

Kalman filtering and sensorimotor integration by recurrent biological neural networks

Sophie Deneve¹ and Alexandre Pouget²,

¹ Gatsby computational neuroscience unit, University college London, London, UK.

² Department of Brain and Cognitive Science, University of Rochester, Rochester NY, USA.

1 Introduction

In order to estimate the position of objects and the posture of the body, the brain must constantly process incoming sensory and postural inputs, taking into account self-movements and their sensory consequences. One way to do this optimally in the presence of noise would be to compute an internal model of body posture and object position, use a forward model and efferent motor commands to update this internal model on-line, and combine the internal model with the sensory feedback to recover the most reliable estimate of position at any time. Assuming gaussian noise and locally linear sensorimotor transformations, the sensorimotor system can be modelled with the following set of discrete dynamical equations:

$$\mathbf{x}(t + dt) = \mathbf{M}\mathbf{x}(t) + \mathbf{B}\mathbf{c}(t) + \mathbf{w}(t) \quad (1)$$

$$\mathbf{s}(t + dt) = \mathbf{x}(t + dt) + \mathbf{v}(t) \quad (2)$$

where \mathbf{x} is the state of the environment and the body, \mathbf{c} is the motor command resulting in a change in that state, dt is a small temporal increment and \mathbf{s} is a vector containing the sensory and postural inputs available to the organism. The motor noise, $\mathbf{w}(t)$, and sensory noise, $\mathbf{v}(t)$ are zero-mean multivariate gaussian random variables, while \mathbf{M} and \mathbf{B} are matrices describing the dynamical system. Kalman filter theory allows one to iteratively compute an optimal estimate of the state vector $\mathbf{x}(t + dt)$, according to:

$$\hat{\mathbf{x}}(t + dt) = \mathbf{M}((\mathbf{I} - \mathbf{K}(t))\hat{\mathbf{x}}(t) + \mathbf{K}(t)\mathbf{s}(t)) + \mathbf{B}\mathbf{c}(t) \quad (3)$$

$\mathbf{K}(t)$ is known as the Kalman gain matrix, which is iteratively computed from \mathbf{M} , the covariance matrices of the motor and sensory noises, and the covariance matrix of the initial state. Note that here we are using the equation to estimate the state at time $t + dt$, based on the observation of the sensory input and motor command at time t , and *before* observing the sensory input at time $t + dt$. In the Kalman filter terminology, this is known as the time update, or *a priori* estimate of the state, sometimes noted $\hat{\mathbf{x}}(t + dt|t)$. We concentrate on the *a priori* estimate, as opposed to the *a posteriori* estimate, for reasons that should become clear later on.

Our goal is to understand how such optimal filtering could be implemented in biological neural networks. What makes this problem difficult is the fact that the cortex uses noisy population codes, not scalar or vector quantities such as $\hat{\mathbf{x}}(t)$. However, we have recently shown that basis function networks with attractor dynamics can behave as a close approximation of a maximum likelihood estimator for static stimuli [3]. We now show that the same architecture can be used for time varying stimuli, implementing a close approximation to a Kalman filter. This model have important implications for neural representations in sensori-motor brain areas. We start by describing one particular example before presenting a sketch of the general theory.

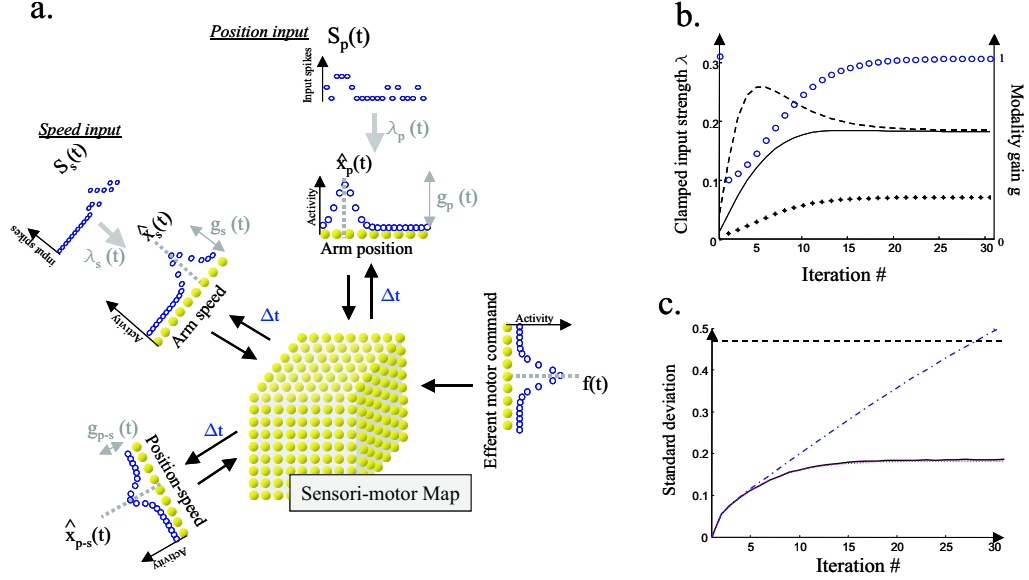


Figure 1: (a) Network implementing an optimal filter for 1D arm kinematics. (b) Parameters $\lambda_s(t)$ (solid line), $\lambda_p(t)$ (dotted line), $g_s(t)$ (circles), $g_{s-p}(t)$ (stars). (c) Standard deviation of arm position estimated by network (solid line), a Kalman filter (dotted line, very close to the solid line), a purely forward estimate (dot-dashed line) and a purely feedback estimate (dashed line).

2 Sensorimotor integration for arm position and velocity.

Consider a subject keeping track of his arm position during movement, while receiving somatosensory feedback about his arm position and velocity. To simplify, the arm is considered as a point-mass whose dynamical equations are described by a set of (discretized) linear dynamical equations:

$$\begin{aligned} x_p(t+dt) &= x_p(t) + 0.1x_s(t) + w_p(t) \\ x_s(t+dt) &= 0.9x_s(t) + f(t) + w_s(t) \end{aligned} \quad (4)$$

$$(5)$$

where x_p and x_s are the arm position and velocity, w_p and w_s are independent gaussian noises with 0 mean and small (time independent) variance describing the motor noise.

Somatosensory feedback about arm position and velocity is available in the form of noisy population codes for arm position and velocity on two distinct somatosensory layers (see figure 1-a). The neural responses in the arm position layer are generated at each time step from a poisson process. Thus,

$$p(S_i^p(t)|x_p(t)) = \frac{[f_i(x_p(t))dt]^{S_i^p(t)} \exp(-f_i(x_p(t)dt))}{S_i^p(t)!} \quad (6)$$

where $S_i^p(t)$ is the number of spike received between t and $t+dt$ by neuron i in the position layer. The tuning curve of this neuron, f_i , is bell-shaped and peaking at this neuron's preferred arm position. The velocity sensory feedback is similarly obtained.

How can the subject estimate his arm position during movement given the sensory feedback and the arm dynamics? One first possibility is to read-out the arm position at time t directly from the feedback $\mathbf{S}_p = [S_i^p(t)]_{i=1\dots N}$, ignoring the sensory feedback received in the past or the

motor commands. However, the temporal window dt being very small, the $\mathbf{S}_p(t)$ correspond to just a few spikes, from which arm position cannot be reliably estimated.

Another possibility is to predict the arm position during the movements iteratively, using a forward model for the arm dynamics and afferent copies of the motor commands. Thus, the arm position at time $t + dt$ can be predicted from the arm position and velocity at time t and the muscular force $\mathbf{f}(t)$ exerted on the arm. This forward estimate is very reliable at the beginning of the movements (if we suppose that the initial state of the arm is known) but accumulate errors during movements due to motor noise.

As in a kalman filter, the internal model could be prevented from accumulating too many errors by combining it on-line with the sensory feedback. However, contrary to equation 3, the sensory feedback is a noisy population code, not a scalar. As shown in the next section, optimal sensorimotor integration in cortical networks can be implemented with a neural version of a forward model in a population code format.

2.1 Neural model

We used a recurrent network model with 4 input layers: the position layer, representing $x_p(t)$, the velocity layer, representing $x_s(t)$, a layer for the dummy "correlation" variable $x_{p-s}(t) = x_p(t) - x_s(t)$ whose role is to encode the correlation between position and velocity (see general theory), and the afferent motor command layer coding for $\mathbf{f}(t)$.

As represented in figure 1-a, all these input layers project to a radial basis function map combining non-linearly the sensory and motor population codes. This radial basis function map has 3 dimensions, one for velocity, one for position and one for force, and sends feedback connections to its input layer, with a delay dt . These feedback connections compute predictive population code for $x_p(t + dt)$, $x_s(t + dt)$, and $x_{p-s}(t + dt)$ from the population codes for $x_p(t)$, $x_s(t)$, $f(t)$ and $x_{p-s}(t)$ received at time t .

As discussed previously, this forward estimate must be combined with the online sensory feedback to keep the most reliable on-line estimate for the arm position during movement. We clamped on-line the somatosensory feedback $\mathbf{S}_p(t)$ and $\mathbf{S}_s(t)$ onto the position and velocity input layers. Thus, the population activity in the position layer are described by:

$$\mathbf{A}_p(t) = g_p(t) \left((1 - \lambda_p(t)) \mathbf{P}_p(t) + \lambda_p(t) \mathbf{S}_p(t) \right) \quad (7)$$

$\mathbf{P}_p(t)$ is the population code for the predicted arm position, computed by the feedback connections from the sensorimotor basis function layer. Thus, $\mathbf{P}_p(t)$ is the forward estimate for arm position. The clamped input contribution $\lambda_p(t)$ and the modality gain $g_p(t)$ control the contribution of the forward estimate relative to the sensory feedback, and are derived from the kalman gain matrix, as described in the general theory.

2.2 Results

For each trial, we measured the network estimates for arm position and velocity, $\hat{x}_p(t)$ and $\hat{x}_s(t)$, by measuring the position of the hill of activities on the position and velocity layer at each time t . The performance of the network is computed by calculating the variance of these estimates compare to the true arm position and velocity.

As shown on figure 1-c, the variance of the network position estimate is very close to the variance of an optimal estimator or kalman filter (only 2% worse). For this particular simulation, we considered that the position and velocity of the arm at time 0 were perfectly known (thus, the initial variance is zero). This performance is much better than the performance of a purely forward model, based on a temporal update of the initial arm position without somatosensory feedback or a purely feedback model which would only take into account the somatosensory input without relying on an internal model.

We chose this example in part to compare our model with experimental data presented by Wolpert et al [5] as support for the implication of an internal model in the control of the arm.

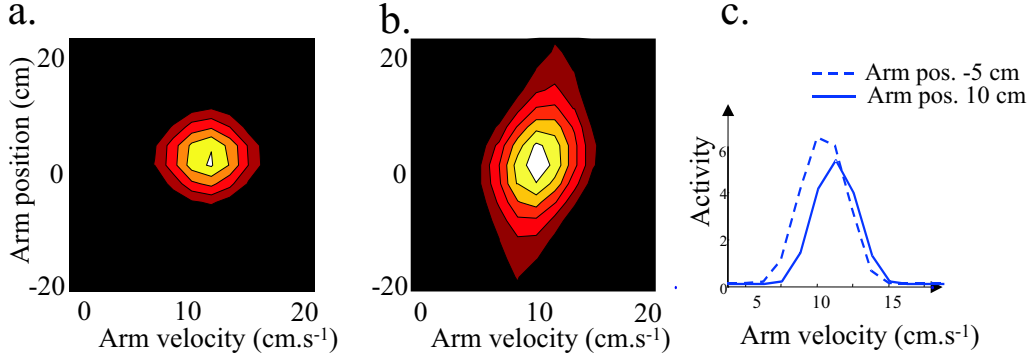


Figure 2: (a) Tuning curve of an intermediate unit to arm velocity and speed at the beginning of the movement ($t=40\text{ms}$) (b) Tuning curve of an intermediate unit to velocity and speed after $t=400\text{ms}$ of movement. (c) Tuning curve of an intermediate unit to velocity for 2 different arm position.

Their experiments measured the propagation of error and bias when humans move their arm on a treadmill for variable lengths of time. The arm is initially visible but invisible during movements. As they observed, we find that the variance of estimated arm position during the movement increases and then stabilizes to a stable state, as in figure 1-c, and closely mimics the performance of the Kalman filter.

The parameters λ and g for this particular initial condition (initial variances at 0) are plotted in figure 1-b. Note that while the initial temporal profile of these control gains depends on the initial variance of the velocity and position estimates, their stable regime, reached after about 100ms, is a function of the sensory and neural noise alone. As such, their stable values are time invariant and independent on these initial conditions. We were able to show that the network acts as an optimal on-line estimator regardless of the arm trajectory and initial state. This result is a direct consequence from the proof out-lined in the general theory section.

2.3 Neural predictions

In figure 2-a and b, we plotted the two-dimensional tuning curves of an intermediate cell to arm position and velocity at 2 different period during the movements. At the beginning of the movement, the tuning curve is symmetric and bell-shaped, and this unit can be described as tuned to velocity and gain modulated by arm position, or vice versa, as tuned to position and gain modulated by velocity. However, after a few hundred milliseconds of movements the response become asymmetric and appears elongated along a tilted axis. In particular, the cell's preferred velocity shift with the arm position so that the cell prefers higher velocity for more rightward position, as illustrated in figure 2-c. Note that, by convention, arm position are increasing from left to right.

Why is the velocity tuning not invariant to changes in arm position? In fact, arm position and velocity might initially be independent random variables but they always become positively correlated during movements. For example, if the velocity at time $t - dt$ is higher than expected due to neural noise, both position and velocity will be higher at time t . In the network, this correlation between position and velocity is taken into account thanks to an input layer coding for the dummy variable x_{p-s} .

Shifts of the velocity tuning curves with position compatible with our prediction have been observed experimentally in motor cortices [2], suggesting that these cells are involved in optimal sensorimotor integration. More generally, this dependance of neural responses on the correlations between sensory or motor variables might have contributed to the on-going controversy about the nature of motor representations.

3 General Theory

This is an outline of the general proof showing that a Kalman filter for any set of linear sensorimotor dynamical equations associated with sensory and motor noise can be implemented in an iterative basis function network.

We assume that the state of the environment and body, $\mathbf{x}(t)$ follows equation 2, where $\mathbf{x}(t)$ is a vector with K components $\mathbf{x}(t) = \{x_k(t)\}_{k=1,K}$. Moreover, the brain receives sensory and postural inputs from K sources which provide information about the state of the system. These sensory inputs are noisy population codes for a set of K input variables: $\mathbf{S}_k(t) = \mathbf{f}(x_k(t)) + \mathbf{N}_t$, where \mathbf{f} describes the tuning curves and \mathbf{N}_t the neural noise at time t .

We construct a network with input layers divided into two groups. The first group corresponds to the layers encoding the efferent motor commands $\mathbf{c}(t)$. The second group contains the sensory layers. These input layers connect to an intermediate radial basis function map which projects back to all input layers. The feedback connections send a prediction, $\mathbf{P}_k(t)$, of the population code to each input layer, computed from the sensory and motor population codes at time $t - dt$.

At each time step the activities on the input layers, $\mathbf{A}_k(t)$ are computed by combining the prediction from the basis function layer, $\mathbf{P}_k(t)$, with the current sensory input, $\mathbf{S}_k(t)$:

$$\mathbf{A}_k(t) = g_k(t) \left((1 - \lambda_k(t)) \mathbf{P}_k(t) + \lambda_k(t) \mathbf{S}_k(t) \right) \quad (8)$$

As discussed earlier, the terms $\mathbf{P}_k(t)$ are population codes for the *a priori* estimate of the state, $\hat{\mathbf{x}}_k(t)$. To compute the next state estimate, we first iterate equation 8 and then compute $\mathbf{P}_k(t+dt)$ from the patterns $\mathbf{A}_k(t)$ (see [3] for equations). The computation involved in this latter step is formally equivalent to iterating the state equation (eq. 1), since the basis function implements the state equation with population codes. In the limit of low noise, the *a priori* estimate of the state vector in the network is updated at each time step according to a concatenation of equation 8 and the state equation:

$$\hat{\mathbf{x}}(t+dt) = \mathbf{M}\mathbf{G}(t) \left((\mathbf{I} - \mathbf{\Lambda}(t)) \hat{\mathbf{x}}(t) + \mathbf{\Lambda}(t) \hat{\mathbf{s}}(t) \right) + \mathbf{B}\mathbf{c}(t), \quad (9)$$

where $\mathbf{G}(t)$ is a matrix completely determined by the modality gains $g_k(t)$, $\mathbf{\Lambda}(t)$ is a diagonal matrix whose diagonal elements are the $\lambda_k(t)$ and $\hat{\mathbf{s}}$ is the maximum likelihood estimate of the sensory inputs, $\hat{s}_k(t) = \arg \max_{x_k(t)} P(\mathbf{S}_k(t)|x_k(t))$. Due to space constraints, we cannot provide the full derivation between equation 8 and 9, but it is a direct extension of our previous analytical work (see supplementary information of [3]).

The last step consists of finding the matrices $\mathbf{G}(t)$ and $\mathbf{\Lambda}(t)$, such that equation 9 is equivalent to the Kalman filter equation (eq. 3). This is not always possible for arbitrary state variables. However, a solution can always be found by adding dummy "correlation" variables to the state vector in the form of appropriately chosen linear combinations of the other variables. This is precisely the role of the variable x_{p-s} .

4 Discussion

We have proposed a general framework for implementing optimal on-line sensorimotor integration in cortical networks by performing dynamical remapping, or iterative predictions, in the presence of time varying noisy sensory inputs. The projection of multiple sources of information on iterative basis function layers allows the system to take advantage of all redundancies and correlations present in the sensory input, as well as prior knowledge of sensory-motor dynamics.

Note that this framework can also be used to compute the motor command from an initial and a desired state (optimal motor control). Moreover as basis function maps can approximate any non-linear function, this approach generalizes easily to non-linear dynamical equations, in which case the network can perform as well as an extended Kalman filter [1]. In a more realistic model of sensorimotor integration and optimal motor control, the sensory delays must be taken

into account. Our model, however, can be used to predict any future state of the sensorimotor system, and thus provide a straightforward compensation for sensory delays.

The network implements the Kalman gain matrix by manipulating the gains of neural responses and the contribution of sensory inputs onto the sensory input layers, in contrast with another work which used local hebbian plasticity [6]. Such gain modulations could be implemented by synaptic depression, sensory suppression during movements, release of neuromodulators, attentional modulations, adaption due to potassium currents, or other biological mechanisms. It is unclear if complex, movement specific temporal modulation as plotted in figure 1-b are indeed implemented in cortical circuits. However, a simplified model could still be very close to optimal. For example, the stable regime in figure 1-b corresponds to the stable state of the Kalman equation and does not depend on the initial covariance matrix or the motor commands. Using these constant gains and input strengths, the network still exhibits optimal performance except at the beginning of movements. Which simplification would be biologically realistic and which appropriate gain modulations could be implemented and learned by neural circuits remains to be explored.

We used basis function maps to implement optimal multisensory integration [3] and optimal sensorimotor integration during saccadic eye movements (not shown because of space constraints) and arm movements. These sensorimotor systems correspond to well studied cortical areas (e.g. LIP and M1). Evidence for the neural implementation of a forward model exists in LIP, where visual receptive fields are remapped *prior* to a saccade to predict the new eye-centered position of a visual object *after* the saccade [4]. The fact that arm velocity tuning curves shift with position suggests that a forward model might also be in use for arm control in M1 [2]. More generally, the model has important implications for the neural representations we expect to find in multi-modal brain areas. Regardless of implementation details, our model predicts that the tuning curves of sensorimotor cells will have the same overall shape than the joint probability of the corresponding state variables. Thus, most neurons in multisensory and sensorimotor areas should represent sensory, postural and motor variables in a non-separable way (meaning, their response should not systematically be a sum or a product of their sensory, postural and motor tuning curves). Instead, their representation should be flexible, partially shifting, movement and task-dependent, dominated by the frame of reference of their most reliable input variables and reflecting the correlations between them.

References

- [1] B. Anderson and J. Moore. *Optimal filtering*. Prentice Hall, Englewood Cliffs, N.J., 1979.
- [2] R. Caminiti, P. Johnson, and A. Urbano. Making arm movements within different parts of space: Dynamic aspects in the primate motor cortex. *Journal of Neuroscience*, 10:2039–2058, 1990.
- [3] S. Deneve, P. Latham, and A. Pouget. Efficient computation and cue integration with noisy population codes. *Nature Neuroscience*, 4(8):826–831, 1999.
- [4] J. Duhamel, C. Colby, and M. Goldberg. The updating of the representation of visual space in parietal cortex by intended eye movements. *Science*, 255(5040):90–92, 1992.
- [5] D. Wolpert, Z. Ghahramani, and M. Jordan. An internal model for sensorimotor integration. *Science*, 269(5232):1880–82, 1995.
- [6] S. Wu, , and S. Amari. Neural implementation of bayesian inference in population codes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, Cambridge, MA, 2002.