Functional, but not anatomical separation of "what" and "when" in prefrontal cortex

Supplementary material

Christian K. Machens, Ranulfo Romo, and Carlos D. Brody

September 7, 2009

Contents

1 Notation

$\langle\cdot\rangle$	averaging over both time t and frequency f
A	mixing or transition matrix of the subspace dynamics
$\mathbf{b}$	external inputs for subspace dynamics
C	covariance matrix
C_{ij}	$(i,j)-$th entry in covariance matrix C
$\mathbf{E}$	external inputs to the linear network model
f	index for the frequency of the first stimulus, runs from $1\ldots N_f$
N	number of neurons in data set
N_z	dimension of subspace, in our case $N_\mathrm{z}=6$
N_t	number of time points
N_f	number of frequencies
$q_k(t,f)$	signal part of the firing rate of the k-th neuron
$r_k^i(t,f)$	firing rate of the k-th neuron in the i-th trial
$\bar{r}_k(t,f)$	trial-averaged firing rate of the k-th neuron (smooth PSTH)
$\bar{\mathbf{r}}(t,f)$	trial-averaged firing rates of all neurons
$\mathbf{r}(t)$	simulated firing rates
U	orthogonal matrix describing the PCA coordinate transform
U_z	the first N_z columns of U
W	connectivity matrix of the linear network model
t	time index, runs from $1\ldots N_t$
$\bar{z}_k(t,f)$	k-th principal component — $\bar{\mathbf{r}}(t,f)$ projected onto the k-th principal axis
$\bar{\mathbf{z}}(t,f)$	the first N_z principal components — $\bar{\mathbf{r}}(t,f)$ projected into the N_z-dimensional subspace spanned by the first N_z principal axes
$\mathbf{z}(t)$	simulated firing rates within subspace
$\eta_k^i(t,f)$	noise part of the firing rate of the k-th neuron in the i-th trial
$\eta_k(t,f)$	noise part of the trial-averaged firing rate of the k-th neuron
H	covariance matrix of the noise
H_{ij}	$(i,j)-$th entry in covariance matrix of the noise

2 Principal component analysis

We will assume that the reader is familiar with the basics of principal component analysis (PCA; see e.g. Hyvärinen et al., 2001; Hastie et al., 2001) and will only focus on the specifics of how we applied PCA to our data set. As described above, the time-varying firing rate of the neurons is estimated as $\bar{r}_k(t,f)$ where the index $k = 1 \ldots N$ denotes the number of the neuron, the index $t = 1 \ldots N_t$ the point in time, and $f = 1 \ldots N_f$ the stimulus frequency used. The bar serves as a reminder that this firing rate is estimated by averaging over trials. We obtain the mean firing rate of neuron k during the delay period as

$$\begin{aligned} \bar{r}_k &= \frac{1}{N_t N_f} \sum_{t=1}^{N_t} \sum_{f=1}^{N_f} \bar{r}_k(t,f) \\ &= \langle \bar{r}_k(t,f) \rangle \quad . \end{aligned}$$

We will use the angular brackets as a short-hand notation for averages over time points and frequencies. The covariance matrix of the neural firing rates then becomes

$$C_{ij} = \left\langle \left[\bar{r}_i(t,f) - \bar{r}_i \right] \left[\bar{r}_j(t,f) - \bar{r}_j \right] \right\rangle \tag{1}$$

which can be diagonalized to yield

$$C = U \Lambda U^T$$

where U is an orthogonal matrix and Λ a diagonal matrix whose entries are the eigenvalues of the covariance matrix. We will assume that these eigenvalues are sorted in reverse order so that the largest eigenvalue corresponds to the first entry in Λ (see Figure 3A).

The columns of the matrix U contain the basis vectors of a new coordinate system whose axes we will call the *principal axes* whereas the coordinates of the data in the new system are called the *principal components*.[1] If the first N_z principal components capture the bulk of the variance in the data, then the data are well described by a projection of the trajectories into the respective N_z-dimensional subspace. If we define the matrix U_z to correspond to the first N_z columns of U (which means that U_z has the size $[N \times N_z]$ and its transpose, ${U_z}^T$, the size $[N_z \times N]$), then

$$\bar{\mathbf{z}}(t,f) = {U_z}^T \left(\bar{\mathbf{r}}(t,f) - \bar{\mathbf{r}} \right) \quad .$$

represents the trajectories within the new N_z-dimensional coordinate system. Here, $\bar{\mathbf{r}}(t,f) = \big((r_1(t,f), \ldots, r_N(t,f) \big)$ denotes the vector of time-varying firing rates. Note that the projected trajectories are centered with respect to the coordinate origin since the vector of

[1] There seems to be some confusion in the literature what the term "principal components" should refer to. Here we follow Hyvärinen et al. (2001) so that a principal axis refers to a (unit) basis vector, i.e., a direction in state space, whereas e.g. the first principal component refers to the coordinate of the data points along the first principal axis, and so on.

mean firing rates, $\bar{\mathbf{r}} = (\bar{r}_1, \ldots \bar{r}_N)$, has been subtracted. Consequently, the principal components, i.e., the coefficients of $\bar{\mathbf{z}}(t,f)$, denoted by $\bar{z}_k(t,f)$, can take both negative and positive values (compare Figure 3B and Supplementary Figure 2a).

To the extent that this projection captures most of the variance in the data, we can reconstruct the activity of each neuron using

$$\bar{\mathbf{r}}_{\text{est}}(t,f) = U_{\text{z}}\bar{\mathbf{z}}(t,f) + \bar{\mathbf{r}} \approx \bar{\mathbf{r}}(t,f) \quad .$$

For every neuron this yields a set of N_{z} reconstruction coefficients which correspond to the rows of U_{z}. If we denote the reconstruction coefficients of the k-th neuron as $\mathbf{u}_k$, i.e., the k-th row of U_{z}, then the set of all reconstruction coefficients constitutes a cloud of points in an $N_{\text{z}} = 6$-dimensional space. Some marginal distributions of this cloud, a two-dimensional and two one-dimensional projections, are shown in Figure 5.

If the neurons were to fall into different groups or categories, then the points $\mathbf{u}_k$ should fall into different clusters as well. Such a scenario can be tested by the random projection method: if such a clustering exists, then random projections of the N_{z}-dimensional space onto one or two dimensions should eventually uncover some of the clusters. However, random projections of the reconstruction coefficients always show a single cloud of points (see Figure 5 and Supplementary Figure 1). Accordingly, we could not find any evidence for separate classes of neurons and therefore conclude that the activities of individual neurons are random combinations of the N_{z} components.

3 Determining the noise floor

Because the firing rates $\bar{\mathbf{r}}(t,f)$ are estimated by averaging over a finite number of trials, they will fluctuate partly through random, non-task locked activity. In turn, these random variations will influence or bias the PCA (see e.g. Sengupta and Mitra, 1999).

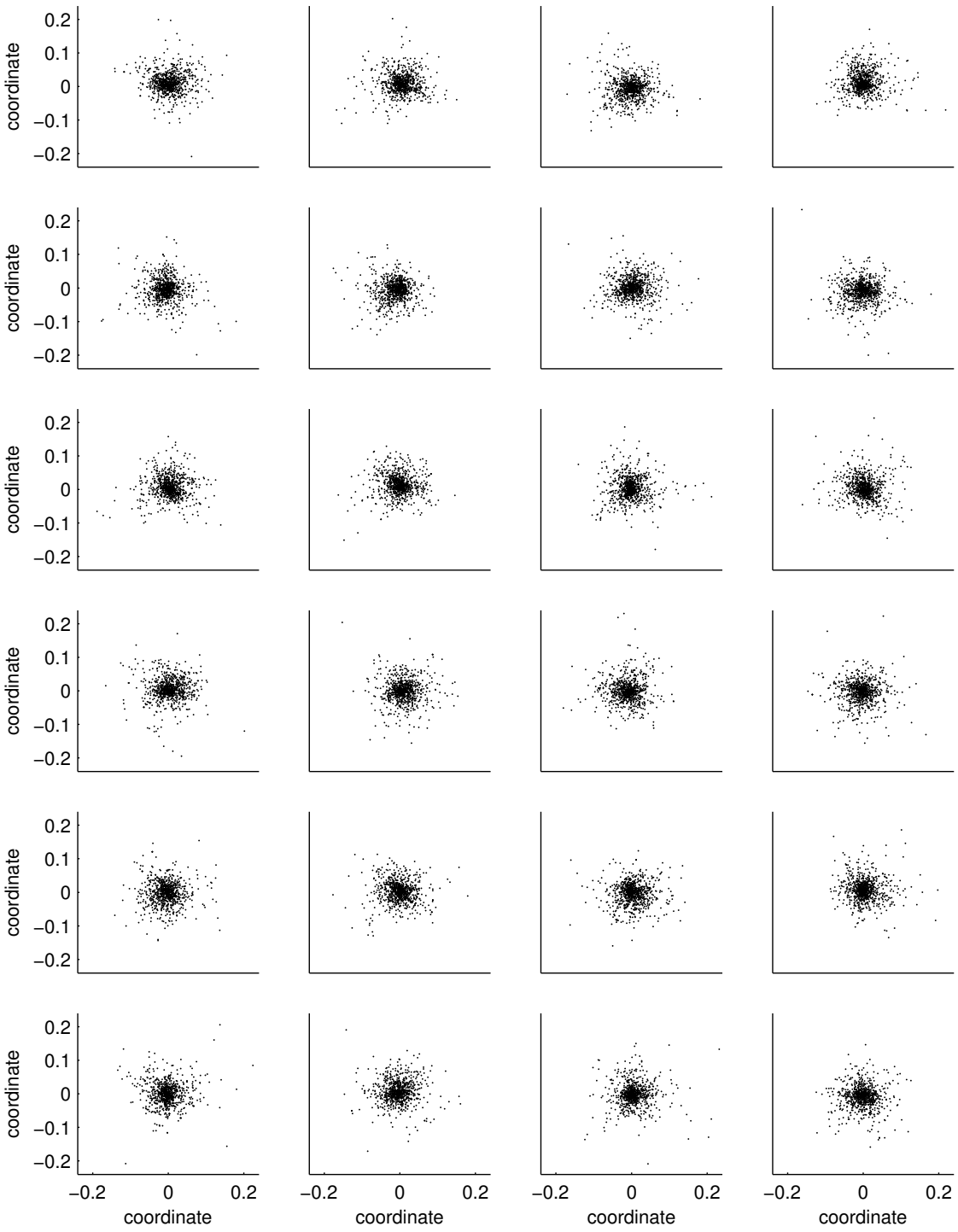
To make these considerations explicit, let us assume that the firing rate of the $k-$th neuron in trial i is determined by some "signal" rate $q_k(t,f)$ and a variable "noise" term $\eta_k^i(t,f)$ that will change from trial to trial,

$$r_k^i(t,f) = q_k(t,f) + \eta_k^i(t,f) \quad .$$

The noise term here incorporates all sources of trial-to-trial variability, be they through the stochastic nature of individual spike trains or through systematic, but uncontrolled-for changes from trial to trial. When averaging over N trials, we obtain

$$\bar{r}_k(t,f) = q_k(t,f) + \bar{\eta}_k(t,f)$$

where we define $\bar{\eta}_k(t,f)$ as the average over N instantations of the noise term $\eta_k^i(t,f)$. Since the overall mean rate $\bar{r}_k$ is given by averaging over time points t and frequencies f,



Supplementary Figure 1: Random projections of the reconstruction coefficients into two dimensions suggest a unimodal distribution of neural activities.

we obtain

$$\begin{aligned}\bar{r}_k &= \langle q_k(t,f)\rangle + \langle \bar{\eta}_k(t,f)\rangle \\ &= q_k + \bar{\eta}_k \quad .\end{aligned}$$

None of the above assumptions are likely to be correct on a strict basis; the noise will not be additive, and it will most likely not be stationary (e.g. it may increase or decrease during the delay period). They are, however, useful approximations to illustrate the kind of problems that PCA is facing when confronted with noisy data.

Under the given assumptions, the estimation of the covariance matrix, Equation (1), also decomposes into a signal and a noise part,

$$\begin{aligned}C_{ij} &= \left\langle \left[\bar{r}_i(t,f) - \bar{r}_i\right]\left[\bar{r}_j(t,f) - \bar{r}_j\right]\right\rangle \\ &= \left\langle \left[q_i(t,f) + \bar{\eta}_i(t,f) - q_i - \bar{\eta}_i\right]\left[q_j(t,f) + \bar{\eta}_j(t,f) - q_j - \bar{\eta}_j\right]\right\rangle \\ &\approx \left\langle \left[q_i(t,f) - q_i\right]\left[q_j(t,f) - q_j\right]\right\rangle + \left\langle \left[\bar{\eta}_i(t,f) - \bar{\eta}_i\right]\left[\bar{\eta}_j(t,f) - \bar{\eta}_j\right]\right\rangle \\ &= Q_{ij} + H_{ij}\end{aligned}$$

where the last row defines the signal covariance matrix Q_{ij} and the noise covariance matrix H_{ij}. The approximation sign indicates a step in which we neglect any covariations between signal and noise that will occur due to the finite length of the time series. These remaining covariations are much smaller than the covariations in the noise or signal terms.

Ideally, we would want to carry out PCA on the signal covariance matrix Q_{ij}. Since most of the cells were not recorded at the same time, however, we cannot compute Q_{ij}, since we do not know the exact noise covariance matrix H_{ij}. To nevertheless investigate the influence of noise on the PCA results, we constructed putative noise matrices based on the simplifying assumption that neurons are firing independently.[2] To generate representative noise traces from the data, we notice that the difference between two single-trial firing rates of a neuron yields

$$r_i^k(t,f) - r_i^l(t,f) = \eta_i^k(t,f) - \eta_i^l(t,f) \quad . \tag{2}$$

Since $\eta_i^k(t,f)$ and $\eta_i^l(t,f)$ have the same statistical properties, then by the rules of adding and subtracting random terms, the right hand side evaluates to

$$\eta_i^k(t,f) - \eta_i^l(t,f) = \sqrt{2}\eta_i^m(t,f) = \sqrt{2M}\bar{\eta}_i(t,f) \tag{3}$$

[2]Some of the neurons have been recorded simultaneously; in these cases, we can estimate the noise correlations between neurons on a trial-to-trial basis. In most of the neuron pairs investigated, we did not find any significant noise correlations. When the data are pooled over all neuron pairs, however, very weak noise correlations between neuron pairs become visible (data not shown; but see Machens et al., 2005, Brody et al, in preparation).

where the last equality follows since the noise on the PSTH estimates, $\bar{\eta}_i(t,f)$, scales with a factor of $1/\sqrt{M}$ with respect to the single-trial-noise. We can therefore generate traces of the "average" noise $\bar{\eta}_i(t,f)$ through the relation

$$\bar{\eta}_i(t,f) = \frac{1}{\sqrt{2M}}\left[r_i^k(t,f) - r_i^l(t,f)\right] \quad \text{for } l \neq k \quad .$$

In turn, we can compute an estimate of the noise covariance matrix,

$$H_{ij} = \left\langle \left[\bar{\eta}_i(t,f) - \bar{\eta}_i\right]\left[\bar{\eta}_j(t,f) - \bar{\eta}_j\right] \right\rangle$$

Diagonalization of this noise covariance matrix yields the eigenvalues shown in the black dots in Figure 3A,B — if there were no signal in the data, then this would be the result of PCA. (Note that several repetitions of this procedure also allow us to estimate error bars on the eigenvalues of H_{ij}. We did not display them in Figure 3A,B because they were generally too small to be visible.)

Subsequently, we can use our estimates of the noise to state how much *signal* variance the principal components capture. (Remember that PCA based on C_{ij} captures variance generated by both signal and noise.) If we denote the eigenvalues of the covariance matrix C_{ij} as $\lambda_{C,i}$, then the *total* variance captured by the first N_z principal components is

$$\begin{aligned}\sum_{i=1}^{N_z} \lambda_{C,i} &= \sigma_{\text{total}}(N_z) \\ &= \sigma_{\text{signal}}(N_z) + \sigma_{\text{noise}}(N_z) \quad ,\end{aligned}$$

where $\sigma_{\text{signal}}(N_z)$ and $\sigma_{\text{noise}}(N_z)$ denote the fractions of the total variance that are due to signal and noise, respectively. Since we do not know $\sigma_{\text{noise}}(N_z)$, we cannot give a precise estimate of $\sigma_{\text{signal}}(N_z)$. We can, however, estimate an upper bound on the noise variance, $\sigma_{\text{noise}}(N_z)$, by performing principal component analysis on the noise traces, i.e., by diagonalizing the covariance matrix H_{ij}. If $\lambda_{H,i}$ denotes the i-th largest eigenvalue of H_{ij}, then

$$\sum_{i=1}^{N_z} \lambda_{H,i} = \sigma_{\text{bound}}(N_z) \geq \sigma_{\text{noise}}(N_z) \quad ,$$

i.e., the N_z-dimensional subspace found by PCA on the noise traces will always capture more noise variance than any other N_z-dimensional subspace. Consequently, the total amount of signal variance captured by the first N_z components is bound from below by

$$\sigma_{\text{signal}}(N_z) \geq \sigma_{\text{total}}(N_z) - \sigma_{\text{bound}}(N_z) \quad .$$

For normalization, we divide this number by the total amount of signal variance, which is given by

$$\begin{aligned}\sigma_{\text{signal}}(N) &= \sigma_{\text{total}}(N) - \sigma_{\text{noise}}(N) \\ &= \text{tr}C - \text{tr}H \quad .\end{aligned}$$

The percentages given in Figure 3 and Figure 4 are the normalized lower bounds on the captured signal variance as determined by the above formulae.

As the number of dimensions N_z included increases, so will the captured signal variance. We decided to cut-off the dimensionality at the point when at least 95% of the signal variance variance have been captured (as determined by the lower bound above). To make the analysis comparable across monkeys, we required this criterion to be fulfilled *on average*, which led to a cut-off at $N_z = 6$ dimensions. Specifically, we obtained $\sigma_{\text{signal}}(6) = 92.9\%$ for monkey *RR*013, $\sigma_{\text{signal}}(6) = 95.9\%$ for monkey *RR*014, $\sigma_{\text{signal}}(6) = 97.3\%$ for monkey *RR*015, and $\sigma_{\text{signal}}(6) = 96.5\%$ for monkey *RR*013 with 6 second delay period. Averaged over data sets, we obtain $\langle \sigma_{\text{signal}}(N_z) \rangle = 95.7\%$.

4 Difference-of-Covariances (DOC) method: Separating time and frequency

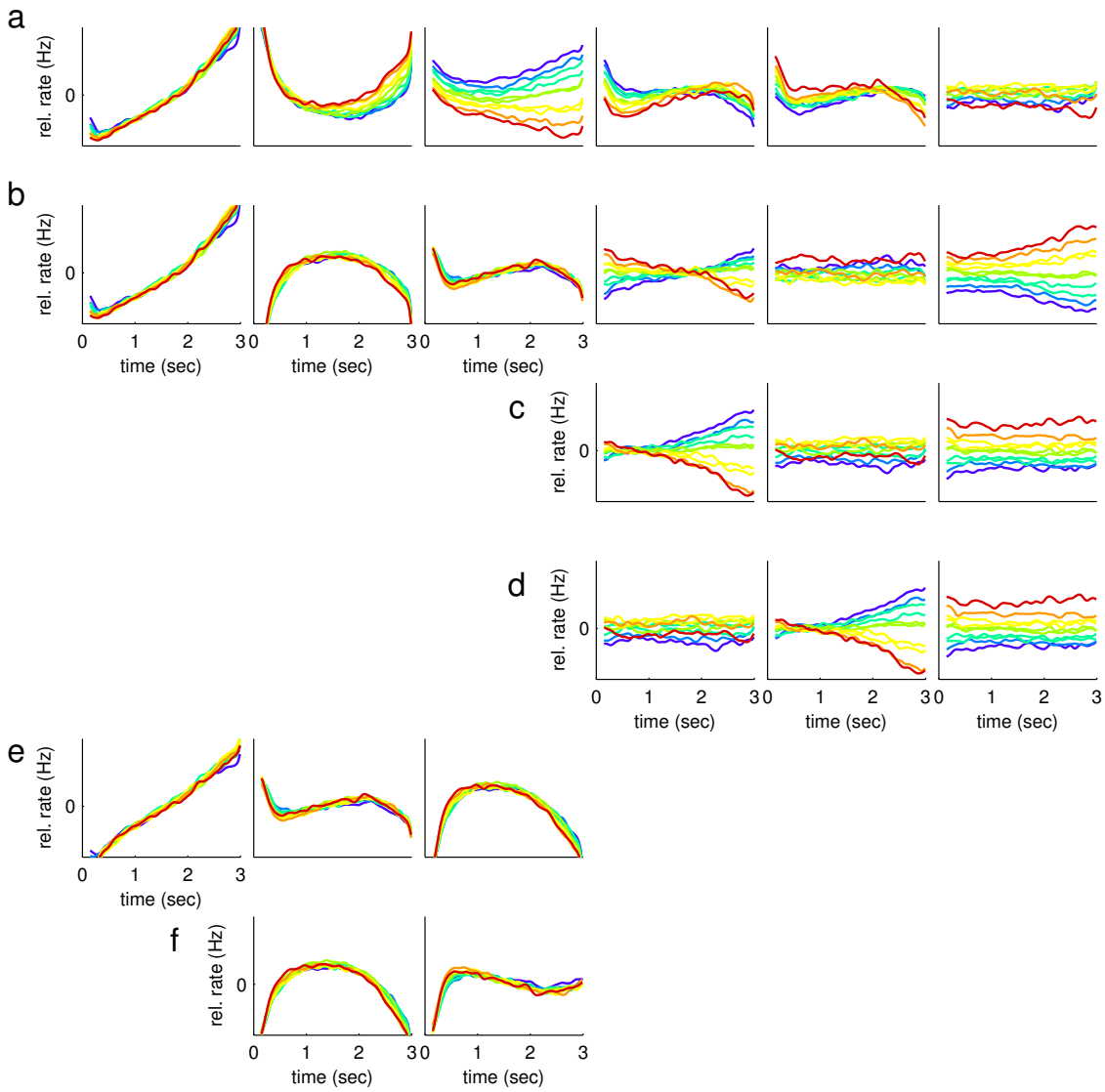
Principal component analysis orders the axes of the new coordinate system by the amount of variance they capture in the data. In our case, variance comes about through any variations in the firing rates of the neurons. Geometrically, we envision variance as the spread of several trajectories — each corresponding to a different stimulus frequency f — in state space. With this intuition in mind, we can distinguish two sources of firing rate variance: (1) Any variance due to changes of the system's state over time, independent of the frequencies f, and (2) any variance due to differences in the system's state across frequencies, yet independent of time.

These two sources of variance could be mixed up in any imaginable way. They could, however, also fall into different subspaces in which case they should be separable by *linear* methods. To investigate this possibility, we performed additional coordinate transformations within the 6-dimensional subspace.

4.1 DOC method in $n = 6$ dimensions.

(a) Principal component analysis

The results from the original PCA are replotted in Supplementary Figure 2a, as taken from Figure 3D.



Supplementary Figure 2: Coordinate transformations that lead from Figure 3B to Figure 4A. All y-axes have the same scaling; their sign, however, is arbitrary. (a) Results from PCA, replotted from Figure 3B. Panels are sorted by their associated eigenvalue. (b) Results from DOC, which separates the time and frequency-components. Panels are sorted according to eigenvalue *in reverse order*, i.e., largest eigenvalue on the right. (c) An additional coordinate transformation in the memory subspace orders axes according to temporal dependencies *within* that subspace. Panels are sorted according to eigenvalue. (d) Re-ordering of panels in (c) by their frequency-dependent variance. (e) Extraction of the optimal linear read-out of time (slight change in first component). (f) Ordering of the remaining two axes by the amount of variance during the last third of the delay period.

(b) DOC method: Separating time and frequency components

To investigate the potential linear separability of the time- and frequency-dependencies, we sought to construct a coordinate system in which coordinates are ordered by the amount of time-dependent or frequency-dependent variance. The time-dependent variance can be measured by averaging over frequencies,

$$\bar{\mathbf{z}}_f(t) = \frac{1}{N_f} \sum_{f=1}^{N_f} \bar{\mathbf{z}}(t,f) \quad , \tag{4}$$

and the frequency-dependent variance by averaging over time points,

$$\bar{\mathbf{z}}_t(f) = \frac{1}{N_t} \sum_{t=1}^{N_t} \bar{\mathbf{z}}(t,f) \quad . \tag{5}$$

To find the desired coordinate system, we subtracted the respective covariance matrices,

$$S_{ij} = \left\langle \bar{z}_{f,i}(t)\bar{z}_{f,j}(t) \right\rangle - \left\langle \bar{z}_{t,i}(f)\bar{z}_{t,j}(f) \right\rangle, \qquad i,j = 1\ldots 6$$

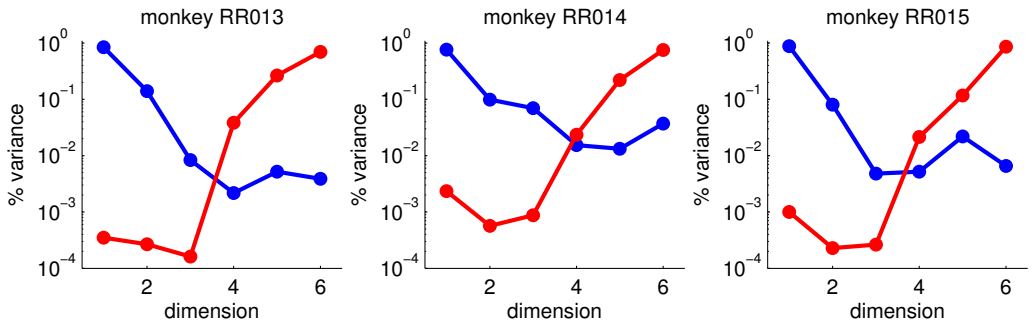
and performed an eigenvalue decomposition on this difference matrix. Note that S_{ij} is symmetric, but no longer positive definite, and hence eigenvalues can be both positive and negative. Due to the competing influence of the two covariance matrices (one positive, one negative), the new coordinate system will order coordinates by their contribution to the time-dependent variance and, in reverse order, by their contribution to the frequency-dependent variance.

The eigenvalue decomposition yields a $[6 \times 6]$ orthogonal matrix U_b which allows to re-represent the original projected data, $\bar{\mathbf{z}}(t,f)$, so that

$$\bar{\mathbf{z}}_b(t,f) = U_b \bar{\mathbf{z}}(t,f) \quad .$$

The results are shown in Supplementary Figure 2b and illustrate graphically a separation between time components (panels 1–3) and frequency (or memory-related) components (panels 4–6).

Aside from the visual inspection, we can also quantify to what extent the new coordinate system "separates" time- and f-information. The results are shown in Supplementary Figure 3. Here the red curves indicate the amount of variance that each coordinate captures with respect to differences between stimuli f, i.e., variance as measured through $\bar{\mathbf{z}}_t(f)$, and the blue curves indicate the amount of variance each coordinate captures with respect to changes over time, i.e., variance as measured through $\bar{\mathbf{z}}_f(t)$.



Supplementary Figure 3: Separation of variance by its sources. Red curves indicate the percentage of variance captured about f1-dependent differences between trajectories, blue curves indicate the percentage of variance due to the temporal evolution of trajectories. The curves show that most of the power through temporal changes lies in coordinates 1–3, most of the power through f-dependent changes lies in coordinate 4–6.

(c,d) Reading out f1: A coordinate system for the "what" space

While the above procedure yielded the desired separation of time- and frequency-dependent variation, we were particularly interested in seeing whether there are axes along which we can obtain read-outs of time and frequency.

In a first step, we focused on the 3-dimensional subspace given by components 4–6 in Supplementary Figure 2b. To find a read-out of frequency that is constant in time, we carried out an additional PCA that orders axes according to any remaining time variations within individual trajectories.[3] We define

$$\bar{\mathbf{z}}_x(t,f) = \bar{\mathbf{z}}_b(t,f) - \frac{1}{N_t}\sum_{t=1}^{N_t} \bar{\mathbf{z}}_b(t,f) \quad ,$$

so that the respective covariance matrix in the subspace given by components 4–6 will measure the variance due to time variations in invididual trajectories,

$$C_{c,ij} = \left\langle \bar{z}_{x,i+3}(t,f)\bar{z}_{x,j+3}(t,f) \right\rangle \qquad i,j = 1\ldots 3 \quad .$$

The eigenvalue decomposition of this covariance matrix yields one large eigenvalue (89.8% of the sum total) and two small eigenvalues (5.6% and 4.5% of the sum total), showing that there is one dimension along which individual trajectories vary strongly in time and two dimensions along which the trajectories vary only weakly in time. PCA also returns

[3] Note that components 1–3 capture only temporal variations that are common to *all* trajectories, whereas here we seek to extract temporal variations that are f-dependent, and therefore *individual* to each trajectory.

a $[3 \times 3]$ orthogonal matrix V_c. If I_3 is the $[3 \times 3]$ identity matrix, then we define an $[6 \times 6]$ orthogonal matrix by

$$U_c = \begin{pmatrix} I_3 & 0 \\ 0 & V_c \end{pmatrix}$$

which allows us to re-represent the data according to

$$\bar{\mathbf{z}}_c(t,f) = U_c U_b \bar{\mathbf{z}}(t,f) \quad .$$

The results are shown in Supplementary Figure 2c, and, re-ordered by their frequency-dependent variance, in Supplementary Figure 2d. (Note that we only plot the coordinates that have changed.)

(e,f) Reading out t: A coordinate system for the "when" space

In a second step, we focused on the subspace spanned by the first three dimensions (1–3) and explicitly searched for the axis with respect to which all firing rates rise or fall as linearly as possible. The covariance matrix in the 3-dimensional subspace is given by

$$C_{d,ij} = \left\langle \bar{z}_{c,i}(t,f) \bar{z}_{c,j}(t,f) \right\rangle \qquad i,j = 1 \ldots 3 \quad .$$

When we project all trajectories onto a centered straight line, $f(t) = t - N_t/2$, we obtain

$$x_i = \left\langle \bar{z}_{c,i}(t,f)(t - N_t/2) \right\rangle \qquad i,j = 1 \ldots 3 \quad .$$

The optimal axis can be found using linear regression which in this case yields

$$\mathbf{a}_1 = C_d^{-1} \mathbf{x} \quad .$$

We normalize this basis vector, $\hat{\mathbf{a}}_1 = \mathbf{a}_1 / ||\mathbf{a}_1||$, and then use it as a building block of a 3-dimensional orthonormal basis $\{\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \hat{\mathbf{a}}_3\}$, which in turn can be used to construct an orthogonal transform V_d. Using the same procedure as before, we obtain yet another 6-dimensional coordinate transform matrix,

$$U_d = \begin{pmatrix} V_d & 0 \\ 0 & I_3 \end{pmatrix}$$

where I_3 is once more an $[3 \times 3]$ diagonal matrix. With this new coordinate system, the data are given by

$$\bar{\mathbf{z}}_d(t,f) = U_d U_c U_b \bar{\mathbf{z}}(t,f)$$

and the respective results are shown in Supplementary Figure 2e.

The last step leaves axes 2 and 3 undetermined since they are simply given by a (random) orthogonalization procedure. To make a specific choice of axes, we decided to order them by the variance they capture about the last third of the delay period,

$$C_{e,ij} = \left\langle \bar{z}_{d,i+1}(t,f)\bar{z}_{d,j+1}(t,f) \right\rangle_{t\in[2N_t/3,Nt]} \qquad i,j = 1,2 \quad ,$$

where the averaging over time here does not include the complete delay period, but only the last third. Eigenvalue decomposition yields an orthogonal transform V_e which can be used to construct a 6-dimensional orthogonal transform

$$U_e = \begin{pmatrix} 1 & 0 & 0 \\ 0 & V_e & 0 \\ 0 & 0 & I_3 \end{pmatrix}$$

The final representation of the data are therefore obtained via

$$\bar{\mathbf{z}}_f(t,e) = U_e U_d U_c U_b \bar{\mathbf{z}}(t,f)$$

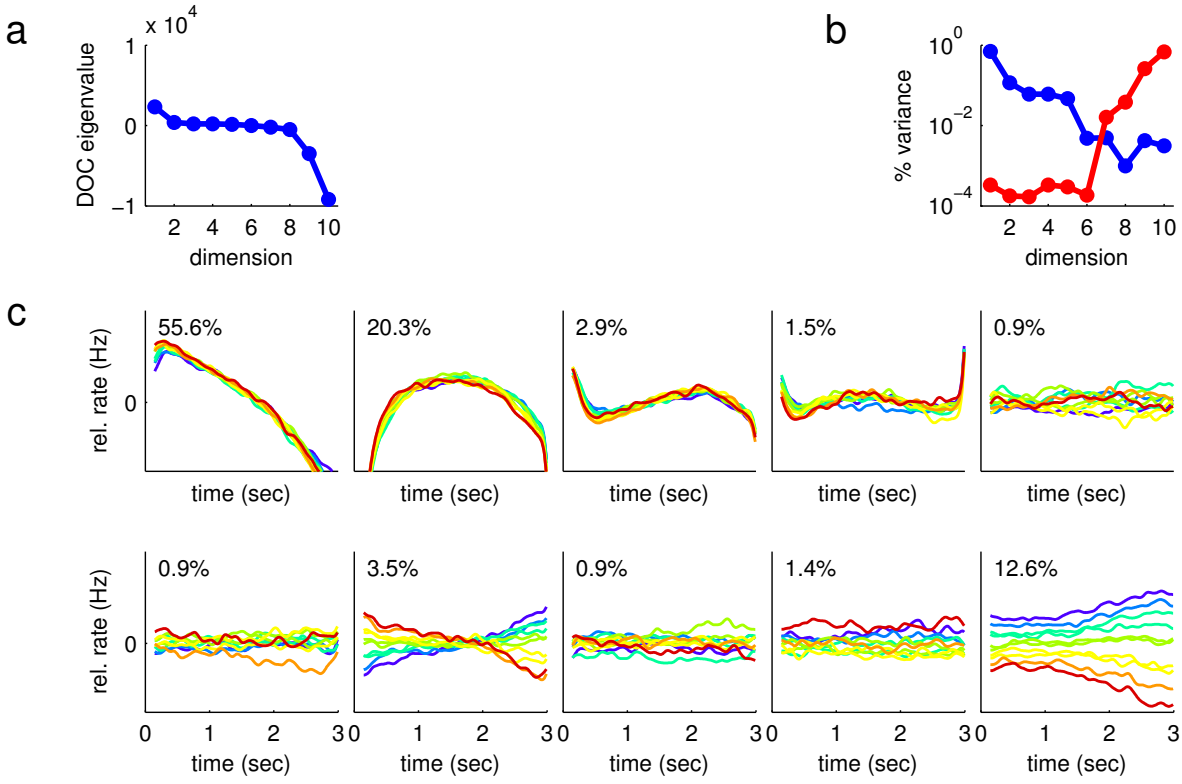
and the respective results are shown in Supplementary Figure 2f. Compare this to the coordinates shown in the main article, Figure 4B.

4.2 DOC method in $n > 6$ dimensions.

The DOC method *only* separates frequency or time-dependent variance, it does not *per se* reduce dimensionality similar to PCA. As a consequence of our definition of time- and frequency-dependent variance, Equation (4) and Equation (5), there could be components with little time- or frequency-dependent variance that still contribute significantly to the actual firing rate variance. This is indeed the case and illustrated in Supplementary Figure 4 where we redid the analysis with a PCA preprocessing step that reduces the dimensionality to $n = 10$ rather than $n = 6$. When sorting components by their eigenvalues (see Supplementary Figure 4a), then one of the original components (here second row, second panel) has a relatively small eigenvalue since it has little contribution to either time or frequency-dependent variance (see Supplementary Figure 4b). However, it does have significant contributions to the overall variance. Similar results hold when the dimensionality n is increased even further. The PCA preprocessig step ensures that a low-dimensional projection is found *before* time- and frequency-dependent variances are separated into different subspaces.

4.3 A counter example

We emphasize that whether or not the DOC method finds a separation of time and frequency variance depends on the data set. There are many ways in which such a separation



Supplementary Figure 4: DOC method applied after an initial PCA preprocessing step that reduces the dimensionality to $n = 10$ (instead of $n = 6$). (a) Eigenvalues from the DOC method. Positive eigenvalues capture time-dependent variance, negative eigenvalues frequency-dependent variance. (b) Percentage of variance captured about frequency and time-dependent variance (compare Supplementary Figure 3). (c) Components from the DOC method (without further processing; compare Supplementary Figure 2b).

could fail. To provide some intuition about that, we constructed a simple counter example in Supplementary Figure 5. We designed a system with an intrinsically two-dimensional dynamics as shown in the central panel. Individual neural responses are random one-dimensional projections of the two-dimensional population dynamics and several examples are shown on the left. Note that their firing rates change both with time and frequency.

When applied to this simulated data set, the DOC method finds two significant components which are shown in the right panels in Supplementary Figure 5. These components do not unmix the temporal and frequency variance.[4] This is to be expected since, by design, there is no temporal component with respect to which the dynamics could be read out linearly. In other words, there is no (linear, cartesian) coordinate along which the firing rates change with time only; such a read-out would have to be non-linear and follow the circular movement of the trajectories.

5 Estimating the parameters of the linear network model

For the network model, we will assume that the data are sampled in time steps of 100 msec. The number of time steps is then $N_t = 30$ for a delay period of 3 seconds. Given the time index $t = 1 \ldots N_t$, we assume that the firing rates of the individual neurons can be generated through a linear model of the form

$$\mathbf{r}(t) = W\mathbf{r}(t-1) + \mathbf{E}$$

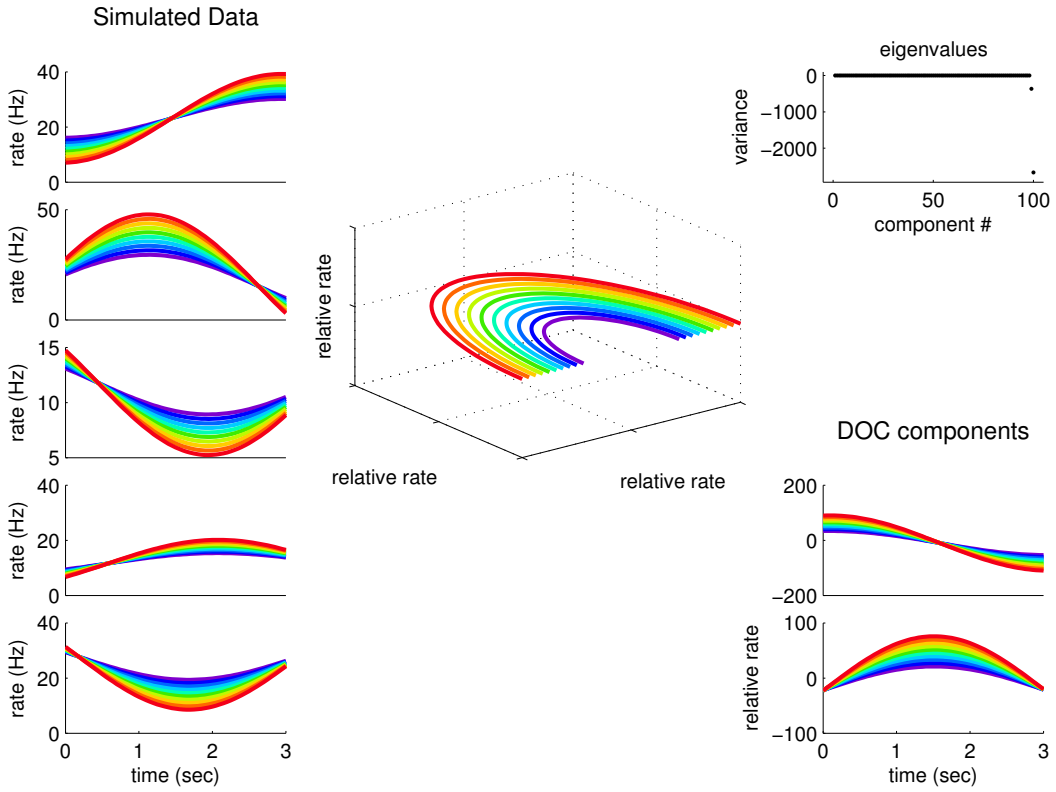
where W is an $[N \times N]$ matrix of effective connectivities between neurons and $\mathbf{E}$ is an N-dimensional vector of external inputs. The initial conditions for this dynamic equation will depend on the first stimulus frequency f,

$$\mathbf{r}(1) = \mathbf{c}(f) \quad .$$

The data analysis shows that the trajectories are confined to a $N_z = 6$-dimensional subspace. If we assume that this observation holds strictly for the simulation, then $\mathbf{r}(t) = U_u U_u^T \mathbf{r}(t)$ where $U_u^T = U_e U_d U_c U_b U_z{}^T$ is the $[N_z \times N]$ matrix that we found after all of the steps in Sections 2 and 4. If we define a new variable within this subspace, $\mathbf{z}(t) = U_u^T \mathbf{r}(t)$, then the dynamics in the subspace are given by

$$\begin{aligned} \mathbf{z}(t) &= U_u^T \left[W\mathbf{r}(t-1) + \mathbf{E} \right] \\ &= U_u^T W U_u \mathbf{z}(t-1) + U_u^T \mathbf{E} \\ &= A\mathbf{z}(t-1) + \mathbf{b} \end{aligned} \tag{6}$$

[4] Note that this result holds both with or without initial dimensionality reduction through PCA.



Supplementary Figure 5: The left panels show the firing rates of several exemplary neurons. The central panel shows the intrinsic dynamics of the system, i.e., the firing rates when embedded in state space. The right panels show the eigenvalues obtained from the DOC analysis (top) and the two DOC components with significant eigenvalues. These components do not unmix time and frequency.

where A is now an $[N_z \times N_z]$ matrix and $\mathbf{b}$ an N_z-dimensional vector,

$$A = U_u^T W U_u$$
$$\mathbf{b} = U_u^T \mathbf{E} \quad ,$$

and the initial conditions are given by

$$\mathbf{z}(1) = \mathbf{d}(f) = U_u^T \mathbf{c}(f) \quad . \tag{7}$$

Note that the parameters of the firing rate model can be retrieved from the reduced subspace model via

$$W = U_u A U_u^T$$
$$\mathbf{E} = U_u \mathbf{b}$$
$$\mathbf{c}(f) = U_u \mathbf{d}(f) \quad .$$

To find the parameters A and $\mathbf{b}$ that best match the observed subspace dynamics, we will seek to minimize the global loss function

$$\text{Err} = \sum_{t=1}^{N_t} \sum_{f=1}^{N_f} \left\| \bar{\mathbf{z}}(t,f) - \mathbf{z}(t,f) \right\|^2 + \lambda \sum_{i=1}^{N_z} \sum_{j=1}^{N_z} \left| A_{ij} - \delta_{ij} \right| \tag{8}$$

where $\bar{\mathbf{z}}(t,f)$ denotes the data and $\mathbf{z}(t,f)$ is a solution to Equation (6) with initial conditions given by Equation (7). Using the same notation as before, we will distinguish between data and model by using a bar on top of all variables that describe the data.

The first term in the loss function captures the difference between model and simulation, summed across the whole delay period and all frequencies, the second term penalizes any coefficients of the state transition matrix A_{ij} that deviate from the identity matrix and thereby expresses our (prior) expectation that the system state does not change over time. Within the Bayesian framework, the choice of the second term, an L1 lasso-regularizer, corresponds to a sparse prior on the parameter space (Hastie et al., 2001).

This regularization term helps to avoid overfitting of parameters, and the parameter λ is consequently chosen so as to minimize the prediction error of the model on new data. To measure the prediction error, we used cross-validation, i.e., we split the data into a training set (data from all frequencies but one, say $f = l$) and an evaluation or prediction set (data from the remaining frequency, $f = l$). The model prediction error for the individual neurons shown in Figure 8C is normalized by the signal variance of each neuron; for neuron k it amounts to

$$p_k = \frac{\frac{1}{N_t N_f} \sum_{t=1}^{N_t} \sum_{f=1}^{N_f} \left(\bar{z}_k(t,f) - z_k(t,f) \right)^2}{\sigma_{\text{total},k} - \sigma_{\text{noise},k}} \quad .$$

where the total and noise variances for neuron k are computed as in Section 3.

For the simulations and connectivity matrices shown in Figure 6–8, the parameters of the model were refitted on the complete data set, using the value of λ determined above. This procedure allowed us to use only one set of parameters and thereby define a single model for the simulations.

Note that with $N_f = 10$, $N_t = 30$, and $N_z = 6$, there will typically be about $N_f \times N_t \times N_z = 1800$ data points. At the same time, there are about 100 parameters to be fitted: 36 parameters for the transition matrix A, 6 parameters for the external inputs $\mathbf{b}$, and 60 parameters for the initial conditions $\mathbf{d}(f)$.

In general, finding the minimum to Equation (8) is a complex, non-linear fitting problem with many local minima. We approached it in two steps: First, we used an approximation to the global loss function to find a first estimate of the parameters. Second, we improved this estimate by using gradient descent on the global loss function.

In the first step, we solved a simplified fitting problem in which the simulation of the system is approximated by the equation

$$\begin{aligned}\mathbf{z}(t,f) &= A\mathbf{z}(t-1,f)+\mathbf{b}\\ &\approx A\bar{\mathbf{z}}(t-1,f)+\mathbf{b}\end{aligned}$$

where we replaced the right hand-sides with values taken from the data. Accordingly, instead of simulating the model from $t = 1$ through all possible times, we only simulate a single time step, but each time step we start with a new initial condition that we take from the data. If, in addition, we set the regularization term to zero, $\lambda = 0$, the loss function becomes

$$\mathrm{Err}_1 = \sum_{t=1}^{N_t}\sum_{f=1}^{N_f}\left\|\bar{\mathbf{z}}(t,f) - A\bar{\mathbf{z}}(t-1,f) - \mathbf{b}\right\|^2 \quad .$$

The problem is thus reduced to a linear regression for the parameters A and $\mathbf{b}$. The initial conditions are no longer true parameters in this scenario, they are simply given through the firing rate values at the first time step.

Let us define a matrix Z_0 such that each row contains both the time and frequency entries of the individual dimensions up to time step $N_t - 1$,

$$Z_0 = \begin{bmatrix} \bar{z}_1(1,1) & \bar{z}_1(2,1) & \dots & \bar{z}_1(N_t-1,1) & \bar{z}_1(1,2) & \bar{z}_1(2,2) & \dots & \bar{z}_1(N_t-1,N_f) \\ \bar{z}_2(1,1) & \bar{z}_2(2,1) & \dots & & & & \dots & \bar{z}_2(N_t-1,N_f) \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \bar{z}_6(1,1) & \bar{z}_6(2,1) & \dots & & & & \dots & \bar{z}_6(N_t-1,N_f) \end{bmatrix}$$

and another similar matrix Z_1, in which the time steps run from 2 to N_t. Furthermore, assume that the mean of the rows of Z_0 and Z_1 is given by the vectors $\mathbf{m}_0$ and $\mathbf{m}_1$, respectively. If we subtract these row means from each row, and indicate the respective matrices

by $\bar{Z}_0$ and $\bar{Z}_1$, then the solution for the parameters is given by

$$\begin{aligned} A &= \bar{Z}_1 \bar{Z}_0^T (\bar{Z}_0 \bar{Z}_0^T)^{-1} \\ \mathbf{b} &= \mathbf{m}_1 - A\mathbf{m}_0 \\ \mathbf{d}(f) &= \bar{\mathbf{z}}(1,f) \end{aligned}$$

where the superscript ^{T} denotes the matrix transpose (see e.g. Hastie et al., 2001).

The above approximation will generally work well if the transition matrix has eigenvalues smaller than one — in this case, the dynamics are decaying and errors due to the one-step approximation will not accumulate. For our data set, however, we found that several eigenvalues of the state transition matrix were close to one, leading to rapid accumulation of errors when the system was simulated over time, often resulting in an exponential explosion of the variables.

To alleviate this problem, we therefore set out to minimize the global loss function, Equation (8), in a second step. We used a conjugate gradient descent algorithm (Press et al., 1992) to update the parameters from an initial value which was given by the estimate from the one-step approximation above. We assume familiarity of the reader with the general principles of conjugate gradient descent and in the following simply reproduce the computation of the gradients.

Gradient with respect to the state transition matrix A

To compute the gradient of the loss function, Equation (8), we applied the chain rule so that (see also Hertz et al., 1991)

$$\begin{aligned} \frac{\partial \mathrm{Err}}{\partial A_{pq}} &= \sum_{t=1}^{N_t} \sum_{f=1}^{N_f} \frac{\partial}{\partial A_{pq}} ||\bar{\mathbf{z}}(t,f) - \mathbf{z}(t,f)||^2 + \sum_{i=1}^{N_z} \sum_{j=1}^{N_z} \frac{\partial}{\partial A_{pq}} |A_{ij} - \delta_{ij}| \\ &= \sum_{t=1}^{N_t} \sum_{f=1}^{N_f} \sum_{k=1}^{N_z} \frac{\partial}{\partial A_{pq}} \big(\bar{z}_k(t,f) - z_k(t,f)\big)^2 + \mathrm{sign}(A_{pq} - \delta_{pq}) \\ &= \sum_{t=1}^{N_t} \sum_{f=1}^{N_f} \sum_{k=1}^{N_z} 2 \frac{\partial z_k(t,f)}{\partial A_{pq}} \big(z_k(t,f) - \bar{z}_k(t,f)\big) + \mathrm{sign}(A_{pq} - \delta_{pq}) \end{aligned}$$

where

$$\begin{aligned} \frac{\partial z_k(t,f)}{\partial A_{pq}} &= \frac{\partial}{\partial A_{pq}} \big[\sum_{j=1}^{N_z} A_{kj} z_j(t-1,f) + b_k \big] \\ &= \delta_{pk} z_q(t-1,f) + \sum_{j=1}^{N_z} A_{kj} \frac{\partial}{\partial A_{pq}} z_j(t-1,f) \quad . \end{aligned}$$

Note that the latter equation is an iterative equation for the gradient.

Gradient with respect to the external inputs b

The computation proceeds along the same lines as above,

$$\begin{aligned}\frac{\partial \mathrm{Err}}{\partial b_p} &= \sum_{t=1}^{N_t}\sum_{f=1}^{N_f}\frac{\partial}{\partial b_p}||\bar{\mathbf{z}}(t,f)-\mathbf{z}(t,f)||^2 \\ &= \sum_{t=1}^{N_t}\sum_{f=1}^{N_f}\sum_{k=1}^{N_z}\frac{\partial}{\partial b_p}\big(\bar{z}_k(t,f)-z_k(t,f)\big)^2 \\ &= \sum_{t=1}^{N_t}\sum_{f=1}^{N_f}\sum_{k=1}^{N_z}2\frac{\partial z_k(t,f)}{\partial b_p}\big(z_k(t,f)-\bar{z}_k(t,f)\big)\end{aligned}$$

where

$$\frac{\partial z_k(t,f)}{\partial b_p} = \delta_{kp} + \sum_{j=1}^{N_z} A_{kj}\frac{\partial}{\partial b_p}z_j(t-1,f) \quad .$$

Gradient with respect to the initial conditions $\mathbf{d}(f)$

To simplify the notation, we will use the short-hand $d_{p,f} = d_p(f)$. Then the computation of the gradient yields

$$\begin{aligned}\frac{\partial \mathrm{Err}}{\partial d_{p,f}} &= \sum_{t=1}^{N_t}\sum_{f=1}^{N_f}\frac{\partial}{\partial d_{p,f}}||\bar{\mathbf{z}}(t,f)-\mathbf{z}(t,f)||^2 \\ &= \sum_{t=1}^{N_t}\sum_{f=1}^{N_f}\sum_{k=1}^{N_z}\frac{\partial}{\partial d_{p,f}}\big(\bar{z}_k(t,f)-z_k(t,f)\big)^2 \\ &= \sum_{t=1}^{N_t}\sum_{f=1}^{N_f}\sum_{k=1}^{N_z}2\frac{\partial z_k(t,f)}{\partial d_{p,f}}\big(z_k(t,f)-\bar{z}_k(t,f)\big)\end{aligned}$$

where

$$\begin{aligned}\frac{\partial z_k(t,f)}{\partial d_{p,f}} &= \frac{\partial}{\partial d_{p,f}}\big[\sum_{j=1}^{N_z}A_{kj}z_j(t-1,f)+b_k\big] \\ &= \sum_{j=1}^{N_z}A_{kj}\frac{\partial}{\partial d_{p,f}}z_j(t-1,f)\end{aligned}$$

and

$$\begin{aligned}\frac{\partial z_k(2,f)}{\partial d_{p,f}} &= \sum_{j=1}^{N_z} A_{kj} \frac{\partial}{\partial d_{p,f}} z_j(1,f) \\ &= \sum_{j=1}^{N_z} A_{kj} \frac{\partial}{\partial d_{p,f}} d_{j,f} \\ &= A_{kp} \quad .\end{aligned}$$

References

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The elements of statistical learning theory*. Springer.

Hertz, J., Palmer, R., and Krogh, A. (1991). *Introduction to the theory of neural computation*. Santa Fe Institute, Westview Press.

Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent component analysis*. Wiley Interscience.

Machens, C. K., Romo, R., and Brody, C. D. (2005). Flexible control of mutual inhibition: A neural model of two-interval discrimination. *Science*, 307:1121–1124.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical recipes in C*. Cambridge University Press.

Sengupta, A. and Mitra, P. (1999). Distributions of singular values for some random matrices. *Phys. Rev. E*, 60:3389–3392.