Supplementary Material for : A Neural Implementation of the Kalman Filter

Robert C. Wilson
Department of Psychology
Princeton University
Princeton, NJ 08540
rcw2@princeton.edu

Leif H. Finkel
Department of Bioengineering
University of Pennsylvania
Philadelphia, PA 19103

1 Relation to Kalman filter - small prediction error case

Here we present the full analysis relating the activity of the network to the Kalman filter. Our approach is to analyze the network in terms of $\mathbf{U}$, which, for clarity, we define here to be the fixed point membrane potential profile of the network when $w = 1$, $\gamma(t) = 0$, $\mathbf{I}(t) = 0$, $S = S_0$ and $\mu = \mu_0$. Thus, the results described here are independent of the exact form of $\mathbf{U}$ so long as it is a smooth, non-uniform profile over the network.

We begin by making the assumption that both the input, $\mathbf{I}(t)$, and the network membrane potential, $\mathbf{u}(t)$, take the form of scaled versions of $\mathbf{U}$, with the former encoding the noisy observations, $z(t)$, and the latter encoding the network's estimate of position, $\hat{x}(t)$, i.e.,

$$\mathbf{I}(t) = A(t)\mathbf{U}(z(t)) \quad \text{and} \quad \mathbf{u}(t) = \alpha(t)\mathbf{U}(\hat{x}(t)) \tag{1}$$

Substituting this *ansatz* for membrane potential into the left hand side of the update equation (equation 4 in the main text) gives

$$LHS = \alpha(t+1)\mathbf{U}\left(\hat{x}(t+1)\right) \tag{2}$$

and into the right hand side of the update equation gives

$$RHS = \underbrace{w\mathbf{J}^{sym}\mathbf{f}\left[\alpha(t)\mathbf{U}\left(\hat{x}(t)\right)\right]}_{\text{recurrent input}} + \underbrace{A(t+1)\mathbf{U}\left(z(t+1)\right)}_{\text{external input}} \tag{3}$$

For the *ansatz* to be self-consistent we require that RHS can be written in the same form as LHS. We now show that this is the case.

We begin by considering the first term in equation 3, the recurrent input term, and show that this implements prediction.

$$\begin{aligned}
\text{recurrent input} &= w\mathbf{J}\mathbf{f}\left[\alpha(t)\mathbf{U}\left(\hat{x}(t)\right)\right] \\
&= \frac{w(\mathbf{J}^{sym} + \gamma(t)\mathbf{J}^{asym})\left[\alpha(t)\mathbf{U}(\hat{x}(t))\right]_+}{S + \mu\sum_i \alpha(t)\left[U_i(\hat{x}(t))\right]_+} \\
&= \frac{w\alpha(t)(\mathbf{J}^{sym} + \gamma(t)\mathbf{J}^{asym})\left[\mathbf{U}(\hat{x}(t))\right]_+}{S + \mu\alpha(t)\sum_i \left[U_i(\hat{x}(t))\right]_+} \\
&= \frac{w\alpha(t)(\mathbf{J}^{sym} + \gamma(t)\mathbf{J}^{asym})\left[\mathbf{U}(\hat{x}(t))\right]_+}{S + \mu\alpha(t)\mathcal{I}}
\end{aligned} \tag{4}$$

where

$$\mathcal{I} = \sum_i \left[U_i\right]_+ \tag{5}$$

Now, by the definition of $\mathbf{U}$,

$$\mathbf{J}^{sym}\left[\mathbf{U}(\hat{x}(t))\right]_+ = (S_0 + \mu_0\mathcal{I})\mathbf{U}(\hat{x}(t)) \tag{6}$$

and, since $\mathbf{J}^{asym}$ is defined as the spatial derivative of $\mathbf{J}^{sym}$, then

$$\mathbf{J}^{asym}\left[\mathbf{U}(\hat{x}(t))\right]_{+} = (S_0 + \mu_0 \mathcal{I})\mathbf{U}'(\hat{x}(t)) \tag{7}$$

where $\mathbf{U}'(\hat{x}(t))$ is the spatial derivative of $\mathbf{U}$ centered at $\hat{x}(t)$. Thus we can write equation 4 as

$$\begin{aligned}
\text{recurrent input} &= \frac{w\alpha(t)(S_0 + \mu_0 \mathcal{I})(\mathbf{U}(\hat{x}(t)) + \gamma(t)\mathbf{U}'(\hat{x}(t)))}{S + \mu\alpha(t)\mathcal{I}} \\
&\approx \frac{w\alpha(t)(S_0 + \mu_0 \mathcal{I})\mathbf{U}(\hat{x}(t) + \gamma(t))}{S + \mu\alpha(t)\mathcal{I}} \\
&= \frac{1}{\frac{S}{w(S_0+\mu_0\mathcal{I})}\frac{1}{\alpha(t)} + \frac{\mu\mathcal{I}}{w(S_0+\mu_0\mathcal{I})}}\mathbf{U}(\bar{x}(t+1)) \\
&= C\mathbf{U}(\bar{x}(t+1))
\end{aligned} \tag{8}$$

To get from the first to the second line, we have assumed that $\gamma(t)$ is small such that the linear approximation for $\mathbf{U}(\hat{x}(t) + \gamma(t))$ is valid. In the third line we have written $\bar{x}(t+1) = \hat{x}(t) + \gamma(t)$ and rearranged the prefactor, assuming $\alpha(t) \neq 0$, into the form of equation 11 from the main text. Thus the recurrent connections implement prediction and this brings us to equation 10 from the main text

$$RHS \approx \underbrace{C\mathbf{U}\left(\bar{x}(t+1)\right)}_{\text{prediction}} + \underbrace{A(t+1)\mathbf{U}\left(z(t+1)\right)}_{\text{external input}} \tag{9}$$

If we define the prediction error to be

$$\delta = z(t+1) - \bar{x}(t+1) \tag{10}$$

and assume that it is small, then we can write

$$\begin{aligned}
RHS &\approx C\mathbf{U}\left(\bar{x}(t+1)\right) + A(t+1)\mathbf{U}\left(z(t+1)\right) \\
&= C\mathbf{U}\left(\bar{x}(t+1)\right) + A(t+1)\mathbf{U}\left(\bar{x}(t+1) + \delta\right) \\
&\approx C\mathbf{U}\left(\bar{x}(t+1)\right) + A(t+1)\left[\mathbf{U}\left(\bar{x}(t+1)\right) + \delta\mathbf{U}'\left(\bar{x}(t+1)\right)\right] \\
&= \left[C + A(t+1)\right]\mathbf{U}\left(\bar{x}(t+1)\right) + A(t+1)\delta\mathbf{U}'\left(\bar{x}(t+1)\right) \\
&\approx \left[C + A(t+1)\right]\mathbf{U}\left(\bar{x}(t+1) + \frac{A(t+1)}{A(t+1)+C}\left[z(t+1) - \bar{x}(t+1)\right]\right)
\end{aligned} \tag{11}$$

which is of the same form as equation 2 and thus the *ansatz* holds. More specifically, equating terms in equations 2 and 11, we can write down expressions for $\alpha(t+1)$ and $\hat{x}(t+1)$

$$\alpha(t+1) \approx C + A(t+1) = \frac{1}{\frac{S}{w(S_0+\mu_0\mathcal{I})}\frac{1}{\alpha(t)} + \frac{\mu\mathcal{I}}{w(S_0+\mu_0\mathcal{I})}} + A(t+1) \tag{12}$$

$$\hat{x}(t+1) \approx \bar{x}(t) + \frac{A(t+1)}{\alpha(t+1)}\left[z(t+1) - x(t+1)\right] \tag{13}$$

which, if we define w such that

$$\frac{S}{w(S_0 + \mu_0\mathcal{I})} = 1 \qquad \text{i.e.} \qquad w = \frac{S}{S_0 + \mu_0\mathcal{I}} \tag{14}$$

are identical to the Kalman filter update equations (equations 2 and 3 in the main text) so long as

$$\text{(a)} \quad \alpha(t) \propto \frac{1}{\hat{\sigma}_x(t)^2} \qquad \text{(b)} \quad A(t) \propto \frac{1}{\sigma_z(t)^2} \qquad \text{(c)} \quad \frac{\mu\mathcal{I}}{S} \propto \sigma_v(t)^2 \tag{15}$$

Thus the network dynamics, when the prediction error is small, map directly onto the Kalman filter equations.

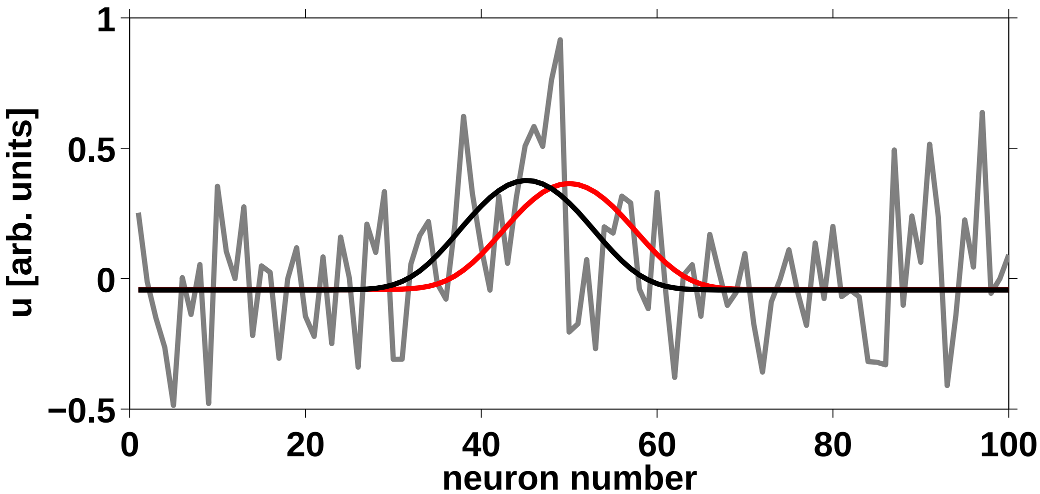

Figure 1: An example of input noise. The red line corresponds to the original, uncorrupted bump centered at $x(t) = 50$. The grey line corresponds to the same bump corrupted by independent Gaussian noise with $\sigma_{noise} = 0.3$. The black line corresponds to, $\mathbf{U}(z(t))$, the maximum likelihood position of the bump given the noisy input. Note that the bump has effectively been shifted to the left by the noise.

2 Effect of input noise

In this section we consider the effect of input noise on the ability of the network to implement a Kalman filter. We model input noise as a random vector, $\epsilon(t)$, that is added to the original noise free input, i.e.

$$\mathbf{I}(t) = A(t)\mathbf{U}(x(t)) + \epsilon(t) \tag{16}$$

and assume that $\epsilon(t)$ is sampled independently at each time step from a Gaussian distribution with mean 0 and covariance matrix Σ, thus we assume that the noise is uncorrelated in time, but may be correlated in space.

We begin by computing the effect of the noise on the position of the input bump which allows us to compute an effective input position from the noisy stimulus that we can feed into our equivalent Kalman filter. Next, we approximate the dynamics of the network and show how their correspondence to the Kalman filter degrades smoothly as the noise level increases.

2.1 Effect of noise on the position of the input bump

Assuming that the true stimulus location is at $x(t)$, we can write the noisy input as

$$\mathbf{I}(t) = A(t)\mathbf{U}(x(t)) + \epsilon(t) \tag{17}$$

Our goal in this section is to characterize the effect of this noise on the position of the input bump. The approach we take is to rewrite $\mathbf{I}(t)$ in the form

$$\mathbf{I}(t) = A(t)\mathbf{U}(z(t)) + \epsilon_{\perp}(t) \tag{18}$$

for some $z(t)$ such that the remaining noise component $\epsilon_{\perp}(t)$ is orthogonal to the line attractor manifold.

This approach is equivalent to asking the question, what position, $z(t)$, would the maximum likelihood decoder choose given the noisy bump of activity $\mathbf{I}(t)$? This idea is illustrated in figure 1. Here the red line corresponds to bump at the location of the actual stimulus, $\mathbf{U}(x(t))$. This is then corrupted by noise to give the noisy input profile shown in grey (note that this example uses a high noise setting for illustrative purposes). Finally, the black line denotes $\mathbf{U}(z(t))$, the maximum likelihood location of the bump given the noisy input activity. Thus, the addition of noise has lead to a shift in the position of the input bump.

In particular we are interested in the conditional distribution $p(z(t)|x(t))$ which we compute as the marginal over $\mathbf{I}(t)$, i.e.

$$p(z(t)|x(t)) = \int p(z(t)|\mathbf{I}(t))p(\mathbf{I}(t)|x(t))d\mathbf{I} \tag{19}$$

Now,

$$\begin{aligned} p(\mathbf{I}(t)|x(t)) &= p\left(\epsilon = \mathbf{I}(t) - A(t)\mathbf{U}(x(t))\right) \\ &\propto \mathcal{N}\left(\mathbf{I}(t); A(t)\mathbf{U}(x(t)), \Sigma\right) \end{aligned} \tag{20}$$

and

$$\begin{aligned} p(z(t)|\mathbf{I}(t)) &\propto p\left(\epsilon = \mathbf{I}(t) - A(t)\mathbf{U}(z(t))\right) \\ &\propto \mathcal{N}\left(\mathbf{I}(t); A(t)\mathbf{U}(z(t)), \Sigma\right) \end{aligned} \tag{21}$$

Thus, $p(z(t)|x(t))$ is given by the convolution of two Gaussians which implies that

$$p(z(t)|x(t)) \propto \exp\left\{-\frac{A^2}{4}\left[\mathbf{U}(z(t)) - \mathbf{U}(x(t))\right]^T \Sigma^{-1}\left[\mathbf{U}(z(t)) - \mathbf{U}(x(t))\right]\right\} \tag{22}$$

Now, for small $z(t) - x(t)$,

$$\mathbf{U}(z(t)) - \mathbf{U}(x(t)) \approx \left[z(t) - x(t)\right]\mathbf{U}'(x(t)) \tag{23}$$

So, $p(z(t)|x(t))$ is approximately a Gaussian distribution over $z(t)$,

$$p(z(t)|x(t)) = \mathcal{N}\left(z(t); x(t), \sigma_z\right) \tag{24}$$

Where σ_z is given by

$$\sigma_z = \frac{1}{A(t)}\sqrt{\frac{2}{\mathbf{U}'\Sigma^{-1}\mathbf{U}}} \tag{25}$$

In figure 2 we numerically test whether the distribution $p(z(t)|x(t))$ is Gaussian for independent Gaussian input noise. In panels A and B we show histograms of the maximum likelihood bump position, z, for bumps originally located at position 0 for two different noise settings. In A, the input noise variance is small 0.067 relative to a maximum bump height, U_{max}, of 0.36. In this case, the deflections in the the bump are also small, less than 1 neural position and the distribution looks fairly Gaussian.

In B, we increase the magnitude of the input noise by two orders of magnitude to 0.607, which is almost three times the height of the original bump. Now the deflections are much larger, on the order of tens of neural positions and the distribution seems to have much heavier tails than a Gaussian.

We investigate the properties of these distributions more methodically in panels C and D. In C we plot the standard deviation, σ_z, of the empirical distributions over z as a function of the standard deviation of the noise input, σ_{noise}. The black crosses represent the results of the simulations while the red line is the prediction from equation 25. Clearly, theory and experiment are in good agreement up to $\sigma_{noise} = 1$, which is large compared to the size of the fixed point activity profile.

In panel D we plot the kurtosis of the empirical distributions as a function of input noise standard deviation. For a true Gaussian distribution the kurtosis should be zero. While this is true for σ_{noise} up to about 0.2, the kurtosis increases as the tails of the empirical distributions become heavier at higher values of σ_{noise}.

2.2 Network dynamics

We now consider the effect of this input on the network and determine the extent to which the network dynamics continue to approximate a Kalman filter. In this case we propose an *ansatz* of the form

$$\mathbf{u}(t) = \alpha(t)\mathbf{U}(\hat{x}(t)) + k(t) + \mathbf{n}(t) \tag{26}$$

For some $k(t)$ that is constant over the network and some zero mean noise vector $\mathbf{n}(t)$. Substituting this and the form for the input (equation 16) into the right hand side of the update equation in the main text gives

$$RHS = \frac{w(\mathbf{J}^{sym} + \gamma\mathbf{J}^{asym})\left[\alpha(t)\mathbf{U}(\hat{x}(t)) + k(t) + \mathbf{n}(t)\right]_+}{S + \mu\sum_{i=1}^{N}\left[\alpha(t)U_i(\hat{x}(t)) + k(t) + n_i(t)\right]_+} + A(t)\mathbf{U}(x(t)) + \epsilon(t) \tag{27}$$

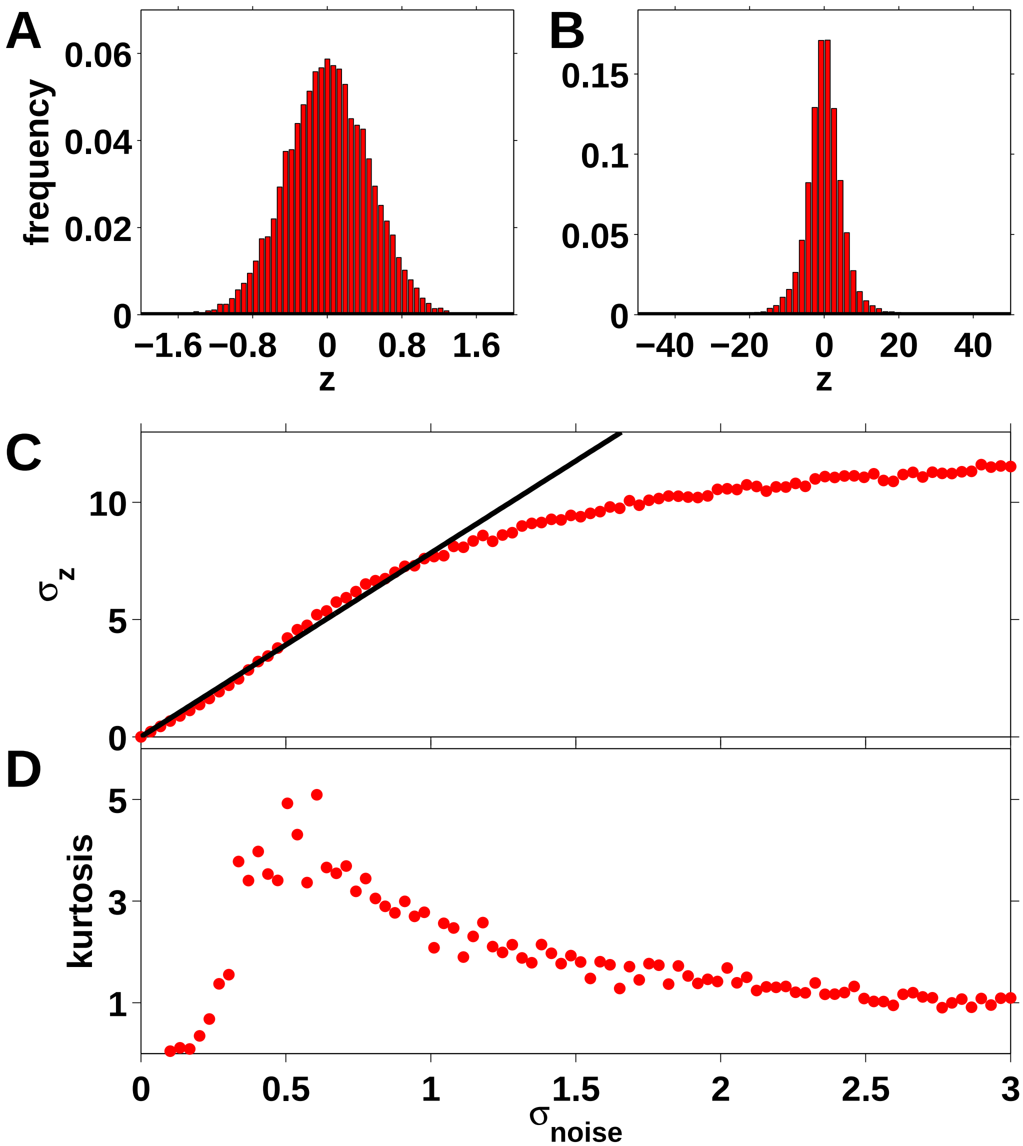

Figure 2: Effect of input noise on the position of the input bump. **A** and **B** Histograms of the decoded position, z, for an input stimulus centered at 0 for two different noise settings. The noise variances are small, $\sigma_{noise} = 0.067$, and large, 0.607, respectively. **C** Standard deviation of the distribution $p(z(t)|x(t))$, σ_z, plotted as a function of the noise variance, σ_{noise}. The red line denotes the theory (from equation 25), while the black crosses are the results of simulations. **D** The kurtosis of the empirical distributions computed from simulations shows that above $\sigma_{noise} = 0.2$, the distributions become heavy tailed.

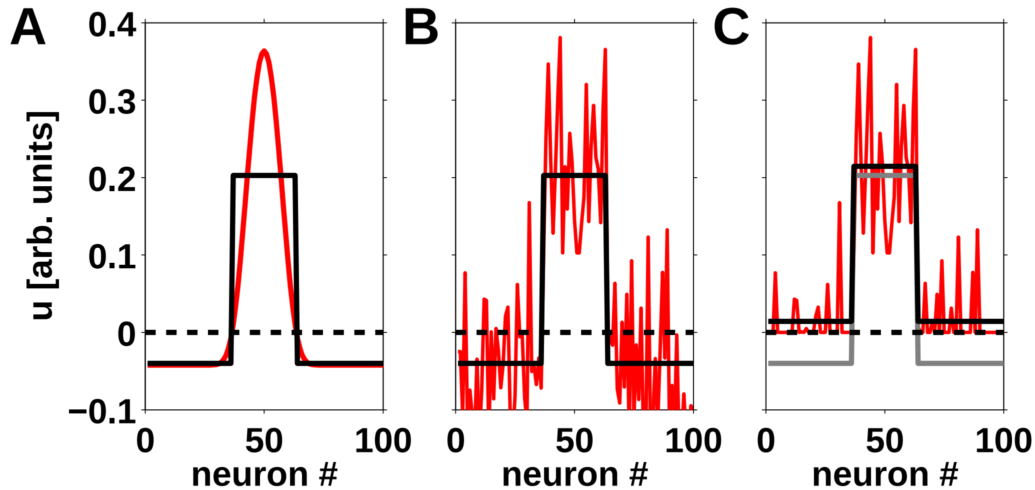

Figure 3: Schematic of the approximation made in section 2.2. **A** The fixed point membrane potential, **U**, (red line) is approximated by a piecewise constant function, equation 29, (solid black line). This approximation splits neural space into two regions, region 1 where $U_i \leq 0$ and region 2 where $U_i > 0$. The threshold for rectification, 0, is highlighted by the dashed black line. **B** The effect of adding noise (red line) to the piecewise constant approximation. **C** After rectification of the noisy signal (red line), the new piecewise constant mean (black line) is higher than the original (grey line). In general, the difference between the new and old means is different in the two regions.

Next we make the approximation

$$[\alpha(t)\mathbf{U}(\hat{x}(t)) + k(t) + \mathbf{n}(t)]_+ \approx g(t)\alpha(t)\left[\mathbf{U}(\hat{x}(t))\right]_+ + k'(t) + \mathbf{n}'(t) \tag{28}$$

For some $g(t)$ and $k'(t)$ that are constant over the network and zero mean noise vector $\mathbf{n}'(t)$. We can get a handle on $g(t)$ and $k'(t)$ in the following way - which is illustrated in figure 3.

We begin by computing an expression for $k'(t)$. To do this, we first approximate $\alpha(t)\mathbf{U}(\hat{x}(t)) + k$ (shown as the red line in figure 3A) in a piecewise constant manner (the solid black line in figure 3A), such that

$$\alpha(t)U_i(\hat{x}(t)) + k \approx \begin{cases} \alpha(t)U_- + k & \text{if } U_i \leq 0 \\ \alpha(t)U_+ + k & \text{if } U_i > 0 \end{cases} \tag{29}$$

where U_+ is the mean value of the components of $\mathbf{U}(\hat{x}(t))$ that are greater than zero, i.e.

$$U_+ = \frac{1}{n} \sum_{i \in \{U_i > 0\}} U_i \tag{30}$$

n is the number of elements in $\mathbf{U}(\hat{x}(t))$ greater than zero. U_- is the average of the elements of $\mathbf{U}(\hat{x}(t))$ that are less than or equal to zero, i.e.

$$U_- = \frac{1}{N-n} \sum_{i \in \{U_i \leq 0\}} U_i \tag{31}$$

This approximation neatly splits the problem into two distinct regions; region 1, where $U_i(\hat{x}(t)) \leq 0$ and region 2, where $U_i(\hat{x}(t)) > 0$.

As illustrated in figure 3C, the process of adding noise and rectifying the signal, effectively increases the mean of the noise *after* rectification (solid black line). The degree to which this occurs is different in each region and we define k'_1 and k'_2 to be the means of the noise distributions in the two different regions after rectification. For large networks these are given by

$$k'_1(t) = \int_{\alpha(t)U_-}^{\infty} \frac{\epsilon}{\sqrt{2\pi}\sigma_{noise}} \exp\left(-\frac{(\epsilon - k - \alpha(t)U_-)^2}{2\sigma_{noise}^2}\right) d\epsilon \tag{32}$$

$$k_2'(t) = \int_{\alpha(t)U_+}^{\infty} \frac{\epsilon}{\sqrt{2\pi}\sigma_{noise}} \exp\left(-\frac{(\epsilon - k - \alpha(t)U_+)^2}{2\sigma_{noise}^2}\right) d\epsilon - \alpha(t)U_+ \tag{33}$$

Solving these integrals gives

$$k_1' = \frac{\sigma_{noise}}{\sqrt{2\pi}} \exp\left(-\frac{(k+\alpha(t)U_-)^2}{2\sigma_{noise}^2}\right) + \frac{k+\alpha(t)U_-}{2} \operatorname{erfc}\left(-\frac{k+\alpha(t)U_-}{\sqrt{2}\sigma_{noise}}\right) \tag{34}$$

and

$$k_2' = \frac{\sigma_{noise}}{\sqrt{2\pi}} \exp\left(-\frac{(k+\alpha(t)U_+)^2}{2\sigma_{noise}^2}\right) + \frac{k+\alpha(t)U_+}{2} \operatorname{erfc}\left(-\frac{k+\alpha(t)U_+}{\sqrt{2}\sigma_{noise}}\right) - \alpha(t)U_+ \tag{35}$$

Finally, we can approximate $k'(t)$ as the weighted sum of these two components, i.e.

$$k'(t) = \left(1 - \frac{n}{N}\right) k_1' + \frac{n}{N} k_2' \tag{36}$$

To compute $g(t)$ we note that within region 2, after rectification, the average activity level is $\alpha(t)U_+ + k_2'$. Equating this to the average activity level in our approximation implies that

$$g(t)\alpha(t)U_+ + k' = \alpha(t)U_+ + k_2' \tag{37}$$

which gives

$$g(t) = 1 + \frac{k_2' - k'}{\alpha(t)U_+} \tag{38}$$

We can now return to the update equation from the main text. Substituting in our approximation we get

$$RHS = \frac{w(\mathbf{J}^{sym} + \gamma\mathbf{J}^{asym})\left(g(t)\alpha(t)\left[\mathbf{U}(\hat{x}(t))\right]_+ + k'(t) + \mathbf{n}'(t)\right)}{S + \mu g(t)\alpha(t)\mathcal{I} + \mu k'(t)N + \sum_{i=1}^N n_i(t)} + A(t)\mathbf{U}(z(t+1)) + \epsilon \tag{39}$$

Now, if we assume that for large N, the noise terms can be safely ignored, i.e.

$$\sum_{i=1}^{N} n_i(t) \approx 0 \tag{40}$$

and

$$\frac{w(\mathbf{J}^{sym} + \gamma\mathbf{J}^{asym})\mathbf{n}'(t)}{S + \mu g(t)\alpha(t)\mathcal{I} + \mu k'(t)N + \sum_{i=1}^N n_i(t)} \approx 0 \tag{41}$$

Then we can write

$$\begin{aligned} RHS &= C'\mathbf{U}(\hat{x}(t+1)) + A(t+1)\mathbf{U}(z(t+1)) + D + \epsilon(t) \\ &\approx (C' + A(t+1))\mathbf{U}\left(\hat{x}(t+1) + \frac{A(t+1)}{C' + A(t+1)}\left(z(t+1) - \hat{x}(t+1)\right)\right) + D + \epsilon_\perp(t) \end{aligned} \tag{42}$$

where

$$C' = \frac{w(S_0 + \mu_0\mathcal{I})g(t)\alpha(t)}{S + \mu g(t)\alpha(t)\mathcal{I} + \mu k'(t)N} \tag{43}$$

and

$$D = \frac{wk'(t)\sum_i J_i^{sym}}{S + \mu g(t)\alpha(t)\mathcal{I} + \mu k'(t)N} \tag{44}$$

Now, the left hand side of the update equation is

$$LHS = \alpha(t+1)\mathbf{U}(\hat{x}(t+1)) + k(t+1) + \mathbf{n}(t+1) \tag{45}$$

Thus, remembering that w is set to obey equation 15 from the main text, we can make the identifications

$$\mathbf{n}(t+1) = \epsilon_\perp \tag{46}$$

$$k(t+1) = D \tag{47}$$

$$\hat{x}(t+1) = \bar{x}(t+1) + \frac{A(t+1)}{C' + A(t+1)}\left(z(t+1) - \hat{x}(t+1)\right) \tag{48}$$

$$\alpha(t+1) = C' + A(t+1)$$
$$= \frac{1}{\left(1 + \frac{\mu k' N}{S}\right)\frac{1}{g(t)\alpha(t)} + \frac{\mu \mathcal{I}}{S}} + A(t+1) \tag{49}$$

These equations are of a similar form to those for the noise free case in section 3 of the main text. However, the correspondence is not exact, in particular,

$$\left(1 + \frac{\mu k' N}{S}\right)\frac{1}{g(t)} \neq 1 \tag{50}$$

and so the height of the bump is not the same as in the noise free case. This also means that the gain term, $\frac{A(t+1)}{C'+A(t+1)}$ is not the same as in the noise free case and therefore, nor is it the same as the equivalent Kalman filter.

However, for low noise settings, $k' \approx 0$ and $g(t) \approx 1$ and so we have

$$\left(1 + \frac{\mu k' N}{S}\right)\frac{1}{g(t)} \approx 1 \tag{51}$$

meaning that $C' \approx C$. Thus for low noise settings we can expect the network to come close to approximating a Kalman filter.

2.3 Discrepancy with Kalman filter

We now investigate exactly how close the network comes to approximating the Kalman filter when the input is noisy and derive an expression for the mean squared error as a function of the magnitude of the noise. As explained above, the reason for this discrepancy between network and Kalman filter is that the scale factor, $\alpha(t)$, in the noisy case (equation 49) is different to that in the noise free case. Thus, the gain factor, $\frac{A(t+1)}{C'+A(t+1)}$, in equation 48 is not the same as the Kalman gain as it was in the noise free case.

In figure 4 we highlight this change in scale factor by computing the fixed point scale factor, α, (in panel A) and noise mean, k, (in panel B) as a function of the noise variance when the input strength is fixed, i.e. $A(t) = 1$. The results of solving equations 49 and 47 correspond to the black line, while the red dots correspond to the results of simulations.

The first thing to note about figure 4 is that α decreases as the variance of the noise increases in both theory and simulation. However, there is a small region at low noise levels where α is essentially constant with respect to σ_{noise}, which suggests that for low noise values, we expect the output of the network to be close to the equivalent Kalman filter.

It is also worth pointing out the similarity between the results of the simulations and the theory in figure 4 - which given the brutality of some of our approximations in the previous section is quite remarkable.

To get a quantitative handle on the magnitude of the error, we make the assumption that the gains in the equivalent Kalman filter and in the network (equation 48) are constant over time. This will be true after long times for the Kalman filter so long as $\sigma_z(t)$ and $\sigma_v(t)$ are constant over time. It will also hold for the neural model at long times so long as $A(t)$ and μ are held constant over time.

If we write G_1 as the long-time gain of the Kalman filter and G_2 for the gain of the neural model, then both models take the form

$$\hat{x}_q(t+1) = \bar{x}_q(t+1) + G_q\left(z(t+1) - \bar{x}_q(t+1)\right) \tag{52}$$

where $q = 1$ or 2 denotes the model number, and $\hat{x}_q(t)$ and $\bar{x}_q(t)$ are the estimate and prediction for model q.

Without loss of generality we assume that the velocity signal is zero, i.e. $v(t) = 0$, in which case we can write

$$\hat{x}_q(t+1) = (1 - G_q)^{t+1}\hat{x}_q(0) + G_q \sum_{i=0}^{t} z(t-i)(1-G_q)^i \tag{53}$$

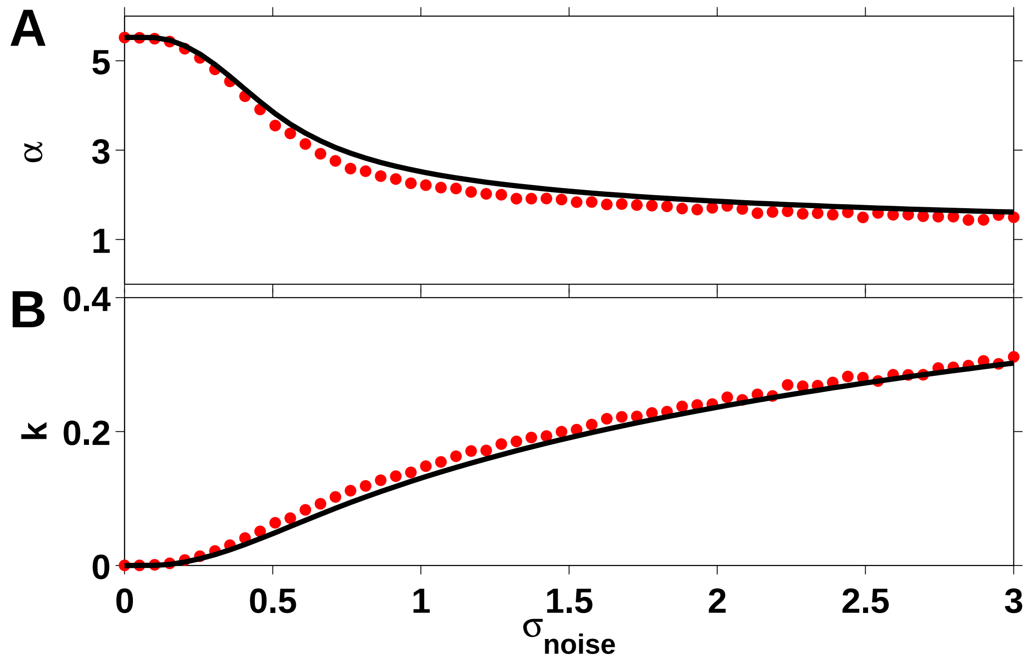

Figure 4: Effect of the variance of input noise, σ_{noise}, on the fixed point values of α in **A** and k in **B**. Simulation results from a network of 100 neurons are shown as red dots. The fixed point solutions of equation 49 and 47 are plotted as the black line. Remarkably, given, the severity of the approximations, there is good agreement between theory and experiment.

If we also assume that the initial estimates are the same, i.e. $\hat{x}_1(0) = \hat{x}_2(0)$, then we can write down the difference between the two models as

$$\begin{aligned} \Delta &= \hat{x}_1(t+1) - \hat{x}_2(t+1) \\ &= \sum_{i=0}^{t} z(t-i) \left[G_1(1-G_1)^i - G_2(1-G_2)^i \right] \end{aligned} \tag{54}$$

We can then average over $\mathbf{z} = \{z(1), z(2), ..., z(t)\}$ given the true input position $\mathbf{x} = \{x(1), x(2), ..., x(t)\}$ to compute the mean and variance of Δ.

$$\langle \Delta \rangle_{\mathbf{z}} = \sum_{i=0}^{t} \langle z(t-i) \rangle_{\mathbf{z}} \left[G_1(1-G_1)^i - G_2(1-G_2)^i \right] \tag{55}$$

where $\langle \cdot \rangle_{\mathbf{z}}$ denotes the average over $p(\mathbf{z}|\mathbf{x})$. For simplicity we assume that the mean of $p(\mathbf{z}|\mathbf{x})$ is zero, thus

$$\langle z(t-i) \rangle_{\mathbf{z}} = 0 \tag{56}$$

and the variance of Δ is given by $\langle \Delta^2 \rangle_{\mathbf{z}}$ which we can compute as

$$\begin{aligned} \langle \Delta^2 \rangle = \sum_{i=0}^{t} \sum_{j=0}^{t} & \langle z(t-i) z(t-j) \rangle_{\mathbf{z}} \\ & \times \left[G_1(1-G_1)^i - G_2(1-G_2)^i \right] \left[G_1(1-G_1)^j - G_2(1-G_2)^j \right] \end{aligned} \tag{57}$$

Assuming that the noise at any two times steps is independent implies that

$$\langle z(t-i) z(t-j) \rangle_{\mathbf{z}} = \delta_{ij} \sigma_z^2 \tag{58}$$

where δ_{ij} is the Kronecker delta. Thus the mean squared error between the two models is

$$\langle \Delta^2 \rangle_{\mathbf{z}} = \sigma_z^2 \sum_{i=0}^{t} \left[G_1(1-G_1)^i - G_2(1-G_2)^i \right]^2 \tag{59}$$

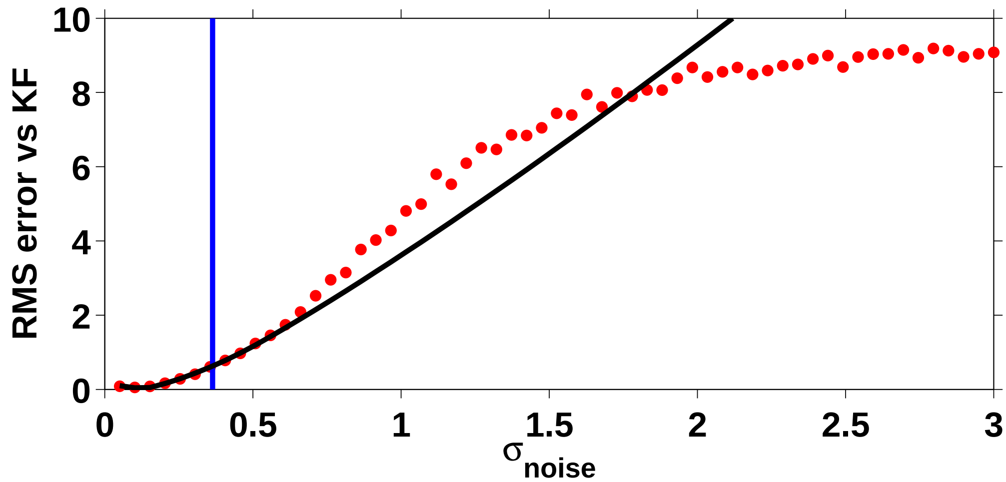

Figure 5: Root mean square difference, $\langle \Delta^2 \rangle$, between the network output and that of the equivalent Kalman filter as a function of input noise in the network. The average results from simulations on a network of 100 neurons are shown as red dots. The black line represents the results of the theory from equation 61. The vertical blue line corresponds to a noise variance equal to the maximum height of $\mathbf{U}$. Thus it is clear to see that the average error is fairly small (less than the increment coded by one neuron) up to quite significant levels of noise and the theory is in excellent agreement with simulation over this range.

Numerically we find that, for large t, this asymptotes to a constant value. Recalling from equation 25 that for independent Gaussian noise we have

$$\sigma_z^2 = \frac{2\sigma_{noise}^2}{A^2 \mathbf{U}'^T \mathbf{U}'} \tag{60}$$

which allows us to write

$$\left\langle \Delta^2 \right\rangle_{\mathbf{z}} = \frac{2\sigma_{noise}^2}{A^2 \mathbf{U}'^T \mathbf{U}'} \sum_{i=0}^{t} \left[G_1(1-G_1)^i - G_2(1-G_2)^i \right]^2 \tag{61}$$

In figure 5 we compare the results of this theory (black line) with simulations (red dots). To help understand the scale, the blue vertical line is placed at the maximum value of the noise-free input, U_{max}. As can be seen, theory and simulation are in excellent agreement for values of σ_{noise} on the order of U_{max} while they diverge quite strongly once the noise gets too high. The reason for this divergence is down to the fact that in this high noise regime (as can be seen in figure 2), the variance of $p(z(t)|x(t))$ is no longer given by equation 25 and thus the input to the equivalent Kalman filter is wrong.

3 Response to change-points (and outliers) - large prediction error case

In this section we derive approximate equations describing the dynamics of the network when the prediction error is large. For simplicity we restrict our discussion to the noise free case.

To get a handle on this analytically we consider inputs of the form

$$\begin{aligned} \mathbf{I}(t) &= A_1(t)\mathbf{U}(z_1(t)) + A_2(t)\mathbf{U}(z_2(t)) \\ &= \sum_{i=1}^{2} A_i(t)\mathbf{U}(z_i(t)) \end{aligned} \tag{62}$$

with $|z_1(t) - z_2(t)| \gg 0$. For general $A_i(t)$, this input is a 'double bump' input with two bumps centered at $z_1(t)$ and $z_2(t)$. However, in our case we will be interested in the case where either $A_1(t) = A(t)$ and $A_2(t) = 0$ or vice versa.

This form of the input captures the two interesting cases of outliers and change-points. For example, if

$$\begin{array}{c} A_1(t) = A(t) \text{ and } A_2(t) = 0 \text{ for } t \neq t_{out} \\ A_1(t) = 0 \text{ and } A_2(t) = A(t) \text{ for } t = t_{out} \end{array} \tag{63}$$

then we have an input that contains an outlier at t_{out}. While an input of the form

$$\begin{array}{c} A_1(t) = A(t) \text{ and } A_2(t) = 0 \text{ for } t < t_{change} \\ A_1(t) = 0 \text{ and } A_2(t) = A(t) \text{ for } t \geq t_{change} \end{array} \tag{64}$$

describes an input with a change-point at time t_{change}.

As before we gain insight into the dynamics of the network by proposing an *ansatz*, this time of the form

$$\mathbf{u}(t) = \sum_{i=1}^{2} \alpha_i(t)\mathbf{U}(\hat{x}_i(t)) \tag{65}$$

Substituting this and the form of the input into the right hand side of the update equation in the main text gives

$$RHS = w[\mathbf{J}^{sym} + \gamma\mathbf{J}^{asym}]\mathbf{f}\left[\sum_{i=1}^{2} \alpha_i(t)\mathbf{U}(\hat{x}_i(t))\right] + \sum_{i=1}^{2} A_i(t)\mathbf{U}(z_i(t)) \tag{66}$$

Now,

$$\begin{aligned} \mathbf{f}\left[\sum_{i=1}^{2} \alpha_i(t)\mathbf{U}(\hat{x}_i(t))\right] &= \frac{\left[\sum_{i=1}^{2} \alpha_i(t)\mathbf{U}(\hat{x}_i(t))\right]_+}{S + \mu\sum_j \left[\sum_{i=1}^{2} \alpha_i(t)U_j(\hat{x}_i(t))\right]_+} \\ &\approx \frac{\sum_{i=1}^{2} \left[\alpha_i(t)\mathbf{U}(\hat{x}_i(t)) + \alpha_{\bar{i}}(t)U_-\right]_+}{S + \mu\sum_{i=1}^{2}\sum_j \left[\alpha_i(t)U_j(\hat{x}_i(t)) + \alpha_{\bar{i}}(t)U_-\right]_+} \end{aligned} \tag{67}$$

where U_- is given in equation 31 and $\bar{i}$ is the 'inverse' of i such that $\bar{i} = 1$ if $i = 2$ and vice versa. Now, if $\alpha_{\bar{i}}U_-$ is small relative to $\alpha_i U_+$, given in equation 30, then we can write

$$\left[\alpha_i(t)\mathbf{U}(\hat{x}_i(t)) + \alpha_{\bar{i}}(t)U_-\right]_+ \approx \left[\alpha_i(t) - \alpha_{\bar{i}}(t)\frac{U_-}{U_+}\right]_+ \alpha_i(t)\left[\mathbf{U}(\hat{x}_i)\right]_+ \tag{68}$$

Thus,

$$RHS \approx \sum_{i=1}^{2} \left[C_i + A_i(t+1)\right]\mathbf{U}\left(\bar{x}_i(t+1) + \frac{A_i(t+1)}{A_i(t+1) + C_i}\left[z_i(t+1) - \bar{x}_i(t+1)\right]\right) \tag{69}$$

where

$$C_i = \frac{w(S_0 + \mu_0\mathcal{I})\left[\alpha_i - \alpha_{\bar{i}}\frac{U_-}{U_+}\right]_+}{S + \mu\mathcal{I}\left(\left[\alpha_1 - \alpha_2\frac{U_-}{U_+}\right]_+ + \left[\alpha_2 - \alpha_1\frac{U_-}{U_+}\right]_+\right)} \tag{70}$$

Equation 69 has the same form form of the *ansatz* and we can make the identification

$$\alpha_i(t+1) \approx C_i + A_i(t+1) \tag{71}$$

$$\hat{x}_i(t+1) \approx \bar{x}_i(t+1) + \frac{A_i(t+1)}{A_i(t+1) + C_i}\left[z_i(t+1) - \bar{x}_i(t+1)\right] \tag{72}$$

Thus the output of the network is of the form of two bumps of activity whose positions approximately implement independent Kalman filters. These equations, in particular equation 71, are used to give the theoretical results in figure 4D of the main text.