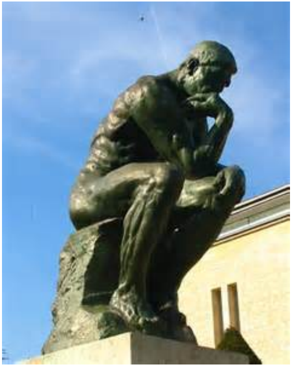
A Dynamic Bayesian Model of Spatial Cognition

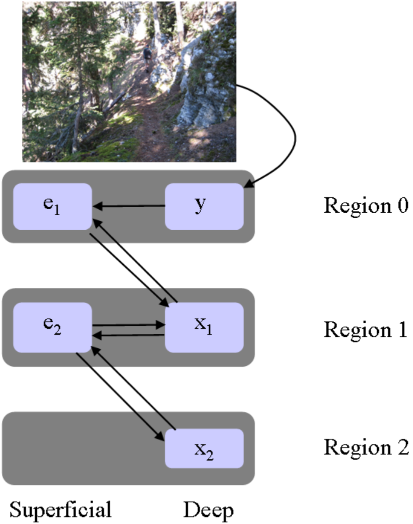
Will Penny



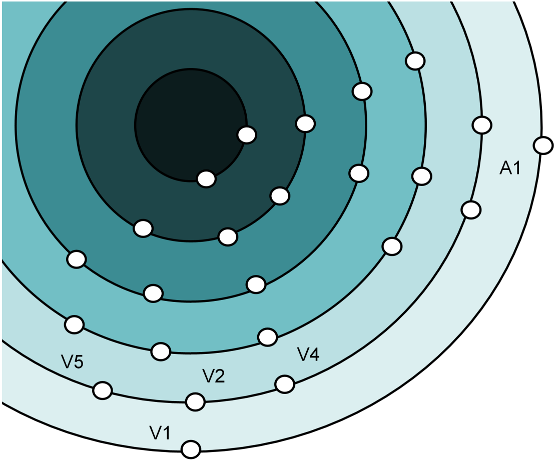
The Anatomy of Choice Workshop, UCL, 2014.

Bayesian Inference in Hierarchical Models

Mumford, Biol Cyb, 1992; **Rao and Ballard**, Nat Neuro, 1999; **Friston**, Neural Networks, 2003

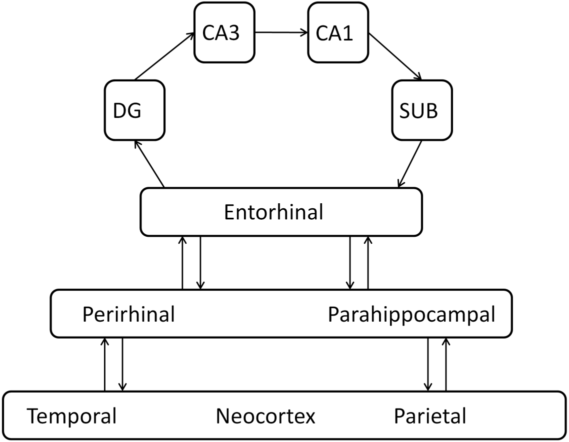


The Centre



Mesulam, Brain (1998)
Friston, Neural Networks (2003)

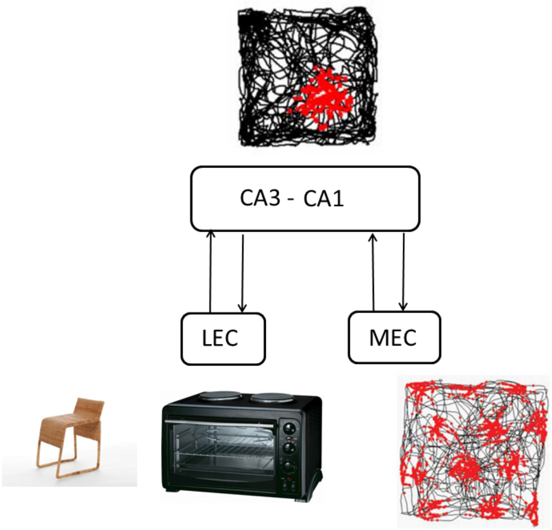
Hippocampal-Neocortical Loop



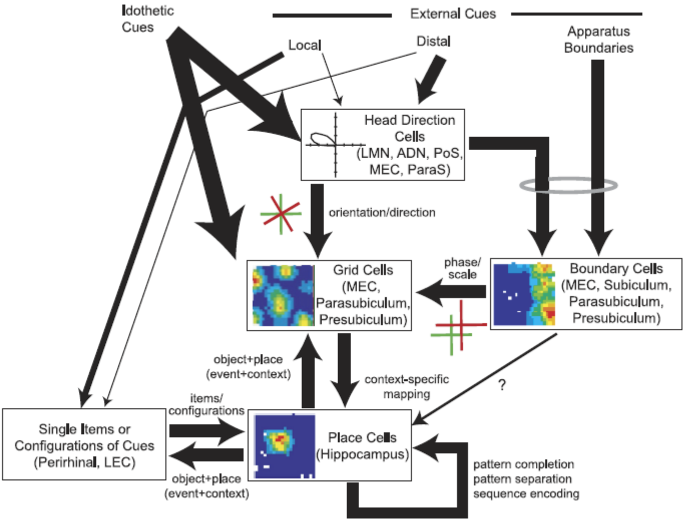
Rolls, Behavioural Brain Research, 2010

Spatial Localisation

Association of LEC object representations with MEC spatial representations.

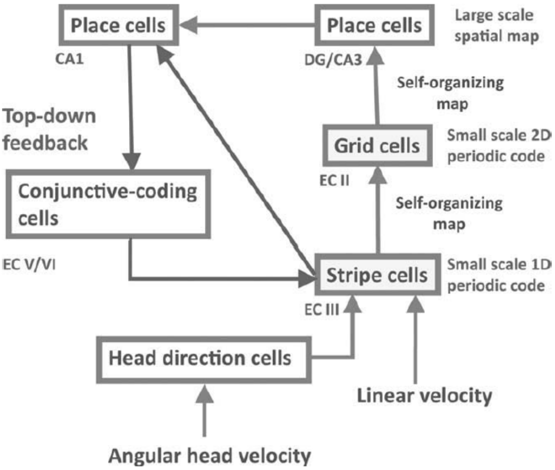


Spatial Localisation



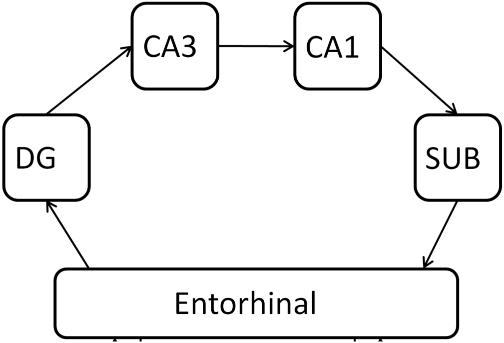
Knierim and Hamilton, Phys Review, 2011

Spatial Localisation



Mhatre and Grossberg, Hippocampus, 2010
Srinavasan and Fiete, Nature Neuroscience, 2011

Bayesian Inference in Dynamical Models



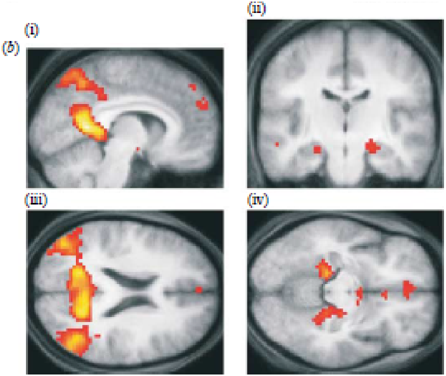
Bousquet et al., Is The Hippocampus a Kalman Filter ? Proc. Biocomputing, 1998.

Fox and Prescott, Hippocampus as unitary coherent particle filter, IJCNN, 2009.

Fuhs and Touretzky, Context Learning in the Rodent Hippocampus, Neural Computation, 2007.

Memory and Imagination

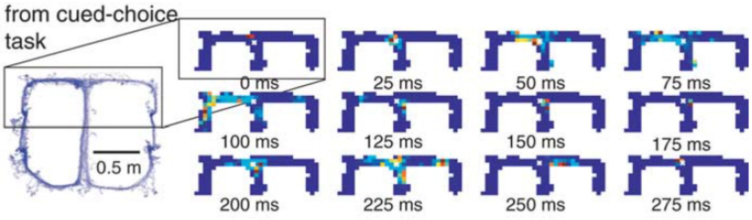
Hassabis and Maguire (Proc. Royal Soc B, 2009) review the set of areas commonly activated in fMRI during recall of recent real memories, recall of imagined experiences and construction of imaginary scenes.



The network includes bilateral hippocampus, parahippocampus, retrospenial and parietal cortex, middle temporal regions and medial prefrontal cortex.

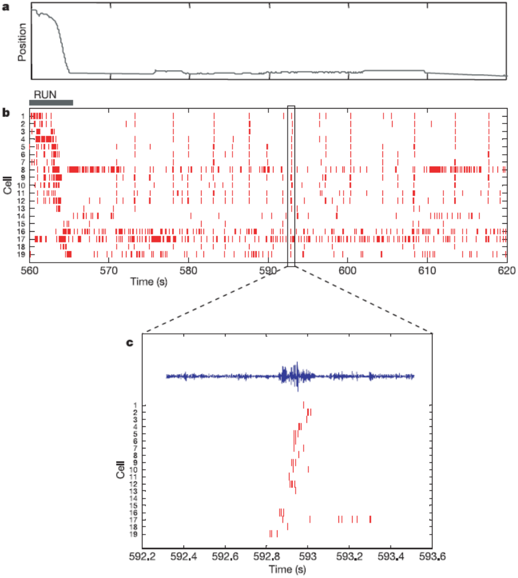
Planning Activity - Forward Sweeps

Small white circle denotes location of rat.



Johnson and Redish, Neural Ensembles in CA3 Transiently Encode Paths Forward of the Animal at a Decision Point, Journal of Neuroscience, 2007.

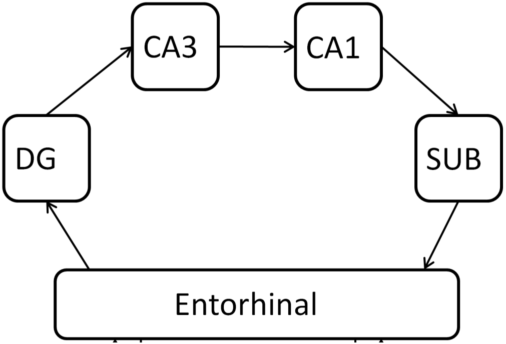
Reverse Sweeps



Foster and Wilson, Nature, 2006.

Forward and Backward Inference

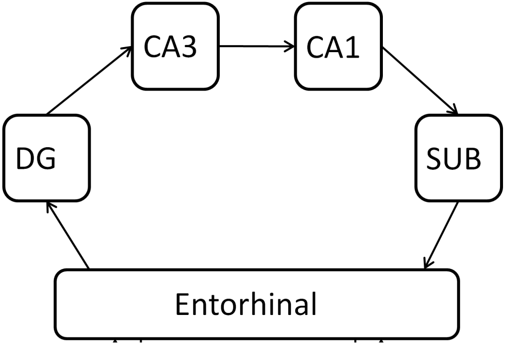
Localisation and Planning are implemented by common set of low-level algorithms based on Forward and Backward Inference over Time.



W. Penny, P. Zeidman, N. Burgess (2013) Forward and Backward Inference in Spatial Cognition, PLoS CB, 9(12) e1003383.

Discrete Latent Space Model

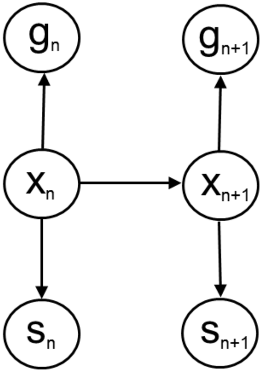
W. Penny, Simultaneous Localisation and Planning. 4th International Workshop on Cognitive Information Processing, Copenhagen, 2014.



W. Penny and K. Stephan. A Dynamic Bayesian Model of Homeostatic Control. 2014 International Conference on Adaptive and Intelligent Systems, UK, 2014.

Generative Model

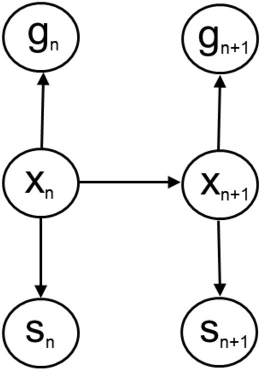
The agent's generative model corresponds to a Hidden Markov Model (HMM) with latent states x_n and two sets of observations: goals g_n and sensory inputs s_n.



Sensory inputs s_n from Entorhinal Cortex:
LEC object identity (from multimodal ventral stream)
MEC spatial location from grid cell code

Generative Model

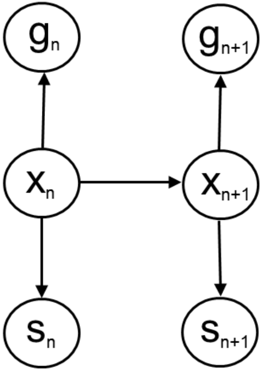
The agent's generative model corresponds to a Hidden Markov Model (HMM) with latent states x_n and two sets of observations: goals g_n and sensory inputs s_n.



Latent states x_n correspond to CA3 activity.

Generative Model

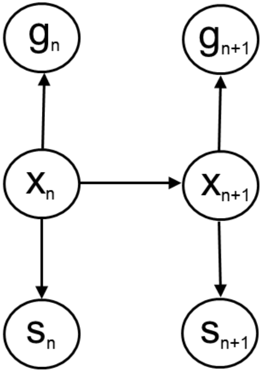
The agent's generative model corresponds to a Hidden Markov Model (HMM) with latent states x_n and two sets of observations: goals g_n and sensory inputs s_n.



Both planning and localisation need access to the same underlying cognitive map instantiated in CA3, $x_n \rightarrow x_{n+1}$.

Generative Model

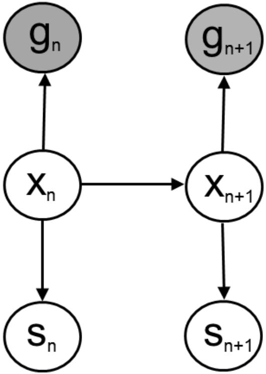
The agent's generative model corresponds to a Hidden Markov Model (HMM) with latent states x_n and two sets of observations: goals g_n and sensory inputs s_n.



Inference is implemented in two separate phases (i) planning and (ii) localisation.

Planning

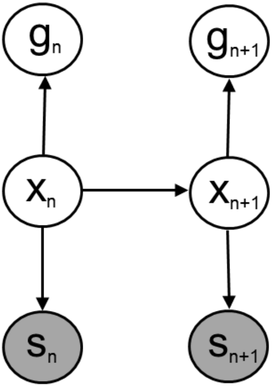
Sensory input is switched off during planning.



Inferences about latent states are made based on observed goal variables.

Localisation

Goal input is switched off during localisation.

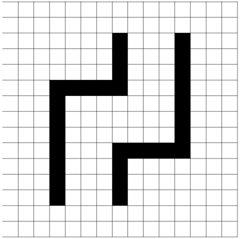


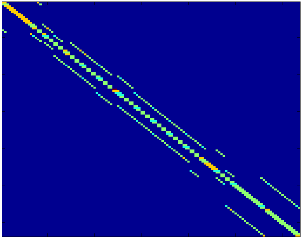
Inferences about latent states are made based on observed sensory variables.

Prior Dynamics

Dynamical Model with state x_n at time index n being one of i=1..K states. Markov state transitions

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



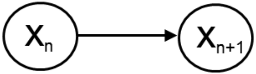


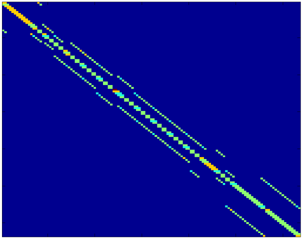
Sparse structure reflects allowed transitions in an environment, the 'prior dynamics'.

Prior Dynamics

Dynamical Model with state x_n at time index n being *one of* i=1..K discrete states. Markov state transitions

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

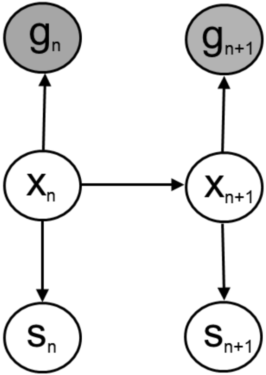




Sparse structure reflects allowed transitions in an environment, the 'prior dynamics'.

Planning

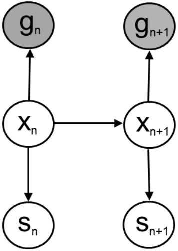
Sensory input is switched off during planning.

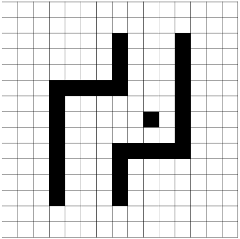


Goals are probabilistic.

Binary Goals

For binary goals, the probability of reaching a goal at location k is $p(g_n = 1 | x_n = k) = r_k$.





Target Distribution

In order to specify to the agent that the 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 $G_N = \{g_1, g_2, ..., g_N\}$.

The prior distribution over hidden states $p(x)$, before receiving goals, is then updated to a posterior distribution, $p(x|G_N)$.

We also refer to this posterior as the **Target Distribution**.

Target Distribution

The Target Distribution can be computed by combining (i) a forward inference step - the alpha recursions, with (ii) a backward inference step - the beta recursions.

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

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

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

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

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

Posterior Dynamics

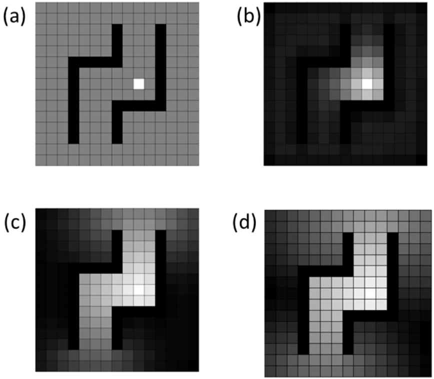
The posterior dynamics are the prior dynamics weighted by the target distribution and renormalised.

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

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

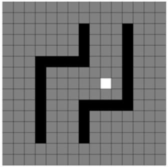
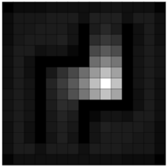
Time to Goal

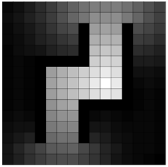
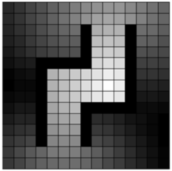
The target distribution, $p(x|G_N)$, for four different times to goal (a) $N = 1$, (b) $N = 16$, (c) $N = 64$ and (d) $N = 1024$. The goal location is $[10, 8]$.



KL Control

Under a uniform prior, $p(x)$, the target distribution corresponds to the exponent of the Optimal Value function of KL control (**Kappen**, PRL, 2005; **Todorov**, NIPS, 2006).

(a)  (b) 

(c)  (d) 

Interim Summary

In the terminology of Markov Chains, an agent following the prior dynamics samples from the prior distribution.

1. Prior Dynamics, $P = p(x_{n+1}|x_n)$. Before goal is provided
2. Prior Distribution, $p(x)$. Equilibrium distribution before goals specified.
3. Posterior Dynamics, $Q = p(x_{n+1}|x_n, G_N)$. After goal is provided
4. Target Distribution, $p(x|G_N)$. Equilibrium distribution after goals specified.

P, is the **Transition Kernel** that leads to the Prior Distribution. Q, is the Transition Kernel that leads to the Target Distribution.

Flows

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 following the posterior dynamics from this initial distribution

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

The state density at subsequent time points can be computed as

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

Iteration of this equation produces 'goal-directed flows' and individual paths to goal are produced by sampling from these flows.

Flows

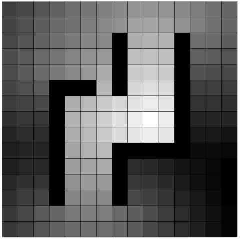
Goal directed flow from $[15, 1]$.
Show movie `known_15_1.avi`

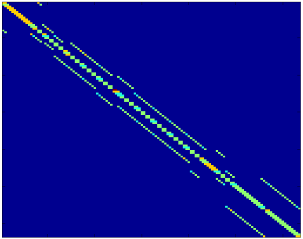
Flows

Goal directed flow from $[2, 8]$.
Show movie `known_2_8.avi`

Nonstationarity

- Small change to environment
- Reflected in small change to prior dynamics
- Recompute target distribution

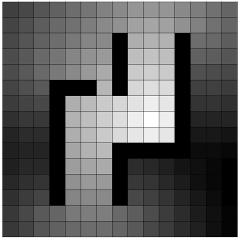


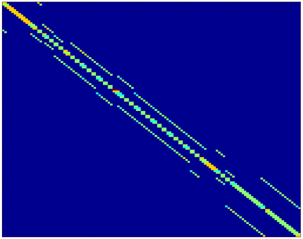


Show movie `hole.avi`

Deliberative versus Habitual Decisions

- Habitual Decisions, state to action mappings, can develop through eg. Reinforcement Learning (RL)
- But small changes to environment are not gracefully dealt with.
- On the other hand, deliberative decisions are slower to make.

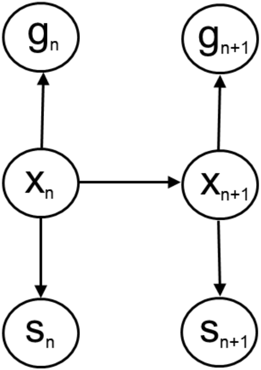




Daw N, Niv Y, Dayan P. Nat Neuroscience, 2005

Multivariate Goals

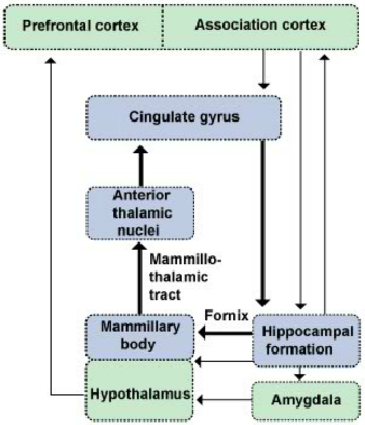
The agent's generative model corresponds to a Hidden Markov Model (HMM) with latent states x_n and two sets of observations: goals g_n and sensory inputs s_n.



Multivariate goals g_n represented in (Augmented) Papez Circuit.

Multivariate Goals

Multivariate goals g_n represented in (Augmented) Papez Circuit.

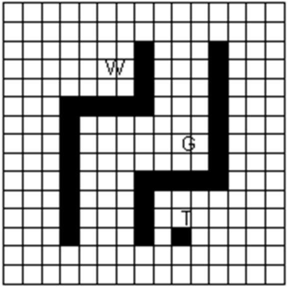


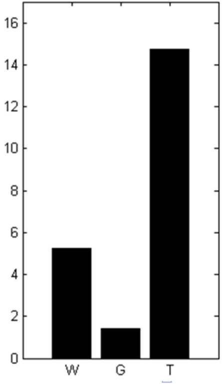
Multivariate Goals

The goal signal can be multivariate. For example if g_n is the homeostatic set point, C encodes the allowed (co-)variance around that set point

$$p(g_n | x_n = k, a) = \mathrm{N}(g_n; a + a_k, C)$$

Here a is the agent's autonomic state, and a_k is the change in autonomic state accrued from visiting state k



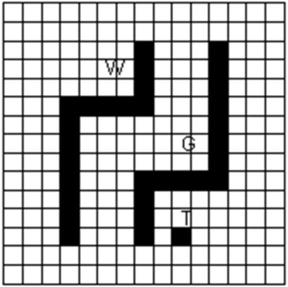


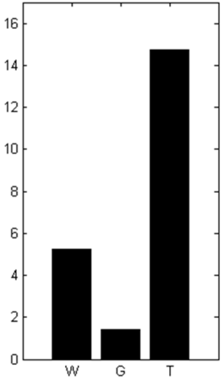
Multivariate Goals

Here we consider autonomic dynamics that evolve as

$$a_n = Ba_{n-1} + a_k$$

such that autonomic states decay with time constants B, but increase by amount a_k upon visiting state k.





Homeostatic Control

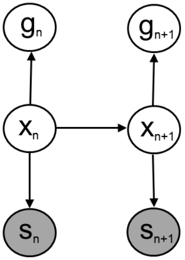
Show `auto.avi`

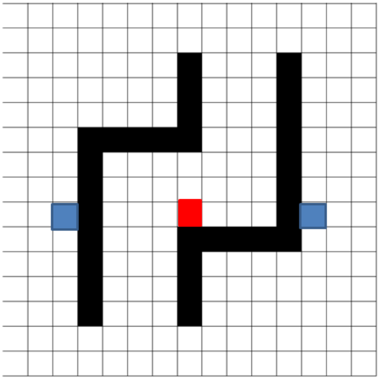
Here the posterior dynamics are recomputed every 32 steps.

Forward and backward hippocampal replay in rats is observed after reaching a goal.

Localisation

Goal input is switched off during localisation. Inferences about latent states are made based on observed sensory variables.





Three olfactory cues at locations $m_1 = [8, 7]$, $m_2 = [3, 7]$ and $m_3 = [13, 7]$. The flanking cues are of the same type (ie same smell).

Olfaction

The agent receives two olfactory inputs

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

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

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

Localisation

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

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

After observing s_n, the posterior distribution is

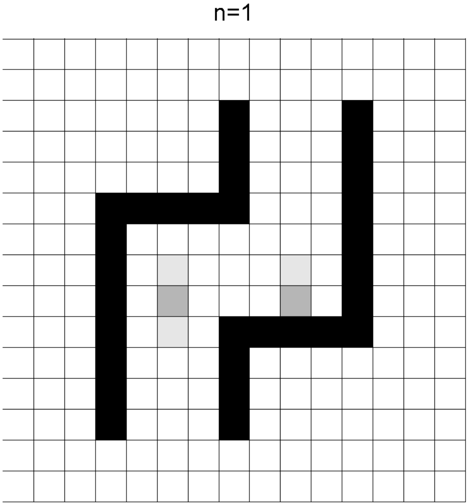
$$p(x_n = k|S_n) = \frac{p(s_n|x_n = k)p(x_n = k|S_{n-1})}{p(s_n|S_{n-1})}$$

with predictive density

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

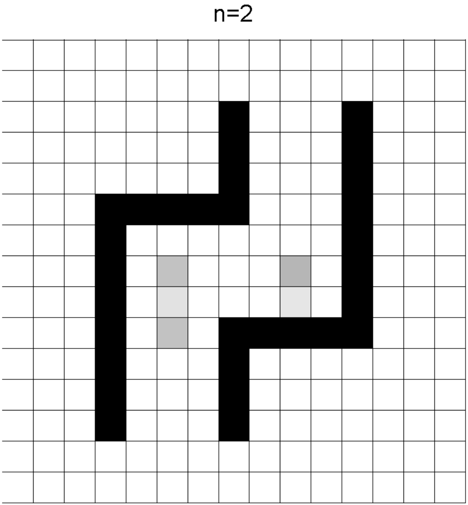
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



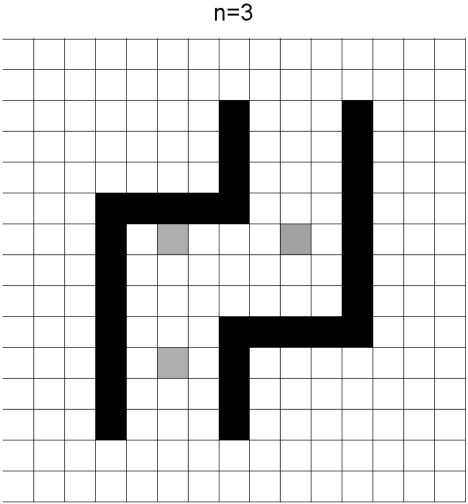
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



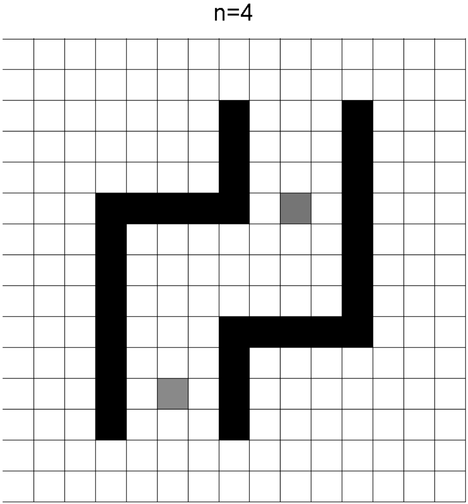
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



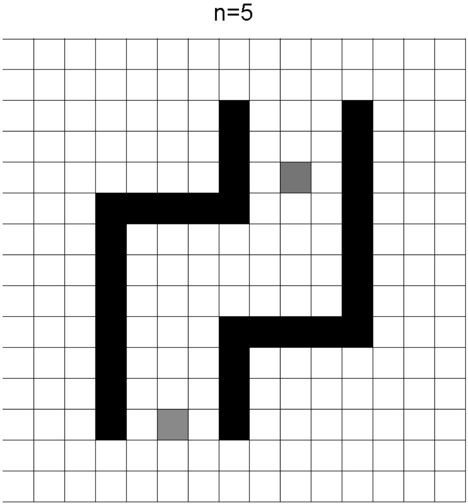
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



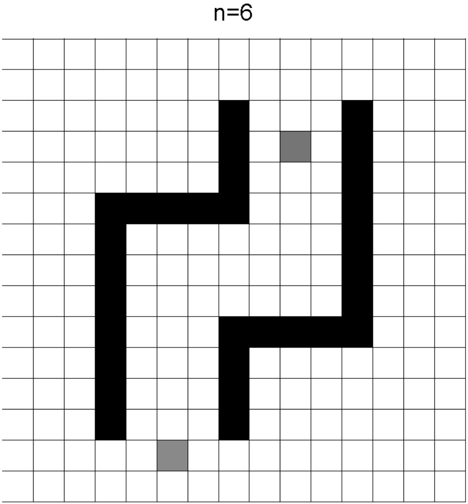
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



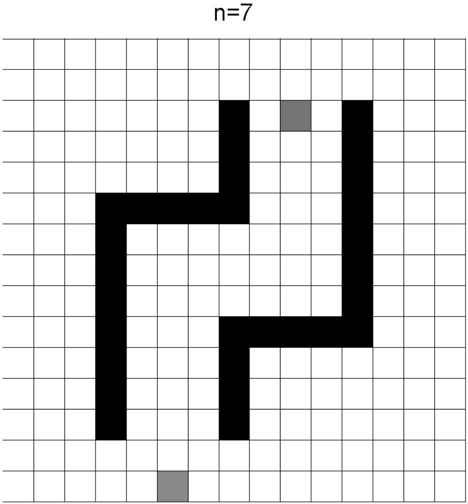
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



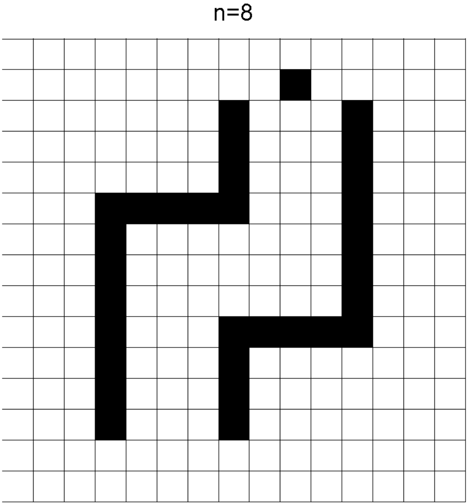
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



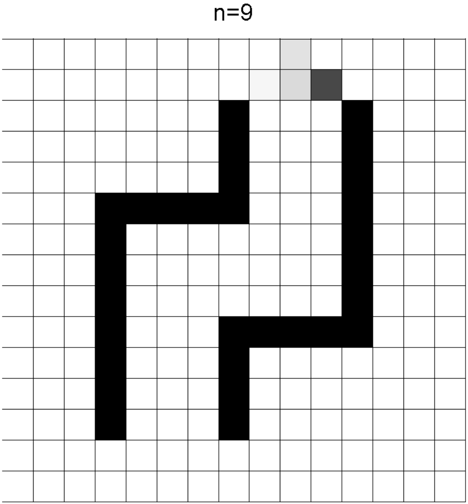
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



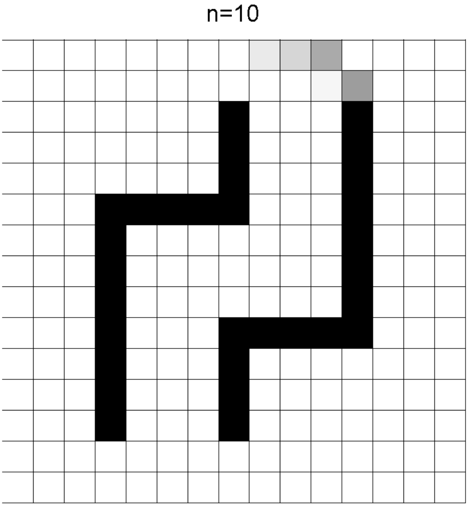
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



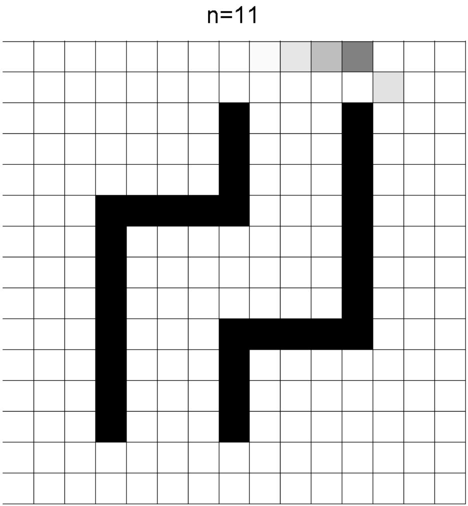
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



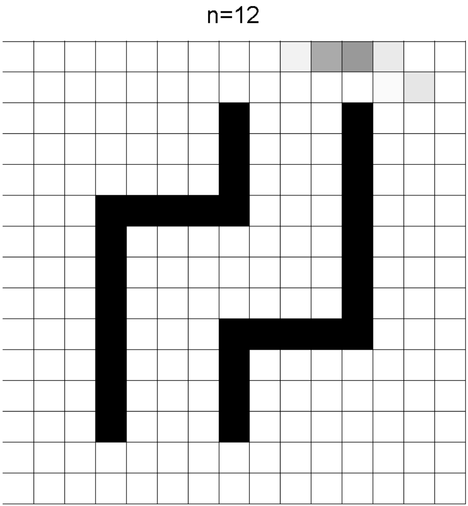
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



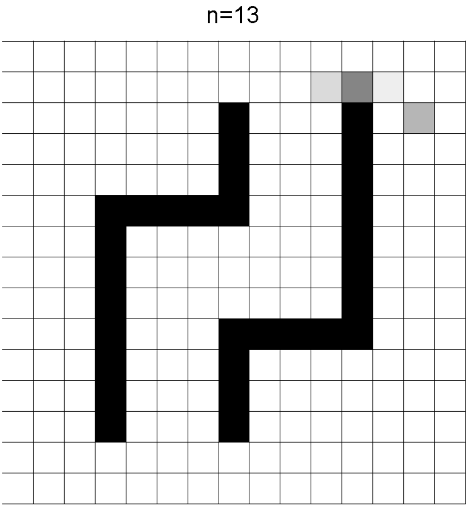
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



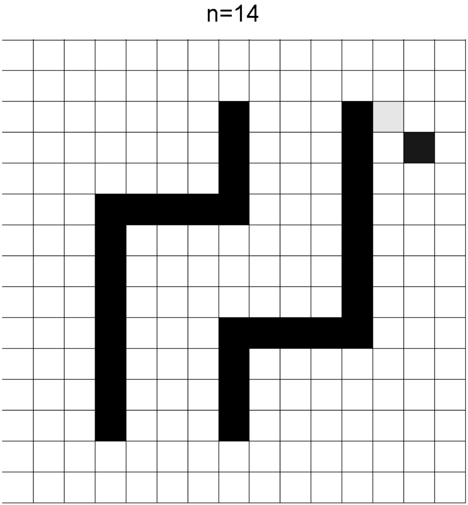
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



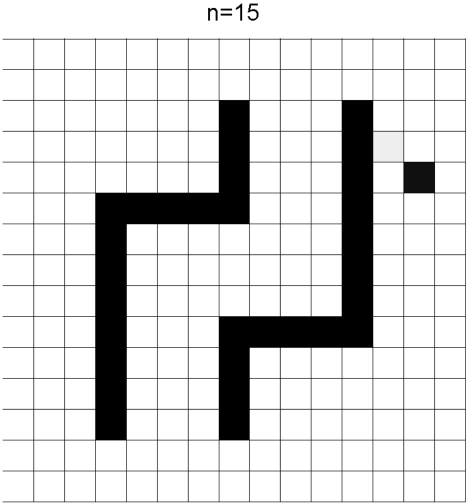
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



Localisation by Olfaction alone

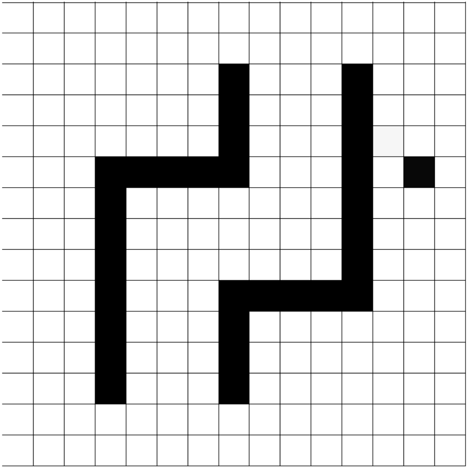
Posterior state density, $p(x_n|S_n)$.



Localisation by Olfaction alone

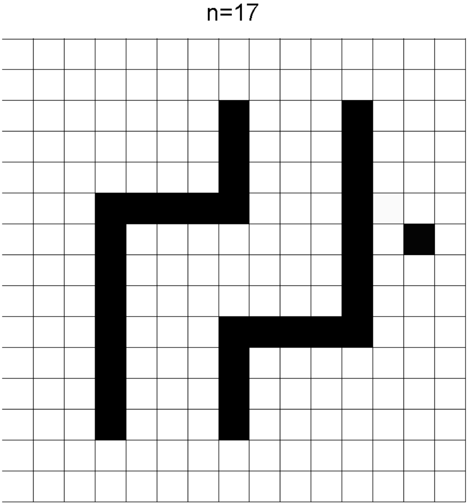
Posterior state density, $p(x_n|S_n)$.

n=16



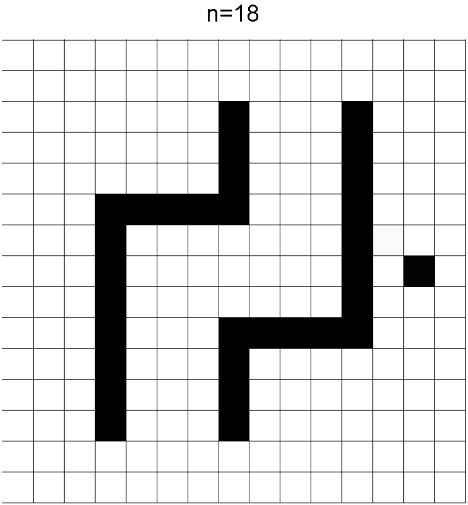
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



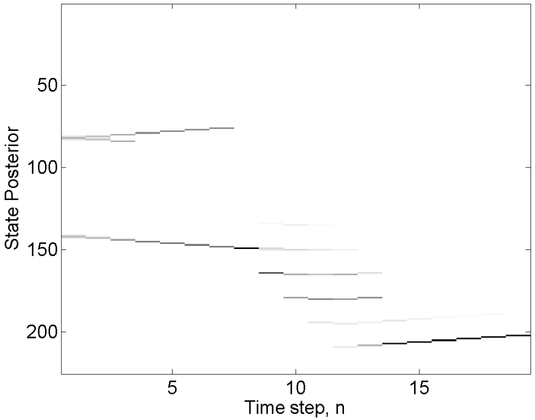
Localisation by Olfaction alone

Posterior state density, $p(x_n|S_n)$.



Multimodality

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.

Population Coding

A neurobiological perspective attributes $a_k = p(x_n = k|S_n)$ to activity of the kth place cell in CA3/CA1.

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

Here, l_k is the centre of the kth place field, and the location encoded by the population of cells is given by the mean of the predictive density

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

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

Path Integration

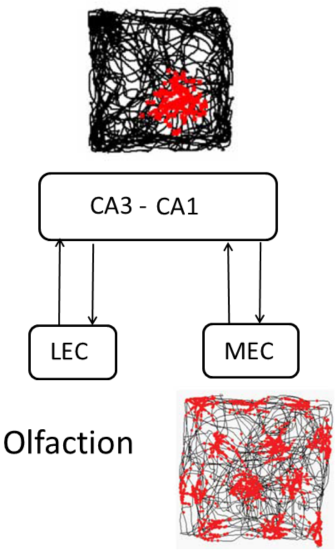
We now augment olfactory input with a spatial signal from the path integration system $r_n = l_n + e_n$ such that the variance of the additive noise is a linear function of time step.

The agent's sensory observation model is augmented as

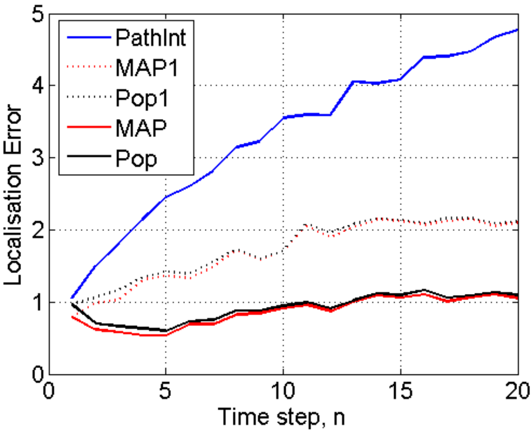
$$p(s_n|x_n = k) \quad = \quad \mathrm{N}(o_n; m_k, \Lambda_o)\mathrm{N}(r_n; l_k, \Lambda_r)$$

Spatial Localisation

Path integral representation (grid cells) can be corrected by Hippocampal feedback.



Olfactory and Path Integral Input



Path Integration only (Pathint), Maximum Posterior estimation based on a single (MAP1) and sequence (MAP) of observations, Population Decoding based on a single (Pop1) and sequence (Pop) of observations.

Planning under Uncertainty

Localisation produces density $p(x_n = k|S_n)$.

A probabilistic planning trajectory can then be found by following the posterior dynamics from this initial distribution

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

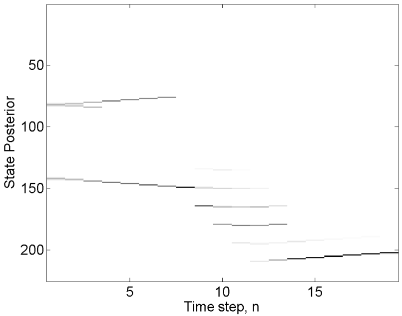
The state density at subsequent time points can be computed as

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

Show `uncertain.avi`

Summary - Discrete Latent States

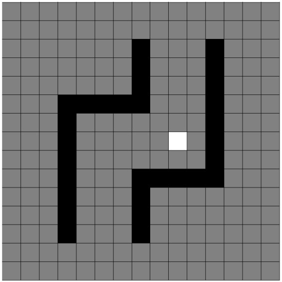
Discrete latent states are good for Localisation. They allow for multiply peaked posteriors with exact inference.



Important for localisation with ambiguous cues.

Summary - Discrete Latent States

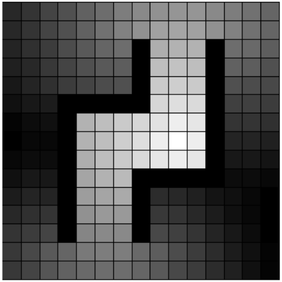
Discrete latent states are good for Planning eg. not susceptible to local minima.



Target distribution readily computed by backward sweep.

Summary - Discrete Latent States

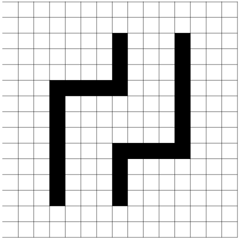
Discrete latent states in 2D are good for planning eg. not susceptible to local minima.

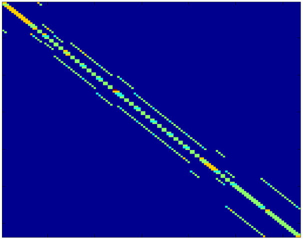


Target distribution readily computed by backward sweep.

Summary - Recurrent Activity in CA3

Muller and Stead (Hippocampus, 1997) have proposed that CA3 solves shortest path problems.

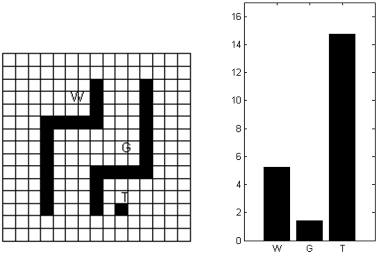




Our proposal shares this view, but with connections implementing prior dynamics.

Summary - Multivariate Goals

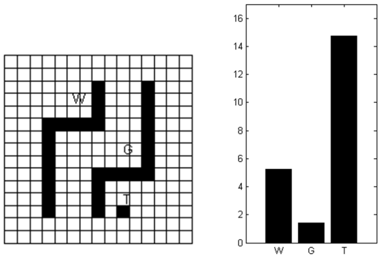
Inference based on multivariate goals allows for homeostatic control.



If actual autonomic states follow their own (decaying) dynamics, and the target distribution is periodically (or otherwise) updated, the agent displays complex autonomous dynamics.

Summary - Allostasis

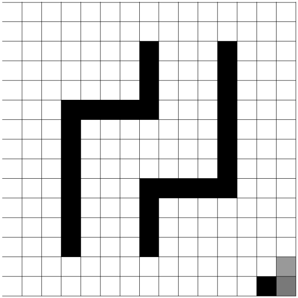
Inference based on multivariate goals allows for homeostatic control.



But if the agent also possessed a predictive model of the autonomic states it could implement predictive regulation or ‘allostatic control’ (**Sterling**, Physiology and Behaviour, 2012).

Summary - Where are the Actions ?

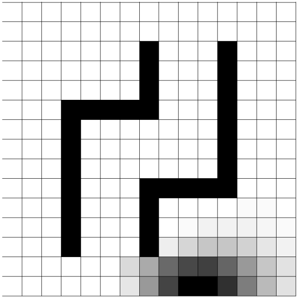
Johnson and Redish (Neural Networks, 2005) propose that HC replays goal-directed sequences for striatum to learn state action-mappings. Model-based system teaching habitual one.



We propose that these 'replays' are samples from posterior dynamics, not simple memories.

Summary - Where are the Actions ?

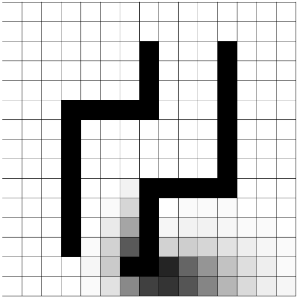
Johnson and Redish (Neural Networks, 2005) propose that HC replays goal-directed sequences for striatum to learn state action-mappings. Model-based system teaching habitual one.



We propose that these ‘replays’ are samples from posterior dynamics, not simple memories.

Summary - Where are the Actions ?

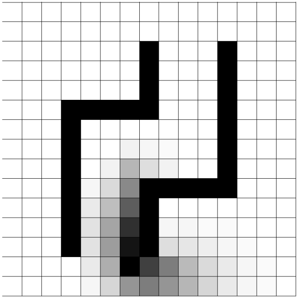
Johnson and Redish (Neural Networks, 2005) propose that HC replays goal-directed sequences for striatum to learn state action-mappings. Model-based system teaching habitual one.



We propose that these ‘replays’ are samples from posterior dynamics, not simple memories.

Summary - Where are the Actions ?

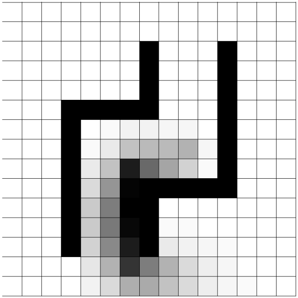
Johnson and Redish (Neural Networks, 2005) propose that HC replays goal-directed sequences for striatum to learn state action-mappings. Model-based system teaching habitual one.



We propose that these 'replays' are samples from posterior dynamics, not simple memories.

Summary - Where are the Actions ?

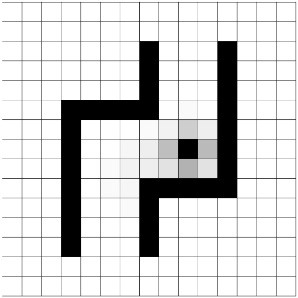
Johnson and Redish (Neural Networks, 2005) propose that HC replays goal-directed sequences for striatum to learn state action-mappings. Model-based system teaching habitual one.



We propose that these ‘replays’ are samples from posterior dynamics, not simple memories.

Summary - Where are the Actions ?

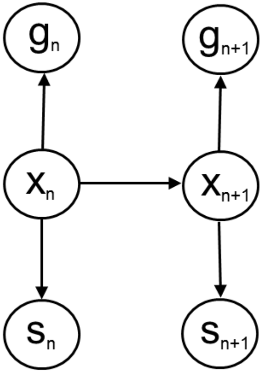
Johnson and Redish (Neural Networks, 2005) propose that HC replays goal-directed sequences for striatum to learn state action-mappings. Model-based system teaching habitual one.



We propose that these ‘replays’ are samples from posterior dynamics, not simple memories.

Summary - Cognitive Map

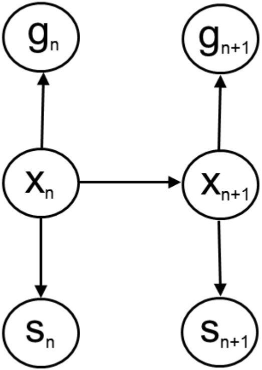
Both planning and localisation need access to the same Cognitive Map ($x_n \rightarrow x_{n+1}$).



Uncertainty from localisation is readily integrated into planning.

Summary - Cognitive Map

Both planning and localisation need access to the same Cognitive Map ($x_n \rightarrow x_{n+1}$).



Localisation is accompanied by a high theta state whereas planning is accompanied by low theta state (and high frequency ripples).

Summary

- Discrete Latent States
- Recurrent activity in CA3
- Multivariate Goals
- Where are the Actions ?
- Both Planning and Localisation need access to CA3
- What about Learning ?

Dynamic Programming

The optimal state value, $V(x_n)$, satisfies the Bellman equation

$$V(x_n) = \max_{u_n} \left(R(x_n, u_n) + \sum_{x_{n+1}} p(x_{n+1}|x_n, u_n) V(x_{n+1}) \right)$$

where $R(x_n, u_n)$ is the instantaneous reward and $p(x_{n+1}|x_n, u_n)$ specifies known Markov Decision Process (MDP) dynamics.

Or in terms of optimal costs $C = -V(x_n)$ and instantaneous costs $L(x_n, u_n) = -R(x_n, u_n)$

$$C(x_n) = \min_{u_n} \left(L(x_n, u_n) + \sum_{x_{n+1}} p(x_{n+1}|x_n, u_n) C(x_{n+1}) \right)$$

KL Control

If the instantaneous cost is

$$L(x_n, u_n) = e(x_n) + KL(p(x_{n+1}|x_n, u_n)||p(x_{n+1}|x_n))$$

then the optimal values can be computed using the Beta Recursions in an HMM (Todorov, 2006).

Here $p(x_{n+1}|x_n)$ are the 'passive dynamics' and $p(x_{n+1}|x_n, u_n)$ are the 'active dynamics'. The KL term embodies a cost for taking an action.

Derivation

The first step is to relate the cost of a state to its 'desirability'

$$\beta(x_n) = \exp(-C(x_n))$$

which is defined such that more costly states are less desirable. We can then write the Bellman equation as

$$-\log \beta(x_n) = \min_{u_n} \left(L(x_n, u_n) - \sum_{x_{n+1}} p(x_{n+1}|x_n, u_n) \log \beta(x_{n+1}) \right)$$

The instantaneous cost is

$$L(x_n, u_n) = e(x_n) + \sum_{x_{n+1}} p(x_{n+1}|x_n, u_n) \log \frac{p(x_{n+1}|x_n, u_n)}{p(x_{n+1}|x_n)}$$

Substituting into the Bellman equation gives

$$-\log \beta(x_n) = e(x_n) + \min_{u} \left(\sum_{x_{n+1}} p(x_{n+1}|x_n, u_n) \log \frac{p(x_{n+1}|x_n, u_n)}{p(x_{n+1}|x)\beta(x_{n+1})} \right)$$

Derivation

From previous page

$$-\log \beta(x_n) = e(x_n) + \min_u \left(\sum_{x_{n+1}} p(x_{n+1}|x_n, u_n) \log \frac{p(x_{n+1}|x_n, u_n)}{p(x_{n+1}|x)\beta(x_{n+1})} \right)$$

We next divide the quantity in the numerator of the log by

$$g(x_n, \beta_n) = \sum_{x_{n+1}} p(x_{n+1}|x_n)\beta(x_{n+1})$$

so that the resulting quantity becomes a probability (ie sums to unity). This division gives

$$-\log \beta(x_n) = e(x_n) - \log g(x_n, \beta_n) + \min_{u_n} \left(\sum_{x_{n+1}} p(x_{n+1}|x_n, u_n) \log \frac{p(x_{n+1}|x_n, u_n)}{p(x_{n+1}|x_n)\beta(x_{n+1})/g(x_n, \beta_n)} \right)$$

Derivation

From previous page

$$-\log \beta(x_n) = e(x_n) - \log g(x_n, \beta_n) + \min_{u_n} \left(\sum_{x_{n+1}} p(x_{n+1}|x_n, u_n) \log \frac{p(x_{n+1}|x_n, u_n)}{p(x_{n+1}|x_n)\beta(x_{n+1})/g(x_n, \beta_n)} \right)$$

We can then finally write

$$\log \left(\frac{g(x_n, \beta_n)}{\beta(x_n)} \right) = e(x_n) + \min_{u_n} \left(KL \left(p(x_{n+1}|x_n, u_n) || \frac{p(x_{n+1}|x_n)\beta(x_{n+1})}{g(x_n, \beta_n)} \right) \right)$$

Importantly we know that the KL divergence is minimized to a value of zero if the two densities are equal

$$p(x_{n+1}|x_n, u_n) = \frac{p(x_{n+1}|x_n)\beta(x_{n+1})}{g(x_n, \beta_n)}$$

Assuming we can set u_n to achieve this we we will be taking the optimal action. Moreover, the Bellman equation then becomes

$$\log \left(\frac{g(x_n, \beta_n)}{\beta(x_n)} \right) = e(x_n)$$

This can then be re-arranged as

$$\beta(x_n) = \exp(-e(x_n)) \sum_{x_{n+1}} p(x_{n+1}|x_n)\beta(x_{n+1})$$