



RADBOD UNIVERSITY NIJMEGEN

BACHELOR THESIS

Exploring the properties of the smoothing distribution of stochastic DCM for fMRI

Author:
Thijs VAN DER PLAS

Supervisors:
Hans-Christian RUIZ
Bert KAPPEN

November 30, 2015

Contents

1	Introduction	2
2	Motivation	2
2.1	Biological context	2
2.2	Dynamic Causal Model for fMRI	2
2.3	Extensions of the model	3
2.4	The utility of the smoothing distribution	3
3	Background Theory	4
3.1	DCM for fMRI model	4
3.2	Peculiarities of DCM for fMRI	6
3.3	Terminology	8
4	Methods	8
4.1	Two smoothers	8
4.2	Euler-Maruyama scheme solver	9
4.3	APIS algorithm	9
4.4	Kalman Smoother with drift	12
4.5	Simulated data	13
5	Results	13
5.1	Parameter sweep singularity	13
5.2	Estimating the smoothing distribution	14
5.3	Difference Kalman and APIS	15
5.4	Number of particles	16
5.5	Number of integration steps	17
5.6	Different controls	19
5.7	Bootstrapping a large N_{obs}	20
5.8	Changing variance during the BOLD transformation	20
6	Discussion	22
6.1	Difference Kalman and APIS	22
6.2	Shapes of control terms	22
6.3	Number of integration steps	23
6.4	Changing variance after BOLD transformation	24
6.5	Conclusion	24

1 Introduction

The objective of this thesis is to explain how one can estimate the probability distribution of a stochastic neuronal state given a set of observations, which is called the smoothing distribution. For this I use the Adaptive Path Integral Smoother by Ruiz and Kappen [2015]. I will consider two different types of observation sets: observations of the neuronal state itself and observations of a nonlinear transformation of the neuronal state. The latter is a realistic observation model, while the former is unrealistic but does provide an interesting comparison between the linear and the nonlinear case.

The smoothing distributions were successfully estimated and this thesis will also comment on the various properties of the APIS method.

2 Motivation

2.1 Biological context

Here I will explain how neuronal activity (local brain activity) induces a hemodynamic response (i.e. fluid dynamics of blood flow), which can be measured using functional magnetic resonance imaging (fMRI) techniques. This is based on review papers by Raichle [1998] and Kim and Ogawa [2012].

fMRI is a very well-known manner to measure brain activity. It is a non-invasive imaging method, which outputs a 3D image consisting of a large amount (10^5) of voxels which represent local brain activity. The signal in a voxel is based on the BOLD response. The BOLD response is a hemodynamic response to neuronal activity. Consequently, neuronal activity is measured in a very indirect way. On top of this, fMRI has a poor temporal resolution (it is much larger than the time scale of neuronal activity), such that an fMRI image shows the average neuronal activity in the fMRI time frame.

In this context neuronal activity mainly consists of local field potentials and spiking activity. Neuronal activity induces changes in the local blood flow and oxygen consumption. Because these two changes are not exactly proportional to each other, the amount of oxyhemoglobin (oxygenated blood, Hb) and deoxyhemoglobin (de-oxygenated blood, dHb) change subsequently. This is expressed in the Blood Oxygen Level Dependent (BOLD) signal, which depends on the local dHb content and blood volume inside a tissue in the brain. Hb and dHb have different magnetic susceptibilities, which is why a fMRI machine can measure the BOLD signal.

This process can be described by a mathematical model, named the Dynamic Causal Model for fMRI. This model allows for interactions between different brain regions. The coupling parameters which model these interactions describe how different brain regions are causally connected and a typical goal of an experiment is to estimate these.

2.2 Dynamic Causal Model for fMRI

The neuronal activity is modelled by a Dynamic Causal Model (DCM), first introduced by Friston et al. [2003]. The DCM describes the neuronal activity $z(t)$ per brain region, influenced by adjacent brain regions and externally controlled inputs $I(t)$ (stimuli). The neuronal activity then undergoes a hemodynamic

transformation which is modelled inside a single blood vein, consisting of two parts: the hemodynamic equations [Friston et al., 2000] and the Balloon model [Buxton et al., 1998]. The hemodynamic equations describe the vasodilatory signal $s(t)$ and the blood inflow in the vein $f(t)$, and the Balloon Model subsequently describes the dHb content $q(t)$ and blood volume $v(t)$ of the vein. These variables are described by a set of four coupled differential equations. The BOLD signal $y(t)$ is dependent on $q(t)$ and $v(t)$. This is summarised in figure 1.

This model is still an active topic of research with both recent further improvements of the original model (Frässle et al. [2015], Havlicek et al. [2015]) and recent applications of the model (e.g. in neurofeedback: Koush et al. [2013], Scharnowski et al. [2014]).

2.3 Extensions of the model

Since its inception, a lot of extensions to the model described above have been proposed in order to create more realistic descriptions of the BOLD signal (e.g. nonlinear DCM [Stephan et al., 2008], two-state DCM [Marreiros et al., 2008]). Some extensions are however not clear improvements and have later been opposed by other articles (e.g. [Buxton et al., 2004] is discommended by Jacobsen et al. [2008]). More importantly, the dynamics of the system do not drastically change, and for the sake of simplicity in this paper I will use the original model dynamics. The only extension that I use is that of a stochastic DCM, by including process and observation noise, as I will explain in section 3. This method was introduced by Riera et al. [2004]. This variation of the DCM for fMRI is also used in papers with similar focusses as mine such as [Osório et al., 2015] and [Murray and Storkey, 2011].

2.4 The utility of the smoothing distribution

The smoothing distribution computes the probability distribution of the neuronal activity given the set of observations (the mathematical definition will be given in section 3). This is an interesting function by itself, but it also has practical use: the smoothing distribution can be of great relevance in or-

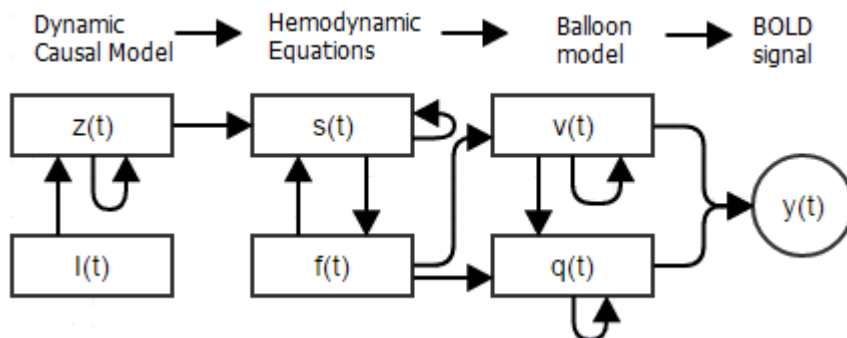


Figure 1: A schematic figure showing how the state variables of the DCM for fMRI model are related.

der to perform an expectation-maximization scheme [Särkkä, 2013]. By such a scheme one can estimate the value of a specific parameter of the model. In the case of DCM for fMRI, the coupling parameters are the relevant parameters to estimate. These determine the strengths and causal directions of connections between different brain regions (A and C in equation 1). In my case I only have two coupling parameters, but for a multi-region DCM the number of coupling parameters scales quadratically with the number of regions. Shortly after DCM for fMRI was first introduced, Penny et al. [2004] proposed a method to compare different DCM model hypotheses by Bayesian model comparison, which became a widely used method. One can however not estimate specific coupling parameter values with this method. Friston et al. [2003] did perform coupling parameter estimation through a different expectation-maximization scheme, which did not make use of a smoothing distribution. The validation of their inference depends directly on the validation of the priors of the coupling parameters, which they described as a “potential weakness”. Later, Chumbley et al. [2007] were able to perform coupling parameter estimation with uninformative priors using a Metropolis-Hastings scheme. Both of these papers did however not implement any process noise. By calculating the smoothing distribution, one is able to perform parameter estimation of a system with process noise.

3 Background Theory

3.1 DCM for fMRI model

All specific values of the constants denoted in the following paragraphs are given in table 1.

Stochastic DCM In this paper only one brain region is considered. Therefore the DCM equation simplifies because no inter-region influences occur [Havlicek et al., 2015]. Also Gaussian noise $dW_z \sim \mathcal{N}(0, \sqrt{\Delta t})$ is added (where Δt is the numerical integration step, see Methods section), such that the stochastic differential equation of the neuronal activity $z(t)$ becomes:

$$dz = (Az + CI)dt + \sigma_z dW_z \quad (1)$$

Here $A < 0$ (such that this system is stable) is the self-interaction term and C determines the coupling strength to the external input $I(t)$. This input function $I(t)$ is typically a boxcar-function for DCM for fMRI experiments (i.e. a stimulus is either present or not).

I have defined the neuronal state $z(t)$ to be stochastic and all latent states (s, f, q, v) to be deterministic. This was first done by Riera et al. [2004], although it is worth pointing out that in this article no DCM was used, such that the input $I(t)$ and process noise dW_z were directly inserted into equation 2. The original paper by Friston et al. [2003] only used observation noise dW_y .

The hemodynamic equations The biophysical response to the neuronal activity $z(t)$ is modelled inside a single blood vein. The first two variables that are introduced are the vasodilatory (i.e. widening of the blood vein) signal $s(t)$ and the normalised blood inflow $f(t)$ of the vein. Please note that ‘normalised’

means ‘compared to resting brain activity’ (i.e. $I(t) = 0$), so $f(t) = \frac{F(t)}{F_0}$. The differential equations of $s(t)$ and $f(t)$ are [Friston et al., 2000]:

$$\frac{ds}{dt} = \left(\epsilon z - \frac{s}{\tau_s} - \frac{f-1}{\tau_f} \right) \quad (2)$$

$$\frac{df}{dt} = s \quad (3)$$

Where τ_s and τ_f are time constants and ϵ is the efficacy constant of $z(t)$.

The Balloon model The variable $f(t)$ is then progressed in the so-called Balloon model, in which two new variables are introduced. The vein is modelled as an expandable compartment/balloon (such that volume can change), hence the name ‘Balloon model’. The variable $q(t) = \frac{Q(t)}{Q_0}$ denotes the normalised content of dHb and $v(t) = \frac{V(t)}{V_0}$ the normalised blood volume inside this ‘balloon’. The differential equations of $q(t)$ and $v(t)$ are given by [Buxton et al., 1998]:

$$\frac{dq}{dt} = \frac{1}{\tau_0} \left(f \frac{1 - (1 - E_0)^{\frac{1}{f}}}{E_0} - v^{\frac{1}{\alpha}-1} q \right) \quad (4)$$

$$\frac{dv}{dt} = \frac{1}{\tau_0} \left(f - v^{\frac{1}{\alpha}} \right) \quad (5)$$

Here τ_0 is a time constant. The term $\frac{1 - (1 - E_0)^{\frac{1}{f}}}{E_0}$ is the normalised fraction of oxygen extracted from the inflowing blood. Finally, α is the stiffness exponent, describing the outflow-volume relationship through the power law $f_{\text{out}}(v) = v(t)^{1/\alpha}$ based on Mandeville et al. [1999], Grubb et al. [1974].

BOLD signal The variables q and v give rise to the BOLD signal S (not to be confused with the vasodilatory signal $s(t)$) of which the relative signal change $\frac{\Delta S}{S} = y(t)$ is measured. $y(t)$ is the signal empirically observed and considered in the rest of this paper (and in literature in general) and will be called the BOLD signal from now on for convenience (as opposed to ‘the relative BOLD signal change’). The formula for $y(t)$ is given by [Buxton et al., 1998]:

$$y(t) = V_0 \left(k_1(1 - q) + k_2 \left(1 - \frac{q}{v} \right) + k_3(1 - v) \right) \quad (6)$$

Note that this is not a differential equation, such that the entire system is 5-dimensional (z, s, f, q, v). V_0 is the blood volume fraction of the balloon at rest (which is the same V_0 as in $v(t) = \frac{V(t)}{V_0}$). k_1, k_2 and k_3 are dimensionless empirically defined constants which depend on E_0 .

Summarising: there is an input $I(t)$ controlled by the experiment. This induces the noisy neuronal activity $z(t)$ which is deterministically transformed into the BOLD signal $y(t)$. This is illustrated in figure 2, where the plots of all six variables are given in response to an input function $I(t)$.

For the parameters of the model I used the values from Murray and Storkey [2011] which were based on Buxton et al. [1998], Friston et al. [2003]. See table 1. In the rest of this paper, unless explicitly noted otherwise, the parameters are the ones listed in this table.

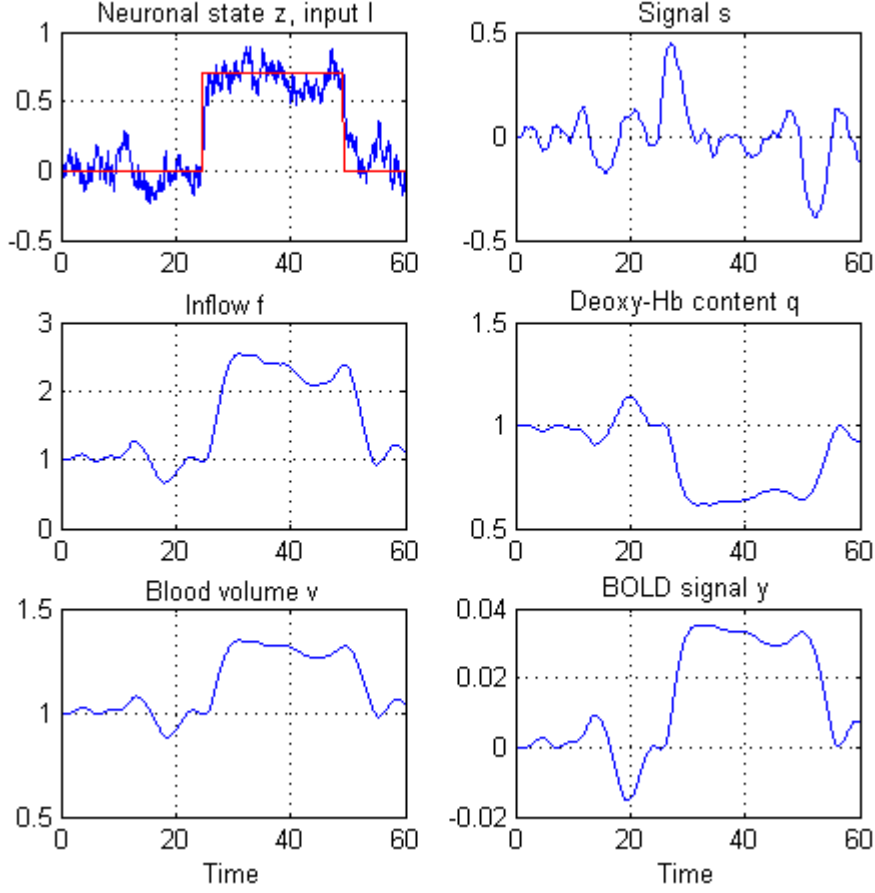


Figure 2: A simulation of the BOLD signal. The plots show (from top to bottom, left to right): the neuronal activity z (blue) and input I (red), the vasodilatory signal s , the blood inflow f , the dHb content q , the blood volume v and the BOLD signal y .

Const.	Value	Const.	Value	Const.	Value
ϵ	0.8	α	0.32	σ_z	0.15
E_0	0.4	V_0	0.018	A	-1
τ_0	1.02	k_1	$7E_0$	C	0.7
τ_f	2.44	k_2	2	$s(0)$	0
τ_s	1.54	k_3	$2E_0 - 0.2$	$f(0), q(0), v(0)$	1

Table 1: Values of parameters of the model taken from Murray and Storkey [2011], Buxton et al. [1998], Friston et al. [2003].

3.2 Peculiarities of DCM for fMRI

Hemodynamic Delay A typical property of the DCM for fMRI model is the hemodynamic delay of 3-5 seconds [Havlicek et al., 2015]. This delay is due to the fact that biophysical changes (blood flow/oxygen consumption) are not

instantaneous but take some time to settle in. This is illustrated in figure 3. Here it clearly can be seen that the peak of the BOLD signal is approximately 3 seconds later than the peak of the neuronal state. Note that small fluctuations in the neuronal state are smoothed out in the BOLD signal.

Singularity In the model described above, specifically in equation 4, a singularity occurs for $f \uparrow 0$:

$$\lim_{f \uparrow 0} f \frac{1 - (1 - E_0)^{\frac{1}{f}}}{E_0} = \lim_{f \uparrow 0} f \frac{1 - (0.6)^{\frac{1}{f}}}{0.4} \rightarrow \infty \quad (7)$$

This formula $(1 - (1 - E_0)^{1/f})$ was proposed by Buxton and Frank [1997], but this singularity is not mentioned. Their paper has a biological focus, such that only physically possible values ($f > 0$) are considered. To my best of knowledge this singularity is only mentioned in an article by Riera et al. [2004], who consider only physically relevant values and therefore show the limit $f \downarrow 0$, which exists and goes to 0.

But for the method I use (the APIS method, see Methods section) this singularity is definitely relevant, as I will explain in the Results section. The value of $f(t)$ is indirectly driven by the input strength $C \cdot I(t)$. To see for which range of

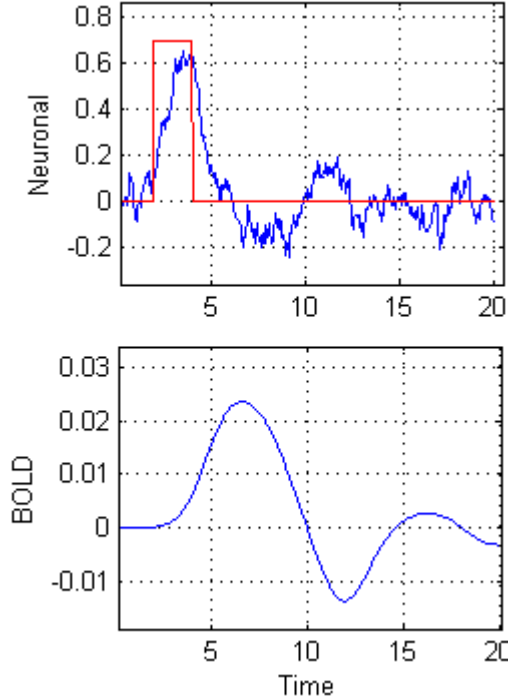


Figure 3: A plot of the neuronal activity $z(t)$ (top) and corresponding BOLD signal $y(t)$ (bottom) to show the hemodynamic delay. The red boxcar function in the top figure shows the (relatively short) input $I(t)$. The peak of the BOLD signal is approximately 3 seconds delayed, this is due to the hemodynamic delay.

$C \cdot I(t)$ this happens, I have performed a parameter sweep (see Results section, figure 5). From this I conclude that for $-0.45 < C \cdot I(t) < 3.97$ a negative value of f will occur (if the system gets enough time to settle in). Note that $f \uparrow 0$ occurs every time $f(t)$ goes through zero.

Ideally this model should be modified, such that negative values of f could not occur, by implementing a biologically plausible modification. This has unfortunately not been done, so that equation 4 is still the state-of-the-art equation [Havlicek et al., 2015]. In order to avoid this problem, I have implemented the following change in the model: $f \rightarrow |f|$. This was the fastest method I have tried computationally wise. It changes the dynamics of the system for $f < 0$, but this does not matter: as I will explain in the Methods section the APIS smoother uses importance sampling, and the importance weights w^u go to zero for unwanted dynamics, such that the smoother eventually learns to avoid negative values of f .

3.3 Terminology

The following notations are used: time is discretized in the numerical integration such that $x(t) = x_t$, furthermore 0 is the start time, T is the end time and $t = 0 : \tau$ denotes all states or observations in between times 0 and τ .

Two relevant probability distributions are considered: the filter distribution (filter) $p(z_t|Y_{0:t})$ and the marginal smoothing distribution $p(z_t|Y_{0:T})$, of which the joint smoothing distribution (smoother) is $p(z_{0:T}|Y_{0:T})$. Both the filter and smoother compute the probability distribution of $z(t)$, but the filter makes use of the observations until time t (an online computation), while the smoother makes use of all the observations given (an offline computation).

4 Methods

4.1 Two smoothers

I will consider two different smoothers, called the neuronal smoother and the BOLD smoother. Both smoothers compute the smoothing distribution of the neuronal state $z(t)$ (which is the stochastic process state, equation 1). The neuronal smoother extracts observations directly from $z(t)$, only adding Gaussian observation noise dW_y . The BOLD smoother however implements the BOLD transformation (equations 2-6) such that an observation is the BOLD signal $y(t)$ plus Gaussian observation noise dW_y . This is summarised in figure 4. Note that the latent states are dependent on their past states, such that this is not an instantaneous transformation. Hence, the neuronal smoother is a linear Gaussian model and the BOLD smoother is a nonlinear model, for which it is naturally more difficult to estimate the smoothing distribution.

The observation noise dW_y is white noise drawn from a normal distribution of which the standard deviation is 0.1 for the neuronal smoother and 0.005 for the BOLD smoother, which is approximately the same signal-to-noise ratio. During all my simulations the prior of z was fixed at $z(t=0) \sim \mathcal{N}(0, \sigma_{pri})$. For the neuronal smoother $\sigma_{pri} = 0.2$ and for the BOLD smoother $\sigma_{pri} = 0.02$.

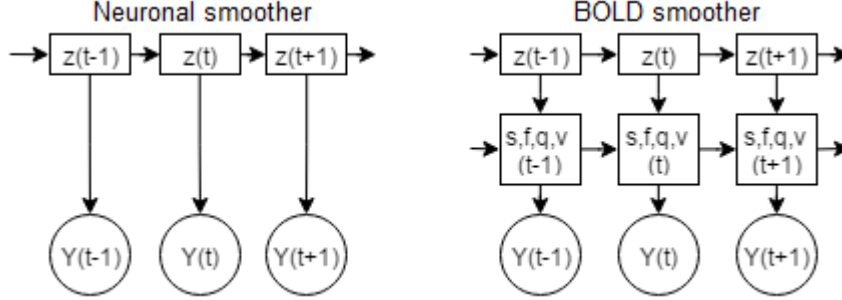


Figure 4: An illustration showing the two different observation models for the neuronal smoother (left) and BOLD smoother (right). Note that both smoothers have the same process state $z(t)$, but that the BOLD smoother has four additional latent states $s, f, q, v(t)$ before an observation is taken.

4.2 Euler-Maruyama scheme solver

In order to integrate this system numerically, I make use of the Euler-Maruyama (E-M) scheme. This is an intuitive way to integrate differential equations, by discretizing the time in an stochastic differential equation, which has the general form:

$$dx = F(x_n, t_n)dt + \sigma_x dW_x \quad (8)$$

Note that for ordinary differential equations $dW_x = 0$. In this E-M scheme $dt \rightarrow \Delta t \rightarrow 0$, $dW_x \rightarrow \Delta W_x$ and $dx \rightarrow \Delta x = x(t_n + \Delta t) - x(t_n)$ such that one obtains the following integration step:

$$t_{n+1} = t_n + \Delta t \quad (9)$$

$$x(t_{n+1}) = x_{n+1} = x_n + F(x_n, t_n) \Delta t + \sigma_x \Delta W_x \quad (10)$$

Please note that $x_n = \{z_n, s_n, f_n, q_n, v_n\}$ and $F(x_n, t_n) \Delta t$ is the set of coupled differential equations (equations 1-5). Equations 9 and 10 describe the change of the system over a small timestep Δt . So in order to calculate the behaviour of the system over the total time interval T these equations have to be iterated $\frac{T}{\Delta t}$ times. This (numerical) integration works well as long as the timestep Δt is sufficiently small. Murray and Storkey [2011] found that the E-M integrator has the same accuracy as the (more sophisticated) Runge-Kutta (R-K) integrator, although the R-K scheme is approximately five times faster (for this specific model). Following these results I chose to use the E-M scheme due to its simplicity.

4.3 APIS algorithm

In order to be able to smooth the neuronal activity $z(t)$ based on noisy measurements I make use of the Adaptive Path Integral Smoother (APIS) algorithm by Ruiz and Kappen [2015]. This smoother makes use of importance sampling in a path integral setting as introduced by Thijssen and Kappen [2015].

Importance Sampling A quick definition of importance sampling is the following [MacKay, 2003]:

One wants to numerically compute the expectation of a function $\phi(x)$ where x is distributed according to $P(x)$. However, $P(x)$ (the target distribution) is too difficult to sample from, but one can evaluate $P^*(x) = Z \cdot P(x)$ where Z is any constant.

Then a proposal distribution $Q(x)$ is introduced, which is simpler than $P(x)$ such that one can sample from $Q(x)$, where $Q^*(x) = Z_Q \cdot Q(x)$ can be evaluated.

Now to calculate the estimator $\hat{\Phi}$ of $\phi(x)$ one uses N samples $x^{1:N}$, sampled from $Q(x)$:

$$\hat{\Phi} = \frac{\sum_n w_n \phi(x^n)}{\sum_n w_n}, \quad w_n = \frac{P^*(x^n)}{Q^*(x^n)} \quad (11)$$

For a sufficient number of samples N and a well (arbitrarily) chosen proposal distribution $Q(x)$ the estimator $\hat{\Phi}$ is correctly estimated. w_n is the importance weight corresponding to sample n .

APIS I will now briefly describe the APIS method, as I implemented it. For the complete and general derivation, see Ruiz and Kappen [2015] and the references therein.

First, a control term $u(x, t)$ is added to the stochastic differential equation 8 as follows:

$$dx = F(x, t)dt + \sigma_x(x, t) (u(x, t) + dW_x) \quad (12)$$

Then, by defining a cost and subsequently minimizing this cost with respect to the control $u(x, t)$ in equation 12, the following relation for the smoothing distribution holds:

$$p(x_{0:T}|Y_{0:T}) \propto p(x_{0:T}) \exp \left(- \int_0^T V(x_t, Y_t, t) dt \right) \quad (13)$$

Here $p(x_{0:T})$ is the prior distribution for the process states, i.e. the uncontrolled dynamics. $V(x_t, Y_t, t)$ is the potential at observation times (and zero at times t where there is no observation). The potential is defined to be the log-likelihood of the observations such that if J is the number of observations then: $-\int_0^T V(x_t, Y_t, t) dt = \sum_{j=1}^J \log(p(Y_j|x_j))$. Using this relation and rewriting equation 13 in terms of the controlled dynamics it is found that:

$$p(x_{0:T}|Y_{0:T}) \propto w_t^u \cdot p(x_{0:T}|u_{0:T}), \quad w_t^u = \frac{\exp(-S^u)}{\mathbb{E}(\exp(-S^u))} \quad (14)$$

$$S^u = \int_0^T \frac{1}{2} \|u(x_s, s)\|^2 ds + \int_0^T u(x_s, s)' dW_s - \sum_{j=1}^J \log(p(Y_j|x_j)) \quad (15)$$

Note that equation 14 is the importance sampling step in the APIS algorithm, as defined in equation 11, where $p(x_{0:T}|u_{0:T})$ is the proposal distribution, $p(x_{0:T}|Y_{0:T})$ is the target distribution and w_t^u are the importance weights. These importance weights weigh the particles, such that the statistics of a great

number of particles (typically in the order of 10^3) make up an accurate estimation of the smoothing distribution. The effective sampling size (ESS) of the particles is defined as:

$$\text{ESS} = \frac{N_{\text{eff}}}{N} = \frac{1}{\mathbb{E}((w^u)^2)} = \frac{1}{1 + \text{Var}(w^u)} \quad (16)$$

The ESS denotes the efficiency of the sampling procedure from the proposal distribution $p(x_{0:T}|u_{0:T})$. This means that for maximum efficiency (ESS=1) the proposal distribution is equal to the target distribution, in other words, that the controlled dynamics are equal to the smoothing distribution. The task of the APIS algorithm is to update the control $u(x, t)$ iteratively such that ideally it converges to the optimal control $u^*(x, t)$, which corresponds to $\text{Var}(w^u) = 0$ and thus $\text{ESS} = 1$. This also means that the ESS is a good measure to check whether the considered controller $u(x, t)$ is the optimal controller.

The optimal control $u^*(x, t)$ is assumed to be parametrised by a linear combination of basis functions $h(x, t)$ through the matrix M_t : $u^*(x, t) = M_t h(x, t)$. In my case, h is chosen to be $h(x) = (1x - \langle \mathbf{x} \rangle, 1)$ where $\langle \mathbf{x} \rangle$ is the weighted mean of all the particles \mathbf{x} .

In the iterative scheme M_t is updated, such that the control and ESS eventually converge. This is done by the following learning rule:

$$M_t^{n+1} = M_t^n + \eta \frac{dQ(h)}{dt} H^{-1} \quad (17)$$

Here, $H = \langle h \otimes h \rangle$ denotes the cross-correlation matrix between the basis functions h with respect to the smoothing distribution. $0 < \eta < 1$ is the learning rate and $dQ(h)/dt$ is defined to be the cross-correlation matrix between the basis function and the noise dW_s :

$$\frac{dQ(h)}{dt} = \lim_{\delta t \rightarrow 0} \frac{\langle \int_t^{t+\delta t} dW_s \otimes h(s) \rangle}{\delta t} \quad (18)$$

The resulting smoothing distribution is given by an ensemble of weighted particles at each integration step.

This iterative scheme is summarised in algorithm 1 below.

Algorithm 1 APIS.m

- 1: **Input** Observations $Y_{0:T}$, number of iterations K_{max} , number of particles N , learning rate η , integration step Δt
 - 2: **while** $k < K_{\text{max}}$ **do**
 - 3: **Generate** particle system $\{z_{0:T}^N, w^{1:N}\}$ according to equation 12
 - 4: **for** $t = 0:T$ **do**
 - 5: **Update** $M_t^{k+1} = M_t^k + \eta \frac{dQ(h)}{dt} H^{-1}$
 - 6: **end for**
 - 7: $k = k + 1$
 - 8: **end while**
 - 9: **Output** Smoothing distribution represented as an ensemble of weighted particles, the controller $u(x, t) = M_{0:T}^{K_{\text{max}}} \cdot h(x)$
-

4.4 Kalman Smoother with drift

The established Kalman smoother is the exact solution for linear Gaussian state-space models [Särkkä, 2013]. So the APIS method can be compared to the Kalman smoother just for the neuronal smoother, as this is a linear Gaussian model. Here I will show how to implement the DCM as a drift function in the Kalman smoother. Recall that the equation for a noisy DCM is:

$$dz = (Az + CI)dt + \sigma_z dW_z \quad (19)$$

In a discrete time integration step this can be written as:

$$z_t = z_{t-1} + (Az_{t-1} + CI_{t-1})\Delta t + \sigma_z \Delta W_z \quad (20)$$

$$z_t = (A\Delta t + 1)z_{t-1} + C\Delta t I_{t-1} + \sigma_z \Delta W_z \quad (21)$$

First, the Kalman filter (by forward propagation) is computed. The Kalman filter is defined according to Särkkä [2013]. The dynamic and observation model are defined by:

$$\mathbf{x}_t = \begin{pmatrix} z_t \\ 1 \end{pmatrix} = \mathbf{A}_{t-1}\mathbf{x}_{t-1} + \mathbf{q}_{t-1} = \begin{pmatrix} A\Delta t + 1 & C\Delta t \cdot I_{t-1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z_{t-1} \\ 1 \end{pmatrix} + \begin{pmatrix} \sigma_z \Delta W_z \\ 0 \end{pmatrix} \quad (22)$$

$$\mathbf{y}_t = \begin{pmatrix} y_t \\ 0 \end{pmatrix} = \mathbf{H}_t \mathbf{x}_t + \mathbf{r}_t = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} z_t \\ 1 \end{pmatrix} + \begin{pmatrix} \Delta W_y \\ 0 \end{pmatrix} \quad (23)$$

Following Särkkä [2013], the mean $m_{z,t}$ and the variance $P_{z,t}$ are predicted every timestep where there is no observation by:

$$m_{z,t} = A_{z,t-1}m_{z,t-1} = (A\Delta t + 1)m_{z,t-1} + C\Delta t \cdot I_{t-1} \quad (24)$$

$$P_{z,t} = A_{z,t-1}P_{z,t-1}A_{z,t-1}^T + Q_{z,t-1} = (A\Delta t + 1)^2 P_{z,t-1} + Q \quad (25)$$

$Q = \sigma_z \sqrt{\Delta t}$ is the standard deviation of the process noise $\sigma_z \Delta W_z$. Every timestep where there is an observation, $m_{z,t}$ and $P_{z,t}$ are updated following the update rule:

$$m_{z,t} = (A\Delta t + 1)m_{z,t-1} + C\Delta t \cdot I_{t-1} + \frac{(A\Delta t + 1)^2 P_{z,t-1} + Q}{(A\Delta t + 1)^2 P_{z,t-1} + Q + R} (Y_t - m_{z,t}) \quad (26)$$

$$P_{z,t} = (A\Delta t + 1)^2 P_{z,t-1} + Q - \frac{(A\Delta t + 1)^2 P_{z,t-1} + Q}{(A\Delta t + 1)^2 P_{z,t-1} + Q + R} \quad (27)$$

Where R is the standard deviation of the observation noise ΔW_y .

Next, the Kalman smoother is computed by a backwards propagation through time and making use of the already calculated Kalman filter. The Kalman smoother ($m_{z,t}^s, P_{z,t}^s$) is calculated according to:

$$m_{z,t}^s = m_{z,t} + \frac{(A\Delta t + 1)P_{z,t}}{(A\Delta t + 1)^2 P_{z,t-1} + Q} \left(m_{z,t+1}^s - (A\Delta t + 1)m_{z,t} - C\Delta t \cdot I_t \right) \quad (28)$$

$$P_{z,t}^s = P_{z,t} + \left(\frac{(A\Delta t + 1)P_{z,t}}{(A\Delta t + 1)^2 P_{z,t-1} + Q} \right)^2 \left(P_{z,t+1}^s - (A\Delta t + 1)^2 P_{z,t-1} + Q \right) \quad (29)$$

4.5 Simulated data

In order to be able to calculate the smoothers, I simulated artificial data. I ran the system for 60 seconds, with the following input function:

$$I(t) = \begin{cases} 1 & 24.6 < t < 49.2 \\ 0 & \text{otherwise} \end{cases} \quad (30)$$

This is based on Murray and Storkey [2011], who use a real fMRI data set. I used one data set for all neuronal smoother experiments, and one other (but similar) data set for all BOLD smoother experiments. These can be seen in figure 6 for the neuronal smoother and figure 7 for the BOLD smoother (the blue and green lines). From this I extracted observations with regular intervals (the black dots in the figures), as I will show in the Results section I used varying numbers of observations N_{obs} .

5 Results

5.1 Parameter sweep singularity

A parameter sweep for f versus $C \cdot I(t)$ was performed in order to find out for which input strengths a singularity occurs in the DCM for fMRI model. This is shown in figure 5. This parameter sweep was performed without noise (i.e. $dW_z = 0$ in equation 1) such that definite bounds were gained. The parameter sweep shows that $f > 0$ for $-0.45 < C \cdot I(t) < 3.97$. Large positive values for $C \cdot I(t)$ can also cause f to become negative because of the post-stimulus undershoot which causes f to decrease [Buxton et al., 1998]. When noise is implemented in the system, f can become negative for even smaller input values (as the noise can coincidentally push z downwards). The probability that this happen is definitely relevant because the APIS smoother works with a large number of particles N , which all have individual trajectories and noise.

The volume v is also plotted because an additional singularity in equation 6 can occur for $v \rightarrow 0$. However, as the parameter sweep shows, v does not go to zero if f does not. This is intuitive: the volume follows the flow, so if the flow is limited then the volume is too.

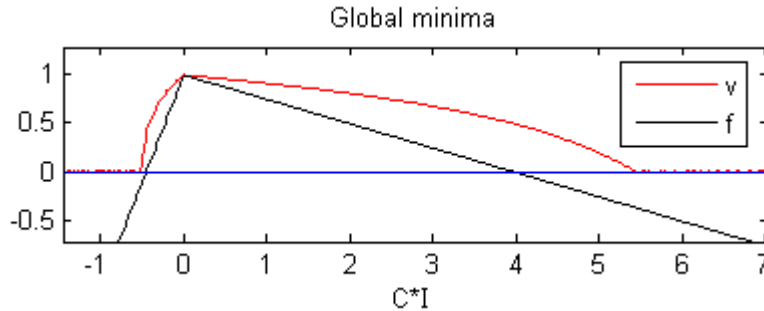


Figure 5: A parameter sweep for f versus $C \cdot I(t)$. Note that when v becomes negative, it also becomes complex due to its dependence on $v^{\frac{1}{\alpha}}$. This is outputted as zero in this Matlab figure.

5.2 Estimating the smoothing distribution

The neuronal smoother and BOLD smoother were both estimated. The outcomes of these computations are dependent on a variety of parameters, of which the most important parameters are: the number of observations N_{obs} , the integration step Δt , the number of APIS iterations K_{max} and the number of particles N . To give an impression of how the smoothing distribution looks, I have plotted the neuronal smoother in figure 6 and the BOLD smoother in figure 7 for $N_{\text{obs}} = 16$, $\Delta t = 0.05$ and $N = 5000$. In the rest of this section I will explore some properties of the smoothers such as the ESS (as defined in the methods section), the shape of the control $u(x, t)$ and bootstrapping difficulties. In particular, the differences between the neuronal and BOLD smoother depending on the parameters will be discussed.

For every simulation, the ESS converges to a maximum value for a growing number of iterations. The number of iterations needed for this saturation differs per simulation (I used a range of $200 \leq K_{\text{max}} \leq 3000$ during my simulations). The more difficult a smoothing problem is the more iterations are needed (i.e. for a higher N_{obs} one needs a smaller η and thus a higher K_{max}). An example of a converging ESS is shown in figure 8 of the neuronal smoother previously shown in figure 6.

I always let the system run until it was converged for at least 50 iterations. These saturated values were consequently used to gain a mean and standard deviation for the ESS (notation: mean \pm standard deviation). I have run a few systems for an extra 1000 iterations after convergence and all simulations remained stable (the ESS values did not change).

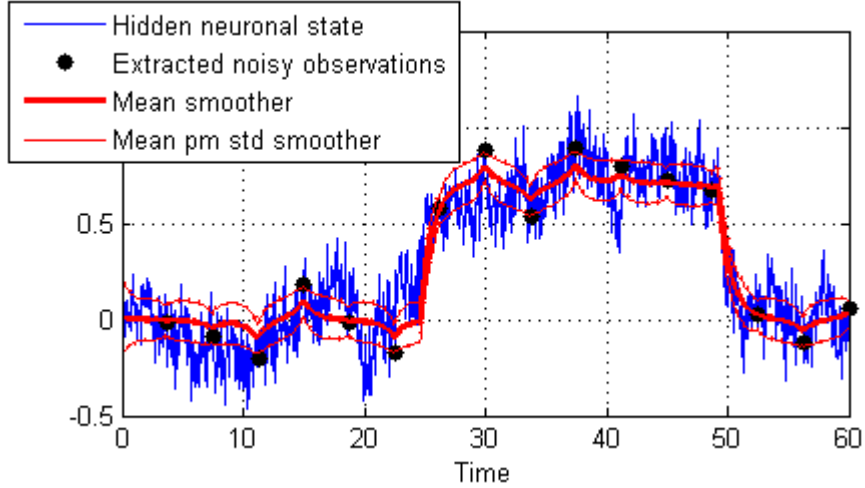


Figure 6: The smoothing distribution of the neuronal smoother.

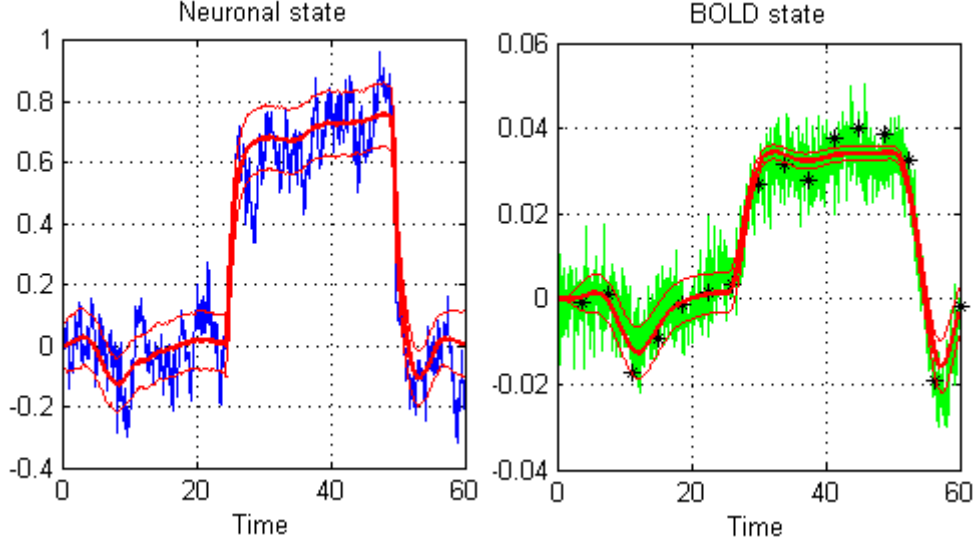


Figure 7: Left: the calculated smoothing distribution (red) of the neuronal activity (blue), based on BOLD observations. Right: The BOLD transformation of the smoothing distribution (red), the original BOLD signal (green) and the extracted observations (black).

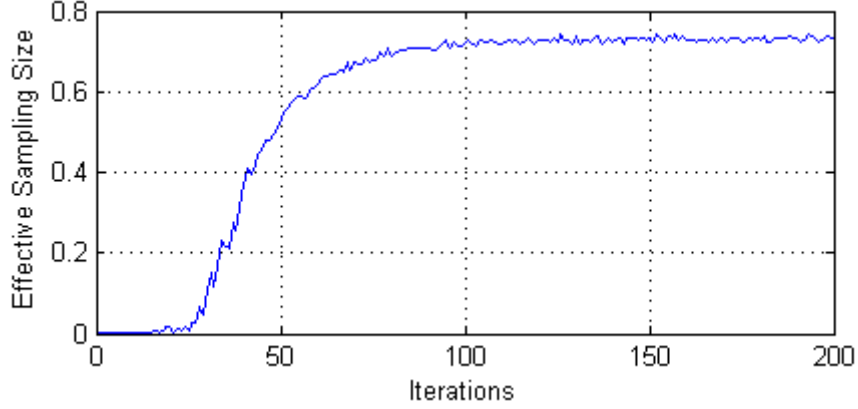


Figure 8: A plot showing the converging ESS for the neuronal smoother shown in figure 6, with parameters $N = 5000$, $\Delta t = 0.05$, $N_{\text{obs}} = 16$, $K_{\text{max}} = 200$.

5.3 Difference Kalman and APIS

While estimating the neuronal smoother a difference between the Kalman method and the APIS method is observed. This difference occurs twice: at the two sudden changes in the drift (when the input is turned on or off). This is illustrated in figure 9 where the Mean Square Error (MSE) between the Kalman smoother and the APIS smoother is plotted. In this simulation the key parameters were set at $N = 5000$, $N_{\text{obs}} = 32$, $\Delta t = 0.05$. In figure 9 it is clearly visible that the

means differ greatly near the discontinuities and variances do not. A close up of the vicinities of the input discontinuities is plotted in figure 10. This figure shows the cause for the peaks of the MSE: the APIS smoother stays closer to the fixed point than the Kalman smoother. I will comment on this matter in the Discussion section. In the rest of my simulations I will only use the APIS method for both the neuronal smoother and the BOLD smoother.

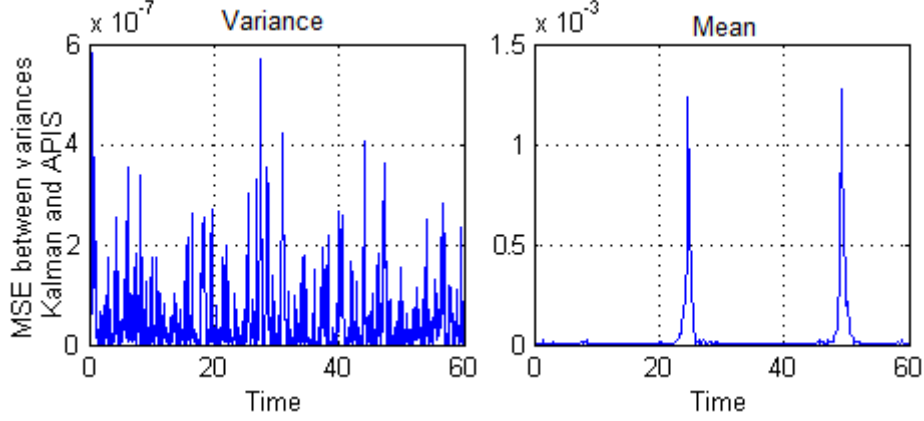


Figure 9: The MSE for the means and variances of the Kalman and APIS smoothers. The left plot shows the MSE of the variances and the right plot shows the MSE of the means.

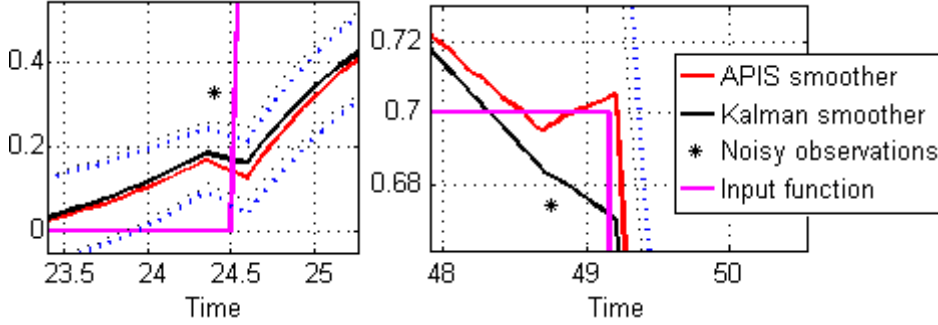


Figure 10: A close up of the difference between the Kalman and the APIS smoother in the vicinity of the discontinuities of the input function. One can see that the smoothers use I_{t-1} for the estimation of z_t , such that they are one integration step delayed in going to the new fixed point.

5.4 Number of particles

In figure 11 the dependence of the ESS on the number of particles N for examples of both the neuronal smoother and the BOLD smoother is plotted with different numbers of observations. The smoothers show equal behaviour. There is a minimum number of particles needed to bootstrap the algorithm (typically

around $N \approx 300$). When it has bootstrapped, the ESS grows for an increasing number of particles and eventually saturates when a sufficient number of particles is reached (typically around $N \approx 2000$). The number of particles needed to either bootstrap or to converge are more or less equal for the two smoothers. These examples show that for a fixed number of observations N_{obs} the ESS saturates for a growing number of particles N . In the following simulations $N = 5000$ is fixed, to ensure that the convergence value for the ESS is reached, and I will show what happens for a varying number of observations.

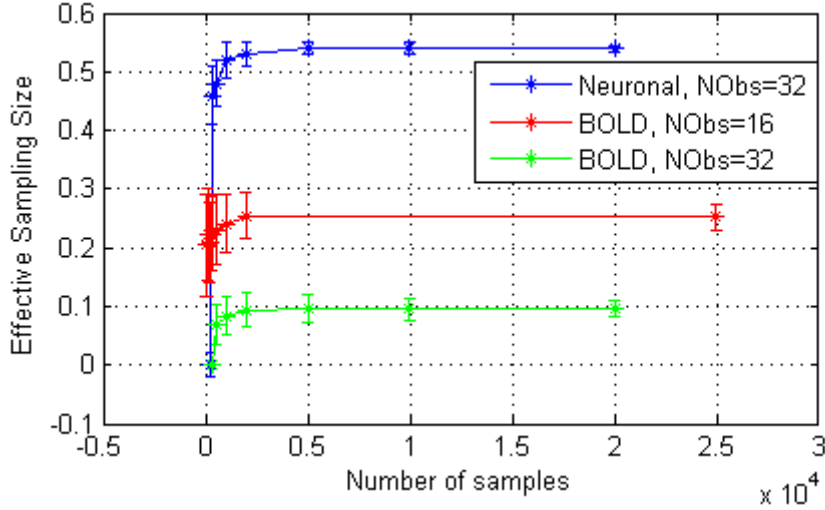


Figure 11: Comparison between the neuronal and BOLD smoother of their saturation of the effective sampling size, dependent on the number of particles. Here $\Delta t = 0.05$ and N_{obs} is given in the legend.

5.5 Number of integration steps

The sparse observations were sampled with regular intervals from the simulated data. Therefore the number of integration steps between each observation is defined to be $\frac{T}{N_{\text{obs}} \cdot \Delta t}$, where $T = 60$ in my case. To see how the ESS depends on the number of integration steps, the smoothers were computed with N_{obs} constant while Δt was varied, and vice versa. The results differ for the neuronal smoother and the BOLD smoother. The neuronal smoother shows very consistent behaviour, see figure 12, because all four simulations follow the same line of convergence. Their ESS show exponential growth and slowly converge to 1. Recall that if $\text{ESS} = 1 \iff \text{Var}(w^u) = 0 \iff u(z, t) = u^*(z, t)$, which means that the control $u(z, t)$ for the neuronal smoother is correctly parametrised by the linear parametrisation $h(z, t) = (z - \langle \mathbf{z} \rangle, 1)$.

The BOLD smoother shows different behaviour, as seen in figure 13. For every fixed N_{obs} the ESS converges to a distinct maximum value smaller than 1. It is thus not possible to gain any ESS for any N_{obs} , as it was for the neuronal smoother. This means that the optimal control cannot be reached and that thus the linear parametrisation of $u(x, t)$ is not correct for this case. However,

this does not mean that a linear controller is not able to improve the sampling efficiency: as I will argue in the Discussion section.

Figure 13 also shows a plot (the pink line) for varying N_{obs} and fixed $\Delta t = 0.05$, where the exponential growth does occur, although it is slower than for the neuronal smoother. This was expected, because this smoothing problem is more difficult due to the hemodynamic transformation. This pink line shows the convergence values for a range of $N_{\text{obs}} = \{2, \dots, 64\}$.

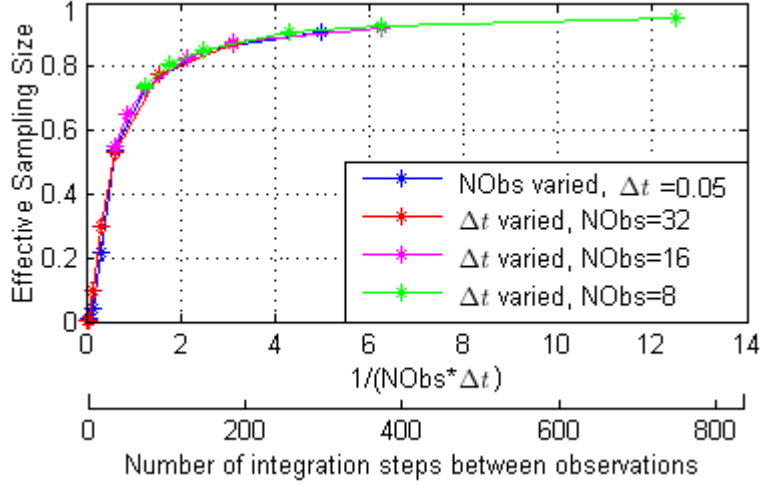


Figure 12: The relation between the ESS and $\frac{1}{N_{\text{obs}}\Delta t}$ and the number of integration steps for the neuronal smoother.

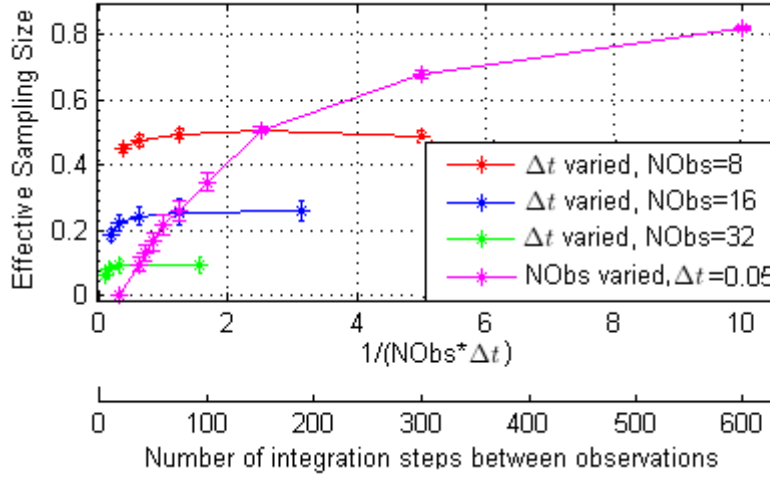


Figure 13: The relation between the ESS and $\frac{1}{N_{\text{obs}}\Delta t}$ and the number of integration steps for the BOLD smoother

5.6 Different controls

The controls for $N_{\text{obs}} = 8$ for both smoothers are plotted in figure 14. The top row shows the controls for the complete time series, while the bottom row shows a close-up of the controls near the first observation. Recall that $u(z, t) = M_t \cdot h(z, t) = (M_t^1, M_t^2) \cdot \begin{pmatrix} z - \langle \mathbf{z} \rangle \\ 1 \end{pmatrix}$. Here, the green line is M_t^1 and the blue line is M_t^2 . A negative M_t^1 steers particles towards the mean, which is what happens at observation times for both smoothers, for this minimizes the cost S^u (equation 15) if the mean itself is close to the observation. A clear difference between the two smoothers is observed in figures 14C and 14D. For the neuronal smoother, the M_t^1 term decays strongly exponentially towards the observation times, and resets to 0 right after the observation has passed. For the BOLD smoother, the M_t^1 term goes to local minima before observation times (taking the hemodynamic delay into account), but not so strongly as for the neuronal smoother. Importantly, the shape of the minimum is very different. Instead of an exponential decay followed by a reset, it shows a very broad peak which has no reset, but appears to be symmetric.

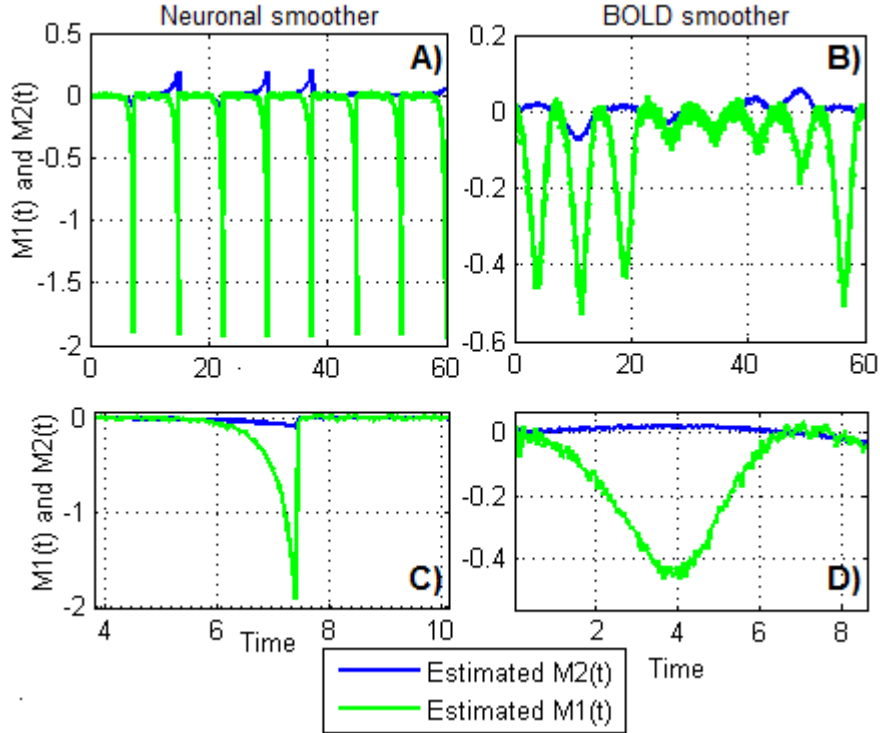


Figure 14: Comparison between the controls of the BOLD smoother (right) and the neuronal smoother (left) for $N_{\text{obs}} = 8$, $\Delta t = 0.05$, $N = 5000$. The first observation lies at $t = 7.5$.

5.7 Bootstrapping a large N_{obs}

For a large number of observations N_{obs} it can be difficult to bootstrap the APIS algorithm, because the problem is so difficult that very few particle trajectories are good enough to learn from. This problem can be overcome by lowering the learning rate η , though the computation will naturally also take longer (one needs more iterations to compensate for a small η). Note that this does not affect the convergence value of the ESS, as once it has bootstrapped the saturation will be at the same value.

However, for very high N_{obs} , for example $N_{\text{obs}} = 128$, a small η still is not sufficient to bootstrap, because the smoothing problem is so difficult that the ESS is stuck at a very small value ($\frac{1}{N} \leq \text{ESS} \leq 0.01$) from which it can not learn. In order to be able to bootstrap such a system, one can use a previously calculated control of a lower N_{obs} (e.g. for $N_{\text{obs}} = 64$) as an initial control. Then the particles are initially pushed into the right direction, such that the ESS is less probable to be too low to bootstrap.

By using this method I was able to bootstrap 128 observations for the neuronal smoother by using the control of $N_{\text{obs}} = 64$, and setting $\Delta t = 0.05$, $\eta = 0.003$, $N = 6000$, $K_{\text{max}} = 1000$, whereas I could not bootstrap this system without using a previously calculated control. I gained an ESS of 0.04 ± 0.02 , which is quite small. This is however in line with figure 12, therefore I expect to be able to gain any $\text{ESS} \leq 1$ if Δt is accordingly small. This however becomes very computationally intensive such that I could not lower Δt any further on my computer.

5.8 Changing variance during the BOLD transformation

The variance of the particles changes inconsistently during the BOLD transformation. This can already be seen in figure 7, but to emphasise this I have plotted the variances of the neuronal smoother in figure 15 and of the BOLD smoother in figure 16. For the neuronal smoother in figure 15 one can see that the variance remains approximately constant at 0.011–0.012 and at observation times goes down because the particles are pushed towards their mean. Importantly, there is no difference between the rest and activation period.

This is different for the BOLD smoother in figure 16. The left figure shows the variance of the smoothing distribution (of the neuronal state). During the rest period a similar effect occurs as for the neuronal smoother: the variance drops when it has to target an observation (taking the hemodynamic delay into account). During the activation period the variance actually increases a little. During the activation period the variance of the BOLD signal drops drastically. This means that the BOLD signal is less sensitive for fluctuations of the neuronal state during the activation period than during the rest period. Therefore the particles tend to stay closer to their mean in the BOLD signal.

Related to this, as I will argue in the Discussion, is the following discrepancy: in figure 7 it appears that in the rest period the smoother targets the observations better than during the activation period. This is emphasised in figure 17. Here two observations were presented, both were placed 0.01 above the stable values (respectively 0 and 0.033). It can clearly be seen that in the rest period the smoothing distribution targets the observations closer than during the activation period.

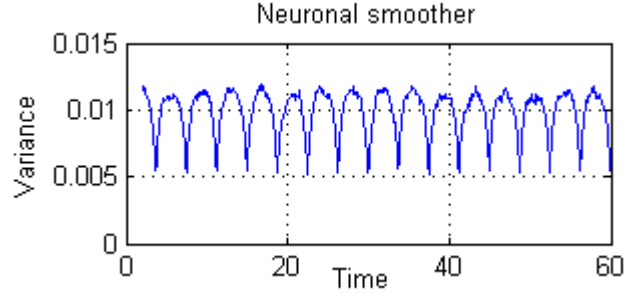


Figure 15: A figure showing the variance of the neuronal smoother corresponding to figure 6. The initial variance (at $t = 0$) was manually fixed at 0.04, which is left out in this plot for viewability purposes.

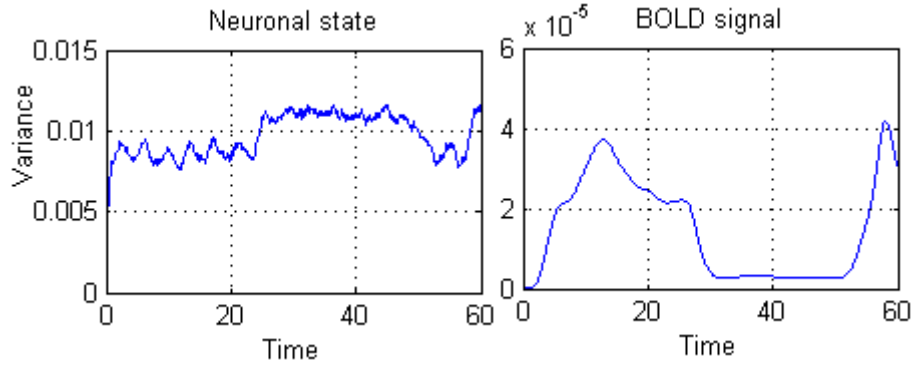


Figure 16: A figure showing the variance of the BOLD smoother corresponding to figure 7; the neuronal smoothing distribution (left) and the variance of the BOLD transformation of the particles (right).

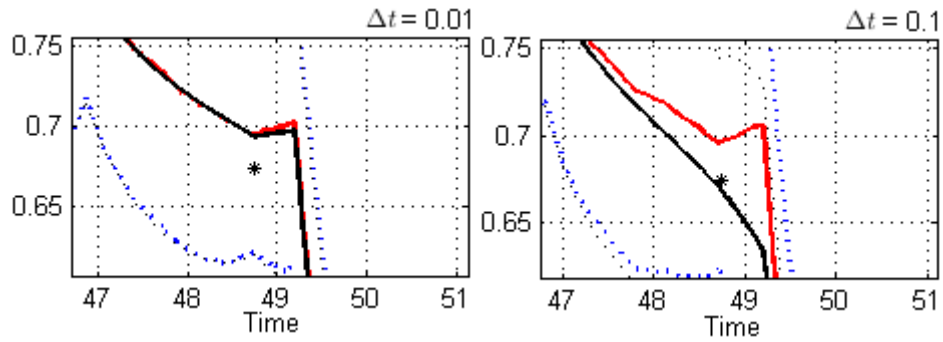


Figure 18: Two plots showing the Kalman method (black) and APIS method (red) for the neuronal smoother with $N_{\text{obs}} = 32, N = 5000$. The integration step differs: $\Delta t = 0.01$ (left) and $\Delta t = 0.1$ (right).

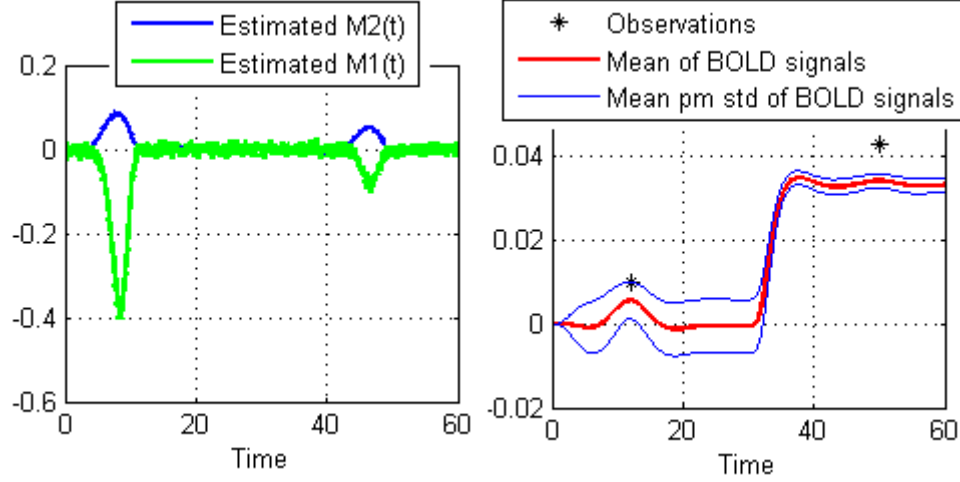


Figure 17: A smoothing simulation done with two observations, both 0.01 above the stable value (of the resting value 0 and of the activated period 0.033). One can clearly see that the smoothing distribution does not steer as strongly to the second observation as to the first one. The left figure shows the corresponding controller.

6 Discussion

6.1 Difference Kalman and APIS

In figures 9 and 10 a difference between the Kalman and APIS method was observed in the vicinities of the edges of the input function. After further evaluation this appears to be a numerical error of the Kalman smoother integration. By lowering Δt , the Kalman smoother comes closer to the APIS smoother. This is illustrated in figure 18 where in the left plot $\Delta t = 0.01$ and in the right plot $\Delta t = 0.1$. The Kalman smoother (black line) clearly approaches the APIS smoother as the integration step decreases.

6.2 Shapes of control terms

The shapes of the control terms differ greatly between the neuronal and the BOLD smoother. The neuronal smoother pushes all particles towards the mean at observation times, because only then the cost is acquired, which should be minimized (see equation 13 where $V(x_t, Y_t, t)$ is only nonzero at observation times). After the observation there is no need to constrain the particles anymore, so the M_t^1 term resets to approximately 0. The M_t^2 term only sometimes shows significant behaviour, in order to shift the mean of the particles to an observation.

The M_t^1 term of the BOLD smoother does not reset after it has reached a minimum, but slowly changes back to 0. I presume this is due to the effect of the hemodynamic delay, note that the the peak of the controller occurs approximately 3.5 seconds earlier than the observation does. Looking at figure 3, it is important to emphasise that although the BOLD signal is 3-5 seconds delayed

until it reaches its maximal magnitude, it already gradually builds up as soon as $z(t)$ is active.

To target the observation, the controller has to act 3.5 seconds in advance. However, all the process noise (in the neuronal state) that will occur in those 3.5 seconds will influence the eventual BOLD signal. I think this is the reason why M_t^1 is so broad, it tries to reduce the influence of the noise by controlling in the period of time between the minimum of M_t^1 and the observation time.

A similar effect occurs before the observation, where a long build-up before the peak of M_t^1 occurs. I presume that this is again necessary to suppress the noise, to ensure that the induced peak in the BOLD signal is at right place in time.

To show the need for the broad peaks I have plotted figure 19, which shows the controllers of BOLD smoother estimations with $N_{\text{obs}} = 16$ (left) and $N_{\text{obs}} = 32$ (right). Here it is visible that if there so many observations such that the controller is unable to have peaks which cover the hemodynamic delay on both sides, the controller stays in the minimum regime (instead of reducing the width of its peaks). Reducing the integration step Δt does not change this.

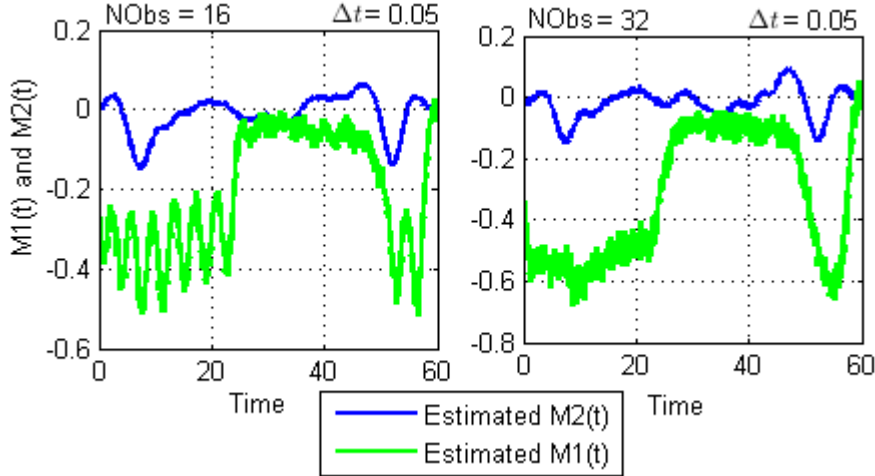


Figure 19: Two plots showing the control terms M_t^1 and M_t^2 for two BOLD smoothers with $N_{\text{obs}} = 16$ (left) and $N_{\text{obs}} = 32$ (right). It can be seen that because the peak of M_t^1 must be broad enough to take the hemodynamic delay into account, it is not able to reset to zero.

6.3 Number of integration steps

The simulations of the neuronal smoother all follow the same line of convergence in figure 12 and the simulations of the BOLD smoother do not in figure 13, I presume this happens because of the following: By lowering Δt , the control terms M_t^1 and M_t^2 are also estimated more precisely. By looking at the neuronal smoother in figure 14C, one can see that this does not matter so much for slow-changing regimes such as $t \in [4, 6]$. It does however matter for fast-changing regimes such as $t \in [7, 7.5]$. Moreover, the integration steps just before the observation time $t = 7.5$ determine how deep the peak of M_t^1 will be (as it is an estimation of the true peak of $u^*(x, t)$). When Δt is smaller, this depth is

estimated more accurately (i.e. it becomes deeper), so that the control $u(x, t)$ comes closer to the optimal control $u^*(x, t)$, and the ESS grows towards 1.

It is then clear why the ESS saturates for the BOLD smoother in figure 13: the shape of the controller term M_t^1 in figure 14D is not a strong decay, which would need a small Δt in order to be estimated accurately. Instead, the peak is broad and continuous, without the sudden reset to 0. Because it is slow-changing, the ESS saturates for a certain Δt , because the noise of the controller is larger than the change in the controller induced by a more precise estimation. Because the ESS cannot become 1, $Var(w^u) > 0$ and thus the linear controller $u(x, t)$ is not the optimal controller $u^*(x, t)$, as one would expect for such a nonlinear model.

But still, for $N_{\text{obs}} \leq 32$ an $ESS \geq 0.1$ is reached, which is a significant gain compared to a non-controlled system, which has ESS typically in the range of $\frac{1}{N} \leq ESS \leq 0.01$. So in this case, it is still useful to control the dynamics with a linear parametrisation. But for a higher N_{obs} , the gain in ESS is very low (and can not be improved) or one is not able to bootstrap the algorithm at all. I have not tried to use a different controller, but this would be the first step of future research in order to improve the ESS of the BOLD smoother.

6.4 Changing variance after BOLD transformation

The fact that the variance of the particles reduces after the BOLD transformation could explain why in figure 14 the M_t^1 term does not show such strong peaks during the activated period; the variance has already reduced, so there is less need for the controller to push the particles towards their mean.

This might also explain the increased variance of the neuronal state during the activated period in figure 15; the particles are less controlled such that their variance increases.

I presume this explains the discrepancy in figure 17: The fact that the variance has reduced in the BOLD signal means that changes in the neuronal state induce relatively small changes in the BOLD signal during the activated period. Therefore it is more difficult for the controller to steer particles towards an observation (i.e. the controller needs to be larger in magnitude in order to gain the same effect as during the rest period). Apparently, it is more cost-expensive in equation 15 to further improve the control than to accept the deviation from the observation. Therefore the system learns to target the observations not as close as during the rest period.

It would be interesting to investigate this matter further, in order to see why precisely the variance reduces during the BOLD transformation.

6.5 Conclusion

I have used the APIS method to smooth neuronal activity based on two differential observational models. The neuronal smoother is a linear Gaussian model and the BOLD smoother is a nonlinear model, defined by the DCM for fMRI model. The two smoothers were compared and it was found that a linear controller $u(x, t)$ was the optimal control for the neuronal smoother but not for the BOLD smoother. The optimal control corresponds to an ESS of 1 and by

evaluating the shapes of the control terms I reasoned that because of the hemodynamic delay the ESS of the BOLD smoother can not be further improved using a linear controller. It would very interesting to investigate which controllers are closer or even equivalent to the optimal control.

I was able to calculate the smoothing distribution for the BOLD smoother, which represents a realistic system. Due to the linear parametrisation of the controller I was limited to $N_{\text{obs}} \leq 32$ observations in a time frame of 60 seconds. I based my simulations on [Murray and Storkey, 2011], where a scanning time (TR) of $4.1s$ was used. This is equivalent to approximately 15 observations in 60 seconds time. However, longer runs (approximately 320 seconds) were performed in [Murray and Storkey, 2011], which would make the smoothing problem more difficult as more observations would be added. It would be interesting to develop a method to deal with this many observations, just as was done with $N_{\text{obs}} = 128$ for the neuronal smoother.

In section 2 I explained that the smoothing distribution is very useful in order to perform parameter estimation, it would naturally be interesting to put this to the test and see how good the estimations would be.

Acknowledgements I would like to greatly thank my daily supervisor, Hans Ruiz, for the time and effort he took to help me. Also I would like to thank my supervisor Bert Kappen and the rest of the SNN group for their warm welcome and advice. In the past three months I have become really enthusiastic about the Machine Learning field and conducting research in general.

References

- Richard B Buxton and Lawrence R Frank. A model for the coupling between cerebral blood flow and oxygen metabolism during neural stimulation. *Journal of Cerebral Blood Flow & Metabolism*, 17(1):64–72, 1997.
- Richard B Buxton, Eric C Wong, and Lawrence R Frank. Dynamics of blood flow and oxygenation changes during brain activation: the balloon model. *Magnetic resonance in medicine*, 39(6):855–864, 1998.
- Richard B Buxton, Kâmil Uludağ, David J Dubowitz, and Thomas T Liu. Modeling the hemodynamic response to brain activation. *Neuroimage*, 23:S220–S233, 2004.
- Justin R Chumbley, Karl J Friston, Tom Fearn, and Stefan J Kiebel. A metropolis–hastings algorithm for dynamic causal models. *Neuroimage*, 38(3):478–487, 2007.
- Stefan Frässle, Klaas Enno Stephan, Karl John Friston, Marlena Steup, Sören Krach, Frieder Michel Paulus, and Andreas Jansen. Test-retest reliability of dynamic causal modeling for fmri. *NeuroImage*, 117:56–66, 2015.
- Karl J Friston, Andrea Mechelli, Robert Turner, and Cathy J Price. Nonlinear responses in fmri: the balloon model, volterra kernels, and other hemodynamics. *NeuroImage*, 12(4):466–477, 2000.
- Karl J Friston, Lee Harrison, and Will Penny. Dynamic causal modelling. *Neuroimage*, 19(4):1273–1302, 2003.
- Robert L Grubb, Marcus E Raichle, John O Eichling, and Michel M Ter-Pogossian. The effects of changes in paco₂ cerebral blood volume, blood flow, and vascular mean transit time. *Stroke*, 5(5):630–639, 1974.
- Martin Havlicek, Alard Roebroeck, Karl Friston, Anna Gardumi, Dimo Ivanov, and Kamil Uludag. Physiologically informed dynamic causal modeling of fmri data. *NeuroImage*, 122:355–372, 2015.
- Daniel J Jacobsen, Lars Kai Hansen, and Kristoffer Hougaard Madsen. Bayesian model comparison in nonlinear bold fmri hemodynamics. *Neural computation*, 20(3):738–755, 2008.
- Seong-Gi Kim and Seiji Ogawa. Biophysical and physiological origins of blood oxygenation level-dependent fmri signals. *Journal of Cerebral Blood Flow & Metabolism*, 32(7):1188–1206, 2012.
- Yury Koush, Maria Joao Rosa, Fabien Robineau, Klaartje Heinen, Sebastian W Rieger, Nikolaus Weiskopf, Patrik Vuilleumier, Dimitri Van De Ville, and Frank Scharnowski. Connectivity-based neurofeedback: dynamic causal modeling for real-time fmri. *Neuroimage*, 81:422–430, 2013.
- David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.
- Joseph B Mandeville, John JA Marota, C Ayata, Greg Zaharchuk, Michael A Moskowitz, Bruce R Rosen, and Robert M Weisskoff. Evidence of a cerebrovascular postarteriole windkessel with delayed compliance. *Journal of Cerebral Blood Flow & Metabolism*, 19(6):679–689, 1999.
- AC Marreiros, Stefan J Kiebel, and Karl J Friston. Dynamic causal modelling for fmri: a two-state model. *Neuroimage*, 39(1):269–278, 2008.
- Lawrence Murray and Amos Storkey. Particle smoothing in continuous time: A fast approach via density estimation. *Signal Processing, IEEE Transactions on*, 59(3):1017–1026, 2011.
- P Osório, P Rosa, C Silvestre, P Figueiredo, et al. Stochastic dynamic causal modelling of fmri data with multiple-model kalman filters. *Methods Inf Med*, 54(3):232–239, 2015.

- Will D Penny, KE Stephan, A Mechelli, and KJ Friston. Comparing dynamic causal models. *NeuroImage*, 22(3):1157–1172, 2004.
- Marcus E Raichle. Behind the scenes of functional brain imaging: a historical and physiological perspective. *Proceedings of the National Academy of Sciences*, 95(3):765–772, 1998.
- Jorge J Riera, Jobu Watanabe, Iwata Kazuki, Miura Naoki, Eduardo Aubert, Tohru Ozaki, and Ryuta Kawashima. A state-space model of the hemodynamic approach: nonlinear filtering of bold signals. *NeuroImage*, 21(2):547–567, 2004.
- Hans-Christian Ruiz and Hilbert Johan Kappen. Particle smoothing of diffusion processes with linear computational cost. 2015. preprint 30-10-2015.
- Simo Särkkä. *Bayesian filtering and smoothing*. Number 3. Cambridge University Press, 2013.
- Frank Scharnowski, Maria Joao Rosa, Narly Golestani, Chloe Hutton, Oliver Josephs, Nikolaus Weiskopf, and Geraint Rees. Connectivity changes underlying neurofeedback training of visual cortex activity. *PloS one*, 9(3):e91090, 2014.
- Klaas Enno Stephan, Lars Kasper, Lee M Harrison, Jean Daunizeau, Hanneke EM den Ouden, Michael Breakspear, and Karl J Friston. Nonlinear dynamic causal models for fmri. *Neuroimage*, 42(2):649–662, 2008.
- Sep Thijssen and HJ Kappen. Path integral control and state-dependent feedback. *Physical Review E*, 91(3):032104, 2015.