Sensors **2013**, *13*, 2929-2944; doi:10.3390/s130302929

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Dynamic Obstacle Avoidance Using Bayesian Occupancy Filter and Approximate Inference

Ángel Llamazares [1,⋆], Vladimir Ivan [2], Eduardo Molinos [1], Manuel Ocaña [1] and Sethu Vijayakumar [2]

[1] Department of Electronics, University of Alcalá, 28871 Alcalá de Henares, Madrid, Spain; E-Mails: eduardo.molinos@depeca.uah.es (E.M.); mocana@depeca.uah.es (M.O.)

[2] Institute of Perception, Action and Behaviour, University of Edinburgh, Edinburgh EH8 9AB, UK; E-Mails: v.ivan@ed.ac.uk (V.V.); sethu.vijayakumar@ed.ac.uk (S.V.)

***** Author to whom correspondence should be addressed; E-Mail: allamazares@depeca.uah.es; Tel.: +34-918-856-726; Fax: +34-918-856-591.

Received: 31 January 2013; in revised form: 14 February 2013 / Accepted: 16 February 2013 / Published: 1 March 2013

Abstract: The goal of this paper is to solve the problem of dynamic obstacle avoidance for a mobile platform using the stochastic optimal control framework to compute paths that are optimal in terms of safety and energy efficiency under constraints. We propose a three-dimensional extension of the Bayesian Occupancy Filter (BOF) (Coué *et al*. *Int. J. Rob. Res.* **2006**, *25*, 19–30) to deal with the noise in the sensor data, improving the perception stage. We reduce the computational cost of the perception stage by estimating the velocity of each obstacle using optical flow tracking and blob filtering. While several obstacle avoidance systems have been presented in the literature addressing safety and optimality of the robot motion separately, we have applied the approximate inference framework to this problem to combine multiple goals, constraints and priors in a structured way. It is important to remark that the problem involves obstacles that can be moving, therefore classical techniques based on reactive control are not optimal from the point of view of energy consumption. Some experimental results, including comparisons against classical algorithms that highlight the advantages, are presented.

Keywords: autonomous navigation; obstacle avoidance; optimal control

1. Introduction

Autonomous vehicles have nowadays become popular for applications such as surveillance and passenger transport. In both cases the safety and efficiency of these systems is depending on the ability of the autonomous navigation system to deal with unpredictable dynamic changes in the environment.

The autonomous navigation systems have been studied extensively in the literature [1,2]. Such systems should be able to perform perception, localisation, planning and actuation. To perceive the state of the environment the on-board sensors are used. The localisation stage is usually based on fusion of sensory information and a priori map to determine the location of the vehicle within the global frame. Once the vehicle's location has been determined, the sequence of actions necessary to reach the goal can be computed within the planning stage. The resulting plan has to satisfy various constraints such as: holonomic constraints, safety, traffic rules, energy efficiency, *etc*. Finally, the actuation stage is responsible for executing the plan.

The localisation stage can be greatly improved by using Global Positioning System (GPS) or its enhanced version: the Differential GPS (DGPS) [3]. This provides a global reference for vehicle with the accuracy of a few centimetres in the case of DGPS. However, when a GPS receiver is deployed in urban environments with high buildings or underground tunnels, the signal can suffer multipath fading or even Line-Of-Sight (LOS) blockage, which renders this sensor inoperative.

For autonomous vehicles to achieve safe operation, understanding safe operation as an obstacle avoidance task that preserves the integrity of the robot and the other objects or people in a dynamic environment, a combination of a priory map with perception information that comes from sensor fusion [4,5] is required. The vehicle can be then safely guided through a mesh of connected way points. Such maps are usually obtained in a semi-autonomous way [6] or using Simultaneous Localization and Mapping (SLAM) techniques [4,7,8] to reduce the uncertainty of localisation and mapping processes by doing both at the same time.

In addition to localisation, a robust autonomous mobile platform requires an obstacle avoidance system. Such system ensures that a vehicle navigates safely around the obstacles while trying to reach its goal. Obstacle avoidance can be divided into global and local approaches. While the former approach assumes a complete model of the environment, such as the potential field methods, the local methods require only partial observability of the environment at the cost of guaranteeing only local optimality. However, computational cost is much lower for local methods and they can be often implemented in the form of reactive controllers. These reactive methods take control of the robot when an obstacle is detected to prevent collision. They use the nearest portion of the environment modelled using the current sensor observation. Some representative examples are Vector Field Histogram (VFH+) [9], Nearness Diagram (ND) [10], Curvature-Velocity Method (CVM) [11], its improved version the Lane-Curvature Method (LCM) [12] and the Beam-Curvature Method (BCM) [13].

One of the major drawbacks of the reactive methods is that they do not take into account the dynamic changes of the environment and assume that all obstacles are static. Therefore, they can not predict their motion. This is, however, an unrealistic assumption especially when the vehicle deals with a high uncertainty over the position, shape and velocity of the obstacles and it is still a challenge for real world applications [14,15]. It is also crucial to combine the available sensory input in a structured manner. In [16] the authors propose to formulate the planning problem as inference on a graphical model. This framework allows to combine sensory information, constraints and multiple goals in the form of task variables that may be represented in different spaces such as configuration space, end-effector space (typical for reaching tasks) or even abstract spaces such as topology based representations of the environment. Hierarchical task variables can be constructed to provide both low level control at the level of dynamics and high level control at the level of task objectives.

This paper extends our previous work recently sketched in [17]. We propose an obstacle avoidance system that takes into account not only the safety of our vehicle around the dynamic obstacles but also optimality of the motion in terms of additional constraints such as the energy consumption. We demonstrate how dynamic obstacles are treated within our navigation system, and how the planning stage can be improved by solving the problem within a stochastic optimal framework. We reduce the energy consumption of our vehicle by, firstly, using a new probabilistic model of the environment inside the perception stage, and secondly by optimising the trajectory using the Approximate Inference Control (AICO). We compare our proposed method with the classical obstacle avoidance algorithms (VFH+, CVM, LCM and BCM).

The rest of the paper is organized as follows: Section 2 shows the proposed method; Section 3 describes the experiments and the actual results; and finally, in Section 4 we conclude and discuss the future works.

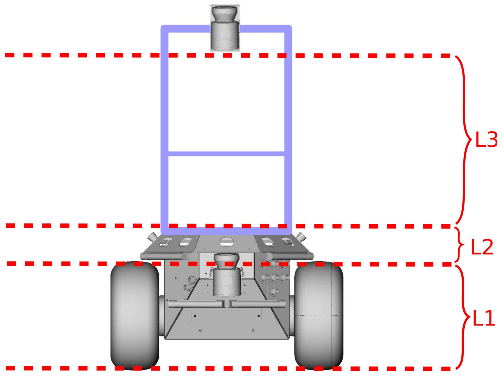
2. Proposed Method

In this section we propose an extension of the method for avoiding dynamic obstacles in three dimensions while minimising the energy consumption. To tackle this problem, we obtain a probabilistic model of the dynamic environment, then we use this model to predict the motion of the obstacles inside our perception stage, and finally, we employ optimal path planning using approximate inference to improve the planning stage based on an energy consumption model.

2.1. Probabilistic Model of the Dynamic Environment

We use the Bayesian Occupancy Filter (BOF) [14] to compute a robust estimate of the position of the obstacles. BOF has been successfully used to detect obstacles in a flat world, but it has not taken into account the detection of obstacles in 3D. In order to add the *height* into the BOF, we use the information provided by our laser range sensor, mounted at an angle with respect the ground plane, as stated in our previous work [18]. The height has been discretised at 3 levels, $L1$ (the wheel height), $L2$ (the mid-body height) and $L3$ (the overhead height), converting the cells into cubes. In order to compute the three-dimensional BOF, all the 3D points inside each cube are projected to the corresponding cell and level. Figure 1 shows the discretisation at three levels with respect to the robot.

Figure 1. Discretisation levels.



The Bayesian Occupancy Filter can be implemented as a loop of a prediction and estimation steps. The authors of [14] suggest that prediction (Equation (1)) and the estimation (Equation (2)) steps can be computed as follows:

$$P(e_n^t|n^t, u^{t-1}) \propto \sum_{n^{t-1}} P(n^t|n^{t-1}, u^{t-1}) P(e_n^{t-1}|n^{t-1}) \tag{1}$$

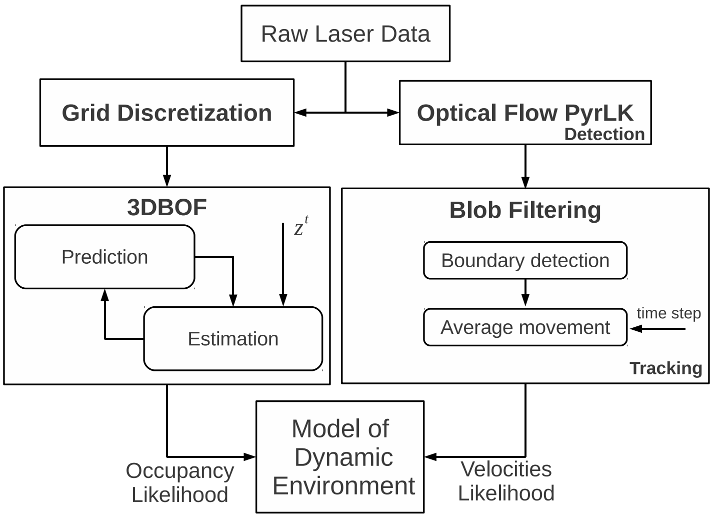
$$P(e_n^t|z^t, n^t) \propto \sum_{m=1}^{S} (\prod_{s=1}^{S} P(z_s^t|e_n^t, m)) \tag{2}$$

where e_n^t is the occupancy of the cell n at time t, u is the command issued at time $t-1$, z are the observations and m is matching between a cell and an observation. $P(n^t|n^{t-1}, u^{t-1})$ is then the transition probability defined by the vehicle dynamics. The standard BOF framework has several issues with the velocity estimation. Firstly, this framework assumes that the velocity of each grid cell is constant [19]. Secondly, the discretisation has to be performed also in the velocity space, meaning that a separate estimate for each pair of velocities (v_x, v_y) is required. This discretisation for a large range of possible velocities together with calculations of static objects result in high computational costs. For these reasons, other authors proposed object detection and clustering techniques to obtain the objects' velocities [20] with an additional constraint that the position of the obstacle has to remain within a bounded neighbourhood.

We have improved the system in terms of efficiency by adding a stage to detect the relative velocities of the obstacles, without assuming constant velocity objects or discretised velocities. Figure 2 shows the flow diagram of our proposed model.

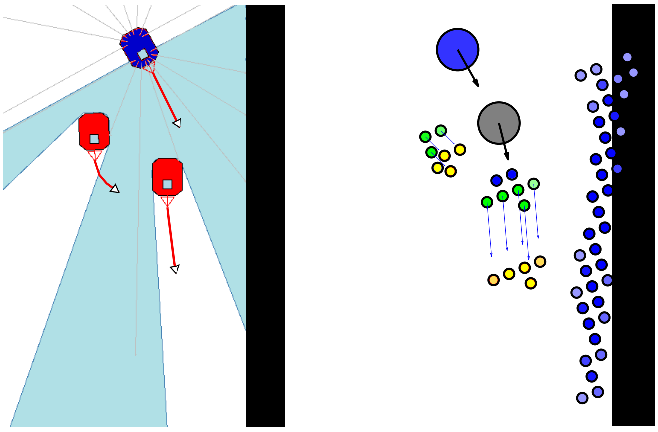
Firstly, we capture a frame using the laser data. This frame has the form of a zenith image of the detected obstacles. We then estimate the movement of the obstacles between two frames by computing the pyramidal implementation of Lukas Kanade optical flow algorithm [21]. Then, we introduce a blob filtering stage in two steps: (1) we detect the boundaries of the objects and (2) we obtain the average motion of all the cells inside each boundary. Each cell has to exceed certain occupancy threshold in order to reduce the noise in the output of optical flow and the computational cost. This average value and the time step are used to compute the relative velocities between the obstacles with respect to the robot's local frame of reference (along local axis x and y), while the velocity along the z axis is not taken into account due to the assumption that the robot moves on a plane. The result of the perception stage is a dynamic occupancy grid providing an estimation of velocity and the occupation probability.

Figure 2. Probabilistic model of the dynamic environment.



The output of our probabilistic model is illustrated in Figure 3. Left image shows the simulated environment. Our robot is represented in blue colour and the moving obstacles are red. Right image shows the sensing results. The blue circle is the robot at the current time step, the grey circle shows the predicted future position of the robot, moving obstacles at current time step are represented with green dots, the predicted future positions of moving obstacles are shown as yellow dots and static obstacles are shown as blue dots. The occupancy probability value is given by the darkness value of each dot.

Figure 3. Simulated environment and sensing results.


(a) Simulated environment. (b) Sensing results.

2.2. Energy Consumption Model

In order to compare our method with classical obstacle avoidance algorithms in terms of energy, we only take into account the power demand of the robot's motors. We assume that the rest of the equipment has constant energy consumption and therefore it cannot be improved any further. We also assume that the power demands of the robot's motors are based on overcoming inertia, road grade, tyre friction, and aerodynamic loss. This road-load methodology was mainly introduced by [22]. The power demand (in Watts) is the tractive power as defined by Equation (3):

$$P = mv[a(1+\varepsilon) + gR_G + gK_R] + \frac{1}{2}\rho K_D A_F v^3 \tag{3}$$

where m is vehicle mass in metric tones (0.077 in our case), v is vehicle speed (assuming no headwind) in m/s, a is vehicle acceleration in m/s^2, ε is a mass factor accounting for the rotational masses and assumed to be 0.1 [23], g is acceleration due to gravity (9.8 m/s^2), R_G is road grade (0.0 in our case), K_R is rolling resistance, this value for radial tires can range from 0.008 to 0.013 for a majority of the on-road passenger car tires but can be larger depending on tire pressure, temperature, ground surface, and speed [24,25] (a medium value in the range $\approx$ 0.009 is assumed [22]), ρ is air density ($\approx$ 1.2 kg/m^3), K_D is aerodynamic drag coefficient ($\approx$ 0.3 [22]) and A_F is the frontal area in square meters ($\approx$ 1 m^2 in our case). These values are obtained based on the references and the robot's specifications, assuming that the goal of this work is to obtain a comparison of energy saving, and not the exact value in each case.

The robot speed that we use to obtain the power demand is provided by the kinematic model of the robot based on the angular speed of the wheels for each time step (100 ms in our case). According to this, we assume that the robot is moving with linear speed between execution steps. On the other hand, our planning and almost all the classical algorithms do not allow the robot to describe sharp turns or spin.

2.3. Optimal Path Planning Using Approximate Inference

We have decided to formulate the path optimisation problem within the Approximate Inference Control (AICO) framework [16]. The state of the robot is defined by $x_t = (r_x, r_y)$, the position of the robot on the ground plane (r_x, r_y) and its derivative in the dynamic case. The transition probability is defined by a linear control process with Gaussian noise:

$$P(x_{t+1}|x_t, u_t) = \mathcal{N}(x_{t+1}|A_t x_t + a_t, Q_t + B_t H^{-1} B_t^T) \tag{4}$$

given state x_t where A_t, a_t, B_t define the linear system that approximates the state transition, Q_t is the covariance of the system noise and H_t is the covariance of the uniform prior over u_t. The control u_t has been integrated out to simplify the equation (refer to [16]). Our goal is to compute a path that minimizes the total expected cost from time t_0 to the final time t_T:

$$C(x_{0:T}, u_{0:T}) = \sum_{t=0}^{T} c_x(x_t) + c_u(u_t) \tag{5}$$

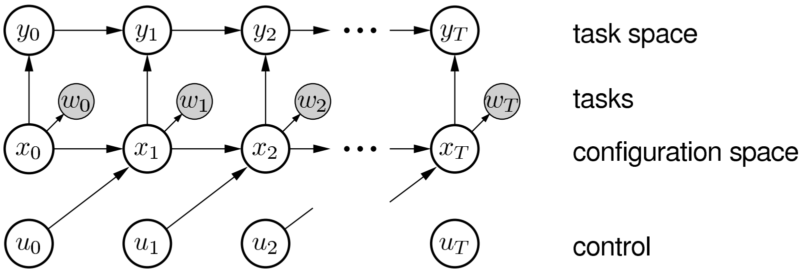
where $c_x(x_t)$ is the state dependent cost (defined by sum of angular velocity and reciprocal distance to static and dynamic obstacles) and $c_u(u_t)$ is the control cost (defined by Equation (3)). The problem can now be described by a graphical model:

$$p(x_{0:T}, u_{0:T}) \propto P(x_0)\prod_{t=0}^{T} P(u_t) \prod_{t=1}^{T} P(x_t|u_{t-1}, x_{t-1}) \prod_{t=0}^{T} P(w_t = 1|x_t) \tag{6}$$

where $P(u_t)$ is the control prior reflecting the control cost and $P(w_t = 1|x_t)$ is the probability of receiving low cost reflected by constantly observing a random variable $w_t = 1$. The binary random variable has the conditional probability $P(w_t = 1|x_t, u_t) = \exp(C(x_{0:T}, u_{0:T}))$. The costs $C(x_{0:T}, u_{0:T})$ play the role of the neg-log-probability of $w_t = 1$, in accordance to the typical identification of energy with neg-log-probability. The distance to obstacles is treated as a separate task and it enters the model through a task variable y_t that represents position in the task space (see Figure 4). For more details on how to couple the task variables with states, refer to [16]. We compute the posterior $P(x_{0:T}|w_{0:T} = 1)$ over the state trajectories to solve the path planning problem using the Gaussian message passing algorithm. This involves combining the forward ($\mu_{x_{t-1}\to x_t}$), backward ($(x_t)\mu_{x_{t+1}\to x_t}$) and cost messages ($(x_t)\mu_{w_t\to x_t}(x_t)$) to compute the posterior marginal belief:

$$b(x_t) = \mu_{x_{t-1}\to x_t}(x_t)\mu_{x_{t+1}\to x_t}(x_t)\mu_{w_t\to x_t}(x_t) \tag{7}$$

Figure 4. Graphical model of AICO in configuration and task space.



The cost function in AICO is defined by task variables. Each task variable defines a task space and a squared metric is used to compute the cost inferred in this space. However, inference-based path planning with the linearised motion model and the holonomic constraint is difficult and suffers from problems with local minima due to the velocity constraints. The reasoning behind this is that the Gaussian distribution over the state space can potentially assign probability mass to states that do not satisfy the holonomic constraint, which either causes sideways slipping in the model or if we constrain the Gaussian itself the distribution becomes degenerate. For this reason, we have excluded the orientation from the state and we have added an additional cost term to penalise for angular velocity instead. We assume that arbitrary angular velocities can be executed but optimise for low angular velocities. This reduces the complexity of the state space by turning the hard holonomic constraint into a soft constraint.

In order to achieve robustness and safety of our vehicle, we use a method that yields free paths which tend to maximise the clearance between the vehicle and the obstacles based on a Voronoi graph. The aim of the global planning is to keep the vehicle at a safe distance from the surrounding obstacles. We compute the initial path using graph search on the Voronoi graph. This path is then used as AICO initialisation and it helps to deal with local minima.

3. Implementation and Results

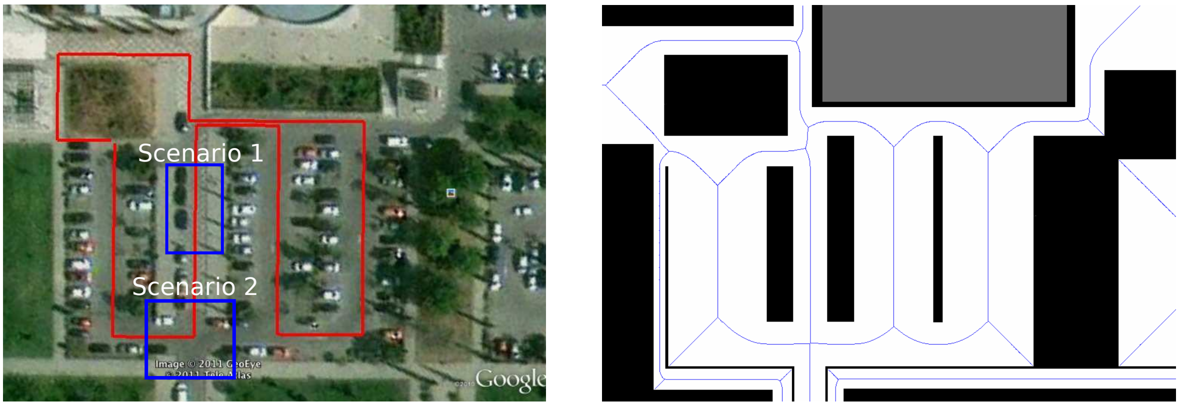
In this section we describe the implementation of our system and the experimental results. The results have been obtained from the real Seekur Jr. platform and the simulator provided by Mobilerobots.

The results have been evaluated in two stages: firstly we evaluate the gain of using the probabilistic model of the environment inside the perception stage of the classical algorithms (VFH+, CVM, LCM and BCM). Secondly, we compare our whole dynamic obstacle avoidance system with the classical algorithms.

3.1. Test-Bed

We have tested our system in an outdoor environment in the South Parking of the Polytechnic School at the University of Alcalá (UAH). The overall area of the environment is approximately 70 × 70 m^2 (Figure 5(a)). In addition, the surveying route has been marked in red colour. The route was a 330 m long, and the blue rectangles represents the scenarios where the system was tested.

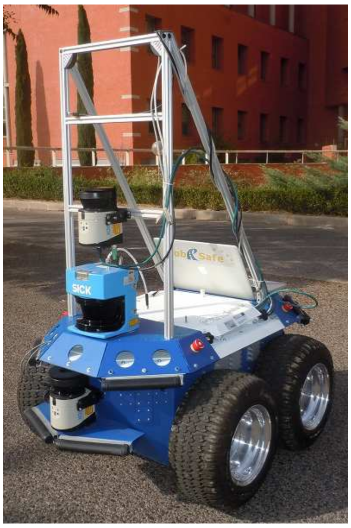
Figure 5. Test environment.



(a) Surveying route and test scenarios in real environment.

(b) Voronoi Diagram.

Figure 6. Seekur Jr. used in the experimentation.



The robot used in the experimentation was a Seekur Jr. by Mobilerobots, with the following configuration: MacBook Pro with Ubuntu 12.04 LTS operating system, Aria/MobileSim control software, RTK-GPS Maxor GGDT by JAVAD, low-cost GPS and stereo camera, two SICK LMS 151 outdoor lasers (the first one parallel to the ground and the other mounted at an angle to obtain the 3D points cloud), bumpers, encoders in the wheels and one Inertial Measurement Unit (IMU) to reduce the odometry errors in the turns. Seekur Jr. is shown in Figure 6.

3.2. Results Using the Probabilistic Model of the Dynamic Environment

We have used four classical algorithms on the real platform and analysed the effect of using our proposed probabilistic model. For simplicity we only take into account the $L2$ level of the three-dimensional BOF estimation (at the height of main body of the Seekur). We have tested the robot in two different scenarios and commanded it to reach a goal 7 meters away from its start position:

1. Parallel: For this test the robot is located in the middle of a 5 meters wide corridor (Figure 5(a): Scenario 1). The obstacle starts moving along the corridor in the same direction as the robot but with a delay and it tries to overtake the robot.
2. Perpendicular: For this test the robot is located at the crossroads (Figure 5(a): Scenario 2). The obstacle is moving along the main road perpendicular to the robot's direction, blocking its path.

For all of these experiments, we have analysed the following parameters:

- Path curvature: The assumption is that the smoother the path the lower the energy consumption.
- Acceleration (a): the positive acceleration in (ms^{-2}) ignoring energy regenerated from breaking.
- Velocity (v): the absolute velocity of the robot in (ms^{-1}).
- Time (t): the time needed to reach the goal in (s).
- Energy (E): the energy consumption of the robot in (J).

As an example, the top part of the Figures 7 and 8 shows the path followed by the robot using the VFH+ algorithm and VFH+ with our probabilistic model in the two scenarios. The yellow diamond is the goal to achieve. The plots show that all paths obtained by our proposed probabilistic method are smoother with lower overall curvature.

Figures 9 and 10 show the velocities and accelerations of the robot using the LCM and our probabilistic model in both scenarios. The results show that also the accelerations and velocities over time are much smoother and lower when using our probabilistic model of the environment.

Figure 11 shows the energy consumption of the robot using the VFH+ algorithm in both scenarios. The results show that using our probabilistic model, the energy consumption is reduced by 20%. A summary of energy consumption is shown in Table 1. We can conclude that the use of our probabilistic model of the environment in classical algorithms reduces the energy consumption up to 45.5%.

Figure 7. Path using VFH+ and AICO: parallel scenario.

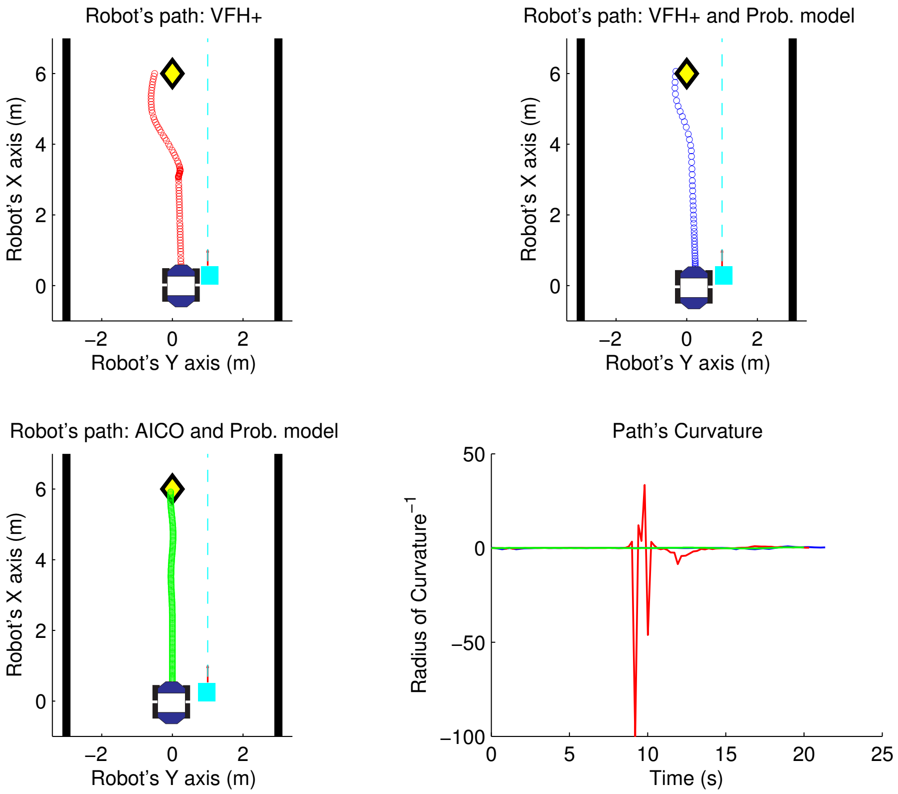


Figure 8. Path using VFH+ and AICO: perpendicular scenario.

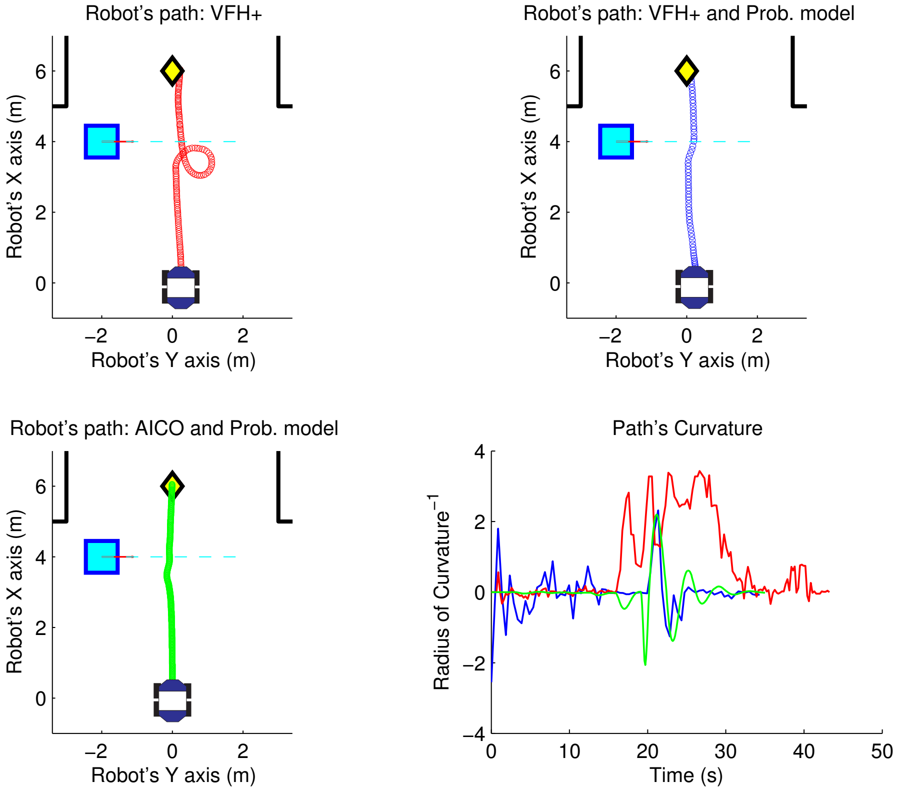


Figure 9. Velocities and accelerations using LCM and AICO: parallel scenario.

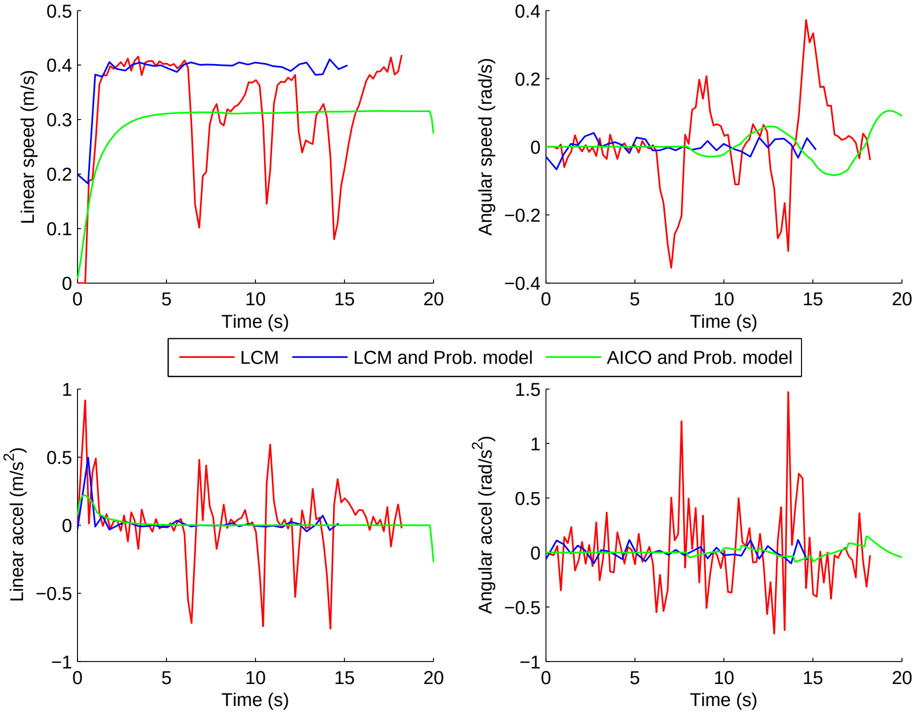


Figure 10. Velocities and accelerations using LCM and AICO: perpendicular scenario.

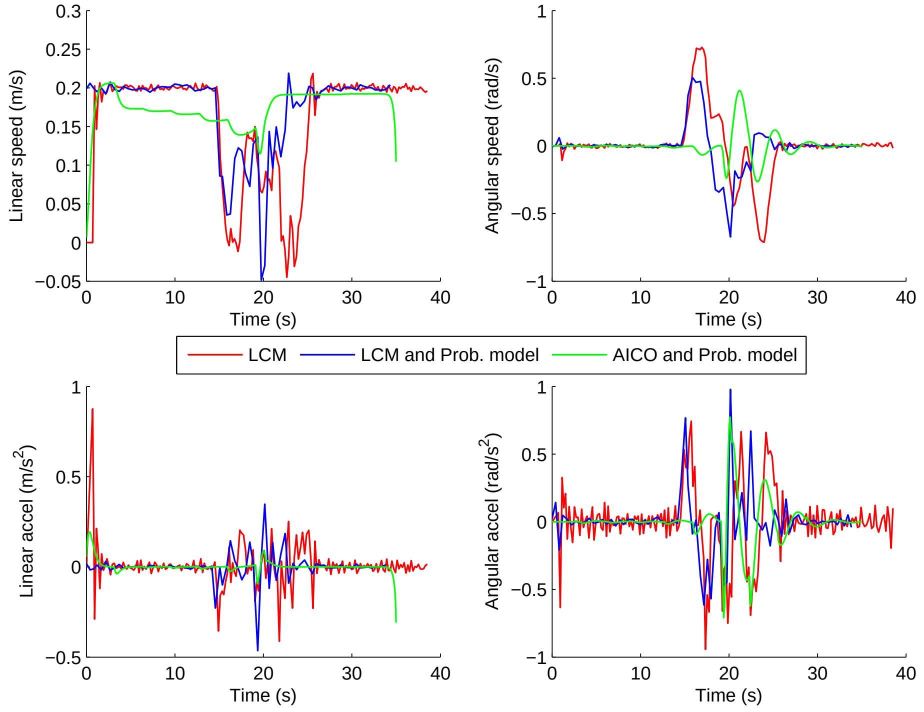
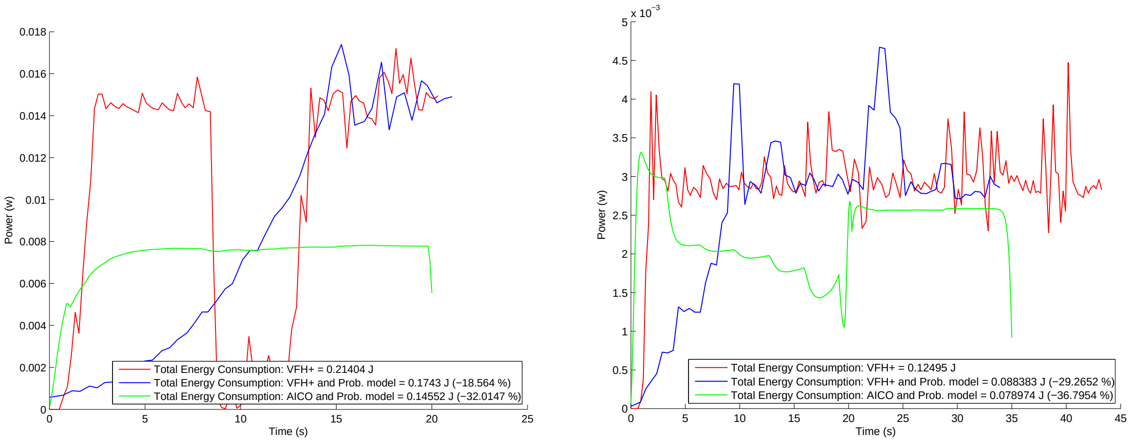


Figure 11. Energy consumption using VFH+ algorithm.



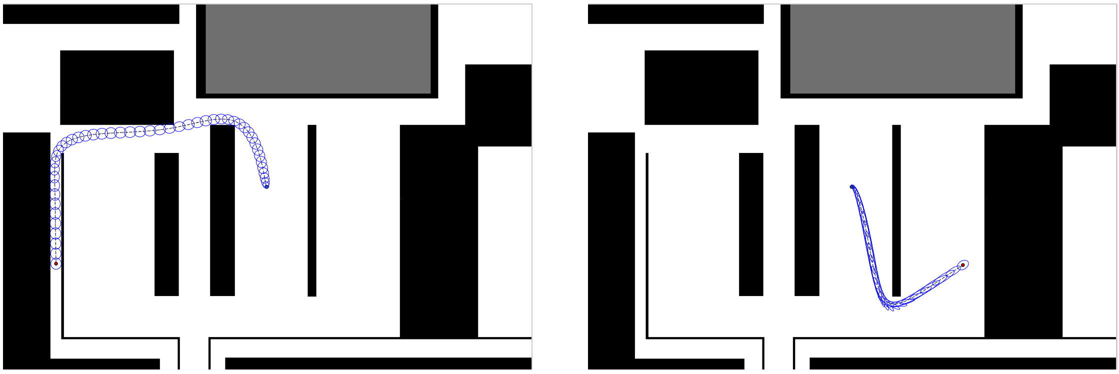
(a) Parallel scenario.

(b) Perpendicular scenario.

3.3. Results of the Dynamic Obstacle Avoidance System

The aim of this experiment is to show that our system computes safe paths reaching the goal configuration optimally with respect to energy consumption. Here we show that the performance of the reactive methods can be further improved by optimising the motion with respect to energy. The inference based planner described in Section 2.3 is initialized using the path computed from the Voronoi graph of the environment including only static obstacles. Then AICO computes the initial optimal path from start to goal positions (Figure 12). Starting and goal positions are marked by the green and red dots respectively. The covariance ellipses are overlaid.

Figure 12. Two examples of optimal paths computed using AICO in a static environment.



We have used our probabilistic model of the environment to detect the position and velocity of dynamic obstacles in the robot coordinate frame which we have then mapped into the global coordinate frame. We use this information to predict the movement of these obstacles. AICO is then used to compute the optimal trajectory around the initial path while using the probabilistic model predictions about the dynamic obstacles. A set of task variables has been used to define the optimality: position, power demand (Equation (3)), turning velocity and collision avoidance. The collision avoidance is achieved by inferring cost for reciprocal distance to the closest obstacle. AICO works under the assumption that the full state of the world, including the motion of the obstacles, is known. This is, however, no longer true when the prediction made by our probabilistic model is inaccurate. We therefore update our prediction using the new observations and discount the occupancy probability over time. We re-plan the path if the prediction error reaches a threshold. The planner therefore behaves similarly to a Kalman filter.

Only small changes in the environment between two time steps are expected. In such situations AICO requires only small number of iterations to converge, which makes re-planning computationally affordable. The replanning time (3 s on average) is however too long to be use as a reactive controller.

We have applied our proposed method to perform the task while avoiding the obstacles and minimizing energy consumption. AICO solves the finite horizon optimization problems, which means that the duration of the trajectory needs to be specified a priori. It is not within the scope of this paper to optimize for time, we have therefore set the trajectory duration to 20 s for parallel scenario and 35 s for perpendicular scenario, which are the respective average durations as computed using the reactive methods with our probabilistic model.

Figures 7 and 8 show the results of the path optimization and demonstrate that optimizing the energy consumption further decreases the curvature of the trajectory. Similarly, Figures 9 and 10 show that the velocity and acceleration profiles are much smoother. As a result, the energy consumption in both scenarios was reduced by approximately 10% when compared with the best results achieved by the reactive methods as shown in Table 1.

Table 1. Summary of results for energy consumption. AICO is being compared with the best reactive method for the given scenario.

	Scenario	Perception stage	Total Energy Consumption (J)	Reduction of Consumption (%)
VFH+	Parallel	Raw laser data	0.21404	–18.56%
		Probabilistic model	**0.1743**	
	Perpendicular	Raw laser data	0.12495	–29.26%
		Probabilistic model	0.088383	
CVM	Parallel	Raw laser data	0.22615	–7.52%
		Probabilistic model	0.20914	
	Perpendicular	Raw laser data	0.10458	–4.39%
		Probabilistic model	0.099981	
LCM	Parallel	Raw laser data	0.20125	+2.89%
		Probabilistic model	0.20709	
	Perpendicular	Raw laser data	0.09097	–8.1295%
		Probabilistic model	**0.083575**	
BCM	Parallel	Raw laser data	0.20546	–7.29%
		Probabilistic model	0.19047	
	Perpendicular	Raw laser data	0.16599	–45.5%
		Probabilistic model	0.090454	
AICO	Parallel	Probabilistic model	**0.14552**	–16.51%
	Perpendicular	Probabilistic model	**0.0789**	–10.91%

4. Conclusions and Future Works

In this paper, we have presented a method for avoiding dynamic obstacles, taking into account not only the integrity of the system, but also the minimization of the energy consumption. We have proposed a probabilistic model of the dynamic environment using: (1) an extension of the Bayesian Occupancy Filter proposed by [14] to a three-dimensional method to detect the obstacles positions and (2) a method to estimate the velocity of these obstacles using a tracking stage based on optical flow and a detection stage based on blob filtering.

We have used the probabilistic model of the environment inside the perception stage of the four classical methods: VFH+, BCM, CVM and LCM. Just by using a single level of the occupancy grid we were already able to improve the perception stage of these avoidance systems. The results have shown an improvement in energy consumption up to 45.5%. We have also shown that the resulting trajectories as well as the velocity and acceleration profiles are much smoother when using our method.

We have implemented a Voronoi graph based global path planner which serves as an initialisation method for our inference based local planner: AICO. Within AICO we combine multiple task variables to obtain the optimal path based on obstacle clearance and energy consumption. This method shows further qualitative improvement against the reactive methods with an improvement in energy consumption of 10.91% and 16.51% respect to the best results of reactive methods in perpendicular and parallel scenarios respectively. The computational cost of using AICO is currently too high for it to be used in real-time planning applications.

In the near future, we intend to improve the accuracy of the probabilistic model by refining the discretisation. We also intend to improve the calculation of the probabilistic model to speed up the algorithm to accommodate for higher number of cells. The planning algorithm can be improved by solving the inference at multiple scales and by exploiting parallel computation. Furthermore, we will address issues of coverage using topology based properties such as winding numbers [26]. This will allow us to reduce energy consumption by planning optimal route to survey an area such as a parking lot when searching for a parking space.

Acknowledgements

This work has been funded by TIN2011-29824-C02-01 and TIN2011-29824-C02-02 (ABSYNTHE project) from the “Ministerio de Economía y Competitividad” and a grant of the Fundación Caja Madrid.

References

1. Montemerlo, M.; Becker, J.; Bhat, S.; Dahlkamp, H.; Dolgov, D.; Ettinger, S.; Haehnel, D.; Hilden, T.; Hoffmann, G.; Huhnke, B.; *et al*. Junior: The stanford entry in the urban challenge. *J. Field Rob.* **2008**, *25*, 569–597.
2. Thrun, S.; Montemerlo, M.; Dahlkamp, H.; Stavens, D.; Aron, A.; Diebel, J.; Fong, P.; Gale, J.; Halpenny, M.; Hoffmann, G.; *et al*. Winning the DARPA grand challenge. *J. Field Rob.* **2006**, *23*, 661–692.

3. Enge, P.; Misra, P. Scanning the issue/technology. Special issue on global positioning system. *Proc. IEEE* **1999**, *87*, 3–15.
4. Schleicher, D.; Bergasa, L.M.; Ocaña, M.; Barea, R.; López, E. Low-cost GPS sensor improvement using stereovision fusion. *Sign. Process.* **2010**, *90*, 3294–3300.
5. Hentschel, M.; Wulf, O.; Wagner, B. A GPS and laser-based localization for urban and non-urban outdoor environments. In *Proceedings of IROS 2008: IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 22–26 September 2008; pp. 149–154.
6. Thrun, S.; Fox, D.; Burgard, W. A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonom. Rob.* **1998**, *5*, 253–271.
7. Newman, P.; Cole, D.; Ho, K. Outdoor SLAM using visual appearance and laser ranging. In *Proceedings of ICRA 2006: IEEE International Conference on Robotics and Automation*, Orlando, FL, USA, 15–19 May 2006; pp. 1180–1187.
8. Bekris, K.E.; Click, M.; Kavraki, E.E. Evaluation of algorithms for bearing-only SLAM. In *Proceedings of ICRA 2006: IEEE International Conference on Robotics and Automation*, Orlando, FL, USA, 15–19 May 2006; pp. 1937–1943.
9. Ulrich, I.; Borenstein, J. VFH+: Reliable obstacle avoidance for fast mobile robots. In *Proceedings of ICRA 1998: IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 16–20 May 1998; pp. 1572–1577.
10. Minguez, J.; Montano, L. Nearness diagram (ND) navigation: Collision avoidance in troublesome scenarios. *IEEE Trans. Rob. Autom.* **2004**, *20*, 45–59.
11. Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Rob. Autom. Mag.* **1997**, *4*, 23–33.
12. Ko, N.Y.; Simmons, R. The lane-curvature method for local obstacle avoidance. In *Proceedings of IROS 1998: IEEE/RSJ International Conference on Intelligent Robots and Systems*, Victoria, BC, Canada, 13–17 October 1998; pp. 1615–1621.
13. Fernández, J.L.; Sanz, R.; Benayas, J.A.; Diéguez, A.R. Improving collision avoidance for mobile robots in partially known environments: The beam curvature method. *Rob. Autonom. Syst.* **2004**, *46*, 205–219.
14. Coué, C.; Pradalier, C.; Laugier, C.; Fraichard, T.; Bessiere, P. Bayesian Occupancy Filtering for multitarget tracking: An automotive application. *Int. J. Rob. Res.* **2006**, *25*, 19–30.
15. Fulgenzi, C.; Spalanzani, A.; Laugier, C. Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid. In *Proceedings of ICRA 2007: IEEE International Conference on Robotics and Automation*, Roma, Italy, 10–14 April 2007.
16. Toussaint, M. Robot trajectory optimization using approximate inference. In *Proceedings of ICML 2009: the 26th International Conference on Machine Learning*, Montreal, QC, Canada, 14–18 June 2009.
17. Llamazares, Á.; Ivan, V.; Ocaña, M.; Vijayakumar, S. Dynamic obstacle avoidance minimizing energy consumption. In *Proceedings of the IEEE Intelligent Vehicles Symposium and Workshop on Perception in Robotics*, Alcalá de Henares, Spain, 3–7 June 2012; pp. P02.1–P02.6.

18. Llamazares, Á.; Molinos, E.; Ocaña, M.; Bergasa, L.M.; Hernández, N.; Herranz, F. 3D map building using a 2D laser scanner. In *Proceedings of 13th International Conference on Computer Aided Systems Theory*, Las Palmas de Gran Canaria, Spain, 6–11 February 2011; pp. 161–164.
19. Yguel, M.; Tay, C.; Mekhnacha, K.; Laugier, C. *Velocity Estimation on the Bayesian Occupancy Filter for Multi-Target Tracking*; Report RR-5836; INRIA: Rocquencourt, France, 2006.
20. Mekhnacha, K.; Mao, Y.; Raulo, D.; Laugier, C. Bayesian occupancy filter based "Fast Clustering-Tracking" algorithm. In *Proceedings of IROS 2008: IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 22–26 September 2008.
21. Bouguet, J. *Pyramidal Implementation of the Lucas Kanade Feature Tracker*; Microprocessor Research Labs, Intel Corporation: Santa Clara, CA, USA, 2000.
22. Sovran, G.; Bohn, M. *Formulae for the Tractive-Energy Requirements of Vehicles Driving the EPA Schedules*; SAE Technical Paper 810184; SAE International: Warrendale, PA, USA, 1981.
23. Jimnez-Palacios, J.L. Understanding and Quantifying Motor Vehicle Emissions with Vehicle Specific Power and TILDAS Remote Sensing. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999.
24. Bauer, H.; Dinkler, F.; Crepin, J.; Dietsche, K. *Automotive Handbook*, 5th ed.; Robert Bosch GmbH: Stuttgart, Germany, 2000.
25. Gillespie, T. *Fundamentals of Vehicle Dynamics*; Society of Automotive Engineers Inc.: Warrendale, PA, USA, 1992.
26. Vernaza, P.; Narayanan, V.; Likhachev, M. Efficiently finding optimal winding-constrained loops in the plane. In *Proceedings of Robotics: Science and Systems VIII*, Sydney, Australia, 9–13 July 2012.