

SIMULTANEOUS LOCALISATION AND PLANNING

Will Penny

Wellcome Trust Centre for Neuroimaging,
University College, London WC1N 3BG, UK.

ABSTRACT

This paper proposes an algorithm for the solution of the Simultaneous Localisation and Planning problem. The solution is based on statistical inference in a Hidden Markov Model which proceeds in separate phases of localisation and planning. Each requires access to the same contextual model operationalised via the ‘prior dynamics’, and is implemented using forward (localisation) and forward and backward (planning) message passing. I propose that this formalism provides a useful computational-level description of aspects of Hippocampal function.

Index Terms— Navigation, Simultaneous Localisation and Mapping, Planning as Inference, KL Control, Hidden Markov Models, Hippocampus, Pattern Replay.

1. INTRODUCTION

An established paradigm in robotics is that of Simultaneous Localisation and Mapping (SLAM) [1] in which an agent explores and maps novel environments. This paper addresses a related problem. Assuming that an environment has already been mapped, what algorithm would allow an agent to localise itself and compute optimal routes to arbitrary goals? We refer to this as the Simultaneous Localisation and Planning (SLAP) problem.

We address this problem using state-space models and probabilistic inference, which allows us to specify both task goals and an agent’s location in probabilistic terms. This builds on a ‘planning as inference’ perspective in which sequential decision making problems that have previously been the domain of Reinforcement Learning (RL), have been recast as problems of statistical inference [2, 3].

More specifically, our approach closely follows the work of Kappen [4] and Todorov [5] who have shown that, if actions are penalised using a Kullback-Leibler (KL) criterion then the associated Bellman equation is linearly solvable. Moreover, this solution corresponds to backward inference in a Hidden Markov Model (HMM) [6]. This formalism,

known as KL control, has been developed for Markov Decision Processes (MDPs) in which the state of the agent is directly observed.

In this paper, we work directly with HMMs and consider two types of observations; goals and sensory inputs. The use of sensory inputs, allows the state of the agent to be inferred, and provides a natural way in which uncertainty from localisation can be incorporated into planning. We view this HMM approach as providing a computational-level description of aspects of Hippocampal function.

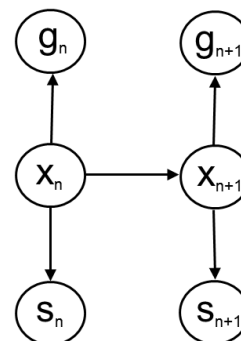


Fig. 1. The agent’s generative model. This corresponds to an HMM with two sets of observations; goals, g_n , and sensory inputs, s_n .

2. METHODS

We consider a dynamical system evolving over time points $n = 1..N$ with discrete latent states x_n , goals g_n and sensory states s_n . The overall generative model is shown in Figure 1 and is fully specified via the definition of three probability distributions (i) the state transition density $p(x_{n+1}|x_n)$, (ii) the sensory observation density $p(s_n|x_n)$ and (iii) the goal observation density $p(g_n|x_n)$.

Inference is implemented in two separate phases (i) planning and (ii) localisation. In the planning phase, goals are provided and the posterior distribution over latent states is computed. At this time sensory states are either not provided

Thanks to the Wellcome Trust for funding.

or their influence on planning is eliminated, for example, by reducing sensory precision.

In the localisation phase, sensory observations are provided and the posterior distribution over latent states is computed. In this phase goals are either not provided or their influence on localisation is eliminated, for example, by reducing goal precision.

In what follows each of the $k = 1..K$ discrete latent states is associated with a location in a 2D environment, l_k , and $N(x; \mu, \Lambda)$ denotes a multivariate Gaussian distribution over x with mean μ and precision Λ .

2.1. Planning from a Known Location

2.1.1. Prior Dynamics

In the planning phase the agent is given information about task goals. Prior to observing these goals the hidden states evolve according to Markovian dynamics

$$p(x_{n+1} = i | x_n = j) = A_{ij} \quad (1)$$

where A is a $K \times K$ state transition matrix. Although high-dimensional this matrix is sparse, reflecting the spatial structure of an environment and allowed transitions within it. We refer to the above equation as describing the ‘Prior Dynamics’.

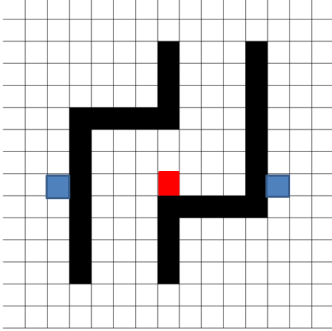


Fig. 2. The environment contains $15 \times 15 = 225$ discrete states, with black squares denoting forbidden locations. At each time step agents may move to cardinal neighbours or remain in the same position but cannot transit across edges of the domain (eg top to bottom). The coloured squares indicate the location of olfactory sources - see Localisation section.

2.1.2. Probabilistic Goals

The probability of observing goal g_n given state x_n is given by the density $p(g_n | x_n)$. This formulation has many interesting properties. First, goals are inherently probabilistic; one specifies the likelihood of obtaining a goal at a given location. Second, the goal signal g_n can be multivariate, allowing an agent to have multiple goals. Third, the current relevance

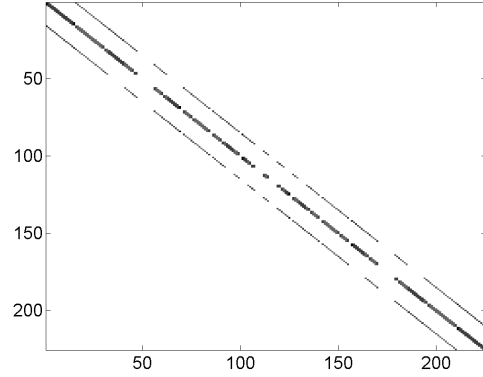


Fig. 3. The prior dynamics are embodied in a state transition matrix A of dimension 225×225 having a highly sparse structure reflecting the allowed transitions in an environment.

of each goal is readily modified (by changing, for example, a precision parameter). Fourth, the observed goal signal can be used to specify set points and thereby implement a system for instantiating homeostatic or autoregulatory control. For example if we have

$$p(g_n | x_n = k) = N(g_n; \mu_k, \text{diag}(\lambda_n)) \quad (2)$$

then $\mu_k(i)$ is the numerical value of goal type i that can be realised at state k , $g_n(i)$ can be set to the desired fixed point for that goal type, and $\lambda_n(i)$ signifies its current relevance.

2.1.3. Posterior Dynamics

We now consider an agent being in receipt of a goal signal g . In order to specify to the agent that this goal is to be reached within a ‘time horizon’ of N steps we set the sequence of observation variables $g_n = g$ for $n = 1..N$. We denote this sequence as $G_N = \{g_1, g_2, \dots, g_N\}$.

The dynamics of the agent after receiving the goal signal, or the ‘Posterior Dynamics’, are defined as

$$\begin{aligned} Q_{ij} &\equiv p(x_{n+1} = i | x_n = j, G_N) \\ &= \frac{p(x_{n+1} = i | x_n = j) p(x_n = j | G_N)}{\sum_{i'=1}^K p(x_{n+1} = i' | x_n = j) p(x_n = j | G_N)} \end{aligned} \quad (3)$$

An agent following the posterior dynamics implements goal-directed navigation, whilst one following the prior dynamics merely obeys the physics of a given environment.

2.1.4. State Posterior

The posterior dynamics constitute a reweighting of the prior dynamics by the density $p(x_n = j | G_N)$. This density can be computed using standard inference algorithms such as the alpha-beta recursions [7]. This requires a forward sweep to

compute

$$\alpha(x_n) = p(g_n|x_n) \sum_{x_{n-1}} p(x_n|x_{n-1})\alpha(x_{n-1}) \quad (4)$$

with $\alpha(x_1 = k) = p(x_1 = k)p(g_1|x_1 = k)$, and a backward sweep to compute

$$\beta(x_n) = \sum_{x_{n+1}} p(g_{n+1}|x_{n+1})p(x_{n+1}|x_n)\beta(x_{n+1}) \quad (5)$$

with $\beta(x_N = k) = 1$. We then have

$$p(x_n|G_N) = \frac{\alpha(x_n)\beta(x_n)}{\sum_k \alpha(x_n = k)\beta(x_n = k)} \quad (6)$$

To avoid numerical underflow [7] we scale the forward and backward messages by $\sum_k \alpha(x_n = k)$. The forward ‘alpha’ recursions embody the prior distribution and provide a normalisation factor for the backward ‘beta’ recursions.

2.1.5. Trajectories from a Known Location

A known state at time n is equivalent to a probability distribution $p(x_n)$ with unit mass at $x_n = k$ and zero elsewhere. A probabilistic planning trajectory can then be found by integrating the posterior dynamics from this initial distribution. The probability mass at time point $n + 1$ is

$$p(x_{n+1} = i) = \sum_{k=1}^K q(x_{n+1} = i|x_n = k)p(x_n = k) \quad (7)$$

The state density at subsequent time points can be computed as

$$p(x_{n+m} = i) = \sum_{k=1}^K q(x_{n+m} = i|x_{n+m-1} = k)p(x_{n+m-1}) \quad (8)$$

or in matrix form as

$$p(x_{n+m+1}) = Q^m p(x_{n+1}) \quad (9)$$

This is the discrete form of the Chapman-Kolmogorov equation. Iteration of this equation produces ‘goal-directed flows’ and individual paths to goal are produced by sampling from these flows.

2.2. Localisation

Given sensory input $S_{n-1} = \{s_1, s_2, \dots, s_{n-1}\}$, the distribution over hidden states prior to a subsequent observation is

$$p(x_n = k|S_{n-1}) = \sum_i p(x_n = k|x_{n-1} = i)p(x_{n-1} = i|S_{n-1}) \quad (10)$$

After observing s_n , the posterior distribution is

$$p(x_n = k|S_n) = \frac{p(s_n|x_n = k)p(x_n = k|S_{n-1})}{p(s_n|S_{n-1})} \quad (11)$$

with predictive density

$$p(s_n|S_{n-1}) = \sum_{j=1}^K p(s_n|x_n = j)p(x_n = j|S_{n-1}) \quad (12)$$

A neurobiological perspective on the above equations attributes $a_k = p(x_n = k|S_n)$ to activity of the k th place cell in the CA3/CA1 hippocampal region. Here, l_k is the centre of the k th place field, and the location encoded by the population of cells is given by the mean of the predictive density

$$l_{pop,n} = \sum_{k=1}^K l_k a_k \quad (13)$$

An alternative decoding scheme would be to simply take the location of the Maximum A Posteriori (MAP) most active cell.

2.3. Planning from an Uncertain Location

Planning under location uncertainty can be implemented by marginalising the active dynamics over that uncertainty

$$p(x_{n+1} = i) = \sum_{k=1}^K q(x_{n+1} = i|x_n = k)p(x_n = k|S_n) \quad (14)$$

This is identical to equation 7 but instead of having a delta function over hidden states on the RHS (corresponding to the known location) we have the posterior state density from localisation. The probability distributions at subsequent time points are then given by the iterations in equation 9.

2.4. KL Control

The state posterior can alternately be expressed as

$$p(x_n = j|G_N) = \frac{p(G_N|x_n = j)p(x_n = j)}{\sum_{j'=1}^K p(G_N|x_n = j')p(x_n = j')} \quad (15)$$

Given a uniform prior $p(x_n = j)$ the equation for the Posterior Dynamics reduces to

$$Q_{ij} = \frac{p(x_{n+1} = i|x_n = j)p(G_N|x_n = j)}{\sum_{i'=1}^K p(x_{n+1} = i'|x_n = j)p(G_N|x_n = j)} \quad (16)$$

Equation 16 corresponds to the ‘Active Dynamics’ of KL control, and $\log p(G_N|x_n = j)$ to the ‘Optimal Value’ function [6]. The ‘Passive Dynamics’ of KL control correspond to our ‘Prior Dynamics’. The scaling of the beta recursions in the HMM implementation (see above) is analogous to the normalisation used in the power method for computing the Optimal Value function [6].

3. RESULTS

This section refers to videos showing goal directed flows that are available from

http://www.fil.ion.ucl.ac.uk/~wpenny/cip14_movies/.

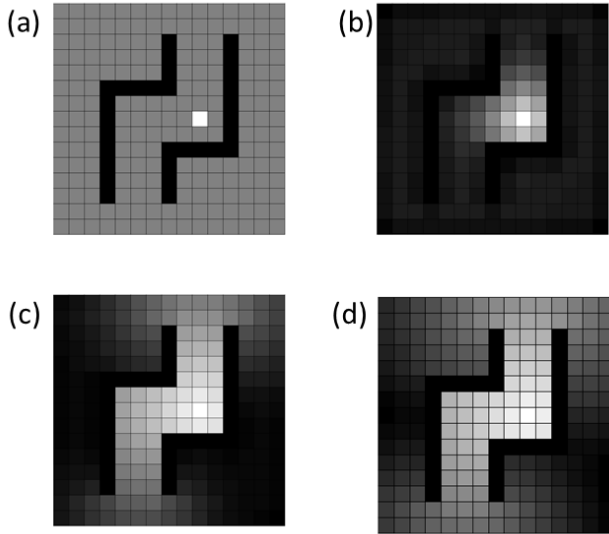


Fig. 4. Time to Goal The figure shows the state posterior, $p(x_1|G_N)$, for four different times to goal (a) $N = 1$, (b) $N = 16$, (c) $N = 64$ and (d) $N = 1024$. The goal location is $[10, 8]$. Under a uniform prior, $p(x_1)$, these plots correspond to the exponent of the Optimal Value function of KL control.

3.1. Planning from a Known Location

Figure 2 shows an example 2D environment and Figure 3 the state transition matrix, A , corresponding to the prior dynamics associated with it. Here A_{ij} has been set to $1/N_j$ if a transition is allowed from j to i , with N_j being the number of allowable transitions from j (fewer next to boundaries and corners). This includes transitions from a state to itself. Transitions are not allowed to or from wall or edge locations (we set $0/0 = 0$, as per usual).

Figure 4 (a) now superimposes a goal onto this environment. The goal observation density $p(g_n|x_n)$ was set to a univariate normal $N(g_n; \mu_k, \lambda_g)$ with parameters and goal signals set so that the probability density is 1 at the goal and zero elsewhere. We then computed the state posterior for three different values for the time to goal N . This computation is implemented using the alpha-beta recursions in equations 4 to 6. For $N = 1$ the state posterior has a single peak at the goal. For small values of N the posterior has similarly small values at locations remote from the goal. As we increase N , the ‘spatial gradient’ of the posterior at these remote locations increases. This gradient informs the posterior dynamics via equation 3, allowing a path to be found from remote sites to the goal. In what follows the posterior dynamics, Q , were computed from equation 3 using $N = 1024$ with the goal at $[10, 8]$. Note that for large N , we have $p(x_1|G_N) \approx p(x_2|G_N) \approx p(x_3|G_N)$ etc., so we can simply use $p(x_1|G_N)$ in equation 3.

The movie `known_15_1.avi` shows the state density evolving according to the posterior dynamics. The initial state

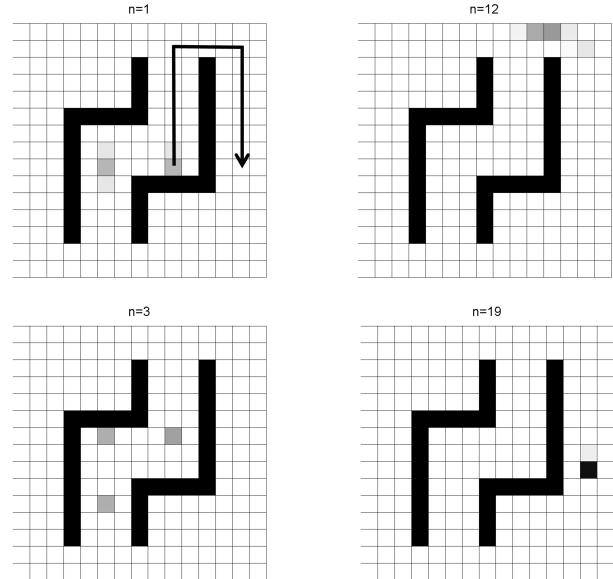


Fig. 5. Localisation with Ambiguous Olfactory Cues The figures plot the posterior state density, $p(x_n|S_n)$, computed using equation 11 at four different time points.

density is a delta function with unit probability mass at location $[15, 1]$ and zero elsewhere. The state density at subsequent time points has been computed using equation 9. We now keep the goal at the same location, hence do not change the posterior dynamics Q , but move the initial position to $[2, 8]$. The evolution of the state density is shown in the movie `known_2_8.avi`.

Finally, we keep the goal at the same location but update the prior dynamics, A , to account for a small change to the environment. This is a hole in a wall appearing at location $[6, 8]$. The posterior dynamics were recomputed based on this new prior and the movie `hole.avi` shows the goal-directed flow from position $[2, 8]$.

The above results show that changes in goal location are accommodated by recomputing the posterior dynamics, Q . Small changes in the environment are readily accommodated by small changes in the prior dynamics (and updating Q). These nonstationarities are less gracefully accommodated in RL which requires complete relearning of state-action mappings.

3.2. Localisation with Ambiguous Olfactory Cues

In this simulation we provide an agent with sensory input from olfactory cues. Olfaction is known to be important in mammalian navigation [8] and here we consider a case where the cues are ambiguous. We provide three olfactory cues at locations $m_1 = [8, 7]$, $m_2 = [3, 7]$ and $m_3 = [13, 7]$, shown in Figure 2, where the flanking cues are of the same type (ie

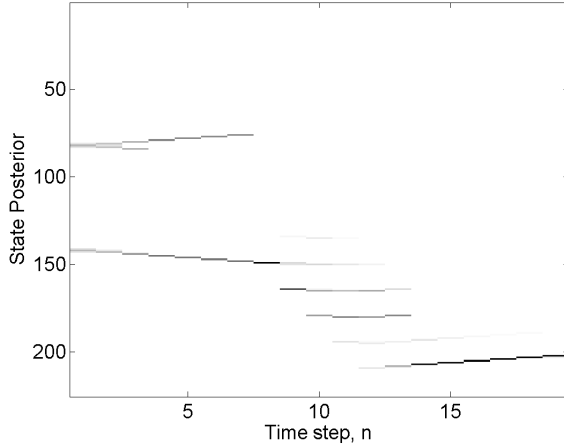


Fig. 6. Localisation with Ambiguous Olfactory Cues. The posterior state density, $p(x_n|S_n)$, as a function of time step, n , shows initial multimodality that is later resolved.

same smell). The agent receives two olfactory inputs

$$\begin{aligned} o_n(1) &= f_o(l_n, m_1) + z_n(1) \\ o_n(2) &= f_o(l_n, m_2) + f_o(l_n, m_3) + z_n(2) \\ f_o(l_n, m_i) &= A \exp\left(-\frac{\|l_n - m_i\|}{2\sigma_o^2}\right) \end{aligned} \quad (17)$$

where $A = 100$, $\sigma_o = 2$, l_n is the location of the agent at time step n , and $z_n(i)$ is zero-mean Gaussian noise with standard deviation 3. The sensory observation density is set to

$$\begin{aligned} p(s_n|x_n = k) &= \mathcal{N}(o_n; m_k, \Lambda_o) \\ m_k &= [f_o(l_k, m_1), f_o(l_k, m_2) + f_o(l_k, m_3)] \end{aligned} \quad (18)$$

We simulate an agent taking a path from a central to a lateral location as shown in Figure 5 (top left), and compute the state posterior using equation 11. Localisation is initially ambiguous, because the same olfactory input can be produced from two locations. This is reflected in the initially bimodal state posterior in Figures 5 and 6. However, as time proceeds all but one of the modes are eliminated as they are not supported by the prior dynamics; because probability mass cannot enter boundaries it is reassigned to other modes.

3.3. Localisation using Olfaction and Path Integration

In this simulation we additionally provide the agent with spatial location cues from a path integration system. Path integration, the estimation of location based on speed, direction of heading, and elapsed time, is an important complementary signal in the mammalian navigation system [9]. Whilst useful, its accuracy tends to decrease over time, which we simulate here by setting the path integral input as $r_n^n = l_n + e_n$ such that the variance of the additive noise is a linear function of time step. Specifically, the initial standard deviation

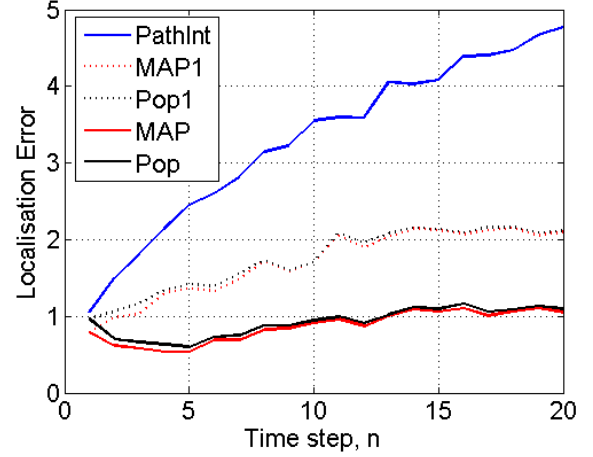


Fig. 7. Localisation via Olfaction and Path Integration. The figure plots localisation error using various strategies: Path Integration only (PathInt), Maximum Posterior estimation based on current observation only (MAP1) and sequence (MAP) of observations, Population Decoding based on current observation only (Pop1) and sequence (Pop) of observations.

was set to 1, and increased by 1 each time step. The agent’s sensory observation model is augmented as

$$p(s_n|x_n = k) = \mathcal{N}(o_n; m_k, \Lambda_o)\mathcal{N}(r_n; l_k, \Lambda_r) \quad (19)$$

We then assessed the localisation accuracy of various strategies using 100 simulated trials. In each trial the initial location (and direction - see below) of the agent was drawn randomly from a uniform distribution over allowed states. Subsequent states were then sampled using a strategy designed to produced directed paths. Specifically, a proposed next state was generated such that, with probability 0.9, the direction of the move was the same as the last. Proposals were accepted if the move was allowed, otherwise a new proposal was made. For each time step, on each trial, the agent’s location was estimated using a variety of strategies. Figure 7 plots the root mean square localisation error averaged over trials.

3.4. Planning from an Uncertain Location

Planning given uncertainty in localisation, can be implemented by marginalising the posterior dynamics over that uncertainty according to equation 14. We illustrate this with the following example. The initial state distributions are given by the density in Figure 5 with $n = 12$. That is, the agent thinks it is located in a small region in the north east corner of the maze. The goal is set to the location $[2, 5]$. We then apply equation 14 to get the state distribution at the next time step, under the posterior dynamics. The resulting goal-directed flow is then given by the iterations in equation 9, and shown in the movie `uncertain.avi`.

4. DISCUSSION

This paper proposes an algorithm for the solution of the Simultaneous Localisation and Planning problem based on inference in an HMM. The HMM approach provides a useful computational-level description of aspects of Hippocampal function. There are several appealing features.

First, the use of a discrete rather than a continuous latent space allows multimodal posteriors to be supported using simple, exact inference, as shown in the results. This is to be contrasted with the approximate inference procedures (particle filtering etc.) required for nonlinear continuous latent space models [9].

Second, discretisation of an otherwise continuous state space is generally unworkable for generic control problems because of the curse of dimensionality. However, spatial navigation problems are inherently two dimensional so discretisation is tenable. Trajectories computed by algorithms such as discrete KL control are, in our experience, more robust than those computed by eg. local linear Gaussian approximations in an equivalent continuous space. Thus, from a computational perspective, if the dimension of continuous state space is sufficiently small, it is preferably discretised.

Third, this work builds on earlier proposals that the hippocampus itself is suited to solving shortest path problems [10]. It has been proposed that CA3 encodes path distances in its connections and may implement a heuristic planning approach such as Dijkstra's algorithm. A similar proposal could be based on the Prior Dynamics where the connection from unit j to i encodes A_{ij} . Our approach shares the advantages of earlier proposals [10] in that the inevitable local changes in an environment can be reflected in a small number of modified connections, but has the additional advantage of being probabilistic, allowing uncertainty in localisation to be incorporated into planning.

Fourth, previous work has proposed that the hippocampus replays goal-directed state sequences so that the striatum can learn the appropriate state-action mappings [11]. However, it may be the case that these 'replays' are not memories of successful episodes but sample trajectories from the Posterior Dynamics.

Fifth, previous work has proposed that a succinct computational description of the hippocampus is that it provides contextual information. The computation of such a contextual signal has previously been operationalised using model selection in an HMM [12]. This is an interesting area for future work eg. using model selection [9] to decide which environmental model is currently appropriate.

Finally, the two phases of inference we have proposed may map onto two distinct modes of hippocampal function, differentiated by the degree of theta activity. Mammalian localisation is accompanied by a high theta state [13], whereas planning related replay activity [14] is accompanied by high frequency ripples in a low theta state. Future work will exam-

ine how, for example, acetylcholine might increase sensory precision and so switch HMM inference from planning to localisation.

5. ACKNOWLEDGEMENTS

I thank Neil Burgess, Christoph Mathys and Karl Friston for useful comments on this work.

6. REFERENCES

- [1] H Durrant-Whyte and T Bailey, "Simultaneous localisation and mapping (SLAM): Part i the essential algorithms," *Robotics and Automation Magazine*, vol. 13, pp. 99–110, 2006.
- [2] H. Attias, "Planning by probabilistic inference," in *Proceedings of the 9th International Conference on Artificial Intelligence and Statistics*, C Bishop and B Frey, Eds., 2003.
- [3] M. Toussaint, "Robot trajectory optimisation using approximate inference," in *Proc. of the 26th International Conference on Machine Learning (ICML 2009)*, A Danyluk, L Bottou, and M Littman, Eds., 2009.
- [4] Kappen H.J., "A linear theory for control of non-linear stochastic systems," *PRL*, vol. 95, pp. 200201, 2005.
- [5] E. Todorov, "Linearly-solvable Markov decision problems," in *Advances in Neural Information Processing Systems*, 2006, vol. 19, pp. 1369–1376.
- [6] E. Todorov, "Efficient computation of optimal actions," *Proceedings National Academy of Sciences*, vol. 106, no. 28, pp. 11478–11483, 2009.
- [7] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.
- [8] L Jacobs, "From chemotaxis to the cognitive map: the function of olfaction.," *PNAS*, vol. 109 Suppl 1, pp. 10693–10700, Jun 2012.
- [9] W. Penny, P. Zeidman, and N. Burgess, "Forward and Backward Inference in Spatial Cognition," *PLoS CB*, vol. 9, no. 12, pp. e1003383, 2013.
- [10] R Muller and M Stead, "Hippocampal place cells connected by Hebbian synapses can solve spatial problems," *Hippocampus*, vol. 6, pp. 709–719, 1997.
- [11] A Johnson and A Redish, "Hippocampal replay contributes to within session learning in a temporal difference reinforcement learning model.," *Neural Netw.*, vol. 18, no. 9, pp. 1163–1171, Nov 2005.
- [12] M Fuhs and D Touretzky, "Context Learning in the Rodent Hippocampus," *Neural Computation*, vol. 19, pp. 3173–3215, 2007.
- [13] D Foster and M. Wilson, "Hippocampal theta sequences.," *Hippocampus*, vol. 17, no. 11, pp. 1093–1099, 2007.
- [14] B Pfeiffer and D Foster, "Hippocampal place-cell sequences depict future paths to remembered goals," *Nature*, vol. 497, pp. 74–79, 2013.