with variable GPS coverage. Therefore, it is necessary to provide the robot with a robust and seamless localization system to perform a safe and trustful navigation. Next, a description of the localization system proposed is given (completely developed in [21]).

4.1 Related work

In [17], the authors integrate, by means of a Kalman filter, a laser-based SLAM method with GPS, IMU and odometry measurements for a continuous localization. However, EKF scheme assumes independence between measurements but SLAM output and GPS measurements are highly correlated as they are both estimations of the same variables of the robot state. These independence assumptions, when real correlations are high, lead to incoherent estimations because they are overconfident. Thus, our approach applies the covariance intersection method [8] to avoid these problems.

In situations where compass cannot be used and the orientation of the robot cannot be directly measured from GPS, it has to be derived from other measurements. [7] proposes a Kalman Filter to fuse odometry/IMU and GPS measurements. To estimate the orientation of the robot they use two consecutive GPS measurements assuming that the robot is moving straight, which may be broken easily as the frequency of the GPS decreases. In [6], using the same estimation, the authors determine the likelihood of the estimated orientation by comparing it to the estimation obtained from the IMU. If they diverge, GPS based estimation is discarded. However, this technique would accept a bad estimation that is coherent with a bad IMU measurement. Thus, we propose a method to calculate the orientation of the robot from several GPS measurements whenever the robot moves straightforward to improve the estimation of the filter.

4.2 Localization framework

The localization technique proposed combines different methods in a robust way to be able to seamlessly localize the robot in very heterogeneous scenarios. Specifically, this method distinguishes among different situations that may be found in such kind of scenario and obtains the best estimation. Table 4.1 sums up the situations detected depending on the current position of the robot (inside or outside the map) and the quality of the GPS (see [21] for details).

	Good GPS	Bad GPS	No GPS
Inside Map	Outdoors	Transition	Indoors
Outside Map	Outdoors	Outdoors (No GPS)	

Table 4.1: Situation detection

To track the best estimation of the localization $\mathbf{x}$ an Extended Kalman Filter with a covariance intersection scheme during the update phase is used. The prediction is computed from the odometry/IMU data $\mathbf{y}_{od}$, as shown in (4.1), where $\mathbf{Q}$ is the covariance of the odometry/IMU error and $\mathbf{F}$ is the Jacobian of the prediction function.

$$\begin{aligned} \mathbf{x}(k{+}1|k) &= \mathbf{x}(k|k) \oplus \mathbf{u}_{od}(k{+}1) \\ \mathbf{u}_{od}(k{+}1) &= \mathbf{y}_{od}(k{+}1) \oplus \mathbf{y}_{od}^{-1}(k) \\ \mathbf{P}(k{+}1|k) &= \mathbf{F}(k)\mathbf{P}(k|k)\mathbf{F}(k)' + \mathbf{Q}(k{+}1) \end{aligned} \tag{4.1}$$

The update phase is based on the covariance intersection method which permits the fusion of different information sources with unknown correlation among them. In this case, three different sources are used: the odometry prediction $\mathbf{x}(k{+}1|k)$, the GPS estimation $\mathbf{y}_{gps}$ and $\mathbf{y}_{map}$ the estimation of a laserscan-based localization, using the method defined in [2], in mapped zones, as shown in (4.2). The final estimation is computed by means of a weighted sum of all the sources. For weighting, besides using the covariance matrices of the estimations, $\mathbf{P}$, $\mathbf{R}_{gps}$ and $\mathbf{R}_{map}$ respectively, three factors (γ_{od}, γ_{gps} and γ_{map}) are applied depending on the situation in which the robot is at each moment as defined in Table 4.2.

$$\begin{aligned} \mathbf{P}(k{+}1|k{+}1)^{-1} =& \gamma_{map}\,\mathbf{R}_{map}^{-1} + \gamma_{gps}\,\mathbf{R}_{gps}^{-1} \\ &+ \gamma_{od}\,\mathbf{P}^{-1}(k{+}1|k) \\ \mathbf{x}(k{+}1|k{+}1) =& \mathbf{P}(k{+}1|k{+}1)(\gamma_{map}\,\mathbf{R}_{map}^{-1}\,\mathbf{y}_{map} \\ &+ \gamma_{gps}\,\mathbf{R}_{gps}^{-1}\,\mathbf{y}_{gps} \\ &+ \gamma_{od}\,\mathbf{P}^{-1}(k{+}1|k)\,\mathbf{x}(k{+}1|k)) \end{aligned} \tag{4.2}$$

4.3 GPS-based orientation

In this framework, the only direct source of absolute orientation estimations is the map-based method from the $\mathbf{y}_{map}$ measurement. GPS only provides directly position estimations,

Situation	γ_{map}	γ_{gps}	γ_{od}
Indoors	0.5	0.0	0.5
Transition (inside map)	0.5	0.0	0.5
Outdoors (no GPS)	0.0	0.0	1.0
Outdoors (good GPS)	0.0	0.5	0.5

Table 4.2: Update weights in different situations

so the orientation when there is not a map must be obtained in another way. Compass can directly measure the orientation of the robot but it is affected by the magnetic fields of the environment, making unusable its estimations in scenarios with sources of magnetic fields. In absence of other information, odometry/IMU estimations can be used for this purpose. But it accumulates error during the movement. It is possible to try estimating the orientation from several GPS measurements to compute $\mathbf{y}_{gps}$ in the expression 4.2, but it is only a precise way when the robot is moving close to a straight line path. The orientation is calculated from a set of consecutive GPS measurements $\{\mathbf{x}_{gps_0}, \cdots, \mathbf{x}_{gps_n}\}$ under some constraints using the mixture of gaussians of equations from (4.3) to (4.5).

$$\hat{\theta} = \sum_{j=1}^{n} w_j \hat{\theta}_j, \quad \sigma^2 = \sum_{j=1}^{n} w_j \left(\sigma_j^2 + \left(\hat{\theta}_j - \hat{\theta} \right)^2 \right) \tag{4.3}$$

$$w_j = \frac{1}{n-1} \left(1 - \frac{\sigma_j^2}{\sum_{k=1}^{n} \sigma_k^2} \right) \tag{4.4}$$

$$\hat{\theta}_j = \arctan \left(\frac{y_{gps_j} - y_{gps_0}}{x_{gps_j} - x_{gps_0}} \right) \tag{4.5}$$

The constraints imposed to consider this estimation to be used in the update step are a minimum distance d between the initial $\mathbf{x}_{gps_0}$ measurement and the following one, a minimum number of GPS measurements needed n such that the distance between $\mathbf{x}_{gps_0}$ and $\mathbf{x}_{gps_n}$ is L for a given GPS measurement frequency, and that all them has taken place while the steering angle ϕ of the robot has a value close to 0, making sure that the robot is moving in a straight line and thus that the estimated orientation is constant. For more details refer to [21].

Depending on whether the GPS-based orientation has been calculated or not, the GPS measurement and covariance in (4.2) take different values. If we have an orientation estimation from the GPS,

$$\mathbf{y}_{gps} = \left(x_{gps},\, y_{gps},\, \hat{\theta} \right)'$$

$$\mathbf{R}_{gps}= \begin{pmatrix} R_{x_{gps}} & 0 & -\frac{w_n}{L} R_{x_{gps}} \sin\hat{\alpha} \\ 0 & R_{y_{gps}} & \frac{w_n}{L} R_{y_{gps}} \cos\hat{\alpha} \\ -\frac{w_n}{L} R_{x_{gps}} \sin\hat{\alpha} & \frac{w_n}{L} R_{y_{gps}} \cos\hat{\alpha} & \sigma^2 \end{pmatrix}$$

Otherwise, the orientation is taken from the prediction step (given by odometry/IMU), and then,

$$\mathbf{y}_{gps} = \begin{pmatrix} x_{gps} \\ y_{gps} \\ 0 \end{pmatrix}, \mathbf{R}_{gps}^{-1} = \begin{pmatrix} \frac{1}{R_{x_{gps}}} & 0 & 0 \\ 0 & \frac{1}{R_{y_{gps}}} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

4.3.1 Improvement of the GPS-based orientation method

The method explained above imposes some constraints to estimate the orientation of the robot based on GPS measurements. The steering angle ϕ of the wheels were used, in order to estimate that the robot was moving in straight line, when $\phi \simeq 0$. But it can not always be a good measure of this behavior. The constraints to use this method to estimate the robot orientation are analyzed from several experiments.

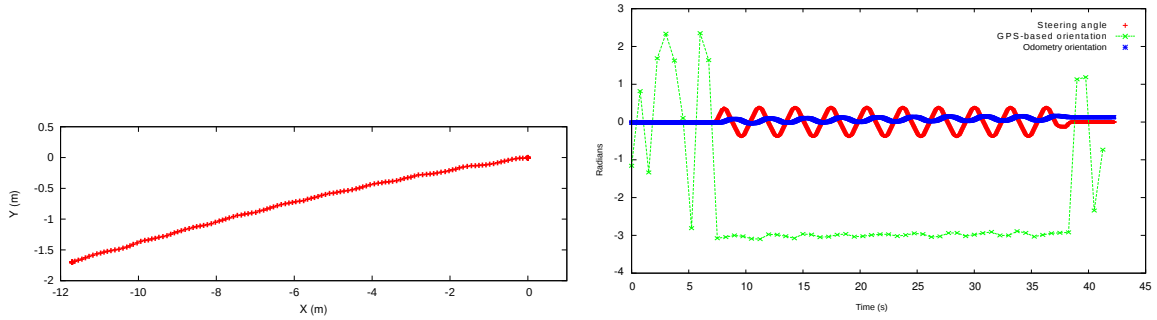
Next, we present the experiments performed to propose different constraints to the method which are less conservative but still achieving good orientation estimations. By weakening the restrictions we are able to obtain these estimations more often, reducing the orientation error accumulated from the odometry/IMU prediction.

In the first experiment the robot moves at constant linear velocity and the steering angle ϕ takes values from equation (4.6). Parameter A may take values from $\{0.1, 0.2, 0.3, 0.4\}$ and ω from the set $\{0.0, 0.5, 1.0, 1.5, 2.0\}$.

$$\phi(t) = A\sin(\omega t) \tag{4.6}$$

Figure 4.1 illustrates the results obtained with $A = 0.4$ and $\omega = 2$. In figure 4.1a, the GPS positions of the robot are depicted, which show the robot is moving nearly in straight line. The steering angle, the orientation from odometry and the orientation computed from GPS measurements are shown in Fig. 4.1b, which reflects that the amplitude of the steering angle is bigger than the one provided by the odometry. So in general the robot orientation is a better source than the steering angle to determine whether the robot is moving in straight line.

Figure 4.1b also depicts that when the robot is not moving the value of the orientation jumps. To analyze the conditions to get a good estimation of the orientation, we have measured the error in the estimations while navigating the robot as straight as possible along $12\,m$ at constant speed. Figure 4.2 shows the square mean error of the estimations. For distances greater than 0.4 there is no improvement in the estimations. Therefore, whenever the procedure for estimating the orientation is initialized, the minimum distance d between the initial $\mathbf{x}_{gps_0}$ data and the next one to estimate the orientation should be of 0.4. In this experiment we found that the threshold th for the maximum variation of orientation accepted



(a) GPS measurements.

(b) Steering angle, orientation from odometry and GPS-based orientation.

Figure 4.1: Results obtained from the experiment following (4.6) with $A = 0.4$ and $\omega = 2$.

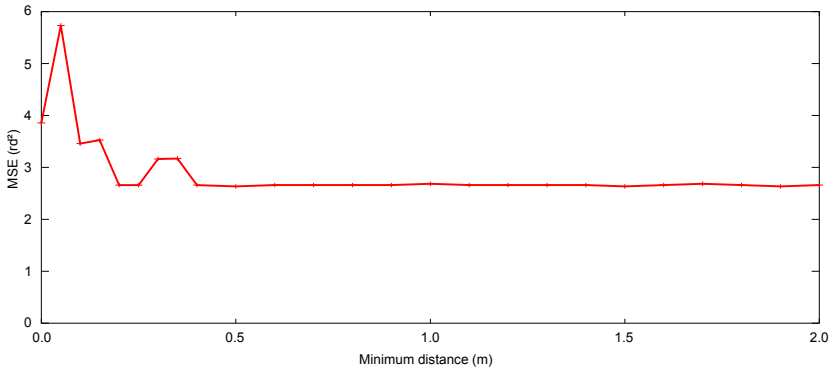
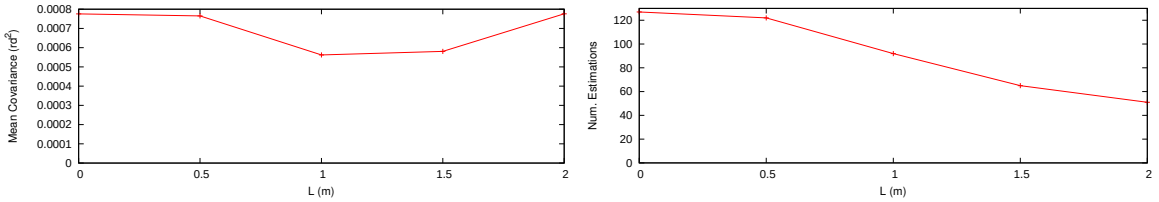


Figure 4.2: Mean square error of the estimation of the orientation using GPS in straight line for different minimum distance constraint.

to determine if the robot is moving in straight line were $1\,^\circ$ every $2\,m$, approximately. Otherwise, the orientation only can be computed from the odometry/IMU estimation.

The constraint on the total distance L between the initial $\mathbf{x}_{gps_0}$ measurement and the last one used to estimate the orientation is examined by performing several experiments in which the robot moves following non-straight paths. The frequency of the GPS measurements is $4Hz$. Figures 4.3a and 4.3b show the number of total estimations and the mean of the covariances of the orientation obtained during the experiment for different values of L. Values higher than $1\,m$ would decrease the frequency of orientation updates as the number of total estimations is considerably reduced, so they are discarded. A value of $L = 1$ reduces almost in 25% the number of estimations computed with respect to $L = 0.5$, whereas there is no much difference in covariance. Thus, we consider that a total distance of 0.5 wolud increase the number of updates of orientation and provide good estimations of the orientation.

The method to estimate the orientation with these new constraints is outlined in Algorithm 1, which follows equations from (4.3) to (4.5). In lines 7 and 8, $\Delta\theta_{odom}$ is the total variation of orientation experimented in odometry during the interval between $\mathbf{x}_{gps_0}$ and



(a) Number of estimations of the orientation.

(b) Mean of covariance of the total estimations.

Figure 4.3: Results of the experiment to determine L.

Algorithm 1 Estimation of the orientation from GPS measurements

Require:

1: $\mathbf{x}_{gps_0}$ is the initial GPS measurement taken as reference
2: $\mathbf{x}_{gps_j}$ is the current GPS measurement
3: th is the maximum variation of orientation in odometry accepted to determine if the robot moves in straight line
4: d is the minimum distance between $\mathbf{x}_{gps_0}$ and $\mathbf{x}_{gps_j}$
5: L is the total distance to estimate the orientation
6: **procedure** EstimateOrientation($\mathbf{x}_{gps_0}$, $\mathbf{x}_{gps_j}$, th, d, L)
7: **if** $\Delta\theta_{odom} > th$ **then**
8: $\mathbf{x}_{gps_0} = \mathbf{x}_{gps_j}$
9: **else**
10: **if** $Distance(\mathbf{x}_{gps_0}, \mathbf{x}_{gps_j}) >= d$ **then**
11: $\hat{\theta}_j(\mathbf{x}_{gps_0}, \mathbf{x}_{gps_j}), \sigma_j^2(\mathbf{x}_{gps_0}, \mathbf{x}_{gps_j})$
12: **if** $Distance(\mathbf{x}_{gps_0}, \mathbf{x}_{gps_j}) >= L$ **then**
13: **return** $\hat{\theta}, \quad \sigma^2$
14: $\mathbf{x}_{gps_0} = \mathbf{x}_{gps_j}$
15: **end if**
16: **end if**
17: **end if**
18: **end procedure**

$\mathbf{x}_{gps_j}$. If $\Delta\theta_{odom}$ is not under th (i.e, the robot is not moving in straight line) we discard the GPS measurements considered for the estimation so far and initialize the GPS measurement of reference with the current one. Line 14 reflects that once the estimation of the orientation has been calculated, the process is initiated and the GPS measurement of reference is initialized with the current one.

4.3.2 Results

This section presents the results obtained from the experiment to compare the GPS-based orientation improvement method with respect to the previous approach. The platform used is a Robucar-TT[1] equipped with IMU, odometry, three range-finder sensors and GPS receiver (figure 4.4).

[1] www.robosoft.com

Figure 4.4: Robucar-TT platform

In [21] a real experiment in a large scenario within indoor and outdoor scenarios was achieved. Figure 4.5 reproduces the experiment, showing the continuity in localization and the limited uncertainty in the whole trajectory using the unified localization technique. This work extends the technique presented to improve the orientation estimation from GPS measurements when it can provide better estimations as explained in section 4.2. We focus our analysis only in a interval of the full experiment where the robot is not moving in a straight line (see green rectangle in figure 4.5).

Figure 4.6a shows the pose estimations with both methods and the GPS data received. The previous method does not update the orientation from GPS because the robot is not moving in straight line. Now, we obtain better results in the pose estimations, which are closer to GPS measurements. In figure 4.6b the detail of the GPS-based orientation estimated from both methods is illustrated. It is clear the improvement in orientation estimation using the GPS-based technique.

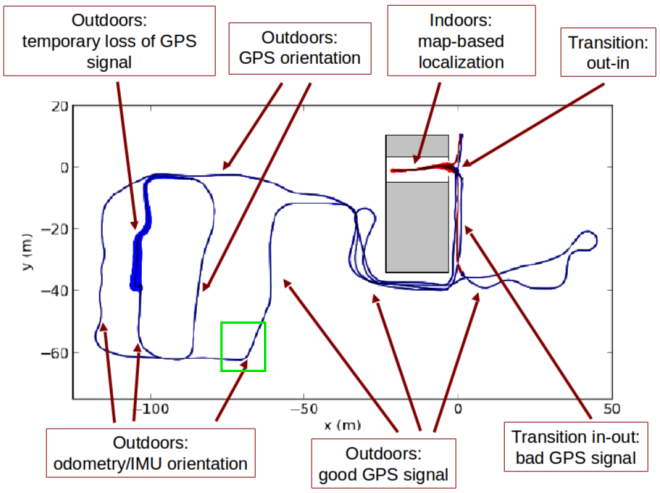
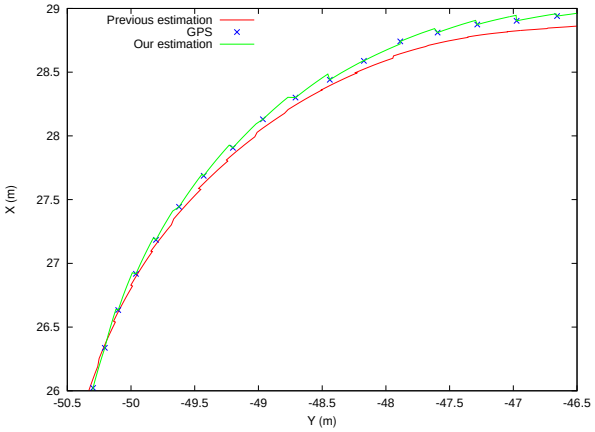
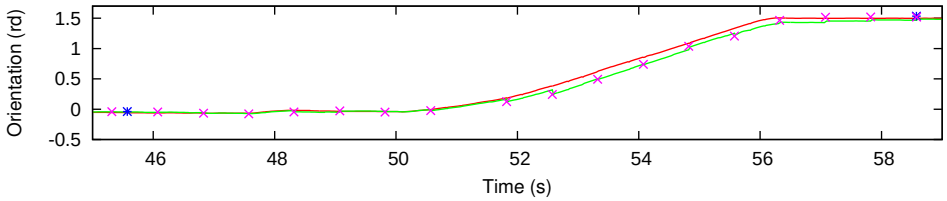


Figure 4.5: Test scenario for evaluating our localization method. It covers an area of 12000 m^2 and the trajectory length is about 1000 meters.



(a) Pose estimations and GPS measurements.



(b) Orientation estimations. Blue and magenta marks are the GPS-based orientation estimated using, respectively, the previous method and our method. Continuous lines represent the resulting orientation integrated with odometry/IMU measurements from previous (in red) method and our method (in green).

Figure 4.6: Robot's pose and orientation estimations in a non-straight stretch.

Chapter 5

Conclusions

Regarding the planning of motion for robots in dynamic environments, a comparison of different planning techniques has been developed. The metrics defined in [15] and its use to quantitatively evaluate the performance of the navigation for *Cost-Optimization* and *Strategies-Optimization* have served to guide the evaluation of *Situation-Strategies* and to elaborate founded conclusions on the performance of the three techniques.

As a result of the evaluation, sets of optimization parameters can be selected to be applied in the *Strategies-Optimization* planning technique. This methodology allows this selection to be made under a well-founded criterion, yielding a reduced number of best control parameters to be applied in different situations and scenarios.

Future work will focus on the extension of the technique to sharing the decision making process among several robots and obtaining optimized plans.

Regarding localization, the previously developed robust seamless continuous localization technique ([21]) for indoor-outdoor environments has been improved. A unified framework for continuous localization in large environments has been presented. The most suitable and accurate sensors in each moment are integrated to provide a continuous localization without discontinuities in the estimations and having a limited uncertainty, even when transitions indoor-outdoor or viceversa are produced. The parameters and conditions allowing the best sensor integration have been obtained from several experiments in real environments.

Appendix A

Simulations

A.1 *Cluttered scenario*

This section shows several steps of the simulation performed to evaluate the *Situation-Strategies* technique.

This scenario contains a high density of moving objects, which reduces considerably the capability of the robot to navigate. Figure A.2 shows several snapshots during the simulation. Some of the objects describe non-straight trajectories to test the robustness of the navigation techniques. In such a scenario, with such amount of moving objects, the velocity space is usually complety occupied and thus the robot has to wait until the moving objects has passed.

Figure A.1 describes the linear and angular velocity profiles during the simulation. The robot tries to maintain the maximum linear velocity when it is possible. However, it has to slow down several times during the simulation and even stop to assure the safety. At the end of the simulation, when almost all the objects have passed the robot can increase the velocity, reaching the goal at maximum linear velocity.

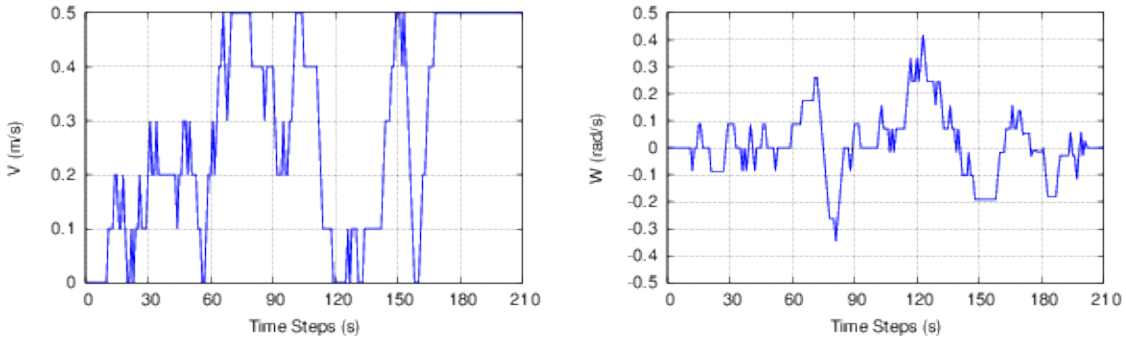


Figure A.1: Velocity profiles in the cluttered scenario.

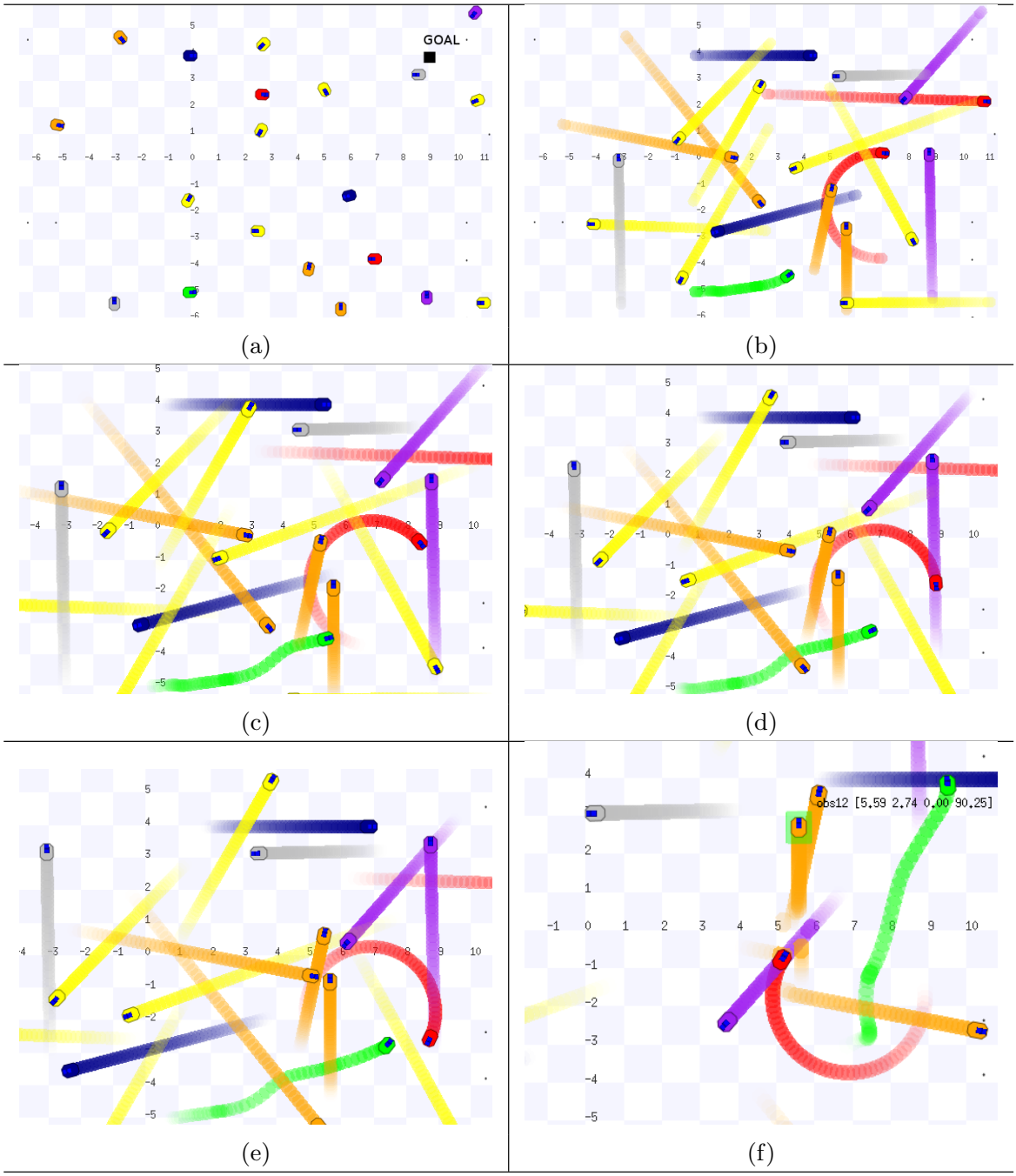


Figure A.2: Snapshots of the simulation in the cluttered scenario

Appendix B

Article: *Seamless indoor-outdoor robust localization for robots*

Seamless Indoor-Outdoor Robust Localization for Robots*

P. Urcola, M. T. Lorente, J. L. Villarroel, and L. Montano

Instituto de Investigación en Ingeniería de Aragón
Universidad de Zaragoza
{urcola, mlorente, jlvilla, montano}@unizar.es

Abstract. In this paper we present a unified localization technique for indoor-outdoor environments that allows a seamless transition between a mapped zone using laser rangefinder on-board sensors and a GPS based localization zone. Different situations are detected during the indoor-outdoor transitions, in which the sensors used change and the localization estimator has to manage them properly for a continuous localization. The quality in the GPS measurements and the zone where the robot is localized are used to determine the best instant for switching the localization parameters for adapting to the situations.

1 Introduction

It is common in indoor robotics applications to assume a limited environment (a room, a building floor, etc) for localization purposes. This limitation is forced by the need of a finite map of the features in the zone to localize the robot. There are many map-based methods for robot localization (see [1] for a selection) that use different sensors achieving good results.

Due to the sparseness of the features needed to get a reliable localization system, the use of maps in outdoor scenarios is uncommon. Instead, outdoor applications usually utilize GPS based localization which avoids any limitation in the environment as it is accessible almost everywhere.

However, there are few systems that provide a continuous localization for both indoor and outdoor scenarios in such a way that the robot is not confined in a limited space.

Main difficulties come from the fact that very different sensors (odometry, IMU, rangefinders, GPS, etc) are needed to get a good estimation. The outputs of some of them completely change depending on the situation of the robot: the GPS gets no measurements indoors, open spaces make rangefinders useless, varying magnetic fields influence compass measurements, etc. Also, the use of different reference frames requires a special consideration of the measurements. All these problems become more obvious during the transitions as measurements are more imprecise.

* This work was partially supported by the Spanish project DPI2012-32100 and by Project DGA T04-FSE.

Some authors have adapted techniques from cell phones and use WPAN and WLAN based localization [2]. These approaches rely on a fixed and known infraestructure and provide results with an accuracy that may not be enough for autonomous robots.

Some approaches, such as [3], rely on the ability of a GPS based Kalman filter to deal with temporary loss of measurements. However, as the duration of this temporary loss increases, the estimation becomes more uncertain.

For transition detection, [4] uses a learning scheme using vision that allows to differenciate indoor and outdoor light characteristics.

In [5], the authors integrate, by means of a Kalman filter, a laser-based SLAM method with GPS, IMU and odometry measurements for a continuous localization. However, EKF scheme assumes independence between measurements but SLAM output and GPS measurements are highly correlated as they are both estimations of the same variables of the robot state. These independence assumptions, when real correlations are high, lead to incoherent estimations because they are overconfident. Thus, in our approach, we use the covariance intersection method [6] to avoid these problems.

In situations where compass cannot be used, the orientation of the robot cannot be directly measured from GPS and has to be derived from other measurements. [7] proposes a Kalman Filter to fuse odometry/IMU and GPS measurements, and the result is integrated into a Monte Carlo localization approach. To estimate the orientation of the robot they use consecutive GPS measurements, assuming that the robot is moving straight in between. This assumption is too strong to be considered in a general scenario. In [8], using the same estimation, the authors determine the likelihood of the estimated orientation by comparing it to the estimation obtained from the IMU. If they diverge, GPS based estimation is discarded. However, this technique would accept a bad estimation that is coherent with a bad IMU measurement. Thus, in our approach we propose a method to calculate the orientation of the robot from several GPS measurements whenever the robot moves straightforward to improve the estimation of the filter.

In real applications where indoor and outdoor scenarios are mixed, it is necessary to have a robust continuous localization without discontinuities when the scenario changes (indoor to outdoor, or vice versa) or when the most accurate sensors have to be chosen. In this paper, we contribute a unified framework for a seamless localization during navigation within different types of environments.

2 Scenario and sensors

The continuous localization system for robots has to work in a scenario composed by zones with different characteristics:

- Indoor and outdoor mapped zones.
- Zones with no GPS coverage.
- Zones covered with imprecise GPS.
- Zones with good GPS coverage.

These zones overlap each other in such a way that the mapped area contains zones of different GPS coverage or not coverage at all. The scenario may not be completely mapped, but the position of all the mapped zones in GPS coordinates is known.

To estimate the localization and orientation of the robot in 2D we have three sources of information:

- Odometry plus IMU with high frequency in the whole scenario, in the sequel odometry/IMU.
- Laser rangefinder based localization in the mapped zone, as proposed in [9].
- GPS based localization.

3 Situations in the environment

We have defined five different situations in which the robot may be, depending on the characteristics of the environment. Each of these situations requires a different measurement policy, as they present different characteristics.

3.1 Description of situations

Indoors. When the robot is in an indoor situation, GPS is unavailable and the localization completely relies on the map-based technique.

Transition Indoors–Outdoors. Inside the map. The robot starts receiving GPS signals, but they are too bad to be used. As the robot is inside the map, the map-based localization can still be used although the uncertainty will be bigger as the robot is leaving the map.

Transition Outdoors–Indoors. Inside the map. In this situation, the GPS measurements become worse and they cannot be used anymore. However, as the robot is inside the map, the map-based localization can be used instead. But previously, we need to give a prior of the position of the robot to the indoor localization because after a while being out of the map, the indoor localization becomes corrupted due to the lack of rangefinder measurements. This action is called a *reset*. The prior information is taken from the current estimation of the robot localization computed using the GPS.

Outdoors. Outside the map with bad GPS. The robot is outside the map, so the indoor localization cannot be used anymore. But there is no GPS signal or it is not good enough to be used so that the robot cannot use any measurement but the prediction from the odometry/IMU.

Outdoors. Outside the map with good GPS. The robot receives good enough GPS measurements and thus they are used to estimate the position of the robot. Notice that the uncertainty of the GPS is still being considered during update. Depending on coverage conditions and on the kind of path, the GPS measurements can be used or not to compute orientation.

3.2 Detection of situations

To detect in which situation the robot is, we use two variables: the GPS quality and the estimated position of the robot at that moment.

Receiving a GPS signal, even with an estimation of the measurement error, it is not enough to consider it as useful. In zones with bad GPS coverage (near high walls, below trees, etc), GPS provides awful estimations moreover if there were no previous good GPS estimations. In these cases, if the robot provides good enough prediction via odometry and IMU, it is better to discard the measurements rather than integrate them into the filter. To decide if a GPS measurement is useful, we define a quality value Q_{gps} which is computed from the variances of the latitude and the longitude measured using (1).

$$Q_{gps} = -log(\sigma_{lat}^2 \sigma_{lon}^2) \tag{1}$$

This quality measurement has been adopted after some experimental tests in different scenarios and it is related to the GPS covariance: the quality increases as the volume of uncertainty decreases. The number of satellites has also been considered as a quality measurement. However, the latter is more volatile than the first one and the uncertainty volume presents bigger discontinuities in the indoor-outdoor transitions and vice versa, as shown in Figure 1.

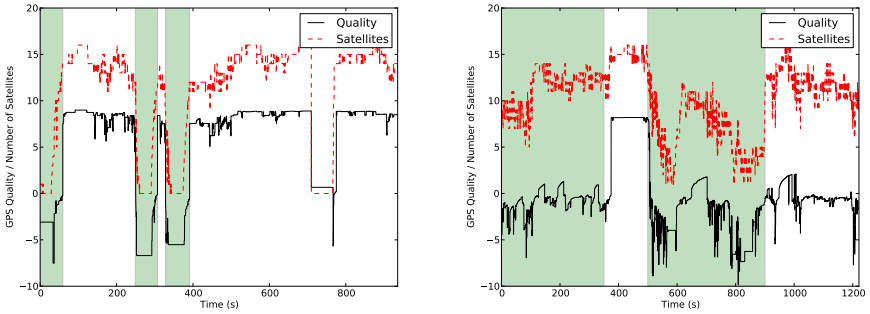


Fig. 1: GPS Quality and number of satellites during two experiments in different scenarios. Shadowed zoned represent indoor zones

However, this is not enough to decide if the robot is indoors or not. It is possible that we receive no GPS at all (see Figure 1 on the left round $t = 700\,s$)

or a really bad quality GPS (as in Figure 1 on the right after $t = 900\,s$) being outside a building. In these cases, we use the best estimation of the localization of the robot to know if it is inside a building or not. We are assuming that we know where the buildings are because they have been previously mapped.

Table 1 sums up the detection of the situations depending on the observed variables.

Table 1: Situation detection

	Good GPS	Bad GPS	No GPS
Inside Map	Outdoors	Transition	Indoors
Outside Map	Outdoors	Outdoors (No GPS)	

4 Continuous Localization

The first problem to deal with when considering different sources of measurements is that each source has a different frame of reference. Thus, we have defined a common frame where all the measurements are transformed into. This frame is defined in such a position that we can estimate its localization using both the map and the GPS.

The method proposed is based on a time discretized bayesian filter approach working in all those changing situations seamlessly. The measurements from different sources are weighted depending on the situation of the robot and on their own covariance. The localization of the robot at time t_k, $\mathbf{x}_k = (x, y, \theta)_k$ and its covariance $\mathbf{P}_k$ is tracked. The filter is divided in two steps, prediction and update.

4.1 Prediction

In the prediction step, the information from the odometry/IMU source $\mathbf{y}_{od}$ is used as the estimation of the movement of the robot since the last measurement from GPS or laser based localization. Odometry/IMU measurements accumulate error as time evolves so that we use $\mathbf{u}_{od}$ the relative movement with respect to a previous measurement instead of the absolute value. This phase corresponds to the prediction step of a Extended Kalman Filter [10], where $\mathbf{F}$ is the jacobian of the prediction function (2).

$$\begin{aligned} \mathbf{x}(k+1|k) &= \mathbf{x}(k|k) \oplus \mathbf{u}_{od}(k+1) \\ \mathbf{u}_{od}(k+1) &= \mathbf{y}_{od}(k+1) \oplus \mathbf{y}_{od}^{-1}(k) \\ \mathbf{P}(k+1|k) &= \mathbf{F}(k)\mathbf{P}(k|k)\mathbf{F}(k)' + \mathbf{Q}(k+1) \end{aligned} \tag{2}$$

Symbol $\oplus$ represents the composition of affine transformations in the plane (translation and rotation) in homogeneous coordinates. As odometry/IMU measurements are always available, the prediction step is performed at each time

step. The uncertainty of the localization increases because of the odometry error $\mathbf{Q}$.

4.2 Update

In the update phase, the measurements from GPS $\langle \mathbf{y}_{gps}, \mathbf{R}_{gps} \rangle$ and from laser-based localization $\langle \mathbf{y}_{map}, \mathbf{R}_{map} \rangle$ are used to correct the predictions made in the previous phase. In case of the map-based localization measurements, the particle filter technique described in [9] is used. According with that approach, $\mathbf{y}_{map}$ is the mass center of the most promising cluster of particles and $\mathbf{R}_{map}$ is the population variance of the particles in the cluster.

All the measurements have three components $\mathbf{y} = (x, y, \theta)$, which corresponds to the components of the state being tracked. Odometry and map based localization provide orientation information but GPS data do not give it directly. In Section 5 we present a procedure to estimate it in some cases.

The general framework we use (3) is based on the covariance intersection filter [6]. This formulation permits the fusion of measurements from different sources that may be correlated, avoiding the statistical independence requirements of EKF approaches. The parameters γ_{map}, γ_{gps}, γ_{od} weight the covariances and their selection is described later, in section 4.3.

$$\begin{aligned} \mathbf{P}(k+1|k+1)^{-1} &= \gamma_{map}\mathbf{R}_{map}^{-1} + \gamma_{gps}\mathbf{R}_{gps}^{-1} + \gamma_{od}\mathbf{P}^{-1}(k+1|k) \\ \mathbf{x}(k+1|k+1) &= \mathbf{P}(k+1|k+1)(\gamma_{map}\mathbf{R}_{map}^{-1}\mathbf{y}_{map} + \\ &\qquad \gamma_{gps}\mathbf{R}_{gps}^{-1}\mathbf{y}_{gps} + \\ &\qquad \gamma_{od}\mathbf{P}^{-1}(k+1|k)\mathbf{x}(k+1|k)) \end{aligned} \tag{3}$$

4.3 Update Tuning

Depending on the situation of the robot, the weighting values $\gamma_{map}, \gamma_{gps}, \gamma_{od}$ in (3) take different values so that the final estimation relies on a mixture of the measurements and the prediction. In addition to that, the measurement covariances are also used to weight the estimation, giving more importance to smaller covariance estimations.

We set to 0.0 the weights of the sensors that are unreliable in each situation (e.g. GPS is set to 0.0 in indoor situations). The rest of the weights are equally set so that the sum of all of them is 1.0. This way, the only weighting factor for the measurements are the covariance matrices of the reliable sensors. The specific weights for the measurements in each situation are defined in Table 2.

5 Outdoor Orientation

Outdoors, the GPS provides the robot position. However, it does not directly provide the orientation of the robot. We propose a technique to estimate the orientation of the robot based on a set of GPS measurements.

Table 2: Update weights in different situations

Situation	γ_{map}	γ_{gps}	γ_{od}	Reset
Indoors	0.5	0.0	0.5	No
In–Out (inside map)	0.5	0.0	0.5	No
Out–In (inside map)	0.5	0.0	0.5	Yes
Outdoors (no GPS)	0.0	0.0	1	No
Outdoors (good GPS)	0.0	0.5	0.5	No

During a distance L in which the robot moves straightforward, i.e. the steering angle is 0 and the orientation θ remains constant, different GPS measurements are stored. For each measurement, an estimation of the orientation $\hat{\theta}_j$ is calculated using (4). The covariances of the measurements $\mathbf{R}_{gps_0}$ and $\mathbf{R}_{gps_j}$ are propagated by means of the jacobian to get the covariance of the orientation σ_j. Then, once the robot has reached the distance L, a final estimation of the robot's orientation is calculated (5) as a weighted sum of the individual computed estimations (Figure 2), based on a mixture probability density function defined in [10]. The weights are valued depending on the covariance of each estimation (6).

$$\hat{\theta}_j = \arctan(\frac{\mathbf{y}_{gps_j} - \mathbf{y}_{gps_0}}{\mathbf{x}_{gps_j} - \mathbf{x}_{gps_0}}) \tag{4}$$

$$\hat{\theta} = \sum_{j=1}^{n} w_j \hat{\theta}_j,\ \sigma^2 = \sum_{j=1}^{n} w_j \left(\sigma_j^2 + \left(\hat{\theta}_j - \hat{\theta} \right)^2 \right) \tag{5}$$

$$w_j = \frac{1}{n-1} \left(1 - \frac{\sigma_j^2}{\sum_{k=1}^{n} \sigma_k^2} \right) \tag{6}$$

Parameter L may be tuned for each scenario. As L increases, more GPS measurements are used to compute orientation, leading to a more accurate estimation. However, it requires that the robot moves long straight trajectories. After several experimental tests, we have set $L = 2.5\,m$ as a trade-off between accuracy and the need of long straight trajectories.

The measurement terms $\langle \mathbf{y}_{gps}, \mathbf{R}_{gps} \rangle$ used in the update phase in (3) are finally obtained as follows:

1. If the robot moves in a straight line (an orientation estimation can be computed),

$$\mathbf{y}_{gps} = \begin{pmatrix} x_{gps} \\ y_{gps} \\ \hat{\theta} \end{pmatrix},\ \mathbf{R}_{gps} = \begin{pmatrix} R_{x_{gps}} & 0 & -\frac{w_n}{L} R_{x_{gps}} \sin\hat{\theta} \\ 0 & R_{y_{gps}} & \frac{w_n}{L} R_{y_{gps}} \cos\hat{\theta} \\ -\frac{w_n}{L} R_{x_{gps}} \sin\hat{\theta} & \frac{w_n}{L} R_{y_{gps}} \cos\hat{\theta} & \sigma^2 \end{pmatrix}$$

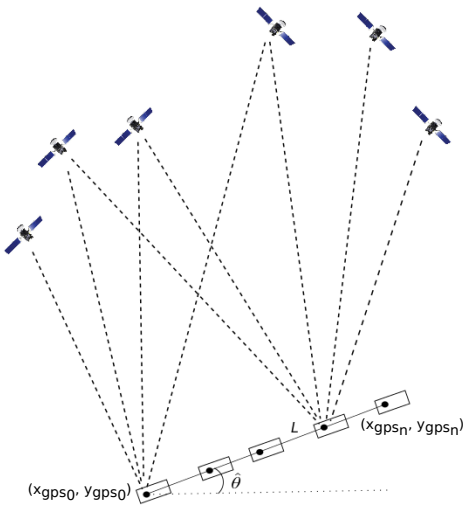


Fig. 2: Orientation of the robot obtained from GPS measurements

2. If the robot does not move in straight line the orientation is taken from the prediction step (given by odometry/IMU), and then,

$$\mathbf{y}_{gps} = \begin{pmatrix} x_{gps} \\ y_{gps} \\ 0 \end{pmatrix}, \mathbf{R}_{gps}^{-1} = \begin{pmatrix} \frac{1}{R_{x_{gps}}} & 0 & 0 \\ 0 & \frac{1}{R_{y_{gps}}} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Note that the error in the position components (x_{gps}, y_{gps}) in GPS measurements are uncorrelated. However, as the orientation is computed from those components, there are non-zero correlation terms when the robot moves in straight line.

6 Results

In this section we present the experimental results obtained to show the robustness of the methods proposed above.

The platform used is a Robucar-TT[1] with all the sensors needed on-board and car-like motion capabilities (see Figure 3).

First, to show the capabilities of the outdoor localization method in good GPS measurement conditions, including orientation estimation, we designed an experiment in which the robot autonomously navigates describing a rectangle defined by four goals on its corners.

The navigation technique is an adaptation of the ORM technique [11] for car-like vehicles. It takes into account the kinodynamic constraints of the robot

[1] www.robosoft.com

Fig. 3: Robucar-TT platform equipped with IMU, odometry, three range-finder sensors and GPS receiver

and permits maneuvers to avoid obstacles and to guide the robot towards the goal. The details of the navigation technique are out of the scope of this paper.

The goals are defined by GPS coordinates so that the localization should permit the robot to repeat the rectangular path with no drift although the odometry accumulates significant errors.

Figure 4 shows the trajectory followed by the robot completing 11 times the rectangular path for a total distance travelled of $2696\,m$ with an average speed of $0.95\,ms^{-1}$.

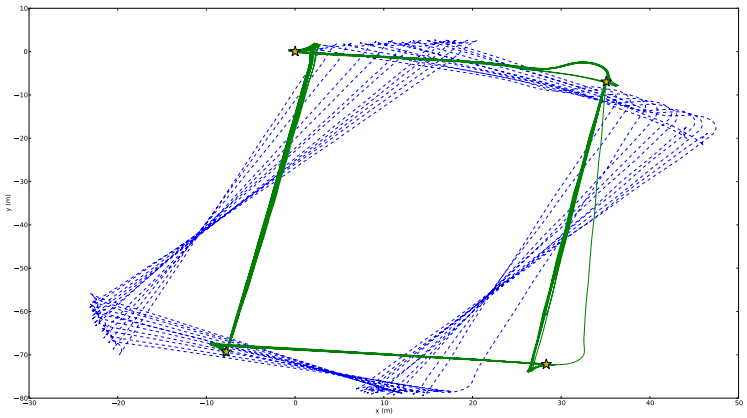


Fig. 4: Estimated localization (solid line) and odometry measurements (dashed line) for the rectangle experiment. The goals are represented by stars.

Figure 5 presents the evolution of the estimation drift during the experiment. We compare the filter localization estimation from the odometry/IMU and the GPS-based one. As known, odometry drift keeps growing with time, being the orientation error the main cause of the position error as well. At the end of the experiment, the odometry reaches a maximum of $18\,m$ of position error and more

than $0.3\,rad$ of orientation error. Meanwhile, the estimation of the localization method is able to keep bounded the error to a small value.

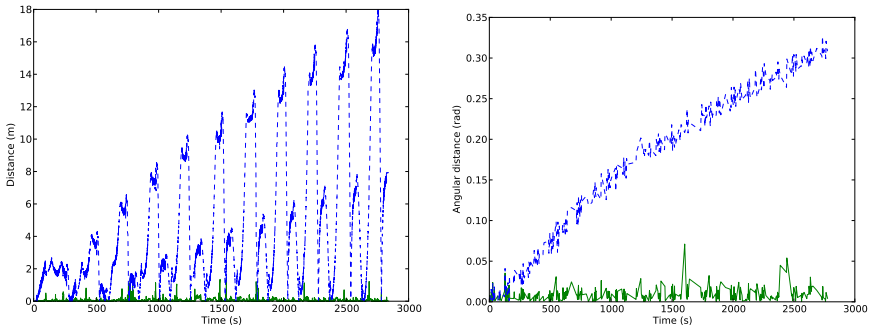


Fig. 5: Position (left) and orientation (right) error of the filter estimation (solid line) and the odometry (dashed line) during the experiment

Second, we designed a full experiment involving all kinds of situations and transitions. Figure 6 shows the trace of the robot during the experiment. The scenario covers an area of $12000\,m^2$ and the robot navigates for more than 1000 meters. In the zones with good localization, the variance of the estimation was below $0.1\,m$ for x and y and below $0.05\,rads$ for the orientation. The maxima in the variances of the estimations occur during transitions, the estimations are reset as well as during the outdoor with no GPS measurements due to the odometry error accumulation. However, transitions are temporal situations and, soon as a measurement is received, variances return to low values. In case of indoor-outdoor transitions, the covariance of the orientation may remain in high values for a longer period because the robot can only measure its orientation when it is moving in straight lines.

Figure 7 show in detail some transition examples of the experiment. On the top left figure, the robot starts inside the building with map-based localization. After leaving the building, the GPS data arrive but their quality is too low to be useful, mainly because the robot navigates close to a wall. The robot leaves the map but still the GPS is imprecise so the odometry/IMU prediction is used until a good enough GPS signal is received.

On the top right, all the trace remains inside the map limits. The robot navigates towards the building and when it is in front of the door, the GPS quality dramatically drops down and thus an outdoor-indoor transition is detected. In that moment, the map-based localization is reset, corresponding to a covariance enlargement. After a while inside the indoor zone, where the robot is continuously localized in the map by means of the laser and odometry/IMU measurements, the robot leaves the building and, when the GPS signal is good enough, an indoor-outdoor transition is detected and the localization system adapts to the outdoor situation.

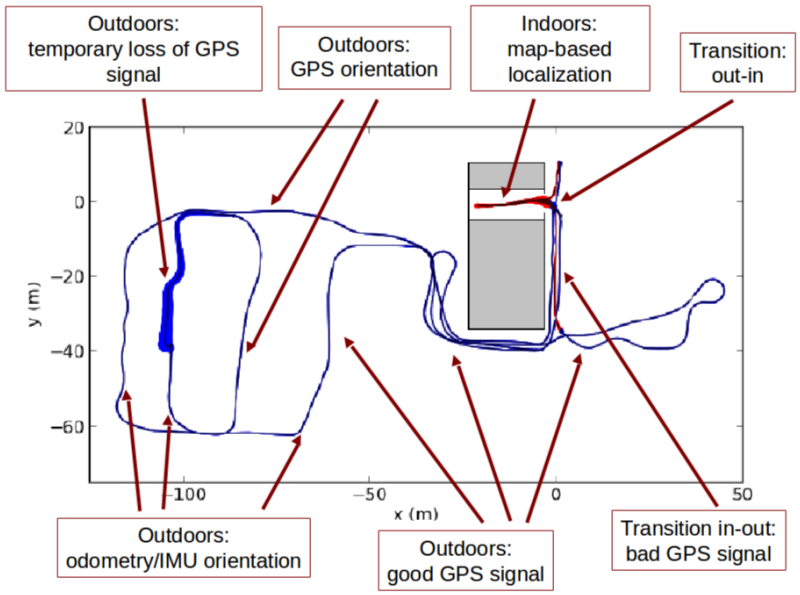


Fig. 6: A complete experiment. All the transitions are shown in which the filter works in the continuous localization.

As can be seen in the second row of Figure 7, localization continuity is kept along the experiment. Only around transitions, some discontinuities may be found as transitions take place in zones where some of the sensors start to provide bad measurements. However, the biggest gap detected is about $0.5\,m$ in x coordinate. A bigger discontinuity was found in $t = 250\,s$ (Figure 7 bottom right) caused by a spurious bad GPS measurement.

7 Conclusions

We present a mobile robot localization method that allows a robot to seamlessly switch between indoor and outdoor environments. We propose an outdoor GPS-based localization method which is able to estimate the position and orientation of the robot bounding the drifting error obtained from the odometry/IMU measurements.

By using a GPS quality estimation and the robot localization (in map or not), we can determine which is the situation of the robot and select the measurements to achieve the best continuous localization.

All the contributions have been tested in different experiments with a car-like platform. The data obtained in those experiments validates their functionality and robustness.

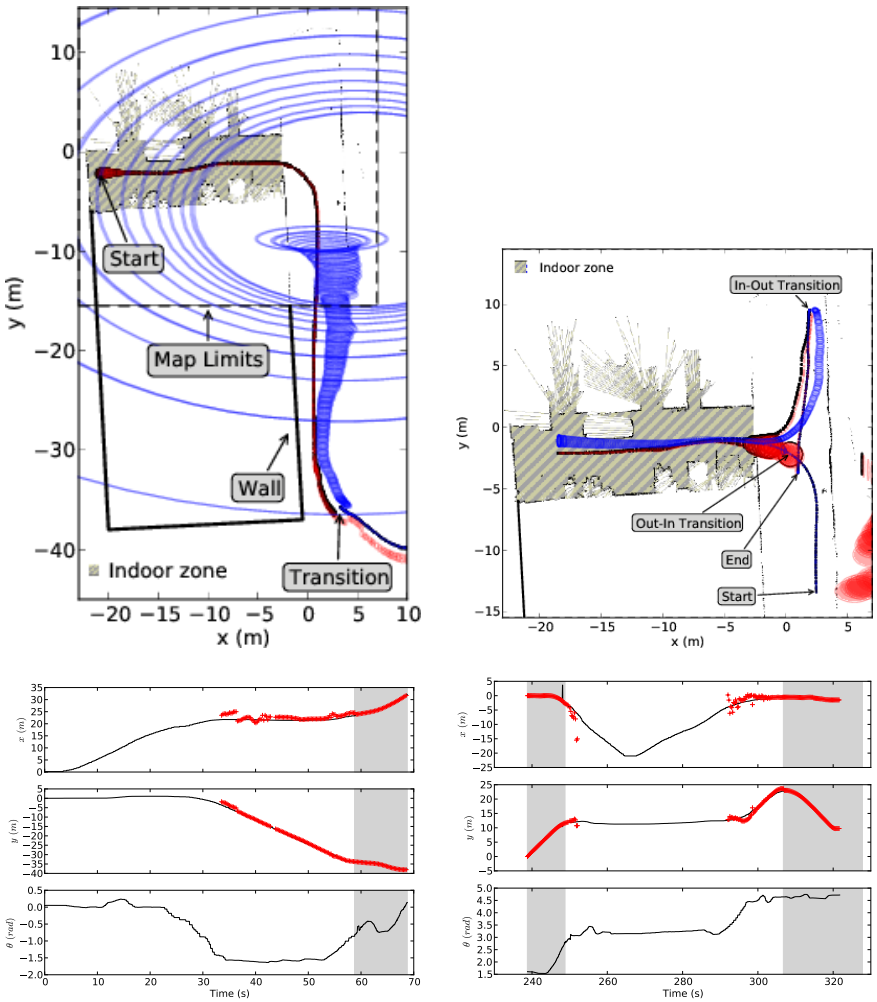


Fig. 7: In the first row, two examples of transitions where it is shown the GPS-based localization (blue), the laser-based localization (red) and the estimation using our method (black). The ellipses represent the uncertainty. In the second row the state variables (in black) during these transitions are depicted. Red crosses represent the GPS data received. The background color indicates whether the robot is using the GPS (gray) or not (white) for localization.

In future works, a smoother approach to the transition method can be studied, by using the weights in the covariance intersection framework to minimize the gaps during transitions. Also, an study of the limits of the GPS-based orientation method will be performed to determine the best parameters such as maximum curvature or minimum distance needed to get a good estimation.

References

1. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. The MIT Press (2005)
2. Kang, J., Kim, D., Kim, E., Kim, Y., Yoo, S., Wi, D.: Seamless mobile robot localization service framework for integrated localization systems. In: 3rd International Symposium on Wireless Pervasive Computing (ISWPC). (May 2008) 175–179
3. Goel, P., Roumeliotis, S., Sukhatme, G.: Robust localization using relative and absolute position estimates. In: Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on. Volume 2. (1999) 1134–1140
4. Collier, J., Ramirez-Serrano, A.: Environment classification for indoor/outdoor robotic matching. In: Canadian Conference on Computer and Robot Vision. (2009) 276–283
5. Pacis, E.B., Sights, B., Ahuja, G., Kogut, G., Everett, H.R.: An adapting localization system for outdoor/indoor navigation. In: SPIE Proc. 6230: Unmanned Systems Technology VIII, Defense Security Symposium, Orlando, EEUU (April 2006)
6. Julier, S.J., Uhlmann, J.K.: Using covariance intersection for SLAM. Robotics and Autonomous Systems (55) (2007) 3–20
7. Hentschel, M., Wulf, O., Wagner, B.: A gps and laser-based localization for urban and non-urban outdoor environments. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on. (2008) 149–154
8. Gu, D., El-Sheimy, N.: Heading accuracy improvement of mems imu/dgps integrated navigation system for land vehicle. In: Position, Location and Navigation Symposium, 2008 IEEE/ION. (2008) 1292–1296
9. Fox, D.: Adapting the sample size in particle filters through KLD-sampling. The International Journal of Robotics Research **22**(12) (December 2003) 985–1003
10. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: Estimation with applications to tracking and navigation. John Wiley & Sons, Inc. (2001)
11. Minguez, J.: The obstacle-restriction method for robot obstacle avoidance in difficult environments. In: Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on. (2005) 2284–2290

List of Figures

2.1 Workspace 8
2.2 *DOVTS* space 8
2.3 Merged objects in *DOVS* 10
2.4 Static objects in *DOVS* 11
2.5 Decision strategies 12
2.6 Decision variables on *DOVS* 13
2.7 Situation tree 14
2.8 *PassingBefore* situation 16
2.9 *PassingAligned* situation 16
2.10 *SlowingDown* situation 17
2.11 *AvoidingObject* situation 18
2.12 Velocity goals in *DOVTS* 19
2.13 *Cost-Optimization* and *Strategies-Optimization* simulations 20

3.1 Evaluation scenario 23
3.2 Metrics assessment 24

4.1 Improving GPS-based orientation technique 29
4.2 Minimum distance d 29
4.3 Distance L 30
4.4 Robucar-TT platform 31
4.5 Experiment. Localization evaluation 32
4.6 Experiment. New method for GPS-based orientation 32

A.1 Simulation - cluttered scenario. Velocity profiles 35
A.2 Simulation - cluttered scenario. Snapshots 36

List of Tables

2.1 Decision variables 13

3.1 Evaluation metrics 22
3.2 Combinations of α parameters 22

4.1 Situation detection 26
4.2 Update weights in different situations 27

Bibliography

[1] P. Fiorini and Z. Shiller. Robot motion planning in dynamic environments. In *International Symposium of Robotic Research*, pages 237–248, 1995.

[2] Dieter Fox. Adapting the sample size in particle filters through KLD-sampling. *The International Journal of Robotics Research*, 22(12):985–1003, December 2003.

[3] T. Fraichard. Dynamic trajectory planning with dynamic constraints: A 'state-time space' approach. In *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on*, volume 2, pages 1393–1400 vol.2, 1993.

[4] T. Fraichard and H. Asama. Inevitable collision states. a step towards safer robots? In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 388–393 vol.1, 2003.

[5] E. Frazzoli, M.A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. In *American Control Conference, 2001. Proceedings of the 2001*, volume 1, pages 43–49 vol.1, 2001.

[6] Dongqing Gu and N. El-Sheimy. Heading accuracy improvement of mems imu/dgps integrated navigation system for land vehicle. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 1292–1296, 2008.

[7] M. Hentschel, O. Wulf, and B. Wagner. A gps and laser-based localization for urban and non-urban outdoor environments. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 149–154, 2008.

[8] S. J. Julier and J. K. Uhlmann. Using covariance intersection for SLAM. *Robotics and Autonomous Systems*, (55):3–20, 2007.

[9] J. Kuffner and S. LaValle. Randomize kinodynamic planning. *International Journal of Robotics Research*, (6):378–400, 2001.

[10] M. T. Lorente. Planificación de la navegación en entornos dinámicos. Master's thesis, Escuela de Ingeniería y Arquitectura, Universidad de Zaragoza, 2011.

[11] C. Menéndez. Navegación de robots autónomos en entornos dinámicos. Master's thesis, Escuela de Ingeniería y Arquitectura, Universidad de Zaragoza, 2012.

[12] J. Minguez and L. Montano. Sensor-based motion generation in unknown, dynamic and troublesome scenarios. *Robotics and Autonomous Systems*, 52(4):290–311, 2005.

[13] Montano L. Montesano L., Minguez J. Modeling dynamic scenarios for local sensor-based motion planning. *Autonomous Robots*, (25(3)):231–251, 2008.

[14] E. Owen. Planificación de movimientos en entornos dinámicos usando Objectos Dinámicos de Velocidad. Master's thesis, Escuela de Ingeniería y Arquitectura, Universidad de Zaragoza, 2012.

[15] E. Owen, C. Menéndez, M. T. Lorente, and L. Montano. Strategies-optimization: Robocentric motion planning for safe navigation in dynamic environments. *Autonomous Robots*. Submitted.

[16] E. Owen and L. Montano. Motion planning in dynamic environments using the velocity space. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 997–1002, 2005.

[17] E. B. Pacis, B. Sights, G. Ahuja, G. Kogut, and H. R. Everett. An adapting localization system for outdoor/indoor navigation. In *SPIE Proc. 6230: Unmanned Systems Technology VIII, Defense Security Symposium*, Orlando, EEUU, April 2006.

[18] Z. Shiller, F. Large, and S. Sekhavat. Motion planning in dynamic environments: obstacles moving along arbitrary trajectories. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, pages 3716–3721 vol.4, 2001.

[19] Fraichard T. Shiller Z., Gal O.

[20] P. Urcola, M. T. Lorente, and L. Montano. Robust navigation and seamless localization for car-like robots in indoor-outdoor environments. In *Robotics and Automation, 2014. IEEE International Conference on*, 2014. Submitted.

[21] P. Urcola, M. T. Lorente, J.L. Villarroel, and L. Montano. Seamless indoor-outdoor robust localization for robots. In *Iberian Robotics Conference*, 2013.