Auton Robot (2009) 27: 93–103
DOI 10.1007/s10514-009-9130-2

A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot

**Ruben Martinez-Cantin · Nando de Freitas ·
Eric Brochu · José Castellanos · Arnaud Doucet**

Received: 8 November 2008 / Accepted: 23 July 2009 / Published online: 4 August 2009

Abstract We address the problem of online path planning for optimal sensing with a mobile robot. The objective of the robot is to learn the most about its pose and the environment given time constraints. We use a POMDP with a utility function that depends on the belief state to model the finite horizon planning problem. We replan as the robot progresses throughout the environment. The POMDP is high-dimensional, continuous, non-differentiable, nonlinear, non-Gaussian and must be solved in real-time. Most existing techniques for stochastic planning and reinforcement learning are therefore inapplicable. To solve this extremely complex problem, we propose a Bayesian optimization method that dynamically trades off exploration (minimizing uncertainty in unknown parts of the policy space) and exploitation (capitalizing on the current best solution). We demonstrate our approach with a visually-guide mobile robot. The solution proposed here is also applicable to other closely-related domains, including active vision, sequential experimental design, dynamic sensing and calibration with mobile sensors.

R. Martinez-Cantin (✉)
Institute for Systems and Robotics, Instituto Superior Técnico,
Lisboa, Portugal
e-mail: rmcantin@isr.ist.utl.pt

N. de Freitas · E. Brochu · A. Doucet
Department of Computer Science, University of British
Columbia, Vancouver, Canada

N. de Freitas
e-mail: nando@cs.ubc.ca

E. Brochu
e-mail: ebrochu@cs.ubc.ca

A. Doucet
e-mail: arnaud@cs.ubc.ca

J. Castellanos
Department of Computer Science and System Engineering,
University of Zaragoza, Zaragoza, Spain

Keywords Bayesian optimization · Online path planning · Sequential experimental design · Attention and gaze planning · Active vision · Dynamic sensor networks · Active learning · Policy search · Active SLAM · Model predictive control · Reinforcement learning

1 Introduction

Online path planning is a fundamental and central problem in mobile robotics. The problem is notoriously hard because robots have to contend with environments that exhibit complex dynamics and unknown uncertainties. Furthermore, robots only have access to a restricted set of partial observations of the world because of their limited field of view and inherent motor constraints.

In this paper, we focus on an optimal sensing scenario where the robot must adaptively plan a path so as to gather observations in an optimal way. More precisely, the objective is for the robot to maximize the information about its location and the location of navigation landmarks in the environment. The main sensor is a simple inexpensive camera. We adopt a model predictive strategy, in which the robot replans the path as new observations are acquired. The robot has to achieve the optimal sensing goals while being subject to limited time and energy budgets, as well as, constraints imposed by its kinematic and dynamic capabilities.

Note that this problem is the same as the one of dynamically deploying a mobile sensor to learn about an environment. It is, therefore, of immediate relevance to the fields of *sensor networks*, *calibration* and *terrain-aided navigation*

(Bergman 1999; Paris and Le Cadre 2002; Meger et al. 2009; Hernandez et al. 2004; Singh et al. 2005). Moreover, since the primary sensor is a camera, this may be also be interpreted as an *active vision* application, where the robot has to decide where to attend to in order to dynamically understand a scene.

Online path planning is essential for proper simultaneous localization and mapping (SLAM) (Sim and Roy 2005; Stachniss et al. 2005). Mobile robots must maximize the size of the explored terrain, but, at the same time, must ensure that localization errors are minimized. While *exploration* is needed to find new features, the robot must return to places where known landmarks are visible to maintain reasonable map and pose estimates. Path planning also plays a key role in the theoretical and practical convergence of SLAM algorithms (Bailey et al. 2006; Martinez-Cantin et al. 2006).

Starting with a fixed horizon, we model the path planning problem with a partially observed Markov decision process (POMDP), with continuous states and actions. The complex robot dynamics and environmental coupling introduce nonlinearity, non-Gaussianity and non-differentiability in the POMDP. Moreover, unlike traditional POMDP models where the reward is a function of the actions and states directly, in our application the reward is a function of the belief state (also known as the information state in control or the posterior filtering distribution in Bayesian inference). This distinction creates additional high-dimensional integrals and recursions, which complicate the solution enormously. Utility (reward) functions that depend on the belief state are commonplace in the field of *experimental design* (Chaloner and Verdinelli 1995; Kueck et al. 2006). There, the formidable computational challenges, which arise when maximising expectations with respect to these utility functions, are well recognized.

Most existing reinforcement learning techniques are unable to cope with our high-dimensional, non-differentiable, continuous POMDPs (see Riedmiller et al. 2009 in this special issue for an introduction to reinforcement learning). Even a toy problem would require enormous computational effort. As a result, it is not surprising that most existing approaches relax the online stochastic path planning problem. For instance, full observability is assumed in Paris and Le Cadre (2002) and Sim and Roy (2005), known robot location and discrete actions are assumed in Leung et al. (2005) and Singh et al. (2007), a small set of actions and myopic planning is adopted in Stachniss et al. (2005); Vidal-Calleja et al. (2006); Bryson and Sukkarieh (2008), and discretization of the state and/or actions spaces is required in Hernandez (2004), Kollar and Roy (2008) and Sim and Roy (2005). *The method proposed in this work does not rely on any of the preceding assumptions.* In Singh et al. (2009), a sound way of exploiting sub-modularity in these planning and sensing domains is proposed. The approach in that publication is however restricted to off-line path planning and makes use of discrete state spaces. It is not clear yet how limiting or useful the sub-modularity assumption will prove to be in the more general problem that we attack here.

We represent the policy of the POMDP with a parameterized path. The robot can easily follow this planned path using either a standard PID controller or any other classical regulator. The robot replans (recomputes the path) as it moves through the environment. The complexity of the model demands the use of simulation techniques to approximate the cost function. However, since the dynamic model is non-differentiable, one cannot use gradient-based stochastic approximations (Konda and Tsitsiklis 2003; Singh et al. 2005) or policy gradient methods (Baxter and Bartlett 2001; Peters and Schaal 2008a, 2008b) to update the parameters of the policy. Instead, we propose here the adoption of Bayesian optimization techniques (Mockus et al. 1978; Jones 2001; Lizotte 2008). Bayesian optimization methods approximate the expected cost function with a surrogate function that is cheaper to evaluate: a Gaussian process in our case. The surrogate function's mean and covariance are used to choose the policy parameter values that should be tried next. After actively selecting a candidate parameter vector, simulations are conducted to obtain a new estimate of the expected cost function and the surrogate function is refit. The decision of what policy parameter to try next trades off exploration (trying parameters where the cost function is very uncertain) and exploitation (trying parameters where the cost function is known to be low). This global optimization technique has the nice property that it aims to minimize the number of cost function evaluations; a fundamental requirement for real-time mobile robotics. Moreover, unlike gradient-based methods, it is likely to do well even in settings where the cost function has many local minima.

2 Goals and model specification

The goal is for a robot to plan a path so as to minimize uncertainty about its pose (location and heading) and the location of environmental landmarks, which are often used for navigation. The typical setup is illustrated in Fig. 1. Initially, the robot has a rough probabilistic estimate of its pose and known landmark locations. As the robot explores, it must reduce the uncertainty in these variables whose existence is known. At the same time, it must recruit new landmarks into its representation of the environment whenever it encounters them for the first time.

The robot has a limited field of view. It can only observe landmarks that fall within its camera sight. Even when visual features are in sight, the robot may fail to detect these because of sensor limitations.

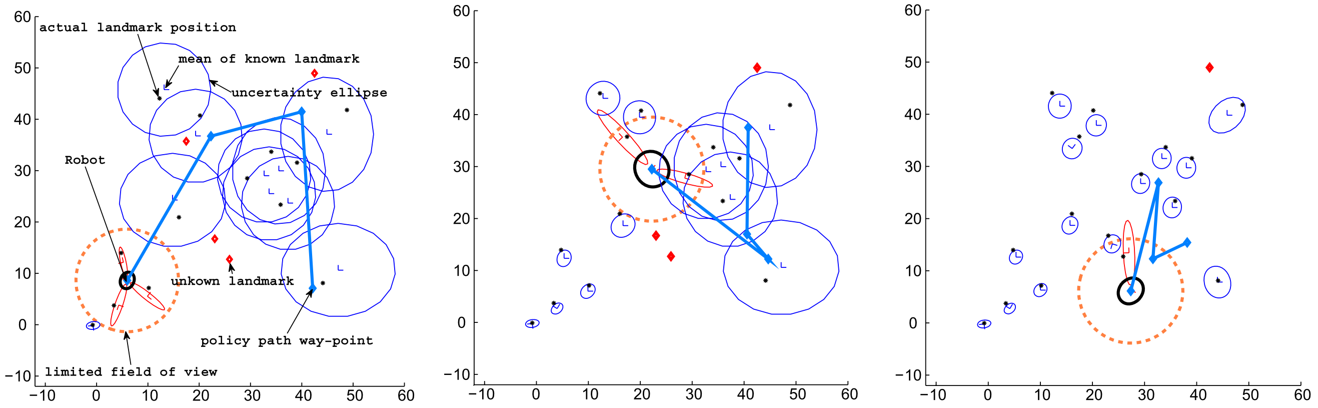

Fig. 1 This simulation shows three stages of the robot exploring an environment. The simulation includes landmarks that the robot does not know a priori. As soon as the robot observes these landmarks, it incorporates them into its model of the world. The robot continuously plans and replans so as to minimize the uncertainty in its pose and in the location of the known landmarks. The figure also shows the robot's limited field of view and the paths that it plans to follow at the three simulation stages

The path of the robot is parameterized in terms of a finite set of ordered way-points θ (which are used by a motion generator to compute a sequence of T commands $\mathbf{u}_{1:T}$) that take into account the kinematic and dynamic constrains of the robot and environment. Every few steps, the robot replans the path using the information gathered in these steps. This adaptive feedback process is necessary to avoid traps that open loop algorithms cannot escape. While exploring, the robot is subject to other constraints such as low energy consumption, limited time, safety measures and obstacle avoidance. However, for the time being, let us first focus on the problem of minimizing posterior errors in localization and mapping as this problem already captures an enormous degree of complexity.

Having restricted the problem to one of improving the information in the joint posterior distribution of the robot's pose and landmarks, a natural cost function for this T-step ahead stochastic planning problem is the average mean square error (AMSE) of the state:

$$C^{\pi}_{AMSE} = \mathbb{E}_{p(\mathbf{x}_{0:T},\mathbf{y}_{1:T}|\boldsymbol{\pi})}\left[\sum_{t=1}^{T}\lambda^{T-t}(\widehat{\mathbf{x}}_t-\mathbf{x}_t)(\widehat{\mathbf{x}}_t-\mathbf{x}_t)'\right], \qquad (1)$$

where $\lambda \in [0, 1]$ is a discount factor, $\pi(\theta)$ denotes the policy (path) parameterized by the way-points $\theta \in \mathbb{R}^{n_\theta}$, $\mathbf{x}_t \in \mathbb{R}^{n_x}$ is the hidden state (robot pose and location of map features) at time t, $\mathbf{y}_{1:T} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T\} \in \mathbb{R}^{n_y T}$ is the sequence of observations along the planned trajectory for T steps, $\mathbf{u}_{1:T} \in \mathbb{R}^{n_a T}$ is the sequence of actions, and $\widehat{\mathbf{x}}_t = \mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t},\boldsymbol{\pi})}[\mathbf{x}_t]$ is the estimate of the state. The expectation is taken with respect to the full path distribution:

$$p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \mid \boldsymbol{\pi}) = p(\mathbf{x}_0)\prod_{t=1}^{T} p(\mathbf{x}_t \mid \mathbf{u}_t, \mathbf{x}_{t-1})\, p(\mathbf{y}_t \mid \mathbf{x}_t, \mathbf{u}_t).$$

We may, alternatively, focus on the uncertainty of the posterior estimates at the end of a planning horizon:

$$C^{\pi}_{AMSE} = \mathbb{E}_{p(\mathbf{x}_T,\mathbf{y}_{1:T}|\boldsymbol{\pi})}[(\widehat{\mathbf{x}}_T-\mathbf{x}_T)(\widehat{\mathbf{x}}_T-\mathbf{x}_T)']. \qquad (2)$$

Note that the true states and observations are unknown in advance and so one has to marginalize over them. Note also that the cost function is a matrix and must be mapped to a scalar. This can be done by either taking the trace or determinant of this matrix. This choice of cost function is a sensible one when the objective is to minimize the uncertainty in the model parameters (Chaloner and Verdinelli 1995; Sim and Roy 2005).

The cost function, transition and observation models, and policy define our POMDP model. This POMDP variant is simpler than classical POMDPs in that the policy is not parameterized explicitely in terms of the belief state. On the other hand, the utility is now a function of the belief state $p(\mathbf{x}_t \mid \mathbf{y}_{1:t}, \boldsymbol{\pi})$. The expensive and difficult problem of estimating the belief state is known as SLAM in robotics (Durrant-Whyte and Bailey 2006).

3 Solving the POMDP with direct policy search

Since the models are not linear-Gaussian, one cannot use standard Linear Quadratic Gaussian (LQG) controllers to solve the problem. Moreover, since the action and state spaces are high-dimensional and continuous, discretization as in Tremois and Le Cadre (1999) and many other works would fail. The discretized POMDP is too large for stochastic dynamic programming techniques derived from the seminal work of Smallwood and Sondik (1973). The fact that the utility depends on the belief state further complicates the problem.

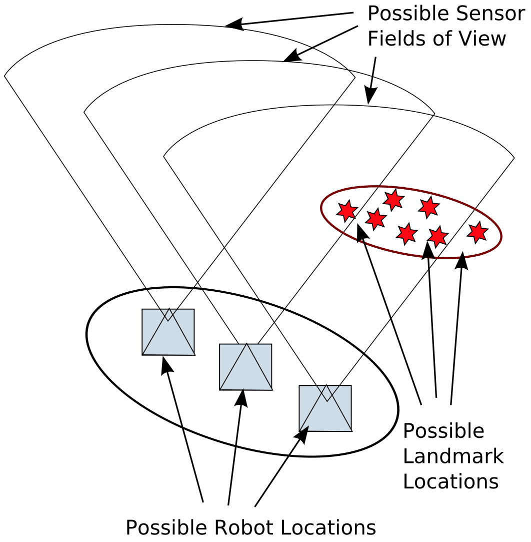

Fig. 2 Simulating future observations using the prior information is not trivial because of discontinuities in the observation model due to the limited field of view and occlusion. The observations are generated by drawing samples from the posterior distributions of the robot pose and landmark locations. If the landmark samples fall within the simulated field of view, they are detected with a predefined probability. That is, we incorporate the detection rates of the sensors

```
1. Choose an initial policy $\pi_0(\theta)$.
2. For $j = 1 : MaxNumberOfPolicySearchIterations$
   (a) For $i = 1 : M$
       i. Sample the prior states $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$.
       ii. For $t = 1 : T$
           A. Use a motion controller regulated about the
              path $\pi(\theta)$ to determine the current action $\mathbf{u}_t^{(i)}$.
           B. Sample the state $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t|\mathbf{u}_t^{(i)}, \mathbf{x}_{t-1}^{(i)})$.
           C. Generate observations $\mathbf{y}_t^{(i)} \sim p(\mathbf{y}_t|\mathbf{u}_t^{(i)}, \mathbf{x}_t^{(i)})$ as
              described in Figure 2.
           D. Compute the belief state $p(\mathbf{x}_t|\mathbf{y}_{1:t}^{(i)}, \mathbf{u}_{1:t}^{(i)})$ using
              a SLAM filter.
   (b) Evaluate the approximate AMSE cost function of equation (3) using the simulated trajectories.
   (c) Choose a new promising set of policy parameters $\theta$ using
       Bayesian optimization.
```

Fig. 3 Direct policy search strategy for T-steps ahead planning. Samples that fail to satisfy time and energy budgets are rejected

To overcome these difficulties, we adopt the direct policy search method for solving POMDPs (Williams 1992; Baxter and Bartlett 2001; Ng and Jordan 2000); see also the papers of Howard et al. (2009); Stolle and Atkeson (2009); Vlassis et al. (2009) in this special issue. In this approach, the cost function is approximated with simulations. This is appealing in our setting because we have reasonable sensor and actuator models, which enable us to simulate trajectories with relatively low variance. Specifically, given M simulated trajectories, as described in Fig. 3, the cost function (2) may be approximated with a Monte Carlo average:

$$C_{AMSE}^{\pi} \approx \frac{1}{M}\sum_{i=1}^{M}(\widehat{\mathbf{x}}_T^{(i)} - \mathbf{x}_T^{(i)})(\widehat{\mathbf{x}}_T^{(i)} - \mathbf{x}_T^{(i)})'. \tag{3}$$

In the simulator, the actions are generated by following the current path (policy) with a simple controller and the states are sampled according to the transition model. The observations $\mathbf{y}$ are hallucinated using the procedure outlined in Fig. 2. After the trajectories $\{\mathbf{x}_{1:T}^{(i)}, \mathbf{y}_{1:T}^{(i)}\}_{i=1}^{M}$ have been obtained, a SLAM filter (EKF, UKF or particle filter) is used to compute the posterior mean of the belief state $\widehat{\mathbf{x}}_{1:T}^{(i)}$.

In policy search, the approximated cost function is used to update the policy parameters θ. Typically, this is done by following stochastic gradients (Baxter and Bartlett 2001; Peters and Schaal 2006). However, in our domain, the cost function is not differentiable. Hence, we must come up with a different approach that does not require differentiability. The new approach must also take computation into account. Note that approximating the cost function with samples requires that we compute the belief state (i.e. solve the SLAM problem) for each sample. This is extremely expensive. The new approach must therefore minimize the number of queries. For these reasons, we chose to carry out Bayesian optimization to update the parameters. This optimization procedure will be discussed in detail in the following section. The overall algorithm for T-steps ahead open-loop planning is shown in Fig. 3.

As the robot moves along the planned path, it is possible to use the newly gathered observations to update the posterior distribution of the state. This distribution can then be used as the prior for subsequent simulations. This process of replanning is known as open-loop feedback control (OLFC), see Bertsekas (1995). We can also allow for the planning horizon to recede. That is, as the robot moves, it keeps planning T steps ahead of its current position. This control framework is also known as receding-horizon model predictive control, see Maciejowski (2002) for a review. We will use the terms *open-loop feedback control* and *model predictive control* interchangeably.

4 Bayesian optimization of the policy parameters

The objective of Bayesian optimization is to find the minimum of the cost function with as few cost evaluations as possible (Kushner 1964; Jones et al. 1998; Locatelli 1997; Mockus et al. 1978). In direct policy search, the evaluation of the expected cost using Monte Carlo simulations is very costly. One therefore needs to find a minimum of this function with as few policy iterations as possible.

Bayesian optimization provides an exploration-exploitation mechanism for finding multiple minima. Unlike traditional active learning, where the focus is often only in exploration (e.g. query the points with the maximum variance, entropy or other information-theoretic measures), here the goal is to balance exploitation and exploration. That is, to save computation, we only want to approximate the cost function accurately in regions where it is profitable to do so. We do not need to approximate it well over the entire state space.

One could apply the Bayesian optimization approach to learn value functions in areas of interest, say areas of high value, but we shall not pursue this approach here. Instead, we will apply the approach to find the minima of the expected cost as a function of the policy parameters θ.

Bayesian optimization involves three stages. First, a prior distribution is defined over the object being analyzed. In our case, the object is the cost function $C^{\pi}(\cdot)$. More precisely, it is the trace of the AMSE matrix, but we drop the trace symbol for ease of notation.

Second, a set of N previously gathered measurements $D_{1:N} = \{\theta_i, C^{\pi}(\theta_i)\}_{i=1}^{N}$ is combined with the prior, through Bayes rule, to obtain the posterior distribution over the object. Note that N corresponds to the number of policy search iterations thus far. At iteration j of policy search, we choose a parameter value θ_j and evaluate the corresponding cost $C^{\pi}(\theta_j)$.

Finally, the posterior risk is minimized so as to determine which new parameters θ should be tried next. Mathematically, the point of maximum expected improvement, as formulated by Mockus et al. (1978), is given by:

$$\theta_{N+1} = \arg\max_{\theta} \mathbb{E}[\max\{0, C^{\pi}_{\min} - C^{\pi}(\theta)\} | D_{1:N}], \tag{4}$$

where $I(\theta) = \max\{0, C^{\pi}_{\min} - C^{\pi}(\theta)\}$ denotes the improvement over a defined standard. Here, our standard is the best (lowest) value of the cost function thus far, $C^{\pi}_{\min}$. The expectation is taken with respect to the posterior distribution $P(C^{\pi}(\cdot)|D_{1:N})$. In contrast with the popular worst-case (minimax) approach, this average-case analysis can provide faster solutions in many practical domains where one does not believe that the worst case scenario is very probable.

To implement the first stage of Bayesian optimization, we place a Gaussian process (GP) prior over the expected cost function: $C^{\pi}(\cdot) \sim GP(m(\cdot), K(\cdot,\cdot))$, see e.g. Rasmussen and Williams (2006) for details on Gaussian process regression. The inherent assumption here is one of smoothness. Although we actually learn the mean function in the manner proposed in Martinez-Cantin et al. (2007a), for presentation clarity let us assume that it is the zero function as it is often the case in the machine learning literature. For details on the actual implementation, see Martinez-Cantin (2008). We adopt the standard Gaussian and Matern kernel functions to describe the components of the kernel matrix $\mathbf{K}$. The parameters of these functions (say kernel width in the Gaussian case) can be learned by maximum a posteriori inference, but since in our active learning setting we don't have many data points, the priors need to be fairly informative. That is, one has to look at the data and get a rough estimate of the expected distance between data points in order to choose the smoothing kernel width.

It is then easy to obtain an exact expression for the mean, μ, and variance, σ^2, of the posterior distribution:

$$\begin{aligned} \mu(\theta) &= \mathbf{k}^T \mathbf{K}^{-1} C^{\pi}_{1:N}, \\ \sigma^2(\theta) &= k(\theta,\theta) - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}. \end{aligned} \tag{5}$$

where $C^{\pi}_{1:N} = (C^{\pi}_1, \ldots, C^{\pi}_N)$, $\mathbf{K}$ denotes the full kernel matrix and $\mathbf{k}$ denotes the vector of kernels $k(\theta, \theta_i)$ for $i = 1, \ldots, N$. Since the number of query points is small, the GP predictions are very easy to compute.

The expectation of the function $I(\theta) = \max\{0, C^{\pi}_{\min} - C^{\pi}(\theta)\}$, with respect to the Gaussian process posterior distribution $\mathcal{N}(C^{\pi}(\theta); \mu(\theta), \sigma^2(\theta))$, can be computed by integrating by parts:

$$\mathbb{E}(I(\theta)) = \int_{I=0}^{I=\infty} I \left[\frac{1}{\sqrt{2\pi\sigma^2(\theta)}} e^{-\frac{(C^{\pi}_{\min} - I - \mu(\theta))^2}{2\sigma^2(\theta)}} \right] dI.$$

This results in the following expression:

$$EI(\theta) = \begin{cases} (C^{\pi}_{\min} - \mu(\theta))\Phi(d) + \sigma^2(\theta)\phi(d) & \text{if } \sigma^2 > 0 \\ 0 & \text{if } \sigma^2 = 0, \end{cases} \tag{6}$$

where ϕ and Φ denote the PDF and CDF of the standard Normal distribution and $d = \frac{C^{\pi}_{\min} - \mu(\theta)}{\sigma^2(\theta)}$. Finding the maximum of the expected improvement function is a much easier problem than the original one because the expected improvement function (also known as the infill function) can be cheaply evaluated. Several refinements of this infill function have been proposed in Schonlau et al. (1998) and Sasena (2002). To optimize the expected improvement function, we used the DIRect algorithm (Jones et al. 1993; Finkel 2003; Gablonsky 2001), though other methods such as sequential quadratic programming could also be adopted.

The overall procedure is shown in Fig. 4 and illustrated in Fig. 5. Many termination criteria are possible, including time and other computational constraints. When carrying out direct policy search, the Bayesian optimization approach has several advantages over the policy gradients method: it is derivative free, it is less prone to be caught in the first local minimum, and it is *explicitly designed to minimize the number of expensive cost function evaluations*.

Bayesian optimization has a long history, starting with the seminal work of Kushner (1964) with one-dimensional Wiener processes. It has been successfully applied to derivative-free optimization and experimental design Jones et al. (1998) and has recently begun to appear in the machine

```
– At policy search iteration $N$:
    – Update the expressions for the mean and variance functions of the GP $\mathcal{N}(C^{\pi}(\theta);\mu(\theta),\sigma^2(\theta))$ using the data $D_{1:N}$.
    – Choose $\theta_{N+1} = \arg\max_{\theta} EI(\theta)$.
    – Evaluate $C^{\pi}_{N+1} = C^{\pi}(\theta_{N+1})$ by running simulations as described in Figure 3.
    – Augment the data $D_{1:N+1} = \{D_{1:N}, (\theta_{N+1}, C^{\pi}_{N+1})\}$.
    – $N = N+1$.
```

Fig. 4 Bayesian optimization algorithm

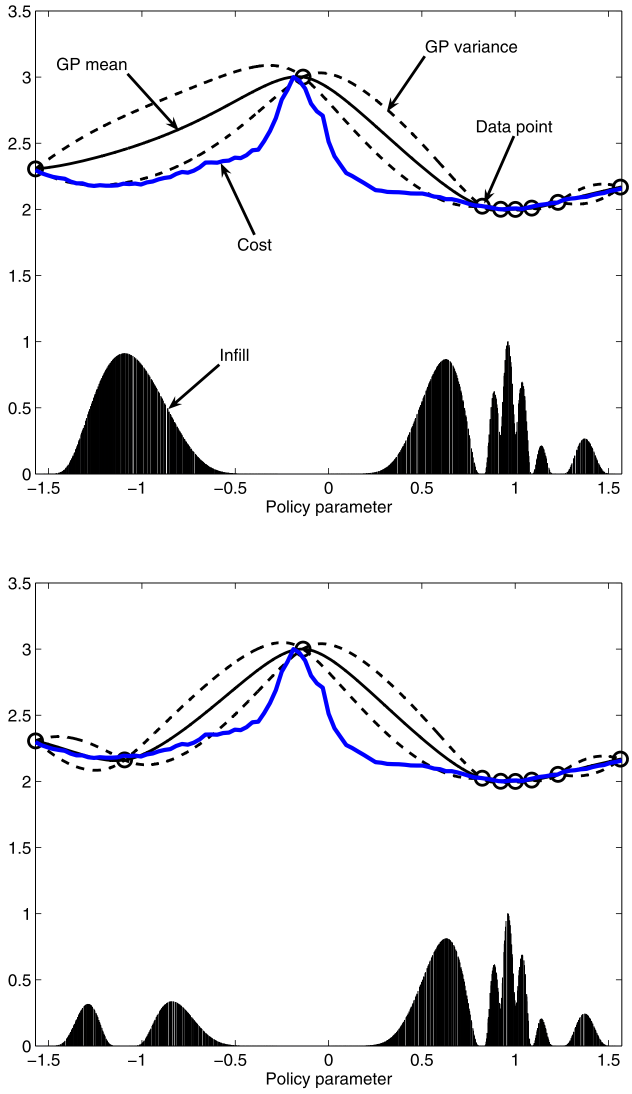

Fig. 5 An example of Bayesian optimization. The figure on top shows a GP approximation of the cost function using 11 simulated values. In reality, the true expected cost function is unknown. The figure also shows the expected improvement (infill) of each potential next sampling location in the lower shaded plot. The infill is high where the GP predicts a low expected cost (exploitation) and where the prediction uncertainty is high (exploration). Selecting and labelling the point suggested by the highest infill in the top plot produces the GP fit in the plot shown below

learning literature (Lizotte et al. 2007; Brochu et al. 2007). There are several consistency proofs for this algorithm in the one-dimensional setting (Locatelli 1997). There are also convergence proofs for a simplification of the algorithm using simplicial partitioning in higher dimensions (Zilinskas and Zilinskas 2002) and, more recently, for GPs with Matern kernels in Vazquez and Bect (2008). The question of obtaining rates of convergence for these algorithms in high-dimensions is still open.

5 Simulations

In our simulation setup, the environment is a free space area with several point features distributed at random (see Fig. 7). The a priory map is known with very high uncertainty, ranging from 1 to 5 m of standard deviation. The simulated robot is a hovering aerial vehicle equipped with inertial sensors, a camera and an altimeter. The field of view is limited to 7 meters. We adopted a detection system that provides observations every 0.5 seconds. The sensor noise is Gaussian for both range and bearing, with standard deviations $\sigma_{range} = 0.2 \cdot range$ and $\sigma_{bearing} = 2^o$. The policy is given by a set of ordered way-points. The motion commands are updated by a controller every 0.1 seconds. The controller guarantees that the robot is heading toward the goal, or hovering over it, for a fixed amount of time.

We compared the behavior of the robot using different planning and acting horizons:

OLC1 : This is an greedy algorithm that select the most informative way-point ahead in open-loop, i.e., the planning horizon is 1.

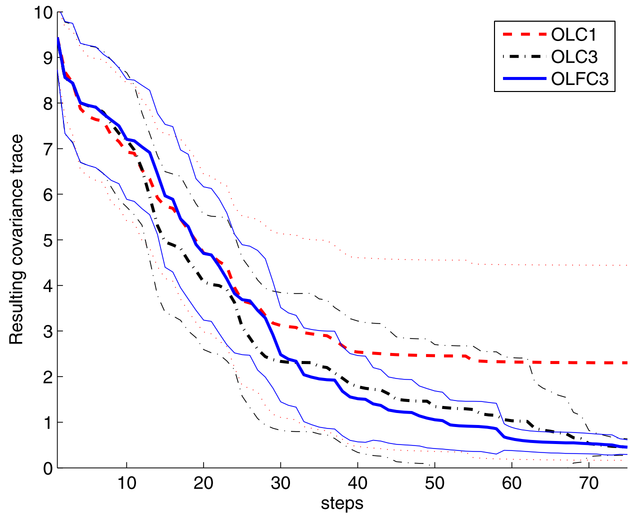

Fig. 6 Evolution of the trace of the state covariance matrix for 15 random maps using OLC1, OLC3 and OLFC3, with 99% confidence intervals

OLC3 : This is an open loop algorithm that plans with 3 way-points ahead. The planning process is still myopic.

OLFC3 : This is an open loop feedback controller with receding horizon, i.e., a model predictive controller. The planning horizon is again 3 way-points, but this time, the robot only executes 1 step before replanning.

It is obvious that the OLC algorithms have a lower computational cost. On the other hand, they can get easily trapped in local minima. As shown in Fig. 6, the open loop approaches have much higher variance. That is, in some of the runs they get trapped exploiting small regions of the map and fails to explore the environment properly. This lack of robustness is unacceptable, so we favor the OLFC approaches.

We tested our algorithms by simulating several maps as shown in Fig. 7. We considered two alternative methods for comparison purposes. The first is a standard heuristic for ex-

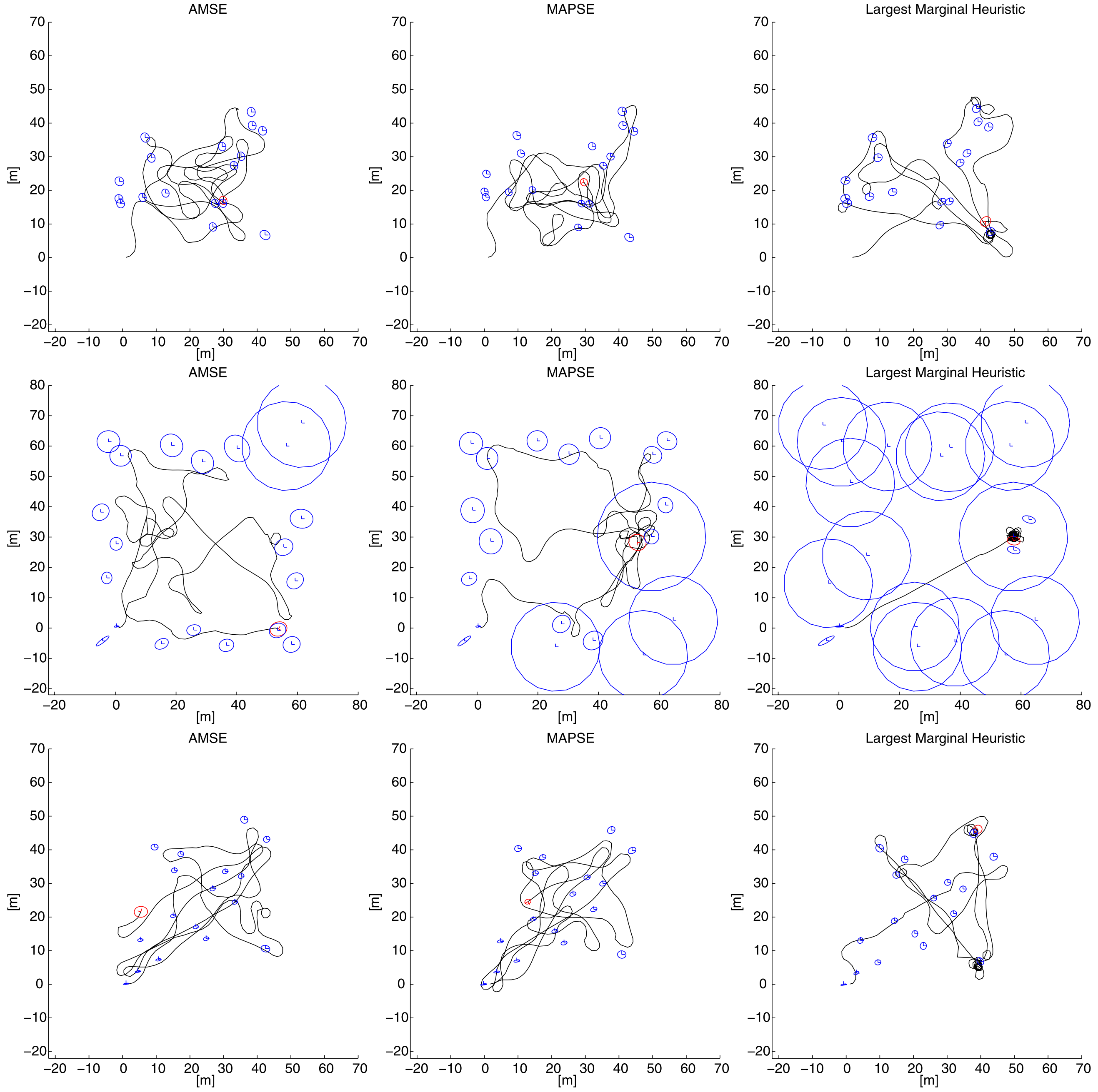

Fig. 7 Exploration trajectories for different environments and different cost functions. Each row is a different scenario while each column is a different cost, from *left* to *right*: AMSE, MAPSE, Largest Marginal Heuristic. The landmarks are plotted in *blue*; the robot, in *red*. The ellipses represent the uncertainty threshold at 95%

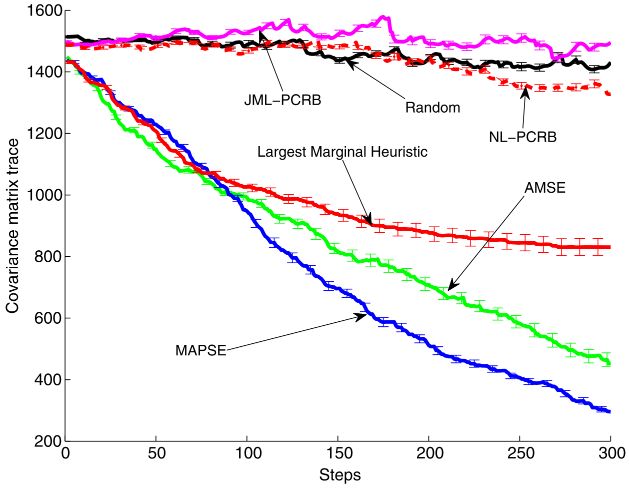

Fig. 8 Evolution of the trace of the state covariance matrix for 15 runs with AMSE, MAPSE and Large Marginal Heuristic cost function using OLFC3. The comparison also shows two posterior Cramer-Rao bounds on the AMSE cost function

ploration in robotics, which we will refer to as the *Largest Marginal Heuristic* approach. Here, the robot follows the shortest path to the landmark of highest uncertainty. We also designed an alternative, which we call *Maximum a Posteriori Square Error* (MAPSE), that uses *only one sample* of the cost function at each policy update. In particular, one uses the maximum a posteriori values of the map and robot location at the current optimization step and feeds these into the simulator to obtain a fast estimate of the cost function. This estimate is used to update the policy parameters in the Bayesian optimization step. As shown in Fig. 7 this procedure seems to do as well as the AMSE approach. However, it is far more efficient computationally because it only requires one simulation to approximate the cost function. Figure 8 shows the results of several simulations, where we have also included the posterior Cramer-Rao Bounds of the AMSE cost function (Martinez-Cantin et al. 2007b). Clearly the MAPSE approach is not only efficient (a requirement for real-time implementation), but also results in the lowest error. The success of this method is a result of the fact that our models are fairly accurate and consequently the simulations have low variance.

Finally, we also tested the MAPSE approach in environments where not all features are known a priori. By enabling the robot to augment its map with new features, we were able to get similar results. Figure 1 shows the results of one typical simulation run.

6 Experiments with a Pioneer robot

The method presented in this paper has been tested on a Pioneer robot with a low-cost web-camera. The map is built using fiducial landmarks with the ARToolkit Tracker of Kato and Billinghurst (1999). The robot and the experimental setup are shown in Fig. 9. The laptop on top on the robot is in charge of all computation: image processing, motion control, planning, SLAM and so on. The navigation is carried out in real-time.

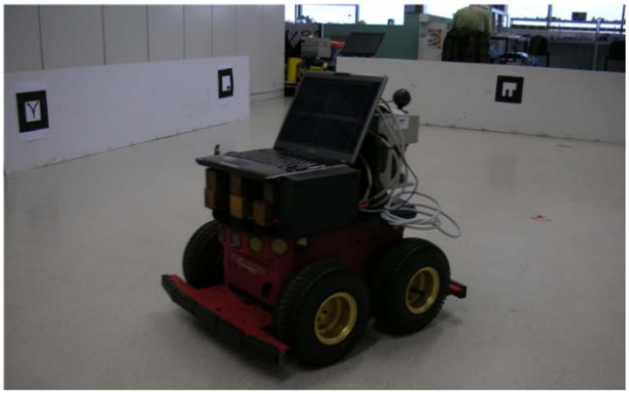

Fig. 9 Mobile robot and experimental environment

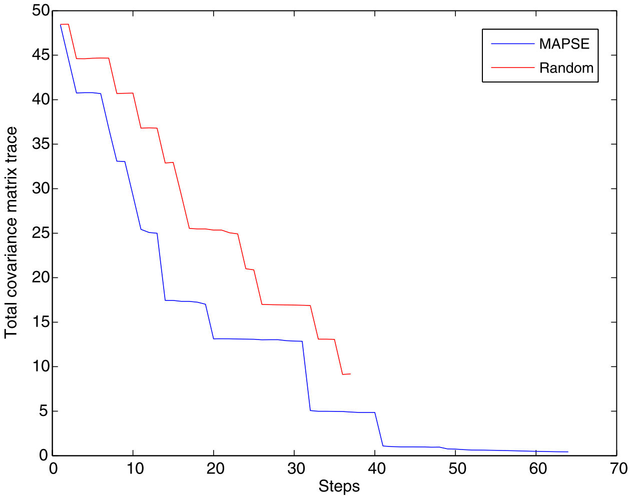

Fig. 10 Total uncertainty (trace of the full covariance matrix) for the Pioneer robot

We chose fiducial landmarks for comparison purposes since, in this case, the algorithms share the same map. Feature detectors using natural images are not very robust and consequently are not suitable for accurate comparison and benchmarking. However, the system was built using the YARP middleware by Metta et al. (2006) and therefore it allows us to use local feature detectors, such as SIFT or SURF.

Figure 10 shows a comparison of the MAPSE approach against a random sampling of the way-points. The random exploration finishes earlier than the active version because, at that point, the robot is lost and cannot navigate properly. When this happens, it tries to navigate outside the map and hits the walls. Figure 11 compares the robot uncertainty again for the active and random exploration. In the random exploration, the robot uncertainty is considerably larger.

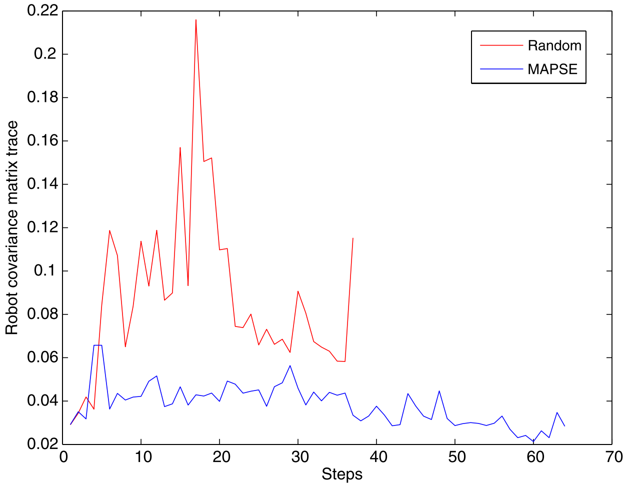

Fig. 11 Robot location uncertainty

7 Conclusions

We presented a computationally efficient method for online path planning. The method was shown to allow a robot to plan a path so as to explore its environment in an optimal way. Comparisons against many existing alternatives show that the method has significant promise. There are many avenues for further work: deciding what landmarks are convenient for navigation, testing other strategies for recruiting new landmarks, and allowing for more sophisticated policies.

Acknowledgements We would like to thank Nick Roy, Hendrik Kueck, Andrew Ng and Luis Montesano for valuable discussions. Also, Matthijs Spaan and Manuel Lopes for their technical advice. This work has been supported by FCT, project PTDC/EEA-ACR/70174/2006, NSERC and MITACS.

References

Bailey, T., Nieto, J., Guivant, J., Stevens, M., & Nebot, E. (2006). Consistency of the EKF-SLAM algorithm. In *Proc. of the IEEE/RSJ int. conf. on intelligent robots and systems*, 2006.

Baxter, J., & Bartlett, P. L. (2001). Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, *15*(4), 319–350.

Bergman, N. (1999). *Recursive Bayesian estimation: navigation and tracking applications*. PhD thesis, Linköping University.

Bertsekas, D. (1995). *Dynamic programming and optimal control*. Nashua: Athena Scientific.

Brochu, E., de Freitas, N., & Ghosh, A. (2007). Active preference learning with discrete choice data. In *Advances in neural information processing systems*, 2007.

Bryson, M., & Sukkarieh, S. (2008). Observability analysis and active control for airborne SLAM. *IEEE Transaction on Aerospace Electronic Systems*, *44*(1), 261–280.

Chaloner, K., & Verdinelli, I. (1995). Bayesian experimental design: a review. *Journal of Statistical Science*, *10*, 273–304.

Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localisation and mapping (SLAM): part I the essential algorithms. *Robotics and Automation Magazine*, *13*, 99–110.

Finkel, D. (2003). DIRECT optimization algorithm user guide. Center for Research in Scientific Computation, North Carolina State University.

Gablonsky, J. (2001). *Modification of the DIRECT algorithm*. PhD thesis, Department of Mathematics, North Carolina State University, Raleigh, North Carolina.

Hernandez, M. (2004). Optimal sensor trajectories in bearings-only tracking. In P. Svensson & J. Schubert (Eds.), *Proc. of the seventh int. conf. on information fusion, international society of information fusion*, Mountain View, CA (Vol. II, pp. 893–900).

Hernandez, M., Kirubarajan, T., & Bar-Shalom, Y. (2004). Multisensor resource deployment using posterior Cramèr-Rao bounds. *IEEE Transactions on Aerospace Electronic Systems*, *40*(2), 399–416.

Howard, M., Klanke, S., Gienger, M., Goerick, C., & Vijayakumar, S. (2009). A novel method for learning policies from variable constraint data. *Autonomous Robots*, *27* (Special issue on Robot Learning, Part B) (this issue).

Jones, D. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, *21*, 345–383.

Jones, D., Perttunen, C., & Stuckman, B. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, *79*(1), 157–181.

Jones, D., Schonlau, M., & Welch, W. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, *13*(4), 455–492.

Kato, H., & Billinghurst, M. (1999). Marker tracking and hmd calibration for a video-based augmentedreality conferencing system. In *Proc. of the 2nd IEEE and ACM int. work. on augmented reality* (pp. 85–94) 1999.

Kollar, T., & Roy, N. (2008). Trajectory optimization using reinforcement learning for map exploration. *International Journal of Robotics Research*, *27*(2), 175–197.

Konda, V., & Tsitsiklis, J. (2003). On actor-critic algorithms. *SIAM Journal on Control and Optimization*, *42*(4), 1143–1166.

Kueck, H., de Freitas, N., & Doucet, A. (2006). SMC samplers for Bayesian optimal nonlinear design. In *Nonlinear statistical signal processing workshop (NSSPW)*, 2006.

Kushner, H. (1964). A new method of locating the maximum of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, *86*, 97–106.

Leung, C., Huang, S., Dissanayake, G., & Forukawa, T. (2005). Trajectory planning for multiple robots in bearing-only target localisation. In *Proc. of the IEEE/RSJ int. conf. on intelligent robots and systems*, 2005.

Lizotte, D. (2008). *Practical Bayesian optimization*. PhD thesis, Dept. of Computer Science, University of Alberta.

Lizotte, D., Wang, T., Bowling, M., & Schuurmans, D. (2007). Automatic gait optimization with Gaussian process regression. In *International joint conference on artificial intelligence*, 2007.

Locatelli, M. (1997). Bayesian algorithms for one-dimensional global optimization. *Journal of Global Optimization*, *10*, 57–76.

Maciejowski, J. (2002). *Predictive control: with constraints*. New York: Prentice-Hall.

Martinez-Cantin, R. (2008). *Active map learning for robots: insights into statistical consistency*. PhD thesis, University of Zaragoza.

Martinez-Cantin, R., de Freitas, N., & Castellanos, J. (2006). Analysis of particle methods for simultaneous robot localization and mapping and a new algorithm: Marginal-SLAM. In *Proc. of the IEEE int. conf. on robotics & automation*, 2006.

Martinez-Cantin, R., de Freitas, N., & Castellanos, J. (2007a). Active policy learning for robot planning and exploration under uncertainty. In *Proc. of robotics: science and systems*, 2007.

Martinez-Cantin, R., de Freitas, N., Doucet, A., & Castellanos, J. (2007b). Active policy learning for robot planning and exploration under uncertainty. In *Robotics: science and systems (RSS)*, 2007.

Meger, D., Marinakis, D., Rekleitis, I., & Dudek, G. (2009). Inferring a probability distribution function for the pose of a sensor network using a mobile robot. In: *ICRA*, 2009.

Metta, G., Fitzpatrick, P., & Natale, L. (2006). Yarp: yet another robot platform. *International Journal on Advanced Robotics Systems*, *3*(1), 140–151.

Mockus, J., Tiesis, V., & Zilinskas, A. (1978). The application of Bayesian methods for seeking the extremum. In L. Dixon & G. Szego (Eds.), *Towards global optimisation* (Vol. 2, pp. 117–129). Amsterdam: Elsevier.

Ng, A., & Jordan, M. (2000). PEGASUS: a policy search method for large MDPs and POMDPs. In *Proc. of the sixteenth conf. on uncertainty in artificial intelligence*, 2000.

Paris, S., & Le Cadre, J. (2002). Planification for terrain-aided navigation. In *Fusion 2002, Annapolis, Maryland* (pp. 1007–1014).

Peters, J., & Schaal, S. (2006). Policy gradient methods for robotics. In *Proc. of the IEEE/RSJ int. conf. on intelligent robots and systems*, 2006.

Peters, J., & Schaal, S. (2008a). Natural actor critic. *Neurocomputing*, *71*(7–9), 1180–1190.

Peters, J., & Schaal, S. (2008b). Reinforcement learning of motor skills with policy gradients. *Neural Networks*, *21*(4), 682–697.

Rasmussen, C., & Williams, C. (2006). *Gaussian processes for machine learning*. Cambridge: The MIT Press.

Riedmiller, M., Gabel, T., Hafner, R., & Lange, S. (2009). Reinforcement learning for robot soccer. *Autonomous Robots*, *27*(1), 55–73 (Special issue on Robot Learning, Part A).

Sasena, M. (2002). *Flexibility and efficiency enhancement for constrained global design optimization with Kriging approximations*. PhD thesis, University of Michigan.

Schonlau, M., Welch, W., & Jones, D. (1998). Global versus local search in constrained optimization of computer models. In N. Flournoy, W. Rosenberger, W. Wong (Eds.) New developments and applications in experimental design (Vol. 34, pp. 11–25). Institute of Mathematical Statistics.

Sim, R., & Roy, N. (2005). Global A-optimal robot exploration in SLAM. In *Proc. of the IEEE int. conf. on robotics & automation*, 2005.

Singh, A., Krause, A., Guestrin, C., Kaiser, W., & Batalin, M. (2007). Efficient planning of informative paths for multiple robots. In *Proc. of the int. joint conf. on artificial intelligence*, 2007.

Singh, A., Krause, A., Guestrin, C., & Kaiser, W. (2009). Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research (JAIR)*, *34*, 707–755.

Singh, S., Kantas, N., Doucet, A., Vo, B., & Evans, R. (2005). Simulation-based optimal sensor scheduling with application to observer trajectory planning. In *Proc. of the IEEE conf. on decision and control and eur. control conference* (pp. 7296–7301) 2005.

Smallwood, R., & Sondik, E. (1973). The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, *21*, 1071–1088.

Stachniss, C., Grisetti, G., & Burgard, W. (2005). Information gain-based exploration using Rao-Blackwellized particle filters. In *Proc. of robotics: science and systems*, Cambridge, USA, 2005.

Stolle, M., & Atkeson, C. (2009). Finding and transferring policies using stored behaviors. *Autonomous Robots*, *27* (Special issue on Robot Learning, Part B) (this issue).

Tremois, O., & Le Cadre, J. (1999). Optimal observer trajectory in bearings-only tracking for manoeuvering sources. *IEE Proceeding Radar, Sonar Navigation*, *146*(1), 31–39.

Vazquez, E., & Bect, J. (2008). On the convergence of the expected improvement algorithm. arXivorg arXiv:0712.3744v2 [stat.CO], http://arxiv.org/abs/0712.3744v2.

Vidal-Calleja, T., Davison, A., Andrade-Cetto, J., & Murray, D. (2006). Active control for single camera SLAM. In *Proc. of the IEEE int. conf. on robotics & automation* (pp. 1930–1936) 2006.

Vlassis, N., Toussaint, G. K. M., & Piperidis, S. (2009). Learning model-free robot control using a Monte Carlo em algorithm. *Autonomous Robots*, *27* (Special issue on Robot Learning, Part B) (this issue).

Williams, R. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, *8*(3), 229–256.

Zilinskas, A., & Zilinskas, J. (2002). Global optimization based on a statistical model and simplicial partitioning. *Computers and Mathematics with Applications*, *44*, 957–967.

Ruben Martinez-Cantin earned his M.S. degree in electrical engineering from the University of Zaragoza, Spain, in 2003. He obtained his Ph.D. degree in 2008 at the University of Zaragoza. He is currently a Post-doctoral researcher at IST in Lisbon. His research interests include mobile robotics, computer vision, sensor data fusion, probabilistic inference and learning, and behaviour models.

Nando de Freitas is an Associate Professor in the Department of Computer Science at UBC. He is also an associate member of the statistics department and teaches in the cognitive systems programme. From 1999 to 2001, he was a visiting post-doctoral scholar with Stuart Russell at the University of California, Berkeley, where he worked on computer vision, probabilistic methods for modern artificial intelligence, and sophisticated inference and learning methods for structured probabilistic models. He obtained his Ph.D. on Bayesian methods for neural networks in 1999 from Trinity College, Cambridge University, following B.Sc. and M.Sc. degrees (with distinction) from the University of the Witwatersrand, Johannesburg.

Eric Brochu is a Ph.D. candidate at the University of British Columbia in Vancouver, Canada. He received a B.A. and B.Sc. from the University of Regina in Saskatchewan, and an M.Sc. from the University of British Columbia. His research interests include applications of machine learning to search, graphics, and user interfaces. Outside of academia, he has worked with a variety of organizations, including IBM, the Vancouver Art Gallery, and most recently, Worio in Vancouver.

José Castellanos earned his M.S. and Ph.D. degrees in industrial-electrical engineering from the University of Zaragoza, Spain, in 1994 and 1998, respectively. He is associate professor with the Departamento de Informática e Ingenierfa de Sistemas, University of Zaragoza, where he is in charge of courses in SLAM, automatic control systems, and computer modelling and simulation. His current research interests include multisensor fusion and integration, Bayesian estimation in nonlinear systems, and simultaneous localization and mapping.

Arnaud Doucet received his M.S. degree from the Institut National des Telecommunications, Paris, France, in 1993 and his Ph.D. degree from University Paris XI, Orsay, France, in 1997. He is an associate professor with the Departments of Computer Science and Statistics at UBC. His research interests include Markov chain Monte Carlo methods, sequential Monte Carlo methods, Bayesian statistics, and Hidden Markov models. He is coeditor of the book Sequential Monte Carlo Methods in Practice (2001).