Stochastic control for maximizing mutual information in active sensing

Stochastic control for maximizing mutual information in active sensing

Mikko Lauri[1] and Risto Ritala[1]

Abstract— **We study an active sensing problem where a robot must decide in which direction to focus the attention of its machine vision system. The robot's objective is to gain a maximum amount of information. The problem is formulated as a model-based finite horizon stochastic control problem with a continuous n-dimensional state space, also known as a partially observable Markov decision process (POMDP). Selection of an appropriate utility function for the control problem is discussed and an approximate solution via open-loop feedback control is presented. The empirical results illustrate the effects of selecting a greedy control strategy only maximizing immediate utility or a non-myopic strategy that also considers possible future utility.**

I. INTRODUCTION

Active sensing is the systematic selection of the focus of attention of an agent to achieve a goal. The goal may be completing an assignment not related to sensing per se: in this case information gathering via sensing is merely a degree of freedom in the operation of the system the same as any other action. Information gathering will only be considered as a means to achieve the ultimate goal. In other cases, the goal may be the explicit gathering of a maximum amount of information on the state of the system. We refer to these two extreme cases as task-achievement and information gathering problems. The goal may also be a combination of the two in which case trade-offs must be considered.

In a robotic context, task-achievement problems with active sensing include e.g. planning how to operate a machine vision system to detect and locate an object to pick it up, or selecting which sensors to activate to avoid obstacles in an uncertain environment. In the first case, active sensing is a problem of how to control a sensor with multiple operating modes, and in the latter case it is a selection problem between several sensors. Conceptually the cases are identical.

Information-gathering problems with active sensing in robotics include finding exploration strategies in a priori unknown environments [1], and active simultaneous localization and mapping (SLAM) [2], [3]. In robotic exploration, future vantage points from which the state of the environment is sensed are selected applying an active strategy. Instead of merely fusing any obtained data into an estimate of the robot's pose and map, future sensing locations are chosen e.g. to maximize the expected area of terrain explored, or minimize the expected uncertainty in the pose estimate.

According to LaValle and Sharma [4], robotic systems planning their actions in the real world face uncertainty from any subset of four sources: uncertainty in predicting and sensing the robot's own configuration, and uncertainty in predicting and sensing the state of the environment. Therefore, active sensing requires planning over uncertain outcomes in the presence of sensor noise.

In the presence of uncertainty, an agent's knowledge of the system state is represented by a probability density function (pdf) over the state, known as the belief state. Whether the problem is of the task-achievement or information-gathering type, the goal of the agent is to act such that its expected utility is maximized. The utility is measured by an appropriately defined reward function. In the context of dynamic systems, these types of planning problems may be thought of as instances of sequential decision-making problems under uncertainty. For Markovian dynamics and rewards that accumulate on a per-decision basis, the planning problem is a partially observable Markov decision process [5], or POMDP. In a POMDP, probabilistic models quantify the uncertainties in action effects and sensing. POMDPs provide a complete framework by which uncertainties in the effects of the agent's actions and sensing can be handled in a principled way. The solution of a POMDP is an optimal control policy specifying which action to execute in any belief state to maximize expected utility.

Active sensing problems have been solved applying POMDP formulations in both discrete and continuous spaces, see e.g. [6], [7], [8]. POMDPs for choosing focus points within a single image was consider in [9]. The active sensing problems are typically formulated as task-achievement problems, where the reward function only depends on the states and actions and thus the expected reward is linear in the belief state. In information-gathering problems, information-theoretic reward functions such as information entropy or mutual information are nonlinear in the belief state. POMDPs with these and other non-standard types of reward functions have been studied e.g. in [10] and [11].

In this paper, we study active operation of a machine vision system. We assume a robotic agent is traversing along a given trajectory. The agent selects the focus of attention of its machine vision system so as to maximize the amount of information obtained on the environment. The problem is formulated as a POMDP with an information-theoretic reward function and solved approximately by applying a receding horizon control strategy. The optimal focus of attention is solved for each time instant. The solution takes into account the uncertainties related to sensing and the effect of current decisions on decisions at future time instants.

The remainder of the paper is organized as follows. In section II, we formulate the active sensing problem, explain

[1] M. Lauri and R. Ritala are with the Department of Automation Science and Engineering, Tampere University of Technology, P.O. Box 692, FI-33101, Tampere, Finland. Email: `mikko.lauri@tut.fi`, `risto.ritala@tut.fi`

how to recursively estimate belief states and discuss the selection of the utility function for the problem. Section III presents an approximate solution method to the problem. Section IV presents results from a simulation experiment, and section V concludes the paper.

II. PROBLEM STATEMENT

Consider a robot in a planar environment equipped with a vision system whose focus of attention may be varied. The robot's state at time instant t consists of its location on the plane defined by x_t^r and y_t^r, its heading angle θ_t^r, and the orientation angle ϕ_t^r of the vision system. The superscript r emphasises that the variables are related to the robot. The vector $x_{t,v} = [x_t^r, y_t^r, \theta_t^r, \phi_t^r]^{\mathrm{T}}$ fully describes the robot's configuration at time t.

The robot is controlled by applying a translational velocity v_{t-1} and rotational velocity ω_{t-1}. The controls are affected by additive Gaussian noise. The joint probability distribution of $\mu_{t-1} = [v_{t-1}, \omega_{t-1}]^{\mathrm{T}}$ is $N(\hat{\mu}_{t-1}, Q_{t-1})$, where the mean $\hat{\mu}_{t-1} = [\,\hat{v}_{t-1}, \hat{\omega}_{t-1}]^{\mathrm{T}}$ denotes the desired control inputs, and $Q_{t-1} = diag(\sigma_v^2, \sigma_\omega^2)$ denotes the diagonal noise covariance matrix. Furthermore, the vision system may be pointed to any angle independent of the robot's heading. The angle ϕ_t^r can be controlled by a scalar control input $u_t \in [\,u_{min}, u_{max}]$ denoting the desired rotational velocity of the orientation angle. Here u_{min} and u_{max} constrain the minimum and maximum rotational velocity of the orientation angle. We assume that control for the orientation angle is exact, although even in the noisy case similar methods as we adopt here could be applied. The dynamics are defined by a nonlinear model [12]

$$x_{t,v} = f_v(x_{t-1,v}, \mu_{t-1}, u_{t-1}) = x_{t-1,v} + \dots \begin{bmatrix} -\frac{v_{t-1}}{\omega_{t-1}} \sin\theta_{t-1}^r + \frac{v_{t-1}}{\omega_{t-1}} \sin(\theta_{t-1}^r + \omega_{t-1}\Delta t) \\ \frac{v_{t-1}}{\omega_{t-1}} \cos\theta_{t-1}^r - \frac{v_{t-1}}{\omega_{t-1}} \cos(\theta_{t-1}^r + \omega_{t-1}\Delta t) \\ \omega_{t-1}\Delta t \\ (\omega_{t-1} + u_{t-1})\Delta t \end{bmatrix}, \quad (1)$$

where Δt is the time discretization interval. Each time index t covers a real-valued time interval $[t, t+\Delta t]$.

There are M stationary features in the environment at locations $\{x_f^i, y_f^i\}_{i=1}^M$. The locations of these features remain unchanged. Let the vector $p = [x_f^1, y_f^1, \dots, x_f^M, y_f^M]^{\mathrm{T}}$ denote the vector of feature locations, and define $x_t = [x_{t,v}^{\mathrm{T}}, p^{\mathrm{T}}]^{\mathrm{T}}$ as the joint state of the robot and the features. Now the dynamics of the whole system are described by

$$x_t = f(x_{t-1}, \mu_{t-1}, u_{t-1}) = \begin{bmatrix} f_v(x_{t-1,v}, \mu_{t-1}, u_{t-1}) \\ p \end{bmatrix}. \quad (2)$$

Using its vision system, the robot may observe features inside its cone of observation $D(x_t, \alpha, r_{max})$, determined by the vision system's view angle α and the maximum observable range r_{max}, see Fig. 1. We denote by $W = \{1, 2, \dots, M\}$ the set of all features, and by $\tilde{W}(x_t) = \{i \in W \mid (x_f^i, y_f^i) \in D(x_t, \alpha, r_{max})\}$ the subset of features observed at state x_t.

An observation $y_{t,i}$ of the ith feature consists of the measured distance d_i and angle β_i of the feature relative to the robot's state. In practice, such observations could be obtained e.g. from a stereo camera system or a laser range finder. Throughout the paper we will assume that data association is known, i.e. it is known from which feature a particular observation originates from. The observations are affected by additive zero-mean Gaussian noise with independent variances σ_d^2 and σ_β^2, respectively. The nonlinear observation model for feature i may be written as $y_{t,i} = h_i(x_t) + r_t^i$, where

$$h_i(x_t) = \begin{bmatrix} d_i \\ \beta_i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_t^r - x_f^i)^2 + (y_t^r - y_f^i)^2} \\ \arctan \frac{y_t^r - y_f^i}{x_t^r - x_f^i} - \phi_t^r - \theta_t^r \end{bmatrix}, \quad (3)$$

and $r_t^i \sim N(0, R_t^i)$ is independent Gaussian noise with covariance $R_t^i = diag(\sigma_d^2, \sigma_\beta^2)$. The complete observation y_t at time t is now the vector of all individual feature observations, $y_t = [y_{t,1}, \dots, y_{t,M}]^{\mathrm{T}}$. The observation model h for the whole system is then given by

$$y_t = h(x_t) + r_t = \begin{bmatrix} h_1(x_t) + r_t^1 \\ \vdots \\ h_M(x_t) + r_t^M \end{bmatrix}, \quad (4)$$

where $r_t^i \sim N(0, R_t^i)$ are i.i.d. and correspondingly $r_t \sim N(0, R_t)$ where R_t is the block diagonal matrix of all R_t^i.

A. State estimation

To maintain a representation of the robot's knowledge of the state x_t of the system we must track the belief state of the robot, i.e. the conditional pdf of the state given the measurements until time t, denoted $p(x_t \mid y_{1:t})$. In general, the belief state may be tracked with any suitable Bayesian filter, for example the extended Kalman Filter (EKF) [13]. In the EKF, belief states are represented as Gaussian pdfs. Recursive formulas are applied to update belief states when new data become available. Nonlinear models f and h are handled by linearization at the means of the belief states. In the following, we briefly describe the EKF implementation we have applied.

Given the previous belief state $N(m_{t-1}, P_{t-1})$ and the desired actions $\hat{\mu}_{t-1}$, u_{t-1}, the EKF first computes a predictive pdf $p(x_t \mid y_{1:t-1}, \mu_{t-1}, u_{t-1})$. The predictive pdf is Gaussian $N(m_t^+, P_t^+)$ with parameters

$$m_t^+ = f(m_{t-1}, \hat{\mu}_{t-1}, u_{t-1}) \quad (5a)$$

$$P_t^+ = F_x P_{t-1} F_x^{\mathrm{T}} + F_\mu Q_{t-1} F_\mu^{\mathrm{T}}, \quad (5b)$$

where $F_{\{x,\mu\}} \equiv F_{\{x,\mu\}}(m_{t-1}, \hat{\mu}_{t-1}, u_{t-1})$ denote the Jacobians of f with respect to the state x_{t-1} and control μ_{t-1}, respectively, evaluated at $(m_{t-1}, \hat{\mu}_{t-1}, u_{t-1})$.

Once a measurement y_t is obtained, the posterior pdf $p(x_t \mid y_{1:t}, \mu_{t-1}, u_{t-1})$ is calculated in the update step via

$$z_t = y_t - h(m_t^+) \quad (6a)$$

$$S_t = H_x P_t H_x^{\mathrm{T}} + R_t \quad (6b)$$

$$K_t = P_t H_x^{\mathrm{T}} S_t^{-1} \quad (6c)$$

$$m_t = m_t^+ + K_t z_t \quad (6d)$$

$$P_t = P_t^+ - K_t S_t K_t^{\mathrm{T}}, \tag{6e}$$

where $(\cdot)^{-1}$ denotes matrix inverse and $H_x \equiv H_x(m_t)$ is the Jacobian of h with respect to the state x_t. The matrix R_t is the block diagonal matrix with diagonal elements R_t^i. Measurements $y_{t,i}$ for which $i \notin \tilde{W}(x_t)$ should have no contribution to the posterior. This is achieved by setting for each $i \notin \tilde{W}(x_t)$ the corresponding rows of H_x to zero.

B. The active sensing problem

Suppose that the robot is following a trajectory defined as a sequence of T expected control inputs $\{\hat{\mu}_{t-1}\}_{t=1}^{T}$. The active sensing problem is defined as follows.

Optimal control of a vision system along a trajectory. *Given the dynamic system defined above, a trajectory $\{\hat{\mu}_{t-1}\}_{t=1}^{T}$ of expected controls and an initial belief state $N(m_0, P_0)$, find a sensing policy of the form $\Pi : (m_t, P_t) \rightarrow u_t$ for controlling the orientation angle of the vision system such that a given value function $V(\Pi)$ is maximized.*

Fig. 1 illustrates the active sensing problem by showing the trajectory from the expected controls and the initial belief state on the robot's position and the location of the features. During the trajectory, the robot must independently decide how to operate its machine vision system by choosing rotational velocities u_{t-1} for the vision system.

C. Selection of the value function

We shall restrict ourselves to consider value functions V consisting of real-valued additive per-step rewards r_t, i.e. $V = \sum_{i=1}^{T} r_t$. This type of value functions are the only type which admit solutions via dynamic programming [14], a key method for solving sequential decision-making problems.

A task-achievement type per-step reward function is of the form $r_t(x_t, u_t)$. It is concerned purely with achieving a favourable state in the system. Such reward functions are reasonable if there is a well-defined target state, such as in a navigation problem. However, in this case the controls u_t have no effect on the robot's pose or the feature location.

Reward functions quantifying information-gathering must depend on the agent's knowledge. Thus they are dependent on the belief state, assuming the form $r_t(x_t, m_t, P_t, u_t)$. This type of reward functions includes e.g. the mean squared error between the state estimate and the true state or the trace of the belief state covariance. However, as noted e.g. in [3], these quantities may not be meaningful in case the state consists of quantities with different units, such as locations, velocities and angles. Information-theoretic quantities, such as mutual information (MI) provide a unit-insensitive quantification of the informativeness of control actions. It has been shown [15] that policies maximizing mutual information can provide good universal performance in problems where there are many information gathering goals, e.g. maximizing target detection probability together with target tracking and identification.

We have chosen to apply the MI of the state and measurement as the reward function. This is equivalent to maximizing the expected Kullback-Leibler (KL) divergence between the posterior distribution $p(x_t \mid y_{1:t}, \mu_{t-1}, u_{t-1})$ and the predictive distribution $p(x_t \mid y_{1:t-1}, \mu_{t-1}, u_{t-1})$ [16], where the expectation is taken w.r.t. the measurement Y_t. Clearly, the posterior depends on the subset of features observed, over which we must likewise take the expectation.

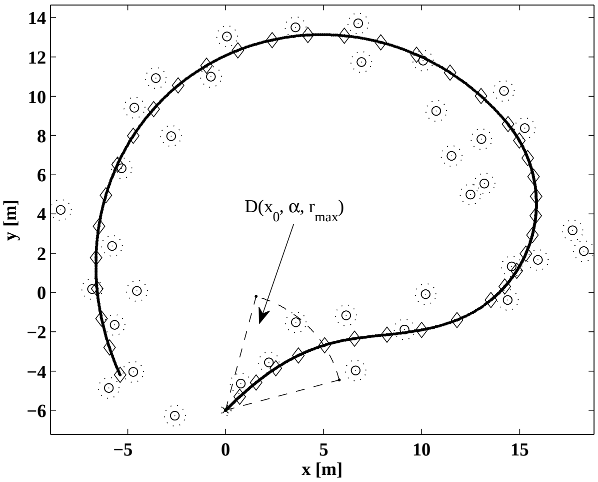

Fig. 1. The active sensing problem along a trajectory. The solid line denotes the expected trajectory of the robot. Diamond markers denote the locations at which control actions must be decided. The circle markers denote the features, and the associated dashed circles represent the 50% confidence intervals in the initial belief state. Features within the cone of observation $D(x_0, \alpha, r_{max})$ limited by dashed lines may be observed.

D. Approximating mutual information

Consider the case with a given predictive distribution $N(m_t^+, P_t^+)$ and a fixed subset $\tilde{W}$ of observed features. The parameters of the posterior distribution may now be calculated via equations (6a - 6e). Notice that only the posterior mean depends on the actual data y_t, while the posterior covariance P_t depends only on the subset $\tilde{W}$ of observed features. We thus denote the posterior distribution as $N(m_t(y_t), P_t^{\tilde{W}})$. Applying the formula for the KL divergence of Gaussian distributions, the expected KL divergence w.r.t. Y_t with a fixed subset $\tilde{W}$ of observed features is

$$D_{KL}^{\tilde{W}}(u_{t-1}) \approx \frac{1}{2}\left(tr\left(\left[P_t^+\right]^{-1} P_t^{\tilde{W}} \right) - l - \log \frac{|P_T^{\tilde{W}}|}{|P_t^+|} \right),$$

where $tr(\cdot)$ denotes the trace operator, l is a constant equal to the dimensionality of m_t^+ and $|\cdot|$ denotes matrix determinant. The approximation is due to the quadratic term of the KL divergence of two Gaussians tending to zero as applying (6d) and (6a) $\mathbb{E}_{Y_t}[m_t(y_t) - m_t^+] = K_t \mathbb{E}_{Y_t}[y_t - h(m_t^+)] \approx 0$.

The mutual information of the predictive state estimate and the observation is now equal to the expectation

$$\mathcal{I}(X_t, Y_t \mid u_{t-1}) \approx \sum_{\tilde{W} \in \mathcal{P}(W)} D_{KL}^{\tilde{W}}(u_{t-1}) P(\tilde{W}). \tag{7}$$

The summation is over the power set $\mathcal{P}(W)$, i.e. the set of all subsets of W. The term $P(\tilde{W})$ is the probability that the subset $\tilde{W}$ of features is observed. The size of the power set for M features is 2^M, an infeasible amount to enumerate to calculate (7) exactly. We instead apply a Monte Carlo approximation based on sampling possible subsets of observed features as follows. A set of samples $\{x_t^{(i)}\}_{i=1}^N \sim p(x_t \mid y_{1:t-1}, \mu_{t-1}, u_{t-1})$ is drawn from the state predictive distribution $N(m_t^+, P_t^+)$. For each sample we find the deterministic subset $\tilde{W}^{(i)} = \tilde{W}(x_t^{(i)})$ of features inside the cone of observation $D(x_t^{(i)}, \alpha, r_{max})$. We obtain

$$\mathcal{I}(X_t, Y_t \mid u_{t-1}) \approx \frac{1}{N} \sum_{i=1}^{N} D_{KL}^{\tilde{W}^{(i)}}(u_{t-1}) \tag{8}$$

that can be applied to practically approximate the MI.

III. SOLVING THE ACTIVE SENSING PROBLEM

The active sensing problem $\max_\Pi V(\Pi)$ can be stated as a maximization problem of finding an optimal policy,

$$\max_{\Pi} \sum_{t=0}^{T-1} \gamma^t \mathbb{E}\left[\mathcal{I}(X_{t+1}, Y_{t+1} \mid \Pi(m_t, P_t)\right], \tag{9}$$

where the action is given through the control policy $u_t = \Pi(m_t, P_t)$ and a discount factor $0 \leq \gamma \leq 1$ determines the relative importance of immediate and future rewards. The belief states evolve conditional on the observations according to the state estimation process outlined above. In general, solving this type of optimization problem requires reasoning over uncertain outcomes in both sensing and action effects. The problem is an instance of a finite-horizon POMDP with an information theoretic reward function. The belief-dependent reward function is in general not linear with respect to the belief state. Standard POMDP algorithms (e.g. [5], [17]) rely on the piecewise linearity and convexity of the value function that follows from linearity of the reward function and are not applicable to the active sensing problem.

We approximate (9) by an online receding horizon control scheme. At time instant k, an optimal open-loop action sequence $u^*_{k:k+H-1}$ of H actions is obtained by solving a finite horizon open-loop equivalent of (9):

$$\operatorname*{argmax}_{u_{k:k+H-1}} \sum_{t=k}^{k+H-1} \gamma^{(t-k)} \mathbb{E}\left[\mathcal{I}(X_{t+1}, Y_{t+1} \mid u_t)\right], \tag{10}$$

where H is the so-called optimization horizon. As we see from the equation, the feedback property of a policy of the form $\Pi : (m_t, P_t) \to u_t$ is lost as we have replaced it with a simple action sequence. Feedback is reintroduced by applying the receding horizon principle: (10) is first solved for the current belief state (m_k, P_k). The first action u_k^* of an optimal sequence is executed and an observation y_{k+1} is perceived. The belief state is revised to (m_{k+1}, P_{k+1}) by the EKF and (10) is solved again at time instant $k+1$, and so on. This open-loop feedback control (OLFC) method has been shown to achieve a value at least as great as the pure open-loop policy of executing $u^*_{k:k+H-1}$ [18].

The selection of the optimization horizon $H \geq 1$ determines how far into the future we consider possible rewards. As the robot is executing a finite control trajectory of length T, at time instant k we limit $H \leq T - k$.

The open loop control problem (10) may be thought of as a search problem over action sequences of length H. Any suitable search algorithm may be applied to find an optimal action sequence. We consider search algorithms that apply Monte Carlo methods for finding optimal action sequences. These search methods may be combined with our approximation of the per-step reward function. In our experiments, we have applied two solution methods. For both methods, the action space is discretized to a finite set of K actions $U = \{u^{(k)}\}_{k=1}^K$.

The first method is a straightforward application of the approximation (8) to evaluate the utility of action sequences of length $H = 1$. Therefore it is a myopic, or greedy, strategy considering only immediate information gain. At time step k we solve

$$u_k^* = \operatorname*{argmax}_{u \in U} \mathbb{E}\left[\mathcal{I}(X_{k+1}, Y_{k+1} \mid u)\right] \tag{11}$$

to find an optimal action. A similar myopic optimization approach has been used e.g. in [3] and [2] for active SLAM.

The second method is a Monte Carlo tree search (MCTS) [19] algorithm over the space of action sequences of length H. MCTS methods iteratively apply Monte Carlo simulations of the dynamic system to construct a search tree over action sequences. The results of the simulations are applied to evaluate the values of the action sequences explored.

In MCTS, each node in the search tree corresponds to a sequence of actions. Initially the tree contains only the root node corresponding to an empty sequence of actions $u_{k:t}$. Additionally, a value estimate $\hat{V}$ computed as the empirical mean and a visitation count N_v are stored in each node. Each simulation starts from the root node. At each simulation step the per-step reward is recorded, in this case obtained from (8) with $N = 1$. An action to execute in the simulation is selected according to the UCT algorithm [20]. At a node corresponding to the action sequence $u_{k:t-1}$, UCT selects an action according to $u_t = \operatorname{argmax}_{u_t} \hat{V}(u_{k:t}) + c \cdot \sqrt{\log(N_v(u_{k:t-1}))/N_v(u_{k:t})}$, where $\hat{V}(u_{k:t})$ is the current estimate of the value of selecting action u_t after $u_{k:t-1}$, $N_v(u_{k:t-1})$ is the number of times the trajectory $u_{k:t-1}$ of actions has been encountered in the simulations so far, and $N_v(u_{k:t})$ is the number of times action u_t has been selected in the simulations after $u_{k:t-1}$. An exploration constant $c > 0$ is set to encourage trying actions that have so far been rarely tried. This selection procedure is followed until a leaf node of the search tree is reached. At a leaf node, child nodes are added to the tree for each possible successor action. The simulation then switches to the rollout stage, where actions are selected uniformly at random until the end of the optimization horizon is reached. Finally, the visitation counts N_v of the nodes traversed during the simulation episode are updated and the mean value estimates $\hat{V}$ are revised according to the recorded per-step rewards.

A fixed number of simulations may be run, or the simulations may be stopped e.g. when a maximum planning time is exceeded. MCTS with UCT as the action selection method has been shown to converge to the optimal solution

[20]. In the case of the active sensing problem (10), the optimal solution found is an optimal open loop sequence $u^*_{k:k+H-1}$ of actions. For a more thorough treatment, we refer the reader to the survey [19], the paper [20] on UCT and [21] on applying MCTS to POMDP problems.

IV. SIMULATION EXPERIMENT

A simulation environment as shown in Fig. 1 was defined with $M = 37$ features to be observed while the robot traverses a trajectory determined by a sequence of $T = 40$ control signals $\{\hat{\mu}_{t-1}\}_{t=1}^{T}$. The belief state was tracked by the EKF. Time was discretized to intervals of $\Delta t = 0.05$ seconds. Each control signal was applied for $20 \cdot \Delta t = 1$ second, thus the total time of the trajectory was 40 seconds.

The machine vision system has a field of view of $\alpha = 60$ degrees and maximum range of $r_{max} = 6$ meters. The vision system outputs observations at 1 Hz rate. The focus of attention is selected by controlling the angular velocity of the vision system orientation angle ϕ^r_t. The angular velocity limits were $u_{min} = -15°/s$ and $u_{max} = 15°/s$, discretized uniformly into 12 distinct values. Means of the robot and feature poses in the initial belief state were as shown in Fig. 1. The initial covariance matrix was diagonal with feature location variances equal to 0.2 and variance of robot pose equal to 0.05 meters or radians squared, respectively. The noise parameters for the problem were $\sigma^2_v = 10^{-2}$ m^2/s^2, $\sigma^2_\omega = 10^{-3}$ rad^2/s^2, $\sigma^2_d = 10^{-3}$ m^2, $\sigma^2_\beta = 2 \cdot 10^{-4}$ rad^2.

We applied the two methods presented above to solve the active sensing problem, and compared their performance with the case where the controls u_t were selected randomly. For each method or parameter set, the experiment was repeated 20 times starting from the same initial belief state and true state. For the myopic strategy (11), $N = 10$, 50 or 100 samples were applied to estimate MI. For MCTS, the exploration parameter $c = 100$ was constant to scale the exploration bonus to a range comparable with typical MI per step. The discount factor was $\gamma = 0.95$. Note that for $\gamma = 0$ only immediate rewards are considered and (10) will produce the same action as (11), and $\gamma = 1$ is only appropriate for finite horizon problems to keep the sum of rewards finite. In general, selection of an appropriate discount factor depends on the problem characteristics; e.g. the uncertainty of future rewards or economic arguments determining how desirable it is to obtain rewards sooner rather than later.

The effect of the optimization method on the differential entropy of the belief state was examined. Lower entropy indicates lower uncertainty. One-sided t-tests were performed to quantify the overall performance of the methods. The entries in Table I indicate whether MCTS with horizon H and number of simulations N_s achieved a lower final entropy (Y) or not (N) with a 5% significance level than the myopic method with $N = 10$, and shows the corresponding p-value from the one-sided t-test. The table shows that more simulations tend to increase the margin by which MCTS is better than the myopic method. However, for longer horizons it is more difficult to adequately sample the space of action sequences even with a large number of simulations. The suboptimality of the open-loop approximation compared to the full problem (9) is more noticeable with $H = 4$ and 5. Comparing against the myopic method with $N = 50$ or 100, results are similar although statistically significant differences start to appear only after $N_s \geq 5000$.

TABLE I
SUMMARY OF THE t-TEST RESULTS.

H	$N_s = 500$	$N_s = 1000$	$N_s = 5000$	$N_s = 10000$
1	N (0.167)	Y (0.016)	Y (0.027)	Y (0.008)
2	N (0.063)	Y (0.004)	Y (0.007)	Y (0.016)
3	Y (0.013)	Y (0.001)	Y (0.028)	Y (0.003)
4	Y (0.003)	N (0.212)	Y (0.021)	N (0.544)
5	N (0.055)	N (0.148)	N (0.156)	N (0.504)

Typical results are shown in Fig. 2. Applying random actions performs significantly worse than the optimization methods. The large confidence intervals for random actions indicate high variation in the results. The myopic strategy and MCTS achieve a lower final differential entropy. In this case, there is no statistically significant difference in the final entropies of the two methods.

The difference between the myopic strategy and MCTS is illustrated by comparing their performance at decisions 10 through 20. These steps correspond to the part of the trajectory shown on the lower right hand side in Fig. 1. Furthermore, Fig. 3 shows the means and confidence intervals for the machine vision system orientation angle ϕ^r_t as function of the number of decisions. Initially, both methods focus attention on the features along the trajectory on steps 1 through 7. The myopic strategy then decides to observe features on the right hand side of the trajectory around x-coordinate greater than 15, and orients its camera towards them. Due to its longer optimization horizon, MCTS anticipates the possibility to observe these features on later time steps and instead prefers to first obtain more information on features around x-coordinate 10 and y-coordinate -1. In Fig. 3, the interval from 9 to 19 decisions is where the orientation angles of the two methods are significantly different from each other. Comparing with Fig. 2, the myopic method first achieves lower belief state entropy at decisions 8 through 11, while afterwards the benefits of the longer-term strategy of the MCTS method become apparent, with it obtaining lower mean differential entropy on decision 12.

V. CONCLUSION

We studied an active sensing problem of selecting the focus of a machine vision system of a robot while the robot traverses a partially known trajectory, formulating the problem as a POMDP. An open-loop approximation was derived and applied together with a receding horizon control strategy. A greedy optimization method and a method based on Monte Carlo tree search were applied to solve the open-loop problems in simulations.

Based on the experimental results, we can identify two advantages of a longer planning horizon. First, in cases where focus of attention cannot be changed rapidly, acting greedily

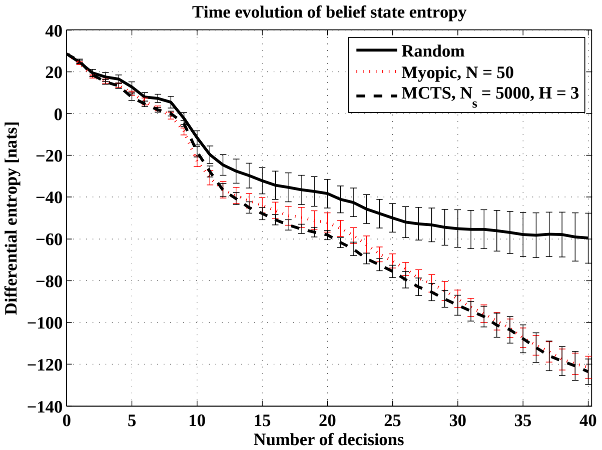

Fig. 2. The differential entropy of the belief state as function of the number of control decisions taken. The lines indicate mean differential entropy over 20 experiments, and the horizontal bars depict 95% confidence intervals.

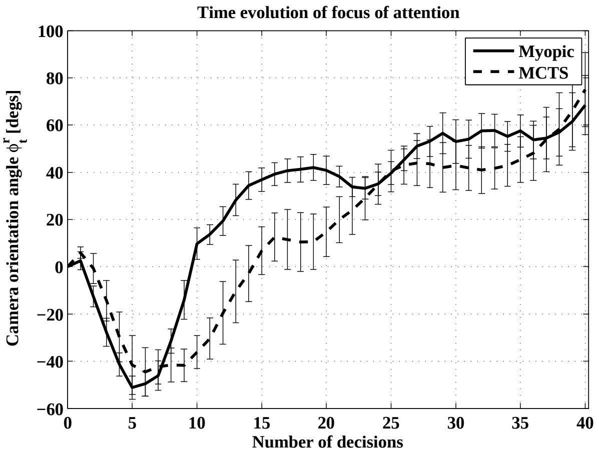

Fig. 3. Time evolution of the orientation angle of the machine vision system during the experiment. The lines show the means and their 95% confidence intervals are indicated by the vertical bars.

may result in poorer performance than longer term planning. Acting greedily considers only immediate information gain and neglects focus change over time. Such a situation arises e.g. if the allowed rate of change in the field of view per time step is small. The second advantage may arise when the features are not stationary. Some features may only be observable for a short period of time when they are within sensor range. The focus of attention must be planned with a longer horizon, as new opportunities to sense a particular feature may not appear again.

The main disadvantage of longer optimization horizons is the computational complexity that increases exponentially with the horizon. We tackled the issue by solving open-loop approximations of the problem and reintroducing feedback by applying the receding horizon principle. However, this approximation is only guaranteed to perform equally well or better than the optimal pure open-loop policy.

We considered a pure information-gathering problem, and did not study the case where sensing or switching the focus of attention has a cost. However, such a problem could also be solved with the methods we applied here. In future work, we are planning to apply similar methods to a wider range of exploration problems in the robotic domain.

References

[1] F. Amigoni, “Experimental evaluation of some exploration strategies for mobile robots,” in *Proc. IEEE International Conference on Robotics and Automation 2008*, Pasadena, California, May 2008, pp. 2818–2823.

[2] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte, “Information based adaptive robotic exploration,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, Oct. 2002, pp. 540–545.

[3] R. Sim and N. Roy, “Global A-Optimal Robot Exploration in SLAM,” in *Proc. IEEE International Conference on Robotics and Automation 2005*, Barcelona, Spain, Apr. 2005, pp. 661–666.

[4] S. M. LaValle and R. Sharma, “On Motion Planning in Changing, Partially Predictable Environments,” *The International Journal of Robotics Research*, vol. 16, no. 6, pp. 775–805, Dec. 1997.

[5] L. Kaelbling, M. Littman, and A. Cassandra, “Planning and acting in partially observable stochastic domains,” *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[6] E. K. P. Chong, C. M. Kreucher, and A. O. Hero, “Partially Observable Markov Decision Process Approximations for Adaptive Sensing,” *Discrete Event Dynamic Systems*, vol. 19, no. 3, pp. 377–422, May 2009.

[7] Y. Li, L. Krakow, E. Chong, and K. Groom, “Approximate stochastic dynamic programming for sensor scheduling to track multiple targets,” *Digital Signal Processing*, vol. 19, no. 6, pp. 978–989, Dec. 2009.

[8] M. Lauri and R. Ritala, “Planning for multiple measurement channels in a continuous-state POMDP,” *Annals of Mathematics and Artificial Intelligence*, vol. 67, no. 3-4, pp. 283–317, May 2013.

[9] N. Butko and J. Movellan, “Optimal scanning for faster object detection,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, June 2009, pp. 2751–2758.

[10] V. Krishnamurthy and D. V. Djonin, “Structured Threshold Policies for Dynamic Sensor SchedulingA Partially Observed Markov Decision Process Approach,” *IEEE Transactions on Signal Processing*, vol. 55, no. 10, pp. 4938–4957, Oct. 2007.

[11] M. Araya, O. Buffet, V. Thomas, and F. Charpillet, “A POMDP Extension with Belief-dependent Rewards,” in *Advances in Neural Information Processing Systems 23*, Vancouver, Canada, Dec. 2010, pp. 64–72.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: The MIT Press, 2005.

[13] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge, UK: Cambridge University Press, 2013.

[14] R. P. Loui, “Optimal paths in graphs with stochastic or multidimensional weights,” *Communications of the ACM*, vol. 26, no. 9, pp. 670–676, Sept. 1983.

[15] C. Kreucher, A. Hero, and K. Kastella, “A Comparison of Task Driven and Information Driven Sensor Management for Target Tracking,” in *Proc. 44th IEEE Conference on Decision and Control*, Seville, Spain, Dec. 2005, pp. 4004–4009.

[16] T. Cover and J. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ: John Wiley & Sons, 2006.

[17] J. Pineau, G. Gordon, and S. Thrun, “Anytime point-based approximations for large POMDPs,” *Journal of Artificial Intelligence Research*, vol. 27, no. 1, pp. 335–380, 2006.

[18] D. P. Bertsekas, “Dynamic Programming and Suboptimal Control: A Survey from ADP to MPC,” *European Journal of Control*, vol. 11, no. 4-5, pp. 310–334, Jan. 2005.

[19] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, “A Survey of Monte Carlo Tree Search Methods,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, Mar. 2012.

[20] L. Kocsis and C. Szepesvári, “Bandit based monte-carlo planning,” in *Proc. European Conference on Machine Learning ECML 2006*, Berlin, Germany, Sept. 2006, pp. 282–293.

[21] D. Silver and J. Veness, “Monte-Carlo Planning in Large POMDPs,” in *Advances in Neural Information Processing Systems 23*, Vancouver, Canada, Dec. 2010, pp. 2164–2172.