

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Bayesian Framework for State Estimation and Robot Behaviour Selection in Dynamic Environments

Georgios Lidoris, Dirk Wollherr and Martin Buss

*Institute of Automatic Control Engineering, Technische Universität München
D-80290 München, Germany*

1. Introduction

One of the biggest challenges of robotics is to create systems capable of operating efficiently and safely in natural, populated environments. This way, robots can evolve from tools performing well-defined tasks in structured industrial or laboratory settings, to integral parts of our everyday lives. However such systems require complex cognitive capabilities, to achieve higher levels of cooperation and interaction with humans, while coping with rapidly changing objectives and environments.

In order to address these challenges a robot capable of autonomously exploring densely populated urban environments, is created within the Autonomous City Explorer (ACE) project (Lidoris et al., 2007). To be truly autonomous such a system must be able to create a model of its unpredictable dynamic environment based on noisy sensor information and reason about it. More specifically, a robot is envisioned that is able to find its way in an urban area, without a city map or GPS. In order to find its target, the robot will approach pedestrians and ask for directions.

Due to sensor limitations the robot can observe only a small part of its environment and these observations are corrupted by noise. By integrating successive observations a map can be created, but since also the motion of the robot is subject to error, the mapping problem comprises also a localization problem. This duality constitutes the Simultaneous Localization And Mapping (SLAM) problem. In dynamic environments the problem becomes more challenging since the presence of moving obstacles can complicate data association and lead to incorrect maps. Moving entities must be identified and their future position needs to be predicted over a finite time horizon. The autonomous sensory-motor system is finally called to make use of its self-acquired uncertain knowledge to decide about its actions.

A Bayesian framework that enables recursive estimation of a dynamic environment model and action selection based on these uncertain estimates is introduced. This is presented in Section 2. In Section 3, it is shown how existing methods can be combined to produce a working implementation of the proposed framework. A Rao-Blackwellized particle filter (RBPF) is deployed to address the SLAM problem and combined with recursive conditional

Source: *Advances in Greedy Algorithms*, Book edited by: Witold Bednorz,
ISBN 978-953-7619-27-5, pp. 586, November 2008, I-Tech, Vienna, Austria

particle filters in order to track people in the vicinity of the robot. Conditional filters have been used in the literature for tracking given an a priori known map. In this paper they are modified to be utilized with incrementally constructed maps. This way a complete model of dynamic, populated environments can be provided. Estimations serve as the basis for all decisions and actions of robots acting in the real world. In Section 4 the behaviours of the robot are described. In Section 5 it is shown how these are selected so that uncertainty is kept under control and the likelihood of achieving the tasks of the system is increased. In highly dynamic environments decision making needs to be performed as soon as possible. However, optimal planning is either intractable or requires very long time to be completed and since the world is changing constantly, any plan becomes outdated quickly. Therefore the proposed behaviour selection scheme is based on greedy optimization algorithms.



Fig. 1. The Autonomous City Explorer (ACE) robotic platform

2. Bayesian framework for state estimation and behaviour selection

The problem of action selection has been addressed by different researchers in various contexts. The reviews of (Tyrrell, 1993) and (Prescott et al., 1999) cover the domains of ethology and neuroscience. (Maes, 1989) addresses the problem in the context of artificial

agents. In robotics, action selection is related to optimization. Actions are chosen so that the utility toward the goal of the robot is maximized. Several solutions have been proposed which can be distinguished in many dimensions. For example whether the action selection mechanism is competitive or cooperative (Arkin, 1998), or whether it is centralized or decentralized (Pirjanian, 1999). Furthermore, explicit action selection mechanisms can be incorporated as separate components into an agent architecture (Bryson, 2000). Reinforcement learning has been applied to selection between conflicting and heterogeneous goals (Humphrys, 1997). A distinction was made between selecting an action to accomplish a unique goal and choosing between conflicting goals.

However, several challenges remain open. Real-world environments involve dynamical changes, uncertainty about the state of the robot and about the outcomes of its actions. It is not clear how uncertain environment and task knowledge can be effectively expressed and how it can be incorporated into an action selection mechanism. Another issue remains dealing with the combinatorial complexity of the problem. Agents acting in dynamic environments cannot consider every option available to them at every instant in time, since decisions need to be made in real-time. Consequently, approximations are required.

The approach presented in this chapter addresses these challenges. The notion of behavior is used, which implies actions that are more complex than simple motor commands. Behaviors are predefined combinations of simpler actuator command patterns, that enable the system to complete more complex task objectives (Pirjanian, 1999). A Bayesian approach is taken, in order to deal with uncertain system state knowledge and uncertain sensory information, while selecting the behaviours of the system. The main inspiration is derived from the human cognition mechanisms. According to (Körding & Wolpert, 2006), action selection is a fundamental decision process for humans. It depends both on the state of body and the environment. Since signals in the human sensory and motor systems are corrupted by variability or noise, the nervous system needs to estimate these states. It has been shown that human behaviour is close to that predicted by Bayesian theory, while solving estimation and decision problems. This theory defines optimal behaviour in a world characterized by uncertainty, and provides a coherent way of describing sensory-motor processes.

Bayesian inference also offers several advantages over other methods like Partially Observable Markov Decision Processes (POMDPs) (Littman et al., 1995), which are typically used for planning in partially observable uncertain environments. Domain specific knowledge can be easily encoded into the system by defining dependences between variables, priors over states or conditional probability tables. This knowledge can be acquired by learning from an expert or by quantifying the preferences of the system designer.

A relationship is assigned between robot states and robot behaviours, weighted by the state estimation uncertainty. Behaviour selection is then performed based on greedy optimization. No policy learning is required. This is a major advantage in dynamic environments since learning policies can be computationally demanding and policies need to be re-learned every time the environment changes. In such domains the system needs to be able to decide as soon as possible. There is evidence (Emken et al., 2007) that also humans use greedy algorithms for motor adaptation in highly dynamic environments. However, the

optimality of this approach depends on the quality of the approximation of the true distributions. State of the art estimation techniques enable very effective and qualitative approximations of arbitrary distributions. In the remainder of this section the proposed Bayesian framework is going to be presented in more detail.

2.1 Bayesian Inference

In terms of probabilities the domain of the city explorer can be described by the joint probability distribution $p(S_t, B_t, C_t, Z_t | U_t)$. This consists of the state of the system and the model of the dynamic environment S_t , the set of behaviors available to the system B_t , a set of processed perceptual inputs that are associated with events in the environment and are used to trigger behaviors C_t , system observations Z_t and control measurements U_t that describe the dynamics of the system. In the specific domain, observations are the range measurements acquired by the sensors and control measurements are the odometry measurements acquired from the mobile robot. The behavior triggering events depend on the perceived state of the system and its goals. The state vector S_t is defined as

$$S_t = \{X_t, m, Y_t^1, Y_t^2, \dots, Y_t^M\} \quad (1)$$

where X_t represents the trajectory of the robot, m is a map of the environment and $Y_t^1, Y_t^2, \dots, Y_t^M$ the positions of M moving objects present at time t . Capital letters are used throughout this chapter to denote the full time history of the quantities from time point 0 to time point t , whereas lowercase letters symbolize the quantity only at one time step. For example z_t would symbolize the sensor measurements acquired only at time step t .

The joint distribution can be decomposed to simpler distributions by making use of the conjunction rule.

$$p(S_t, B_t, C_t, Z_t | U_t) = p_0 \prod_{j=1}^t \{p(s_j | S_{j-1}, U_j) p(z_j | S_j) p(b_j | B_{j-1}, C_j, S_j)\} \quad (2)$$

Initial conditions, $p(s_0, b_0, c_0, z_0, u_0)$, are expressed for simplicity by the term p_0 . The first term in the product represents the dynamic model of the system and it expresses our knowledge about how the state variables evolve over time. The second one expresses the likelihood of making an observation z_t given knowledge of the current state. This is the sensor model or perceptual model. The third term constitutes the behaviour model. Behaviour probability depends on behaviours selected previously by the robot, on perceptions and the estimated state at the current time.

The complexity of this equation is enormous, since dependence on the whole variable history is assumed. In order to simplify it, Bayes filters make use of the Markov assumption. Observations z_t and control measurements u_t are considered to be conditionally independent of past measurements and control readings given knowledge of the state s_t . This way the joint distribution is simplified to contain first order dependencies.

$$p(S_t, B_t, C_t, Z_t | U_t) = p_0 \prod_{j=1}^t \{p(s_j | s_{j-1}, u_j) p(z_j | s_j) p(b_j | b_{j-1}, c_j, s_j)\} \quad (3)$$

As discussed previously, the goal of an autonomous system is to be able to choose its actions based only on its perceptions, so that the probability of achieving its goals is maximized. This requires the ability to recursively estimate all involved quantities. Using the joint distribution described above this is made possible. In the next subsection it will be analyzed how this information can be derived, by making use of Bayesian logic.

2.2 Prediction

The first step is to update information about the past by using the dynamic model of the system, in order to obtain a predictive belief about the current state of the system. After applying the Bayes rule and marginalizing irrelevant variables, the following equation is acquired.

$$p(s_t | Z_{t-1}, U_t) \propto \sum_{s_{t-1}} p(s_t | s_{t-1}, u_t) p(s_{t-1} | Z_{t-1}, U_{t-1}) \quad (4)$$

More details on the mathematical derivation can be found in (Lidoris et al., 2008). The first term of the sum is the system state transition model and the second one is the prior belief about the state of the system. Prediction results from a weighted sum over state variables that have been estimated at the previous time step.

2.3 Correction step

The next step of the estimation procedure is the correction step. Current observations are used to correct the predictive belief about the state of the system, resulting to the posterior belief $p(s_t | Z_t, U_t)$. During this step, all information available to the system is fused.

$$p(s_t | Z_t, U_t) \propto \sum_{s_{t-1}} p(z_t | s_t) p(s_t | Z_{t-1}, U_t) \quad (5)$$

It can be seen from (5) that the sensor model is used to update the prediction with observations. The behaviour of the robot is assumed not to have an influence on the correction step. The effect of the decision the robot will take at the current time step about its behaviour, will be reflected in the control measurements that are going to be received at the next time step. Therefore the behaviour and behaviour trigger variables have been integrated out of (5).

2.4 Estimation of behaviour probabilities

Finally, the behaviour of the system needs to be selected by using the estimation about the state of the system and current observations. That includes calculating the probabilities over the whole set of behaviour variables, $p(b_t | S_t, C_t, Z_t, U_t)$ for the current time step. The same inference rules can be used as before, resulting to the following equation

$$p(b_t | S_t, C_t, Z_t, U_t) \propto \sum_{s_t} p(z_t | s_t) p(b_t | b_{t-1}, c_t, s_t) p(s_t | Z_{t-1}, U_t) \quad (6)$$

By placing (5) in (6) an expression is acquired which contains the estimated posterior.

$$p(b_t | S_t, C_t, Z_t, U_t) \propto \sum_{s_t} p(b_t | b_{t-1}, c_t, s_t) p(s_t | Z_t, U_t) \quad (7)$$

As mentioned previously, system behaviours are triggered by processed perceptual events. These events naturally depend on the state of the system and its environment. Therefore the behaviour selection model $p(b_t | b_{t-1}, c_t, s_t)$ can be further analyzed to

$$p(b_t | b_{t-1}, c_t, s_t) = p(b_t | b_{t-1}, c_t) p(c_t | s_t) \quad (8)$$

and replacing equation (8) to (7) leads to

$$p(b_t | S_t, C_t, Z_t, U_t) \propto \sum_{s_t} p(b_t | b_{t-1}, c_t) p(c_t | s_t) p(s_t | Z_t, U_t) \quad (9)$$

The behaviour model is weighted by the estimated posterior distribution $p(s_t | Z_t, U_t)$ for all possible values of the state variables and the probability of the behaviour triggers. The term $p(b_t | b_{t-1}, c_t)$ expresses the degree of belief that given the current perceptual input the current behaviour will lead to the achievement of the system tasks. This probability can be pre-specified by the system designer or can be acquired by learning.

3. Uncertainty representation and estimation in unstructured dynamic environments

In the previous section a general Bayesian framework for state estimation and decision making has been introduced. In order to be able to use it and create an autonomous robotic system, the related probability distributions need to be estimated. How this can be made possible, is discussed in this section. The structure of the proposed approach is presented in Fig. 2. Whenever new control (e.g. odometry readings) and sensor measurements (e.g. laser range measurements) become available to the robot, they are provided as input to a particle filter based SLAM algorithm. The result is an initial map of the environment and an estimate of the trajectory of the robot. This information is used by a tracking algorithm to obtain a model of the dynamic part of the environment. An estimate of the position and velocity of all moving entities in the environment is acquired, conditioned on the initial map and position of the robot. All this information constitutes the environment model and the estimated state vector s_t . A behaviour selection module makes use of these estimates to infer behaviour triggering events c_t and select the behaviour b_t of the robot. According to the selected behaviour, a set of actuator commands is generated which drives the robot toward the completion of its goals. In the following subsections each of the components and algorithms mentioned here are going to be further analyzed.

3.1 Simultaneous localization and mapping

The problem of simultaneous localization and mapping is one of the fundamental problems in robotics and has been studied extensively over the last years. It is a complex problem because the robot needs a reliable map for localizing itself and for acquiring this map it requires an accurate estimate of its location. The most popular approach (Dissanayake et al., 2002) is based on the Extended Kalman Filter (EKF). This approach is relatively effective

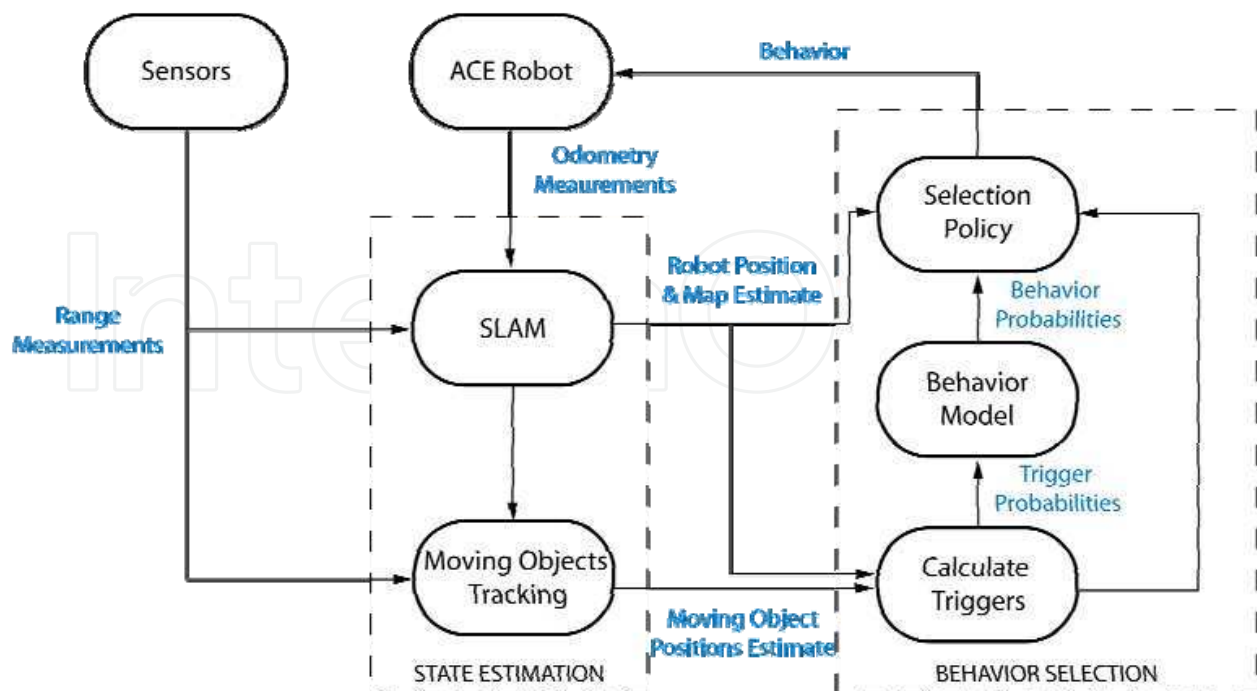


Fig. 2. Proposed approach for modelling dynamic environments and behaviour selection.

since the resulting estimated posterior is fully correlated about landmark maps and robot poses. Its disadvantage is that motion model and sensor noise are assumed Gaussian and it does not scale well to large maps, since the full correlation matrix is maintained. Another well known approach (Thrun et al., 2004) corrects poses based on the inverse of the covariance matrix, which is called information matrix and is sparse. Therefore predictions and updates can be made in constant time. Particle filters have been applied to solve many real world estimation and tracking problems (Doucet et al. 2000), (Murphy, 1999) since they provide the means to estimate the posterior over unobservable state variables, from sensor measurements. This framework has been extended, in order to approach the SLAM problem with landmark maps in (Montemerlo et al., 2002). In (Grisetti et al., 2005) a technique is introduced to improve grid-based Rao-Blackwellized SLAM. The approach described here is similar to this technique, with the difference that scan-matching is not performed in a per-particle basis but only before new odometry measurements are used by the filter.

This approach allows the approximation of arbitrary probability distributions, making it more robust to unpredicted events such as small collisions which often occur in challenging environments and cannot be modelled. Furthermore it does not rely on predefined feature extractors, which would assume that some structures in the environment are known. This allows more accurate mapping of unstructured outdoor environments. The only drawback is that the approximation quality depends on the number of particles used by the filter. More particles result to increased required computational costs. However if the appropriate proposal distribution is chosen, the approximation can be kept very accurate even with a small number of particles. In the remainder of this section the approach is briefly highlighted.

The idea of Rao-Blackwellization is that it is possible to evaluate (Doucet et al., 2000) some of the filtering equations analytically and some others by Monte Carlo sampling. This results in estimators with less variance than those obtained by pure Monte Carlo sampling.

In the context of SLAM the posterior distribution $p(X_t, m | Z_t, U_t)$ needs to be estimated. Namely the map m and the trajectory X_t of the robot need to be calculated based on the observations Z_t and the odometry measurements U_t , which are obtained by the robot and its sensors.

The use of the Rao-Blackwellization technique, allows the factorization of the posterior.

$$p(X_t, m | Z_t, U_t) = p(X_t | Z_t, U_t) p(m | X_t, Z_t) \quad (10)$$

The posterior distribution $p(X_t | Z_t, U_t)$ can be estimated by sampling, where each particle represents a potential trajectory. This is the localization step. Next, the posterior $p(m | X_t, Z_t)$ over the map can be computed analytically as described in (Moravec, 1989) since the history of poses X_t is known.

An algorithm similar to (Grisetti et al., 2005) is used to estimate the SLAM posterior. Only the main differences are highlighted here. Each particle i is weighted according to the recursive formula

$$w_t^i = \frac{p(z_t | m_{t-1}, x_t^i) p(x_t^i | x_{t-1}^i, u_t)}{q(X_t^i | X_{t-1}^i, Z_t, U_t)} \quad (11)$$

The term $p(x_t^i | x_{t-1}^i, u_{t-1})$ is an odometry-based motion model. The motion of the robot in the interval $(t-1, t]$ is approximated by a rotation δ_{rot1} , a translation δ_{trans} and a second rotation δ_{rot2} . All rotations and translations are corrupted by noise. An arbitrary error distribution can be used to model odometric noise, since particle filters do not require specific assumptions about the noise distribution.

The likelihood of an observation given a global map and a position estimate is denoted as $p(z_t | m_{t-1}, x_t^i)$. It can be evaluated for each particle by using the particle map constructed so far and map correlation. More specifically a local map, $m_{local}^i(x_t^i, z_t)$ is created for each particle i . The correlation to the most actual particle map, m_{t-1}^i , is evaluated as follows:

$$\rho = \frac{\sum_{x,y} (m_{x,y}^i - \bar{m}^i) (m_{x,y,local}^i - \bar{m}^i)}{\sqrt{\sum_{x,y} (m_{x,y}^i - \bar{m}^i)^2 \sum_{x,y} (m_{x,y,local}^i - \bar{m}^i)^2}} \quad (12)$$

Where \bar{m}^i is the average map value at the overlap between the two maps. The observation likelihood is proportional to the correlation value.

An important issue for the performance and the effectiveness of the algorithm is the choice of the proposal distribution. Typically the motion model is used, because it is easy to compute. In this work, the basis for the proposal distribution is provided by the odometry motion model, but is combined with a scan alignment that integrates the newest sensor measurements and improves the likelihood of the sampled particles. More specifically, new odometry measurements are corrected based on the current laser data and the global map, before being used by the motion model, through scan matching. This way information from the more accurate range sensors is incorporated. It must be noted here, that this is not performed on a per particle basis like in other approaches (Grisetti et al. 2005), since no great improvement in the accuracy of the estimator has been observed, compared with the higher computational costs involved.

3.2 Conditional particle filters for tracking

The methods mentioned above focus on the aspects of state estimation, belief representation and belief update in static environments. More specifically, an estimate of the most likely trajectory X_t of the robot, relative to an estimated static map, m_t , is provided. To estimate the full state of the environment as defined by (1), the position of moving objects needs also to be estimated.

Until now, no complete Bayesian framework exists for the dynamic environment mapping problem. One of the first attempts was introduced in (Wang et al., 2003). However it is based on the restrictive assumption of independence between static and dynamic elements in the environment. In (Haehnel et al., 2003) scan registration techniques are used to match raw measurement data to estimated occupancy grids in order to solve the data association problem and the Expectation-Maximization algorithm is used to create a map of the environment. A drawback is that the number of dynamic objects must be known in advance. Particle filters have been used to track the state of moving objects in (Montemerlo et al., 2002). However the static environment is assumed known. Particle filters have also been used in (Miller & Campbell, 2007) to solve the data association problem for mapping but without considering robot localization.

A similar approach as in (Montemerlo et al., 2002) is used here, extended to handle unknown static maps. The full state vector can then be estimated by conditioning the positions of moving objects on the robot trajectory estimate provided by tackling the SLAM problem.

$$p(S_t | Z_t, U_t) = p(X_t, m_t | Z_t, U_t) \prod_{m=1}^M p(Y_t^m | X_t, Z_t, U_t) \quad (13)$$

Each conditional distribution $p(Y_t^m | X_t, Z_t, U_t)$ is also represented by a set of particles. The particles are sampled from the motion model of the moving object. Several dynamics models exist, including constant velocity, constant acceleration and more complicated switching ones (Wang et al., 2003). Since people move with relatively low speeds and their motion can become very unpredictable, a Brownian motion model is an acceptable approximation.

Every particle of each particle filter, $y_t^{m,i}$, is weighted according to the measurement likelihood.

$$w_t^{m,i} = p(z_t | x_t, y_t^{m,i}) \quad (14)$$

In order to calculate the likelihood, each sensor reading needs to be associated to a specific moving object. However, measurements can be erroneous, objects might be occluded and the environment model might not be accurate, therefore leading to false associations. Persons are modelled as cylindrical structures during data association of the 2D laser data. The radius of the cylinder has been chosen experimentally. A laser measurement is associated with a person if its distance from a person position estimate is smaller than a maximum gating distance. In this case it is additionally weighted according to its distance from the position estimate. Therefore if the gating regions of two persons overlap, the person closest to a laser point is associated with it.

4. Robot behaviour description

In this section the application of the proposed general Bayesian framework to the Autonomous City Explorer (ACE) robot is going to be presented. The set of available behaviours consists of *Explore*, *Approach*, *Reach Goal* and *Loop Closing*. A detailed description of each one of them follows.

4.1 Explore

The ability to explore its environment in order to find people to interact with and increase its map knowledge, is fundamental for the robot. The robot performs greedy optimization in order to choose its next goal so that a trade-off is achieved between maximizing its information gain and minimizing traveling costs. Given an occupancy grid map, frontier regions between known and unknown areas are identified, as described in (Yamauchi, 1998). The subset of cells of the grid m that belong to a frontier region f , are denoted by m_f . The expected information gain $I(m_f, x_t)$ acquired by reaching a frontier region from the current robot position x_t can be calculated as in (Stachniss et al., 2005). The traveling costs associated with reaching a frontier region, $cost(m_f, x_t)$, are proportional to the path length to it. In order to achieve the aforementioned trade-off, the autonomous explorer chooses its next goal, on the frontier region that maximizes the following objective function

$$m_f^* = \arg \max_{m_f} \{I(m_f, x_t) - \alpha \text{cost}(m_f, x_t)\} . \quad (15)$$

The parameter α is used to define how much the path cost should influence the exploration process and it can be chosen experimentally.

4.2 Approach

In order to interact with a person the robot needs first to approach her. This behaviour generates a target within a safety distance to a person. The person nearest to the robot is chosen in case more than one person is present simultaneously. Estimated positions from the tracker are used.

4.3 Reach goal

If the robot has been instructed a target through interaction, it needs to navigate safely to the specified target. An A* based planner is utilized that takes into account the motions of moving objects. A more detailed description is given in (Rohrmuller et al., 2007).

4.4 Loop closing

As the robot moves, the uncertainty about its position and its map grows constantly, therefore increasing the risk of failure. It is necessary for the robot to find opportunities to close a loop, therefore correcting its estimates. A way to acquire an estimate for the pose uncertainty $H(p(X_t | Z_t, U_t))$ of the robot, is to average over the uncertainty of the different poses along the path as in (Stachniss et al., 2005).

Since the distribution of the particle set can be arbitrary, it is not possible to efficiently calculate its entropy. A Gaussian approximation $N(\mu_t, \Sigma_t)$ can be computed based on the weighted samples with covariance Σ_t . The entropy can then be calculated only as a function of the covariance matrix. Such an approximation is rather conservative but absolutely

eligible, since a Gaussian probability distribution has higher entropy than any other distribution with the same variance.

In order to detect and close a loop, an approach similar to the one described in (Stachniss et al., 2004) is chosen. Together with the occupancy grid map a topological map is simultaneously created. This topological map consists of nodes, which represent positions visited by the robot. Each of these nodes contains visibility information between itself and all other nodes, derived from the associated occupancy grid. For each node the uncertainty of the robot $H_{init}(p(x_t | z_t, u_t))$ when it entered the node for the first time is also saved. To determine whether or not the robot should activate the loop-closing behaviour the system monitors the uncertainty $H(p(x_t | z_t, u_t))$ about the pose of the robot at the current time step. The necessary condition for starting the loop-closing process is that the geometric distance of the robot and a node in the map is small, while the graph distance in the topological map is large. If such a situation is detected the node is called entry point. Then the robot checks the difference between its initial uncertainty at the entry point and its current uncertainty, $H(p(x_t | z_t, u_t)) - H_{init}(p(x_t | z_t, u_t))$. If this difference exceeds a threshold then the loop is closed. This is done by driving the robot to the nearest neighbour nodes of the entry point in the topological map. During this process the pose uncertainty of the vehicle typically decreases, because the robot is able to localize itself in the map built so far and unlikely particles vanish.

5. Behaviour selection

As seen in the previous section, each of the behaviours available to the system has an objective which contributes to the achievement of the overall system goal. The robot needs to efficiently combine these behaviours by deciding when to activate which one and for how long. The proposed behaviour selection scheme is based on (9). This equation can be further analyzed by using the results of the state estimation process as summarized in (13).

$$p(b_t | S_t, C_t, Z_t, U_t) \propto \sum_{x_t} \left\{ p(b_t | b_{t-1}, c_t) p(c_t | x_t) p(x_t, m_t | Z_t, U_t) \prod_{m=1}^M p(y_t^m | x_t, Z_t, U_t) \right\} \quad (16)$$

It must be noted that the summation is done only over the state of the robot, x_t , since both the states of the moving objects and the map are conditioned on it. Particle filters have been used to approximate the posterior distributions $p(x_t, m_t | Z_t, U_t)$ and $p(y_t^m | x_t, Z_t, U_t)$. Therefore they can be approximated according to their particle weights (Arulampalam et al., 2002), given in (11) and (14), leading to the following equation:

$$p(b_t | S_t, C_t, Z_t, U_t) \propto \sum_{i=1}^N \left\{ p(b_t | b_{t-1}, c_t) p(c_t | x_t) w_t^i \delta(x_t - x_t^i) \prod_{m=1}^M \sum_{j=1}^K w_t^{m,j} \delta(y_t^m - y_t^{m,j}) \right\} \quad (17)$$

δ is the Dirac delta function, N is the number of particles used by the Rao-Blackwellized SLAM algorithm, M is the number of persons tracked by the robot and K is the number of particles of each conditional particle filter. After the probability of each behaviour is calculated, the behaviour with the maximum posterior probability is chosen.

$$b_t^* = \arg \max_{b_t} p(b_t | S_t, C_t, Z_t, U_t) \quad (18)$$

Greedy optimization of task completion probability is performed. The order of calculation for this equation is $O(NMK)$, which is significantly lower than the complexity of existing methods for action selection under uncertainty, like POMDPs, that typically have complexity exponential to the number of states. This allows the system to take decisions more often, in order to cope with fast changes and the occurrence of unpredictable events in the environment. The behaviour selection scheme is described in the next section in more detail.

5.1 Behaviour selection model

The term $p(b_t | b_{t-1}, c_t)p(c_t | x_t)$ in equation (18) is the behaviour model and it plays a crucial role in the optimality of the behaviour selection. It depends on the previous behaviour of the robot, the perceptual events that activate system behaviours and the estimated system state. This model supplies an expert opinion on the applicability of each behaviour at the present situation, indicating if it is completely forbidden, rather unwise, or recommended. This is done according to the information available to the system.

The results of state estimation are used to evaluate if behaviour triggering events have occurred and how certain their existence is. During this step the term $p(c_t | x_t)$ in (16) is calculated. Triggers and behaviours can have *high*, *medium*, *low* probability or be inactive. These values are predefined in this implementation and encode the goals of the system. They can also be acquired by letting a human operator decide about which behaviour the robot should choose, according to the situation. These decisions are then modelled to probability distributions. Bayesian decision theory and decision modelling provide the theoretical background to achieve that. Interesting works in this direction are (Ahmed & Campbell, 2008) and (Hy et al., 2004).

Three triggers exist that are used to calculate the probabilities of the behaviour model. These are:

- The existence of a person in the vicinity of the robot denoted by *person*. If a person has been detected then this trigger is activated. Its probability, $p(\text{person} | x_t)$, increases as the robot comes closer to a person.
- The existence of a goal for the robot to reach, which is given through interaction with people, denoted by *goal*. The probability $p(\text{goal} | x_t)$ increases as the distance of the given target from the current most likely, estimated robot position decreases.
- The existence of a loop closing opportunity, *loop*. It depends as explained in Section 4.4 on the existence of an entry point for loop closing and the difference between current position uncertainty and the initial position uncertainty at the entry point. The probability $p(\text{loop} | s_t)$ increases as the difference in uncertainty from the current position to the initial uncertainty at the entry point position becomes larger.

It remains now to explain how $p(b_t | b_{t-1}, c_t)$ is constructed. At each time step the robot knows its previous behaviour b_{t-1} and the triggers that are active. Using Table 1, behaviours are proposed as recommended and are assigned high probability. The rest of the behaviours that are possible receive lower recommendations and some are prohibited (denoted by "-" in the table). For example, if the previous behaviour of the robot, b_{t-1} , was *Loop Closing*, the trigger *loop* has probability *low* and the robot has no goal assigned, then the most recommended behaviour for the current time step, b_t , will be *Explore*. No other behaviour is possible.

$b_t \backslash b_{t-1}$	Explore	Loop Closing	Approach	Reach Goal
Explore	$\neg person$	$p(loop x_t) < medium$ & $\neg goal$	$\neg person \parallel$ $p(person x_t) < medium$	$p(goal x_t) < medium$
Loop Closing	$p(loop x_t) > medium$	$p(loop x_t) > low$	-	$p(loop x_t) > medium$
Approach	person	-	$p(person x_t) < high$	$\neg goal \& person$
Reach Goal	-	$p(loop x_t) < medium$ & goal	Goal	Goal

Table I. Behaviour Selection Model

A recommended behaviour is assigned *high* probability value and all other possible behaviours a *low* value. Finally values are normalized. If only one behaviour is possible as in the example given, then it receives a probability of 1. This way, $p(b_t | b_{t-1}, c_t, s_t)$ is acquired and is used to calculate the behaviour that maximizes (15).

6. Results

In order to evaluate the performance of the proposed behaviour selection mechanism, experiments were carried out. The robot was called to find its way to a given room of the third floor of our institute, without any prior knowledge of the environment. The floor plan as well as the starting position of the robot and the given target room is shown in Fig. 3. The robot must interact with people in order to ask for directions.

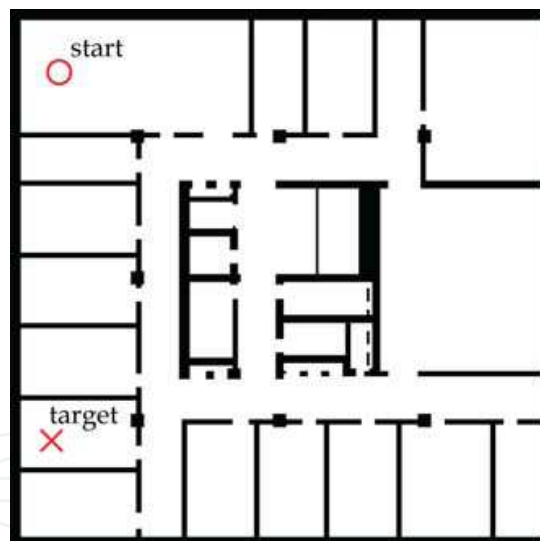


Fig. 3. Ground truth map of the third floor of the Institute of Automatic Control Engineering, Munich is illustrated. The robot starts without map knowledge and is required to reach the depicted final target location, which is acquired by interaction with humans.

All algorithms described in this paper have been implemented in C++ and have been tested on-line on the robot, using an AMD Athlon Dual Core 3800+ processor and 4GB of RAM. For the Rao-Blackwellized particle filter 200 particles were used and the conditional particle filters for people tracking used 30 particles each. Behaviour selection was performed at 1Hz. The SLAM and tracking module was running at 2Hz and the path planner at 1Hz. It has been found experimentally that at this frequency the tracker can track up to 15 moving objects.

In Fig. 4 the decisions taken by the robot in different situations during the experiment are illustrated. At first the robot decides to explore in order to acquire information about where the target room is. Two persons are detected and the robot decides to approach the one nearest to it in order to interact with. A goal position is acquired in the form of a waypoint "10m in the x direction and 3m in the y direction". The robot decides to reach this goal. After the intermediate goal is reached, a decision is made to explore in order to acquire new direction instructions. Another person is approached and new instructions are given which this time will lead to the final goal. As the robot moves its uncertainty grows. At some point an opportunity to close a loop is recognized. Therefore the robot decides to change its behaviour to *Loop Closing*, in order to reduce its uncertainty. After the loop is closed, the robot reaches its final goal.

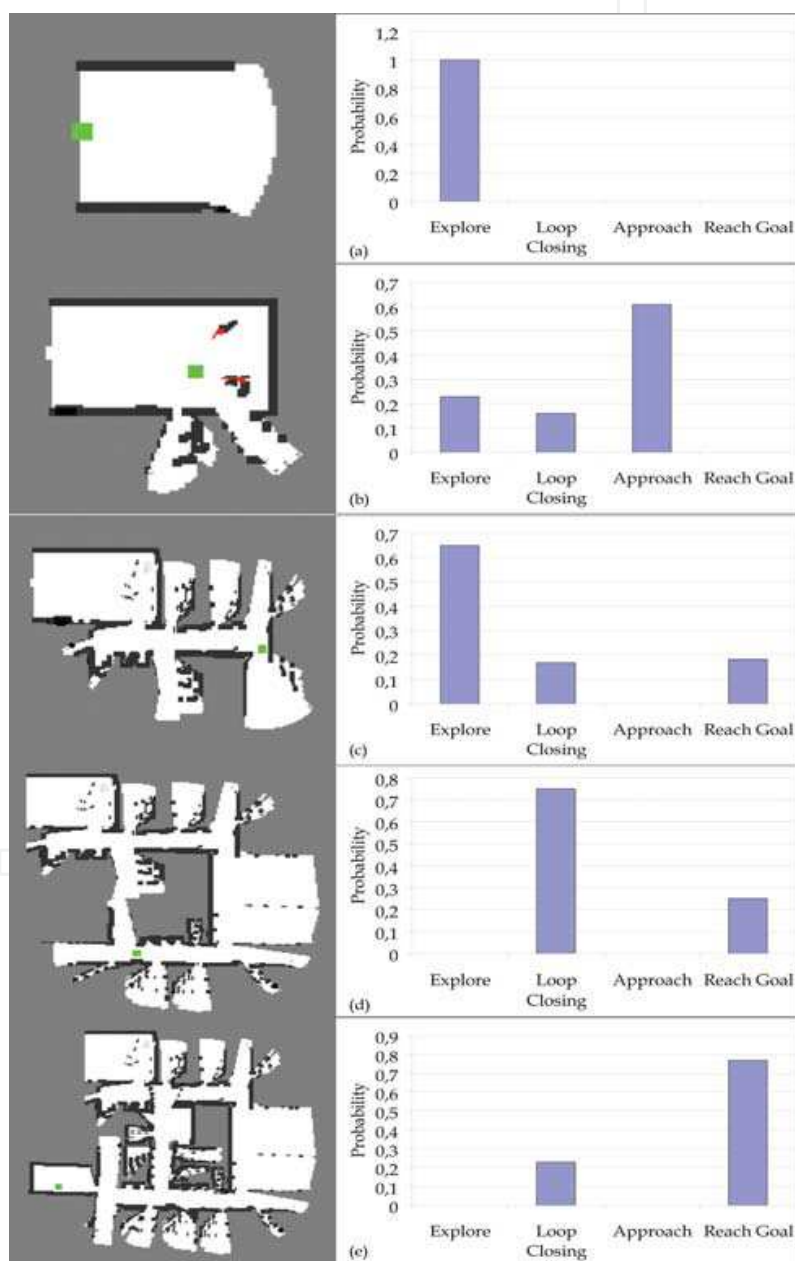


Fig. 4. The robot is called to find its way to a given goal, without prior map knowledge. All information is extracted by interaction. The decisions of the behaviour selection scheme are

shown in different situations. (a) The robot starts without any prior map information and decides to explore in order to find persons to interact with. (b) Two persons are found and the robot chooses the one closest to it in order to interact. (c) A goal was given to the robot by the first interaction and was reached by the robot. Now it chooses to explore in order to find a person to acquire a new target. (d) The robot has a target but its position uncertainty is high. It detects an opportunity to close a loop and decides to do so. (e) The robot reaches its final goal.

By taking uncertainty into account in action selection, the robot can anticipate unforeseen situations and increase the likelihood of achieving its goal. In Fig. 5 the overall uncertainty of the robot during this experiment is illustrated by the red line. The uncertainty of the robot trajectory when it reaches the target directly, without being controlled by the proposed scheme, is illustrated by the blue dashed line. It can be seen that at the early phases of the experiment the uncertainty of the system is larger with the proposed scheme, since the robot drives more complex trajectories in order to approach people, but it is not critical. At some point it decides to close the loop and its uncertainty is reduced notably. When it reaches its final goal the overall system uncertainty is much lower than without behaviour selection. Lower uncertainty is equivalent to safer navigation and increased task completion likelihood.

The presented system is capable of deciding when it should pursue its given target, in which situation interaction with humans is needed in order to acquire new target information and finally when its overall uncertainty has reached a critical point. In this last case it tries to reduce it by taking actions that improve its state estimates.

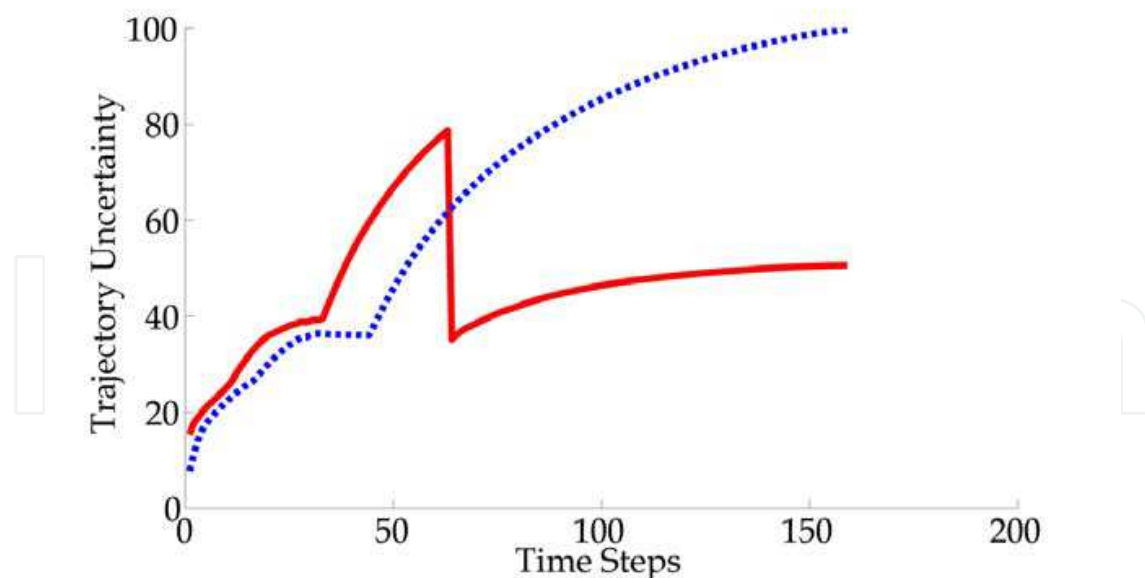


Fig. 5. Trajectory uncertainty as it evolves with the time. With red the uncertainty of the robot is illustrated, while it is controlled with the proposed behaviour selection scheme. The uncertainty of the robot trajectory when it reaches the target directly, without being controlled by the proposed scheme, is illustrated with blue dashed line.

7. Conclusion

In this Chapter a probabilistic framework has been introduced, that enables recursive estimation of a dynamic environment model and action selection based on these uncertain estimates. The proposed approach addresses two of the main open challenges of action selection. Uncertain knowledge is expressed by probability distributions and is utilized as a basis for all decisions taken from the system. At the same time the complexity of the proposed action selection mechanism is kept lower than of most state-of-the-art algorithms. The probability distributions of all associated uncertain quantities are approximated effectively and no restrictive assumptions are made regarding their form. More specifically, a Rao-Blackwellized particle filter (RBPF) has been deployed to address the SLAM problem and conditional particle filters have been modified to be utilized with incrementally constructed maps for tracking people in the vicinity of the robot. This way a complete model of dynamic, populated environments is provided. The computational costs depend only on the required approximation accuracy and can be defined according to the requirements of the application domain.

The estimated uncertain quantities are used for coordinating the behaviours of the robot so that uncertainty is kept under control and the likelihood of achieving its goals is increased. A greedy optimization algorithm is used for behaviour selection, which is computationally inexpensive. Therefore the robot can decide quickly in order to cope with its rapidly changing environment. The decisions taken may not be optimal in the sense of POMDP policies, but are always responding to the current state of the environment and are goal oriented. The goals of the system are expressed by the behaviour selection model.

Results from the implementation of all proposed algorithms on the ACE robotic platform demonstrate the efficiency of the approach. The robot can decide when to pursue its given goal or when to interact with people in order to get more target information. If its uncertainty becomes large, it takes actions that improve its state estimates. It is shown that overall system uncertainty is kept low even if the robot is called to complete complex tasks.

Human decision making capabilities are remarkable. Therefore, future work will focus on learning the behaviour selection model from data provided by a human expert. This way the quality of the decisions taken by the system can be improved. Formal evaluation criteria for action selection mechanisms need to be developed. This is challenging since such criteria must consider many conflicting requirements and since in almost every study different physical robots are used in variable experimental conditions. Finally, more experiments are going to be conducted in unstructured, outdoor, dynamic environments.

8. Acknowledgements

This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems -- CoTeSys*, see also www.cotesys.org.

9. References

- Ahmed, N.; Campbell, M. (2008). Multimodal Operator Decision Models. *IEEE American Control Conference (ACC)*, Seattle, USA.
- Arkin, R.C. (1998). Social behavior. *Behavior-Based Robotics*, MIT Press, Cambridge, MA.

- Arulampalam, S.; Maskell, S.; Gordon, N.; Clapp, T. (2002). A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50, p. 174-188.
- Bryson, J.J.; Stein, L.A. (2000). Architectures and Idioms: Making Progress in Agent Design. In: *The Seventh International Workshop on Agent Theories, Architectures and Languages*. Springer.
- Dissanayake, G.; Newman, P.; Clark, S.; Durrant-Whyte, H.; Csorba, M. (2001). A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Transactions on Robotics and Automation*, 17(3), p. 229-241.
- Doucet, A.; de Freitas, J. F. G.; Gordon, N. J. (2000). Sequential Monte Carlo Methods in Practice. *Springer-Verlag*, New York.
- Emken, J. L.; Benitez, R.; Sideris, A.; Bobrow J. E.; Reinkensmeyer D.J. (2007). Motor Adaptation as Greedy Optimization of Error and Effort. *Journal of Neurophysiology*. p. 3997-4006.
- Grisetti, G.; Stachniss, C.; Burgard, W. (2005). Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling *International Conference of Robotics and Automation (ICRA)*, Barcelona, Spain.
- Hähnel, D.; Triebel, R.; Burgard, W.; Thrun, S. (2003). Map Building with Mobile Robots in Dynamic Environments. *Proceedings of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Taipei, Taiwan.
- Humphrys, M. (1997). Action Selection Methods Using Reinforcement Learning. *PhD Thesis*. University of Cambridge, Computer Laboratory, Cambridge, England.
- Hy, R. L.; Arrigoni, A.; Bessiere, P.; Lebeltel, O. (2004). Teaching Bayesian behaviours to video game characters. *Robotics and Autonomous Systems*, 47, p. 177-185.
- Körding, K.; Wolpert, D. (2006). Bayesian Decision Theory in Sensorimotor Control. *Trends in Cognitive Sciences*, 10(7), p. 319-326.
- Lidoris, G.; Klasing, K.; Bauer, A.; Xu, T.; Kühnlenz, K.; Wollherr, D.; Buss, M. (2007). The Autonomous City Explorer Project: Aims and System Overview. *Proceedings of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, San Diego, USA.
- Lidoris, G.; Wollherr, D.; Buss, M. (2008). Bayesian State Estimation and Behavior Selection for Autonomous Robotic Exploration in Dynamic Environments. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, Sept, 22-26.
- Littman, M.; Cassandra, A.; Kaelbling, L. (1995). Learning Policies for Partially Observable Environments: Scaling Up. *Proceedings of the 12th Intl. Conf. on Machine Learning*. p. 362-370, San Fransisko, USA.
- Maes, P. (1989). How to do the right thing. *Connection Science Journal*, 1(3): p. 291-323.
- Miller, I.; Campbell, M. (2007). Rao-Blackwellized Particle Filtering for Mapping Dynamic Environments. *IEEE International Conference on Robotics and Automation (ICRA)*, Rome, Italy.
- Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. (2002). FastSLAM: A Factored Solution to Simultaneous Localization and Mapping. *National Conf. on Artificial Intelligence (AAAI)*, Edmonton, Canada.
- Montemerlo, M.; Whittaker, W.; Thrun, S. (2002). Conditional Particle Filters for Simultaneous Mobile Robot Localization and People-Tracking. *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Washington, DC, USA.

- Moravec, H. (1989). Sensor fusion in certainty grids for mobile robots. *Sensor Devices and Systems for Robotics*, p. 243-276.
- Murphy, K. (1999). Bayesian map learning in dynamic environments. *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, p. 1015-1021.
- Pirjanian, P. (1999). Behavior coordination mechanisms -- state-of-the-art. *Technical Report IRIS-99-375*, Institute of Robotics and Intelligent Systems, School of Engineering, University of Southern California.
- Prescott, T.J.; Redgrave P.; Gurney, K. (1999). Layered control architectures in robots and vertebrates. *Adaptive Behavior*, 7:99-127.
- Rohrmüller, F.; Althoff, M.; Wollherr, D.; Buss, M. (2005). Probabilistic Mapping of Dynamic Obstacles Using Markov Chains for Replanning in Populated Environments. *IEEE Intl. Conf. on Robotics and Automation (IROS)*, Nice, France, Sept, 22-26.
- Stachniss, C.; Haehnel, D.; Burgard, W. (2004). Exploration with Active Loop-Closing for FastSLAM. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Sendai, Japan.
- Stachniss, C.; Grisetti, G.; Burgard, W. (2005). Information Gain-based Exploration Using Rao-Blackwellized Particle Filters. *Robotics: Science and Systems (RSS)*, Philadelphia, USA, p. 65-72.
- Thrun, S.; Liu, Y.; Koller, D.; Ng, A.; Ghahramani, Z.; Durrant-Whyte, H. (2004). Simultaneous Localization and Mapping with Sparse Extended Information Filters. *Int.J. Robotics Research*, 23(7-8), p. 693-716.
- Tyrrel, T. (1993). Computational Mechanisms for Action Selection. *PhD Thesis*, University of Edinburgh.
- Wang, C.; Thorpe, C.; Thrun, S. (2003). Online Simultaneous Localization And Mapping with Detection And Tracking of Moving Objects: Theory and Results from a Ground Vehicle in Crowded Urban Areas. *IEEE Int. Conf.on Robotics and Automation (ICRA)*, Taipei, Taiwan.
- Yamauchi, B. (1998). Frontier-based Exploration Using Multiple Robots. *Second Intl. Conference on Autonomous Agents*, Minneapolis, USA.

IntechOpen



Greedy Algorithms

Edited by Witold Bednorz

ISBN 978-953-7619-27-5

Hard cover, 586 pages

Publisher InTech

Published online 01, November, 2008

Published in print edition November, 2008

Each chapter comprises a separate study on some optimization problem giving both an introductory look into the theory the problem comes from and some new developments invented by author(s). Usually some elementary knowledge is assumed, yet all the required facts are quoted mostly in examples, remarks or theorems.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Georgios Lidoris, Dirk Wollherr and Martin Buss (2008). Bayesian Framework for State Estimation and Robot Behaviour Selection in Dynamic Environments, Greedy Algorithms, Witold Bednorz (Ed.), ISBN: 978-953-7619-27-5, InTech, Available from:

http://www.intechopen.com/books/greedy_algorithms/bayesian_framework_for_state_estimation_and_robot_behaviour_selection_in_dynamic_environments

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen