BOLD Parameter Estimation Using Particle Filters

Micah Chambers

Virginia Polytechnic Institute and State University

Contents

Chapter 1

Introduction

For the past twenty years Functional Magnetic Resonance Imaging (FMRI) has been at the forefront of cognitive research. Despite it's limited temporal resolution, FMRI is the standard tool for localizing neural activation. Whereas other methods of analyzing neural signals can be invasive or difficult to acquire, FMRI is relatively quick and cheap, and its analysis relatively straight forward. In the past fifteen years, a steady stream of studies have built on the original Blood Oxygen Level Dependent (BOLD) signal derivation first described by [Ogawa et al., 1993]. The seminal work by [Buxton et al., 1998] attempted to explain the time evolution of the BOLD signal using a windkessel model to describe the time local changes in Deoxygenated Hemoglobin content. Various papers made substantial improvements to this model until [Friston et al., 2000] brought all the changes together into a single complete set of equations. And while there have been numerous adaptations in the model, many of them summarized in [Deneux and Faugeras, 2006], even the most basic version has been shown to have less bias error than the convolution based "Canonical Hemodynamic Model" [Deneux and Faugeras, 2006], [Handwerker et al., 2004]. On the other hand many of the BOLD signal models have far more parameters than a simple scaling parameter. In fact, the number of parameters range from seven [Riera et al., 2004] to 50 [Behzadi and Liu, 2005] per signal time course; a signal which may be as short as 100 samples long. Thus, even in a small FMRI image of the brain (20x20x20), the number of parameters could easily exceed 10,000. Clearly this number of parameters presents a significant risk of being under-determined and could suffer catastrophic variance error; to make no mention of the computation cost. In this work a method of countering these problem is presented with the use of a particle filter.

1.1 Overview

Detecting neural activity using the changes in FMRI images is based on the so called Blood Oxygen Level Dependent (BOLD) signal. The BOLD signal is caused by minute changes in the ratio of Deoxygenated Hemoglobin to Oxygenated Hemoglobin in blood vessels throughout the brain. Because Deoxygenated Hemoglobin (DHb) is paramagnetic, higher concentrations attenuate the signal detected during T2-weighted Magnetic Resonance Imaging (MRI) techniques. The most common FMRI imaging technique, due to its rapid repetition time (TR), is Echo Planar Imaging (EPI). When axons becomes active, a large amount of ions quickly flow out of the cell. In order for this action potential to occur again (and thus for the neuron to fire again), an active pumping process must move ions back into the axon. This process of recharging the axon requires extra energy, which temporarily increases the metabolic rate of oxygen. On a massive scale (cubic millimeter) this activation/recharge process happens continuously. However, when a particular region of the brain is very active, the action potentials occur significantly more often, resulting in significant local increase of the Cerebral Metabolic Rate of Oxygen (CMRO2). Thus, blood vessels in a very active area will tend to have less oxygenated hemoglobin (due to the increased rate at which oxygen is being consumed), and more deoxygenated hemoglobin, resulting in an attenuated FMRI signal. In compensation for activation, muscles that control blood vessels relax in that region to allow more blood flow, which actually overcompensates. This ultimately results in lower than average concentration of deoxyhemoglobin. Thus, the BOLD signal consists of a short initial dip in the MR signal, followed by a prolonged increase in signal that slowly settles out. It is this overcompensation that is the primary signal detected with FMRI imaging. This cascade of events is believed to consist of increased the local metabolism, blood flow, blood volume, and oxygenated hemoglobin. The differences in onsets of these various effects is what causes the overcompensation that is observable in FMRI. Unfortunately, FMRI has no inherent unit of measurement, and thus signal levels are all relative: within a particular person, scanner and run.

1.2 FMRI

Magnetic Resonance Imaging, MRI, is a method of building 3D images non-invasively, based on the difference between nuclear spin relaxation times in various molecules. First, the subject is brought into a large magnetic field which causes nuclear spins to align. Radio Frequency (RF) signals may then be used to excite nuclear spin away from the base alignment. As the nuclei precess back to the alignment of the magnetic field, they emit detectable RF signals. Conveniently, the excitation of nuclear spins return their original state at different rates, called the T1 relaxation time, depending on the atoms excited. Additionally, the coherence of the spins also decay differently (and quite a bit faster than T1 relaxation) based on the properties of the region. This gives two

primary methods of contrasting substances, which form the basis of T1 and T2 weighted images. Additionally, dephasing occurs at two different rates, the T2 relaxation time, which is unrecoverable, and $T2^*$ relaxation, which is much faster, but possible to recover from via special RF signals. T1 relaxation times are typically on the order of seconds if a sufficiently strong excitation was applied. In order to rapidly acquire entire brain images, as done in Functional MRI, a single large excitation pulse is applied to the entire brain, and the entire volume is acquired in a single T1 relaxation period. Because the entire k-space (spatial-frequency) volume is acquired from a single excitation, the signal-to-noise-ratio is very low in this type of imaging (Echo Planar Imaging).

Increasing the spatial resolution of EPI imaging necessarily requires more time or faster magnetic field switching. Increasing magnet switching rates though is difficult, because it can result in more artifacts, or even lower signal to noise ratios. The result is that at *best* FMRI is capable of 1 second temporal resolution. The signal is further diluted because each voxel contains the signal from a large number of neurons, capillaries and veins. Thus, the FMRI signal, which is sensitive to the chemical composition of materials, is the average signal from various types of tissue in addition to the blood. As mentioned in section 1.1, and explored in depth in section 1.3, the usefulness of FMRI comes from the discerning of changes in Deoxyhemoglobin/Oxyhemoglobin. Therefore, it is necessary to assume that in the short term the only chemical changes will be in capillary beds feeding neurons. In practice this may not be the case, for instance near significant veins, and it may explain some of the noise seen in FMRI imaging (see **??**. Because MRI lacks units and certain areas will have a higher base MR signal, all FMRI studies deal with percent change from the base signal; rather than raw values. This also removes most of the structural data which is not helpful in determining neural activity.

1.3 BOLD Physiology

It is well known that the two types of hemoglobin act as a contrast agents in EPI imaging [Buxton et al., 1998], [Weisskoff et al., 1994], [Ogawa et al., 1993], however the connection between Deoxyhemoglobin/Oxygenated Hemoglobin and neural activity is non-trivial. Intuitively, increased metabolism will increase Deoxyhemoglobin, however blood vessels are quick to compensate by increasing local blood flow. Increased inflow, accomplished by loosening capillary beds, precedes increased outflow, driving increased blood storage capacity. Since the local MR signal depends on the ratio of Deoxyhemoglobin to Oxygenated Hemoglobin, increased volume of blood can effect this ratio if metabolism doesn't exactly match the increased inflow of oxygenated blood. This was the impetus for the ground breaking balloon model ([Buxton et al., 1998]) and windkessel model ([Mandeville et al., 1999a]). These works derive from first principals the changes in deoxyhemoglobin ratio and volume of capillaries based on a given flow. These were the first two attempts to quantitatively account

for the shape of the BOLD signal as a consequence of the lag between the cerebral blood volume (CBV) and the inward cerebral blood flow (CBF). In fact [Buxton et al., 1998] went so far as to show that a simple, well chosen blood flow waveform coupled with a square wave cerebral metabolic rate of oxygen (CMRO2) curve, in the context of a balloon model, could fully account for the BOLD signal.

Although [Buxton et al., 1998] showed that a well chosen flow waveform could explain much of the BOLD signal, there was still a matter of proposing a realistic waveform for the CBF and for the CMRO2. [Friston et al., 2000] gave a reasonable and simple expression for CBF input, f, based on a flow inducing signal, s,

$$\dot{s} = \epsilon u(t) - \frac{s}{\tau_s} - \frac{f-1}{\tau_f} \tag{1.1}$$

$$\dot{f} = s \tag{1.2}$$

where ϵ is a neuronal efficiency term, $u(t)$ is a stimulus, and τ_f, τ_s are both time constants. In [Buxton et al., 2004] the final piece of the balloon model was put into place, by describing the CMRO2 as a constant multiple of the CBF (the inflow of blood). This completed the basic balloon model, and was well summarized in [Riera et al., 2004].

$$\dot{v} = \frac{1}{\tau_0}(f - v^{\alpha}) \tag{1.3}$$

$$\dot{q} = \frac{1}{\tau_0}(\frac{f(1-(1-E_0)^f)}{E_0} - \frac{q}{v^{1-1/\alpha}}) \tag{1.4}$$

where v is normalized cerebral blood volume (CBV), and q is the normalized local deoxyhemoglobin/oxygenated hemoglobin ratio, E_0 is the resting metabolic rate and α is Grubb's parameter controling the balloon model. [Obata, 2004] refined the readout equation of the BOLD signal based on the deoxyhemoglobin content (q) and local blood volume (v), resulting in the final BOLD equation:

$$y = V_0((k_1+k_2)(1-q) - (k_2+k_3)(1-v)) \tag{1.5}$$

$$k_1 = 4.3 \times \nu_0 \times E_0 \times TE = 2.8 \tag{1.6}$$

$$K_2 = \epsilon_0 \times r_0 \times E_0 \times TE = .57 \tag{1.7}$$

$$k_3 = \epsilon_0 - 1 = .43 \tag{1.8}$$

Where $\nu_0 = 40.3s^{-1}$ is the frequency offset in Hz for fully deoxygenated blood (at 1.5T), $r_0 = 25s^{-1}$ is the slope relating change in relaxation rate with change in blood oxygenation, and $\epsilon_0 = 1.43$ is the ratio of signal MR from intravascular to extravascular at rest. Although, obviously these constants change with experiment (TE, ν_0, r_0), patient, and brain region (E_0, r_0), often the estimated values taken from [Obata, 2004]

are used as constants ($k_1 + k_2 = 3.4$, and $k_2 + k_3 = 1$) in 1.5 Tesla studies.. While this model is in a sense complete, it is far from perfect. The major problem often brought up with this version of the BOLD model is that it does not represent the so called "post-stimulus undershoot" well. The post-stimulus undershoot is the name for a prolonged sub-normal BOLD response for a period of 10 to 60 seconds after stimulus has ceased ([Chen and Pike, 2009a], [Mandeville et al., 1999b]).

There are two theories for the cause of the post stimulus undershoot. Recall that a lower than base signal means that there is an increased deoxyhemoglobin content in the voxel. The first and simplest explanation is that the post-stimulus undershoot is caused by a prolonged increase in CMRO2 after CBV and CBF have returned to their base levels. This theory is justified by quite a few studies that show CBV and CBF returning to the baseline before the BOLD signal ([Frahm et al., 2008], [Donahue et al., 2009], [Buxton et al., 2004], [Lu et al., 2004], [Shen et al., 2008]). Unfortunately, because of limitations on FMRI and en vivo CBV/CBF measurement techniques it is difficult to isolate whether CBF and CBV truly have returned to their baseline. Other research seems to indicate that there can be a prolonged residual supernormal CBV ([Mandeville et al., 1999b], [Behzadi and Liu, 2005], [Chen and Pike, 2009b]), although none of these papers completely rule out the possiblity of increased CMRO2. Additionally, in [Yacoub et al., 2006], it was found that the post-stimulus undershoot varried across regions of the brain, which could further explain the contradictions found elsewhere. [Chen and Pike, 2009b] makes a compelling case that most of the post stimulus undershoot could be explained be a prolonged CBV increase, and a prolonged CBF undershoot, and that many of the previous measurements showing a quick recovery of CBV may have been dominated arterial CBV's return to baseline.

Because of the significant possility of a completely independent CMRO2, extremely complex models for metabolism exist ([Zheng et al., 2005]), although most recent studies have focused on their ability to explain the prolonged BOLD post stimulus undershoot [Zheng et al., 2005], [Buxton et al., 2004]. This is because [Buxton et al., 2004] and later [Riera et al., 2004] showed that the main portion of the signal may be accurately estimated by a simple blood flow locked expression of the CMRO2.. Although [Deneux and Faugeras, 2006] did not deal extensively with prolonged post stimulus undershoot, the comparisons made in that publication showed minimal improvement from separate expressions of CMRO2, in comparison to the much increased complexity. [Deneux and Faugeras, 2006] did show that by simply adding viscoelastic terms, first proposed in [Buxton et al., 2004], that a slowed return to baseline for the BOLD signal is possible to model. However, viscoelastic effects primarily control CBV, which, as mentioned already, many studies have claimed cannot be responsible for the BOLD post-stimulus undershoot. Another extensive model that attempts to quantify the post-stimulus undershoot is the compliance model proposed by [Behzadi and Liu, 2005]. Although through a somewhat different means than the [Zheng et al., 2005] and [Buxton et al., 2004] papers, its possible the increased model flexibility ultimately is the key reason for the improvements, as opposed to increased plausibility.

Because of these controversies, and because this is the first time a particle filter has been used for this problem in this way, our aim to keep the model simple was best met by using the original balloon model with the possible addition of the visco-elastic effects from [Buxton et al., 2004].

Even more advanced versions of the Balloon model exist. In fact [Buxton et al., 2004] introduced several additional state variables, including the CMRO2, the O2 extraction fraction, which is closely related to CMR02, and the neural response, which causes the stimulus to decay toward some steady state value. The neural response is intended to emulate neural habituation, wherein neurons become less sensitive to a prolonged stimulus. While these advanced may be more capable of capturing a more exact version of the BOLD signal, the difference between the model will often times be below the noise floor. In essence this is a classic bias-variance dilemma: at some point increased model flexibility, and thus variance, is not worth the decrease in model bias. For now it remains to be seen where this line may be drawn in the BOLD signal, although [Deneux and Faugeras, 2006] does not show a significant improvement from the additional parameters added in [Buxton et al., 2004].

1.4 Previous Studies of Parameters

There have been quite a few efforts to quantify the parameters of the various BOLD models. Although [Buxton et al., 1998] and [Friston et al., 2000] both proposed physiologically reasonable values for the model parameters, [Friston et al., 2002] was the first paper to calculate the parameters based on actual FMRI data. In that paper, Friston et. al. used a variation of Expectation Maximization to find a normal distribution for the parameters:

$$\begin{aligned}
\epsilon &= N(.54, .1^2) \\
\tau_s &= N(1.54, .25^2) \\
\tau_f &= N(2.46, .25^2) \\
\tau_0 &= N(.98, .25^2) \\
\alpha &= N(.33, .45^2) \\
E_0 &= N(.34, .1^2) \\
V_0 &= .03(not\ estimated)
\end{aligned}$$

Since then, several other methods of have been used to calculate BOLD parameters from FMRI timecourses. In [Riera et al., 2004], a maximum likelihood method for innovation processes was used, as described by [Ozaki, 1994]. [Ozaki, 1994] uses a similar construction to a Kalman filter, to break the time series into a

series of innovations, for which Maximum Likelihood was performed. While this is in some since the "right" way to find the solution, it comes with several caveats. First, every step in parameter space requires a recalculation of all the state variables. With two or three parameters this is fine, more than that, and calculations could go on indefinitely. Second, it still assumes the parameters and noise are Gaussian, and will only be optimal in that case. Third, depending on the nonlinearities present in the system, local minima may be extremely common. Later [Hu et al., 2009] used an Unscented Kalman Filter over all the parameters and state variables to find the parameter set/variable time series. While this method has the drawback of not necessarily being optimal, unfortunately there is no general optimal solution to non-linear non Gaussian models. Hu et. al.'s technique also will run significantly faster than ML based techniques, since it does not require recalculating the entire timeseries for every step in parameters. Both [Hu et al., 2009] and [Friston et al., 2002] came to results very similar to the expected values stated in [Buxton et al., 1998] and [Friston et al., 2000]. One potential problem with all these techniques are that the depend heavily on the priors. The starting point of the parameters could have a huge impact on the results, and while one can be hopeful that this isn't the reason for the agreement between [Friston et al., 2000] and later results, there is no way to know.

In [Johnston et al., 2008], a hybrid particle filter/gradient descent algorithm was used to simultaneously derive the static (classicly called parameters) and dynamic parameters (classically known as state variables). Essentially a particle filter is used to calculate the state variables at some time, and then the estimated distribution of the particles was used to find the most likely set of parameters that would give that distribution of state variables. [Johnston et al., 2008] comes to a very different set of parameter estimates as compared to the original [Friston et al., 2000] guesses:

$$
\begin{aligned}
epsilon &= .069 \pm .014 \\
tau_s &= 4.98 \pm 1.07 \\
tau_f &= 8.31 \pm 1.51 \\
tau_0 &= 8.38 \pm 1.5 \\
alpha &= .189 \pm .004 \\
E_0 &= .635 \pm .072 \\
V_0 &= .0149 \pm .006
\end{aligned}
$$

Notably, the time constants are significantly longer. This could be a result of some preprocessing to the stimulus timeseries performed in [Friston et al., 2002] and later works but not in [Johnston et al., 2008], or it could be that [Johnston et al., 2008] depended less on the priors.

Its possible, although unlikely that the balloon model is not possible to learn without detrimental cost in variance error. I do not think this is the case, however, and I will work to dispell this possibility in the results with simulated time series.

1.5 Noise

Thus, despite some small discrepancies, the cause of the BOLD signal is relatively well known. However, FMRI doesn't detect this happening in one neuron or one capillary bed, but rather as the aggregate over a space of several millimeters. Though local neurons act "together" (i.e. around the same time), the density of neurons, the density of capillaries, and slight differences in activation across a particular voxel can all lead to signal attenuation or noise.

A particularly insidious type of noise present in FMRI is a low frequency drift, often characterized as a Wiener process ([Riera et al., 2004]). Though not present in all regions, as much as ten to fifteen percent of voxels can be affected, thus it is prevalent enough to cause significant inference problems [Tanabe et al., 2002]. It is still not clear what exactly causes this noise noise comes from, although it is possible it is the result of magnets heating up, or some distortion in magnetic fields [Smith et al., 2007]. It is clear that this drift signal is not solely due to a physiological effects, given its presence in cadavers and phantoms [Smith et al., 1999]. Interestingly, it is usually spatially correlated, and more prevalent at interfaces between regions, although by no means limited to such areas. Often times the noise is written off as movement, however given that co-registration of volumes to a single time point is standard, this seems unlikely. Regardless, the problem mandates the use of some sort of high pass filter to make models useful at all [Smith et al., 2007].

Given the complexity of the noise, it seemed prudent to characterize it. In order to do so we took sample time-series during resting state, which should theoretically be all noise, and analyzed the type of noise present. Both Q-Q plots, and the autocorrelation are useful in determining the noise distribution observed. Because most methods (including the one used in this paper) assume the noise realization are independent of each other, the auto- correlation is of particular interest. Gaussianity is also a common assumption made in studies of FMRI data, though that assumption is not made here. Anytime the assumptions are violated, the error rate will be significantly harder to predict and may in fact skyrocket. Because the noise is often considered to have Wiener noise with Gaussian steps, as first described in [Riera et al., 2003], we also performed the same tests again, but on the steps, rather than the direct measurements.

Finally, removal of the so called "drift" is often performed with some variation of a high pass filter, so we checked the noise distribution after applying such a filter (in this case the subtraction of a spline, see section 4.2). Here we wanted to know whether the subtraction of a spline from the data will result in complete removal of

Wiener noise, and whether the resulting timeseries better conforms to the common assumption of Gaussianity.

Figure 1.1 shows the results with a regression line fit to the points on the Q-Q plot. Recall that in a Quartile-Quartile (Q-Q) plot, if the points plotted on the x-axis and the points on the y-axis come from the same *type* of distribution then all the points will fall on a line. Differences in the variance will cause the line to have a slope other than 1, while differences in the expected value will cause the fitted line to be shifted. In these Q-Q plots, the points are being compared to the standard Gaussian distribution, so the quality of the line fit determines how closely the points fit the Gaussian distribution. Note that in Figure 1.1 the points have all been normalized (changed to percent difference).

A Wiener process should still conform to the normal distribution, albeit with a variance proportional to the run-time. Note that 1.1(a) and 1.1(b) are relatively well described by a Gaussian process with a small autocorrelation, 1.1(c) and 1.1(d) aren't. In particular the tails don't seem to fit, which could be well explained by the much more significant drift in the signal. Thus, its possible a Wiener process does not well describe the error, since overall the distribution does not tend toward a Gaussian. Still, the steps still seem to be relatively normal, meaning at the very least that the process is close to Wiener. As one would expect most of the autocorrelation disappears for the step data, indicating that the steps are relatively independent. Therefore, the assumption of I.I.D. Normal steps is relatively good, even though the accumulated result seems to be less varied than Wiener.

[todo analysis after detrending]

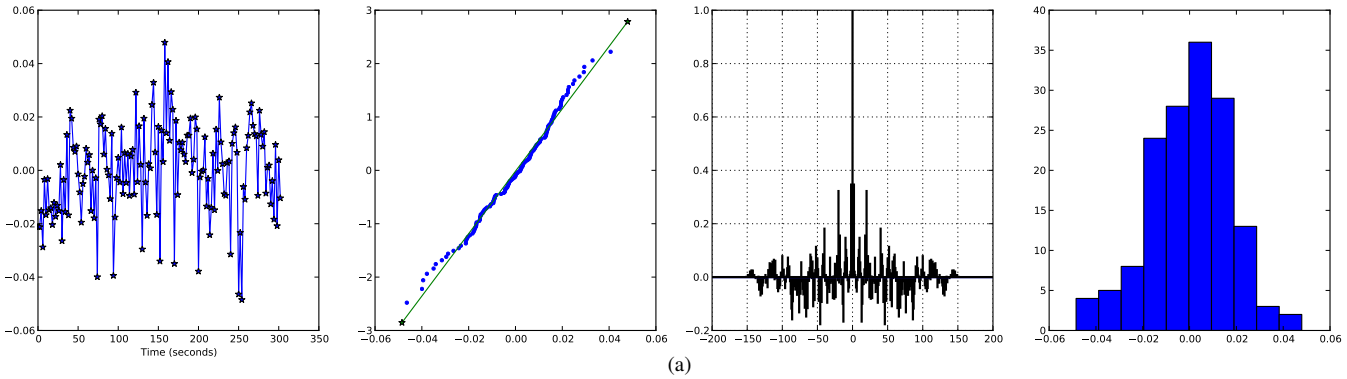
1.6 Analysis of the BOLD Signal

Several studies have endeavored to analyze the BOLD equations, which we will review here. The most obvious analysis is to calculate the steady state signal, when u is held constant long enough for all the transients to settle out.

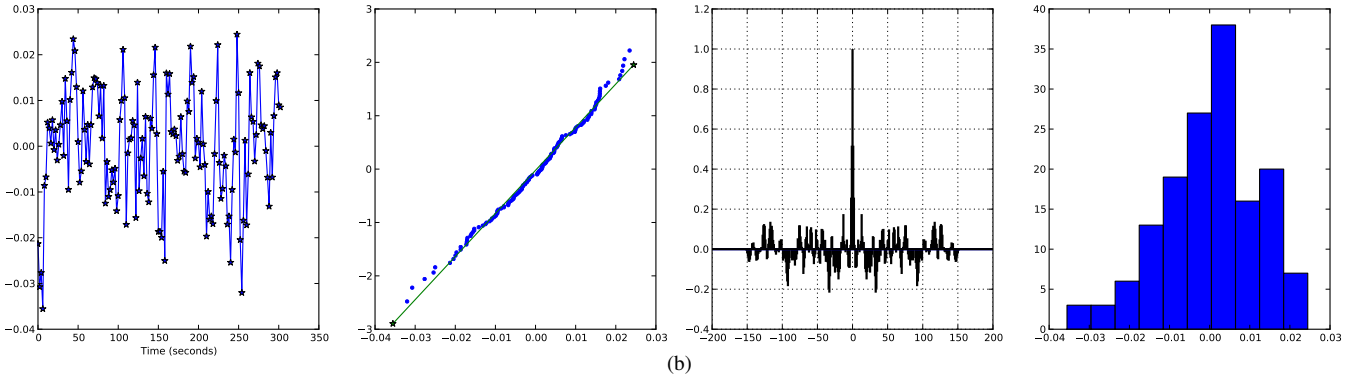
$$\begin{aligned} s_{ss} &= 0 \\ f_{ss} &= \tau_f \epsilon u + 1 \\ v_{ss} &= (\tau_f \epsilon u + 1)^{\alpha} \\ q_{ss} &= \frac{(\tau_f \epsilon u + 1)^{\alpha}}{E_0}(1 - (1 - E_0)^{1/(\tau_f \epsilon u + 1)}) \\ y_{ss} &= V_0((k_1 + k_2)(1 - q_{ss}) - (k_2 + k_3)(1 - v_{ss})) \end{aligned} \tag{1.9}$$

todo [Image with two different α's]

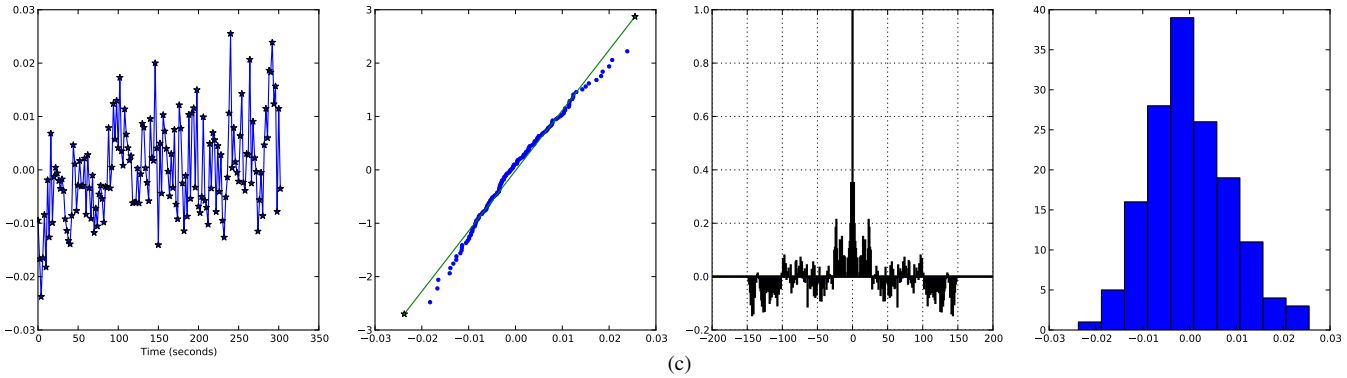
[image comparing the results of 10% changes in various signals]



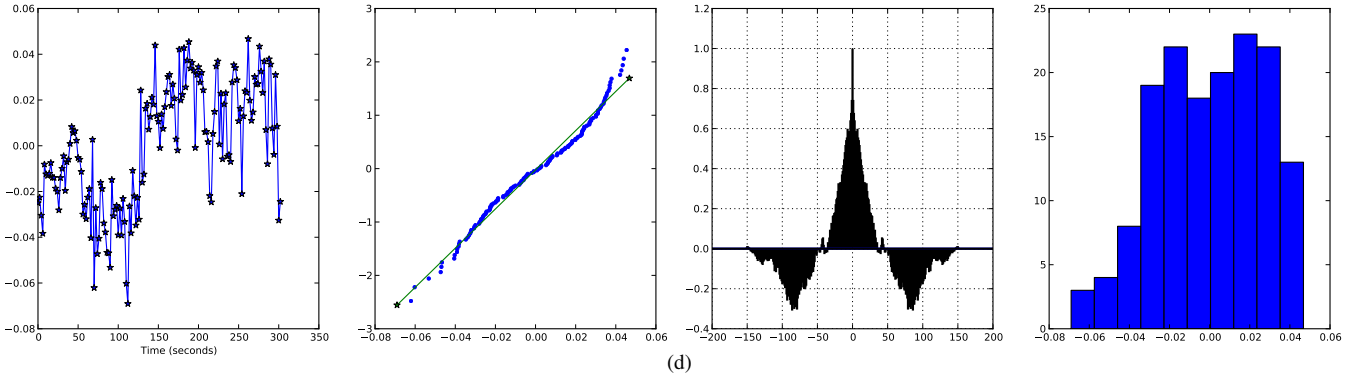
(a)



(b)

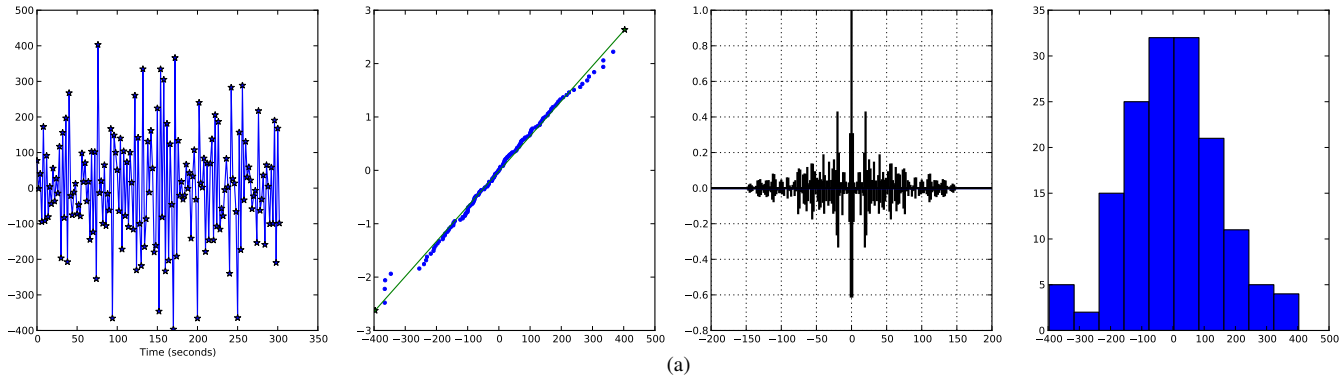


(c)

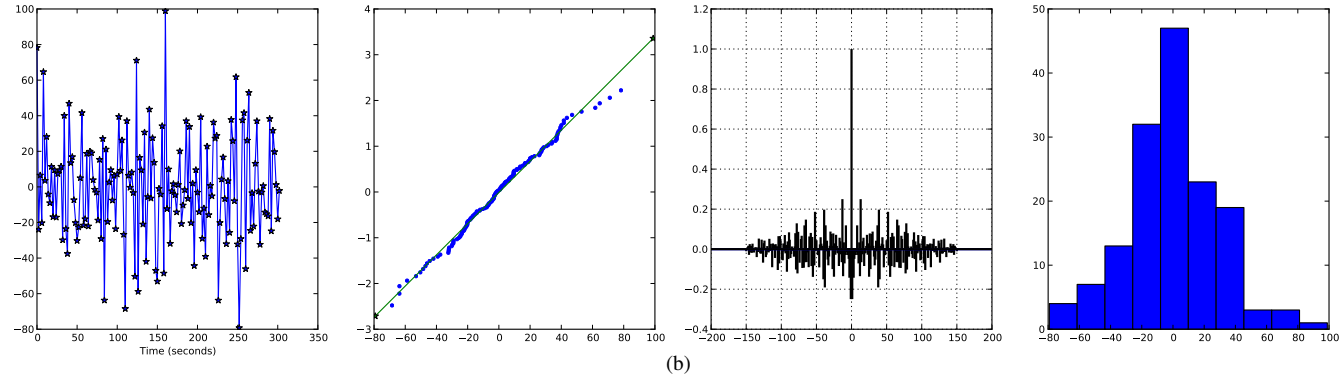


(d)

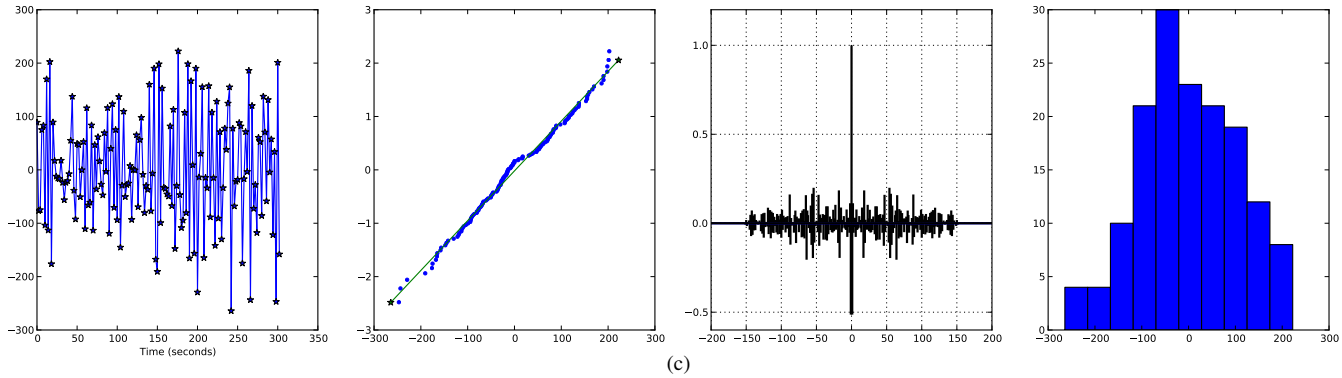
Figure 1.1: Q-Q Plots of normalized resting state data



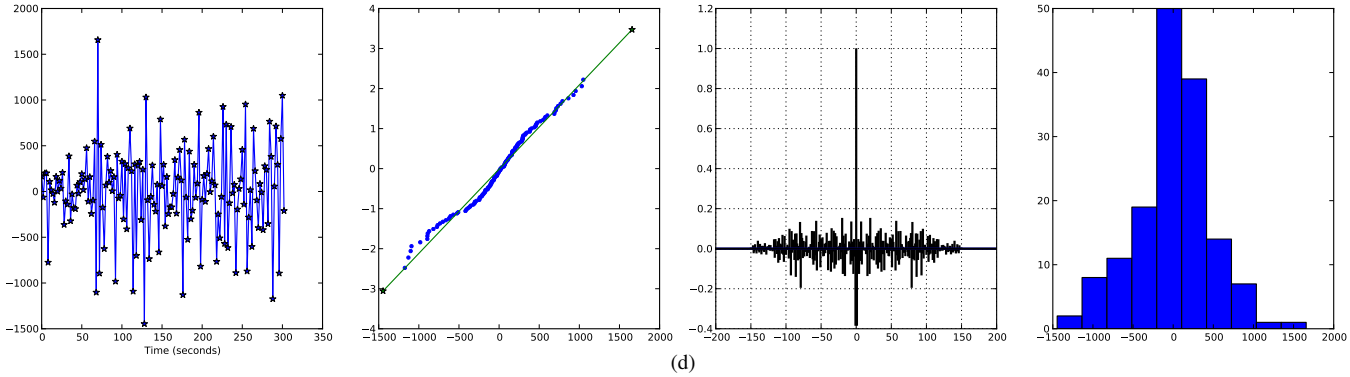
(a)



(b)

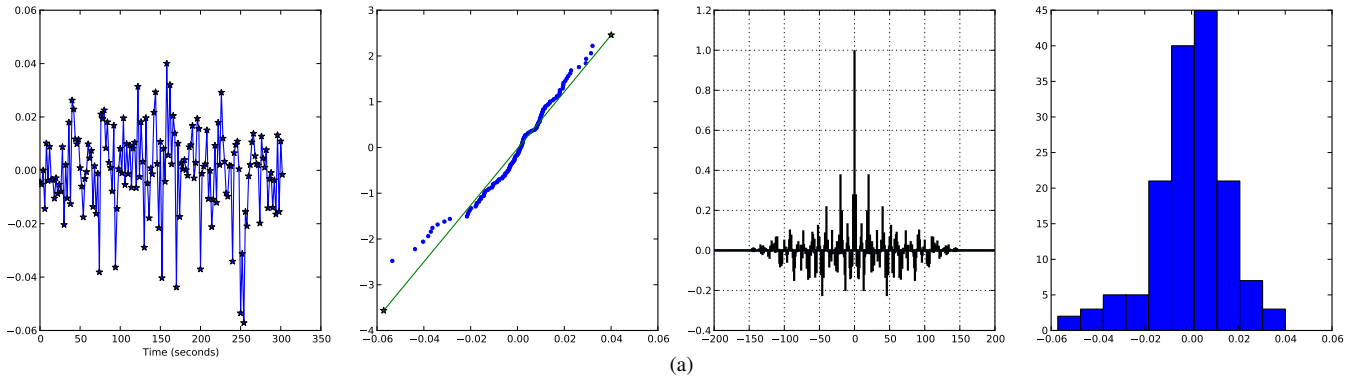


(c)

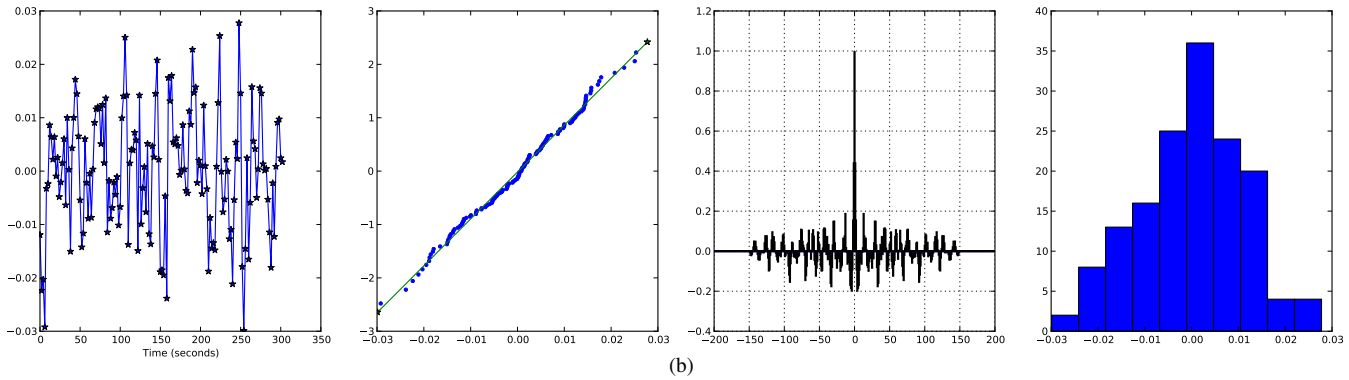


(d)

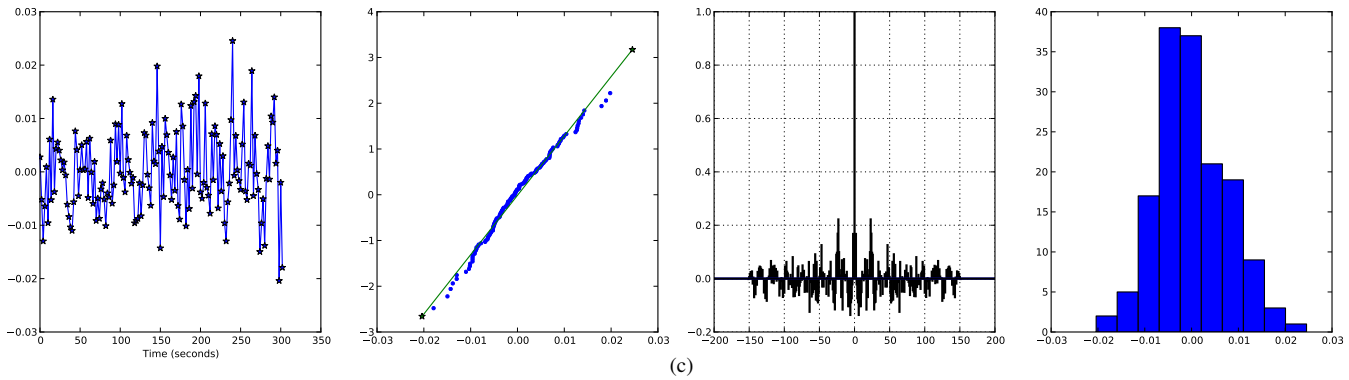
Figure 1.2: Q-Q Plots of resting state data, using the BOLD signal changes



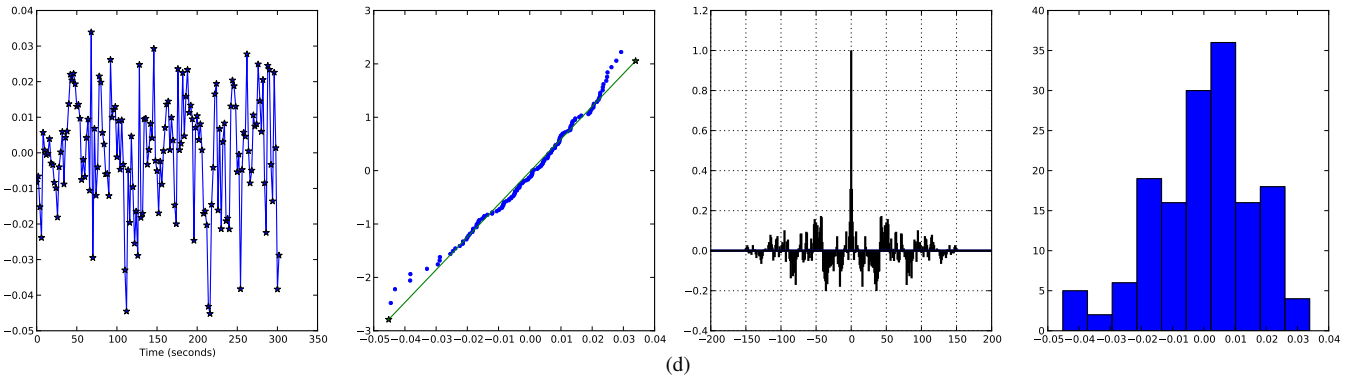
(a)



(b)



(c)



(d)

Figure 1.3: Q-Q Plots of resting state data, after the de-trending

Chapter 2

Current Techniques

The ultimate purpose of this work is to provide a new set of tools for analyzing FMRI data. Whereas esisting techniques have been successful at finding macroscopic regions of activation, :hese techniques are not robust to significant noise and thus linear modeling carries a significant bias error due to lack of model flexibility. While adding parameters can significantly increase error due to model variance, this effect should be mitigated first by the use of a highly constrained model based on first principals and also by the calculation of a full posterior distribution, rather than a single estimate. The purpose of this paper is thus to evaluate the potential of using a particle filter along with the BOLD model to derive physical parameters. In so doing, we hope to be able to show that one or more parameters are a suitable replacement for estimating voxel activation from a standard FMRI image. We also hope to show that estimated posterior distribution of the parameters, derived from the particle filter, is able to provide an accurate measure of the confidence interval.

2.1 Basic Statistical Parametric Mapping

Although not strictly the same thing as parameter calculation from FMRI, activation detection is very similar. In fact, estimation of parameters is somewhat a generalization of the idea of activation detection. Therefore, it is important to draw a comparison between the methods proposed in this thesis with existing methods of activation detection.

The most basic method of analyzing FMRI data is through a standard T-test between ”resting state” and ”active state” samples. This is done by taking the average and variance of the inactive period, and the period during which the stimulus was activate separately then treating them both as gaussian distributions. If they are in fact Gaussian distributions, then a basic t-test will give the probability that the samples came from the same distribution (the null hypothesis). Of course, this test is fraught with problems; even if the drift mentioned

Figure 2.1: GLM todo

earlier has been removed, there is little reason to believe that the noise is Gaussian, or even stable. Additionally, even if the noise were Gaussian, a t-test with a p-value of .05 over 50000 or more samples is on average going to generate $.05 * 50000$ false positives. To compensate for this, bonferoni correction, also known as multiple comparison tests are performed; essentially p-values are divided by the number of independent tests being run. This, however, leads to extremely low p-values, so low that it would be impossible for any biological system to satisfy. To compensate, a Gaussian kernel is applied to the image, thus reducing variance (and thus separating the active and inactive distributions) as well as decreasing the effective number of voxels. Since t-tests are now no longer being applied to n ¡I need to define n¿ independent voxels, the factor by which the p-value must be divided by can be decreased. ¡Do I need to mathematically define all this?¿ The derivation and application of random field theory, and its use can be found in various papers [Worsley et al., 2004].

2.1.1 General Linear Model

The most used form of FMRI analysis is Statistical Parametric Mapping, but is able to account for several different levels or types of stimulus (see [Hofmann, 1997]). By using hierarchical models the output signal timeseries is considered the weighted sum of the various input timeseries. Essentially every experimental factor is considered as another level inputs (ex. different type of stimulus, different run with the same patient, different patient). The equation for a general linear model is then

$$Y(t) = X(t)\beta + \epsilon(t) \tag{2.1}$$

where $Y(t)$ is the smoothed or detrended timeseries of measurements, $X(t)$ is a row vector of input, β is a column vector of weights, and ϵ is the error. Thus for every time, the measurement is assumed to be a weighted sum of the columns of X plus some error. The calculation of β is then performed using a maximum likelihood or gradient descent search to minimize the error.

It is well known of course that a square wave stimulus does not result in a square wave in the activation of brain regions. Thus, various methods are used to smooth the columns of X, and thus bandlimit the input. The best technique is convolving the stimulus input with a Hemodynamic Response Function (HRF), which mimics the basic shape of BOLD activation, including a delay due to rise time and fall time. The downside of this method, however is that the shape of the Hemodynamic Signal is static, meaning the same Hemodynamic Function is used for every region of the brain, and $Y(t)$ must linear combination of scalar the columns of X to be appropriately identified. Additionally, it is well known that different Hemodynamic Response Functions

Figure 2.2: Hemodynamic Response Function todo

are necessary for different regions of the brain. The ”Canonical” HRF that is most used, has been fitted for the visual cortex. Thus the shape of the HRF may differ significantly in terms of onset and fall time in other areas of the brain [Handwerker et al., 2004]. The inability to fit parameters other than scale certainly hinders SPM's ability to locate voxels that do not conform to the ”canonical” hemodynamic response function. If there are different HRF's for different regions of the brain, might there also be more subtle differences even within a particular region? It seems reasonable to think so. As a result, the most common use of SPM will be heavily biased toward the visual cortex.

The GLM is extremely powerful at determining the linear dependence of a set of regressors on the output. Unfortunately, there is significant evidence that many of these dependencies are nonlinear, which means they may be difficult or impossible to detect using linear techniques. This presents a significant and unknown problem that is often left un-addressed in neural studies.

A static linear model is also unable to incorporate other forms of physiological data that may be available. Combined FMRI CBF or CBV imaging methods are not uncommon and they could shed much light on neural activation. However, there is no way of actively combining that data into a static HRF. The fact of the matter is that the relation between the BOLD signal and CBV/CBF simply cannot be described by any linear relationship. While it may be possible to determine some regionally varying set of parameters for the HRF based on CBV/CBF, the lack of physiologically inspired parameters impairs this. The importance of regional differences in the HRF cannot be over-stated; it is well known that capillary beds are far from uniformly distributed and thus blood perfusion and regional oxygenation cannot possibly be uniform either.

Finally, all these techniques require the noise to be Gaussian to reach an optimal solution. In fact there is no known optimal solution to nonlinear models with non-Guassian noise, so obviously some assumptions are going to have to be made to reach a solution. However, it would be nice to have an algorithm that is robust to these effects, and could still give a good solution when Gaussianity is violated.

Because of these limitations, its entirely possible that activation exists in regions that SPM doesn't detect, but because the activation does not conform to the set HRF, it is impossible for these signals to be detected. Perhaps because of this, it is not uncommon for data to be thrown out or averaged together in FMRI studies because no significant activation could be seen in a single run [Riera et al., 2004] [Johnston et al., 2008]. This practice highlights limitations in strictly linear approaches; and suggests that a single HRF is insufficiently flexible to account for relatively common variations of neural activation [Handwerker et al., 2004].

In general activation detection type methods also don't have the ability to find pathologies based on state variables or parameters. It is quite possible that physical properties such as decreased compliance of blood

vessels could indicate a neurological condition that is not easily seen in a T1 or T2 map. In essence, this could make FMRI a much more useful clinical tool than it is now.

2.2 Nonlinear Least Squares

Rather than localizing activation, by using the physiologically plausible BOLD model, it is possible to determine the values of governing parameters.

Although there are certainly benefits to using a derived model, rather than a purely empirical model, there are serious implications. The first problem is that all the powerful classical techniques of gradient descent are off limits; since the model is a true nonlinear dynamical system with no closed form solution. The implication of this is that the calculation of a Jacobian for residuals won't work; and thus powerful techniques such as the Gauss-Newton method, which are helpful in many nonlinear problems, are off limits. Additionally, a gradient descent is difficult to perform without the ability to form partials of the output with respect to all the parameters.

Although anything requiring a Jacobian is out, there are other heuristic techniques that could potentially illuminate the BOLD response. Simulated Annealing (SA) is a common method of optimizing high dimensional regression problems. The idea is to pick a random start, and then at each iteration pick a random nearby point, and if that point is below some energy constraint (energy is a function of the residual), called the temperature, the algorithm moves to that point and continues with the next iteration. The temperature is slowly lowered until no nearby points below the temperature can be found (or the temperature drops below the current point). There are variations of this, for instance it is common to require every movement to be in the downward direction (in terms of energy). Like most nonlinear optimization problems, there is no guarantee of an optimal solution, although the longer the algorithm is allowed to run, the better the solution will get. Since every step requires an entirely new run of the BOLD model, it can be extremely time consuming, which is why we are not using it here.

Algorithm 2.1 Simulated Annealing Algorithm

```
Initialize Θ, or if there exists a decent estimate start there
Initialize temperature, T to value above initial energy
while E(Θ) < T do
  repeat
    Pick θ near Θ
    Calculate energy, E, of θ
  until E > T
  Move to new estimate: set Θ = θ
end while
```

[simulated annealing image?]

Another potential method of interest is the use of Genetic Algorithms (GA). Genetic algorithms are similar

to Simulated Annealing, in that they randomly move to better solutions based on a cost function. However; in genetic algorithms a single point estimate isn't used. Instead a population of estimates is generated, each with distinct parameters, and then each set of parameters is rated with a fitness function. Parameter sets that are good get a higher "fitness"; then new parameter sets are generated by randomly combining pieces of the old parameter sets. The pieces are typically chosen at a rate proportional to the fitness of the donor; thus "fit" parameter sets tend to pass on their properties. In addition to this, random mutations may be introduced that come from no existing parent. The new generation is then rated with the fitness function again, and the entire process starts over. The stop condition for a genetic algorithm is typically based on some threshold for fitness or a maximum number of generations. This is problematic, since it doesn't really provide any guarantee of even reaching a local minima, although in practice it can be quite effective.

[genetic algorithm picture]

Algorithm 2.2 Genetic Algorithm

Initialize N estimates, $E = \{\Theta_0, \Theta_1, ...\Theta_N\}$
for G generations **do**
  Calculate fitness for each Θ, Ex. for residual R, $1/R$ or, e^{-R}
  for i in N **do**
    Randomly select two parents (with higher probability for more fit Θ's)
    Randomly merge parts of the two parents to form a new Θ_i
    At some low probability change one or two parameters in Θ_i
  end for
end for

Although both these methods can be highly effective, they have the downside of requiring very high computation time. In this case of the BOLD model, each time the energy or fitness needs to be calculated, a large number of cycles must be spent re-simulating the BOLD model for the set of parameters. As we'll discuss in **??**, the Particle Filter method is able to circumvent this re-calculation to some degree.

It would not be unreasonable at this point to back off and work with a linearized or more static version of the model. This is the approach taken by the standard General Linear Model discussed in subsection 2.1.1.

2.3 Unscented Kalman Filter

Although the classical methods mentioned in section 2.2 won't work for the nonlinear model presented in section 1.3, there is another Bayesian technique that is worth considering. The Unscented Kalman Filter (UKF) is a powerful Gaussian/Bayes filter that attempts to model the posterior distribution of dynamical systems as a multivariate Gaussian. The Unscented Kalman Filter (UKF) generalizes the Extended Kalman Filter by allowing the state update to be a function, g,

Figure 2.3: Examples of update distributions, using an Kalman Filter, [Thrun et al., 2005]

$$X(t) = g(u(t), X(t-1)) \tag{2.2}$$

$$Y(t) = h(X(t) \tag{2.3}$$

In order to estimate the posterior at t, a deterministic set of "sigma" points (often 2 per dimension, plus 1 at the mode of the multivariate distribution) weighted according to a Gaussian estimate of $X(t-1)$ are passed through the update equation. This set of points are then used to estimate the mean and covariance of $X(t)$. The benefit of this, is that it requires no Jacobian and only a few extra calculations to get a decent esimate of a posterior Gaussian. In the BOLD case, the set of equations we are modeling have no closed form solution, and finding the Jacobian is impossible without approximations. Although [Riera et al., 2004], [Hu et al., 2009] mention a Jacobian of the BOLD response, this is not strictly the case and is rather $\frac{\partial J}{\partial t}$ rather than a true Jacobian. This is important because the Extended Kalman filter depends on the Jacobian to map a Gaussian through the advancement of time. Thus the popular Extended Kalman Filter won't work in this case, whereas the Unscented Kalman Filter still does. In fact [Hu et al., 2009] uses the UKF to perform a similar type of analysis to the one performed in this work.

The difficulty of using a Kalman Filter, however, is that it assumes a multivariate Gaussian for the state variables, $X(t-1)$. The problem with this is that when a system is significantly nonlinear, the state at $X(t)$ will almost certainly be non-Gaussian, and thus estimating $X(t+1)$ with $X(t)$ as a multivariate Gaussian will perpetually introduce error in the distribution. Furthermore, it is not really known what sort of underlying distributions may exist in such a mixed biological, mechanical, chemical system such as the brain. Assuming the parameters all to be Gaussian may in fact be a gross error. On the other hand, for small variances and short time steps the gaussiann distribution is a good fit, and so in some limited cases the Unscented Kalman Filter could work well. These are non-trivial issues given that the assumption of Gaussianity is what allows the UKF to estimate the posterior using only the first and second moments; two parameters that don't uniquely describe most distributions.

To determine the amount of error incurred in a Gaussian estimate during a typical sample period, the states of BOLD equations were assigned according to four dimensional Gaussian. The states were then propagated through two seconds of simulation (a typical TR in FMRI) and then the resulting marginal distributions were compared with a Gaussian distribution. The purpose is to determine the degree to which the results of simulation will result in non-Gaussian output, given a Gaussian input. This also demonstrates the degree of nonlinearity

Parameter	Run 1
τ_0	.98
α	.33
E_0	.34
V_0	.03
τ_s	1.54
τ_f	2.46
ϵ	.54
V_t	N(1, .3)
Q_t	N(1, .3)
S_t	N(1, .3)
F_t	N(1, .3)

Table 2.1: Parameters used to test Gaussianity of variables after being transitioned through the BOLD model

present in the system. The parameters used are shown in Table 2.1

Notably s_t has intentionally been set to a non-equilibrium, but physiologically plausible value. The value of u is left at zero the entire time, so the system will decay naturally (see section 1.3), though initializing s at a non-zero level will drive the system for several seconds. Figure 2.4 shows the results when the system is essentially left on for 100 milliseconds after setting the variables according to Table 2.1. The Q-Q plots fit very will with a Gaussian, demonstrating that at this short time interval nonlinearities have not yet begun to effect the distribution. However, Figure 2.5 us the result after 1 second, which is faster than most FMRI scanners are capable of sampling at. At that range the tails of the distributions for v and q are clearly starting to deviate from the Gaussian distribution. As a result the uncertainty in y is deviating from the Gaussian distribution as well. This is important, because although approximating the distribution with a Gaussian based on the first two moments will work in the short run, there will be residual error in the distribution.

On the other hand, this effect is more limited if the initial variance is somewhat smaller. In tests with those cases, it took much longer for the nonlinearities to skew the distribution. That result could be encouraging to those looking to use the UKF, if the distributions are kept relatively thin.

Another potential problem with the UKF is that typically the sigma points, used to estimate the posterior probability, $P(X(t)|X(t-1), u(t))$, are located on the main axes. As a result, the covariances are not allowed to be effected by the state transition the same way the variances are. While this may be reasonable in low-dimensional systems, high dimensional systems have a much greater potential for interplay between the variables. While this problem is relatively easy to fix (by using more sigma points off the main axes), there is a definite cost in complexity.

Ultimately, there is a distinct possibility that the nonlinearities of the BOLD model make Gaussian estimates unrealistic and thus less effective. More advanced tests where static variables such as α are varied as well could shed even more light on the issue. The trouble with using the UKF then to estimate parameters is that all eleven

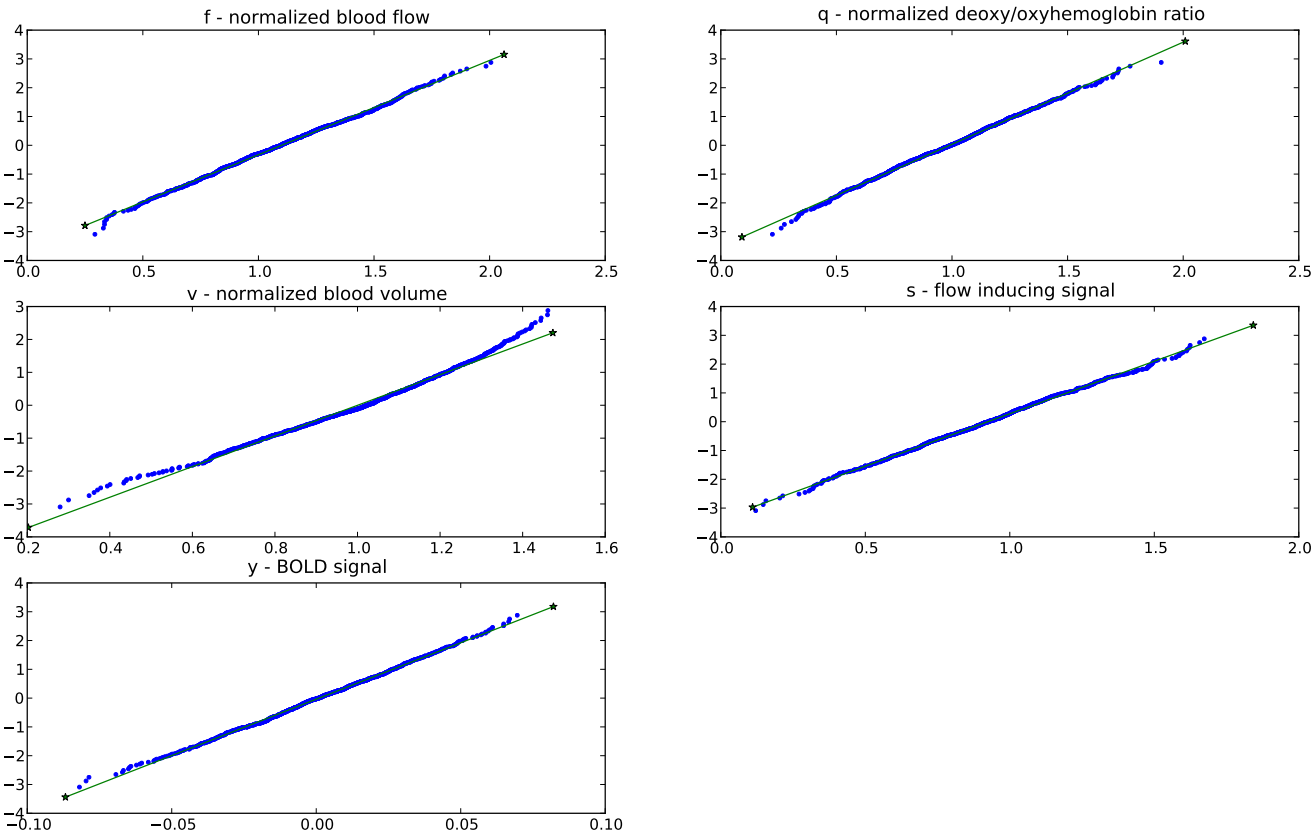


Figure 2.4: Distributions of state variables after simulating for .1s

members of X would be treated like a single joint Gaussian distribution which certainly compound the issues of nonlinearity seen in Figure 2.4 and Figure 2.5

2.4 Variational Filtering

[Friston et al., 2008], [Friston, 2008]

2.5 Hybrid Methods

A large number of hybrid methods have been tried, in order to elicit parameter estimates from the BOLD model. [Vakorin et al., 2007], [Johnston et al., 2007], [Hu et al., 2009]

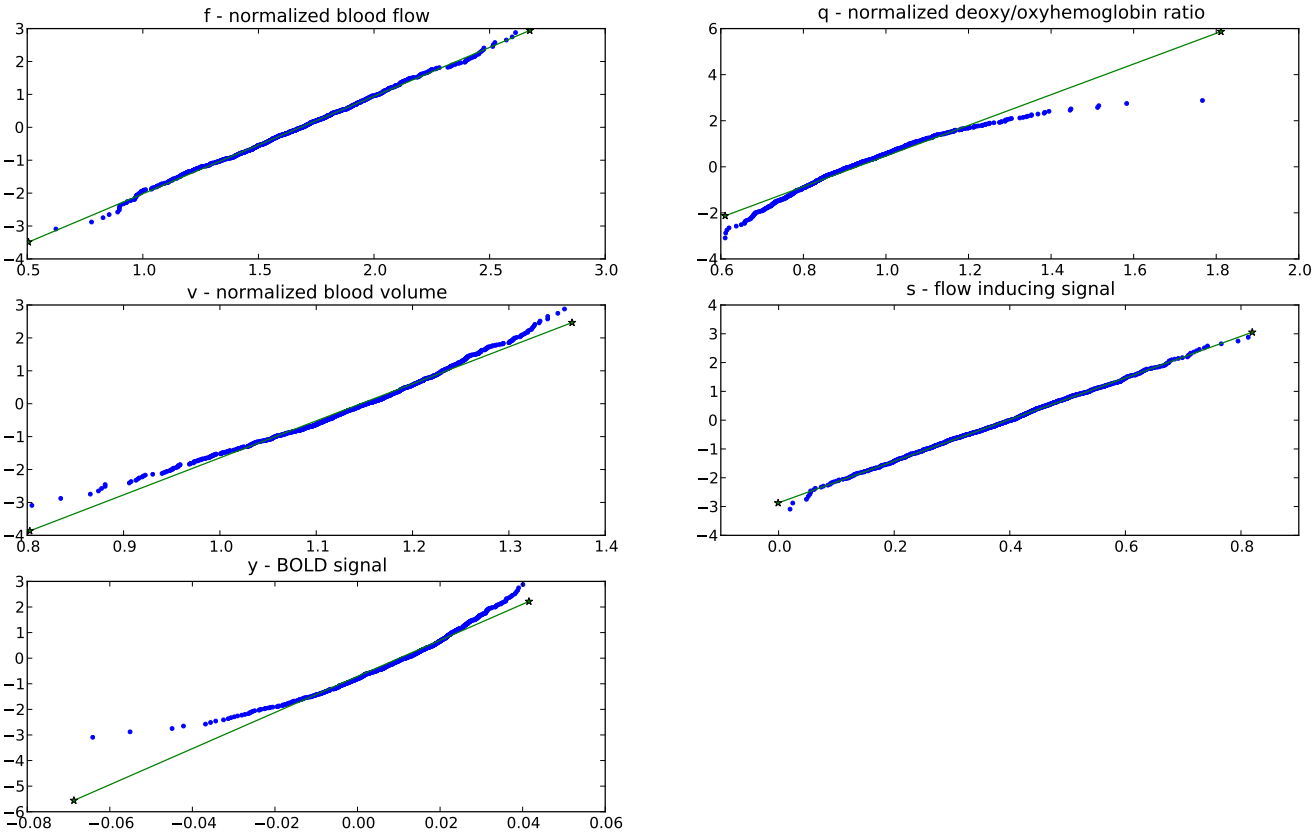


Figure 2.5: Distributions of state variables after simulating for 1s

2.6 Conclusion

In summary, several other approaches could be taken

Chapter 3

Particle Filters

3.1 Introduction

Particle filters, a type of Sequential Monte Carlo (SMC) methods are a powerful way of estimating the posterior probability distribution of a set of parameters give a timeseries of measurements. Unlike Markov Chain Monte Carlo estimation, Sequential Monte-Carlo methods are designed to be used with parameters that vary with time. Unlike variations of the Kalman filter, particle filters do not make the assumption that noise is Gaussian. Thus particle filters are often the best solution to bayesian tracking for non-linear, non-gaussian systems.

3.1.1 Model

The idea of the particle filter is to start with a wide mixture PDF of possible parameter sets, and then, as measurements come in, to weight more heavily parameter sets that tend to give good estimations of the measurements. The reliance on an initial mixture PDF can introduce bias; however, this effect can be minimized by alterring the initial weights in the mixture pdf. Of course every gradient descent must choose starting points and it is often quite easy to establish a reasonable range of parameters, especially when the model being used has a physical meaning. Suppose a set or stream of measurements are given, $\{y(t), t = 1, 2, 3, ...T\}$, where T is permitted to go to infinity. Then the goal is to find the parameters, $\hat{\theta}$, and underlying state time series, $\hat{x}[0:T]$ that minimize the difference between $\hat{y}[0:T]$ and $y[0:T]$. In our case, we will assume that we know the form of the model, which is based on first principals, and that there is some true θ and a true time-series of underlying state variable, $x[0:T]$ that drives $y[0:T]$. Assuming a model form such as we do here reduces model variance, potentially at the cost of increased bias (or systematic) error. We will assume a basic state space model:

$$\dot{x}(t) = f(t, x(t), u(t), \theta, \nu_x) \tag{3.1}$$

$$y(t) = g(t, x(t), u(t), \theta, \nu_y) \tag{3.2}$$

Where $x(t)$ is a vector of state variables, θ is a vector of system constants, $u(t)$ is a stimulus, $y(t)$ an observation, and ν_x and ν_y are random variates. Obviously any one of these could be a vector, so for instance $u(t)$ could encode multiple types of stimuli.

Although not strictly necessary for particle filters, we will make a few assumptions based on the particular type of systems faced in biological processes. First, the systems are assumed to be time invariant. This assumption is based on the idea that if you froze the system for Δt seconds, when unfrozen the system would continue as if nothing happend. Few biological systems are predictable enough for them to be summarized by a time varying function. Even systems that one might assume work in that way are actually much more complicated. The heart may seem like an obvious exception, however, the period between heartbeats vary often enough that prediction would necessitate another state-space model. In short, we assume no parameters are time varying, because not enough information exists to describe any of theme in that way. Luckily particle filters are capable of dealing with non-white, non-Gaussian noise, so unanticipated influence may be re-factored as noise. Secondly we assume that input cannot directly influence the output, which in the case of the BOLD signal is a good assumption. Third, we will assume noise is additive, and that ν_x may be projected into a Weiner, or other summing process that is additive with g and ν_y, which will be named ν_d. Finally, $x(t)$ will encapsulate θ, the unknown model constants, which means that the vector $\dot{x}$ will always have members that are 0. The results of these assumptions are a simplified version of the state space equations:

$$\dot{x}(t) = f(x(t), u(t)) \tag{3.3}$$

$$y(t) = g(x(t)) + \nu_y + \nu_d \tag{3.4}$$

Because ν_d is something akin to and additive Weiner process $y[0:T]$, it will include low frequency noise. ν_y on the other hand will cause i.i.d. noise in $y[0:T]$. For some of the tests, I will use de-trending methods to reduce the effects of ν_d, the remainder of which will be re-factored into ν_y. Both ν_d and ν_y have biological and non-biological sources. MR can lead to both types of noise, as demonstrated in [Smith et al., 1999]. Meanwhile changes in metabolism, heart rate, or other biochemical intervention could all lead to either ν_d or ν_y.

3.1.2 Prior

The goal of the particle filter is to evolve a probably distribution $Pr(\hat{x}(T)|u[0:T], y[0:T])$, that asymptotically approaches the probability distribution $Pr(x(T)|u[0:T])$. Considering that y contains measurement noise as well as noise from x, it is clear that $Pr(x(t)|u[0:T])$ is not a single true value but a true posterior. To begin with, the particle filter starts with a prior distribution, and N_p particles need to be drawn from that distribution, $\alpha(x)$:

$$\{\hat{Pr}x_i(0), w_i] : x_i(0) \sim \alpha(x), w_i = \frac{1}{N_p}, i \in \{1, 2, ..., N_p\}\} \tag{3.5}$$

Where N_p is the number of particles or points used to describe the prior using a Mixture PDF.

$$\hat{Pr}(x(0) = \hat{x}) = \sum_{i=1}^{N_p} w_i \delta(\hat{x} - x_i(0))dx \tag{3.6}$$

Where $\delta(x - x_0)$ is 1 if and only if $x = x_0$ (the Kronecker delta function).

If a true prior is preferred, then the weights should all be $1/N_p$, and since x_i was drawn from the prior, this will be an approximation of the prior distribution. If a relatively flat prior is preferred, then each particle's weight could be divided by the density, $\alpha(x_i)$, which creates a flat prior with support points in the region of $\alpha(x)$. Either way, $\alpha(x)$ should be much broader than the true posterior, $Pr(x(0))$, since the choice of support points is crucial to the convergence of any sampling importance algorithm. For the BOLD signal all the parameters have been studied and have relatively well known mean and variance, so a prior could be very helpful. We ran simulations for both normalized and un-normalized priors, although we believe in cases such as this, where a good prior exists, it should be used. For strictly positive parameters (members of x) we used a gamma distribution, whereas for parameters that could be negative, we used a Gaussian distribution. In both cases standard deviations twice that found in previous studies were used.

Note that all the probabilities implicitly depend on $u[0:T]$, so those terms will be left off for simplicity. Once the probability, $\hat{Pr}(x(T)|x[0:T-1], y[0:T-1])$ has been found (initially this is just Mixture approximating the prior since no measurements are available and no previous probabilities are available), its possible to approximate the probability for short times between times when measurement is available, by shifting the probability according the progression of the state equations. This is only an approximate, since integrating ν_d should increase uncertainty as time without a measurement passes.

$$\hat{Pr}(x(T + \Delta t)) \approx \sum_{i=1}^{N_p} w_i \delta \left(x - (x_i(T) + \int_T^{T+\Delta} \dot{x}_i(t)dt) \right) \tag{3.7}$$

3.1.3 Weighting

When a measurement becomes available it is incorporated into the probability. This process of incorporating new data is called sequential importance sampling, and eventually causes the probability to converge. The weight is defined as

$$w_i(T) \propto \frac{\hat{Pr}(x_i[0:T]|y[0:T])}{q(x_i[0:T]|y[0:T])} \tag{3.8}$$

where q is called an *importance density*, meaning it decides where the support points for $x(T)$ are located. To remove the bias due to the location of the support points, we divide by $q(x_i[0:T]|y[0:T])$. By dividing by the posterior density of the support points (particles), the effect of the particle distribution may be removed from the posterior density. As a result the weight is dependent solely based on $\hat{Pr}(x_i[0:T]|y[0:T])$, the probability of the i^{th} particle's measurements being different from $y[0:T]$ due to noise alone. An example of an importance density would be drawing a large number of points from the standard normal, $N(0,1)$ and then weighting each point, l by $1/\beta(l), \beta \sim N(0,1)$. Of course if there is a far off peak in the posterior that q does not allocate support points in, there will be a quantization error, and that part of the density can't be modeled. This is why it is absolutely necessary that q covers $\hat{Pr}(x_i[0:T]|y[0:T])$.

$q(x_i[0:T]|y[0:T])$ may be simplified by assuming that $y(T)$ doesn't contain any information about $x(T-1)$, which is more practical since knowledge of future measurements is impractical.

$$\begin{aligned} q(x[0:T]|y[0:T]) &= q(x(T)|x[0:T-1], y[0:T])q(x[0:T-1]|y[0:T]) \\ &= q(x(T)|x[0:T-1], y[0:T])q(x[0:T-1]|y[0:T-1]) \\ &= q(x(T)|x(T-1), y[0:T])q(x[0:T-1]|y[0:T-1]) \end{aligned} \tag{3.9}$$

In this paper we will use $q(x_i(T)|x_i(T-1), y[0:T]) = \hat{Pr}(x_i(T)|x_i(T-1))$, based on the Markov assumption, and the belief that the state space model is able to approximate the true state. This means that prior to re-weighting particles, the particles will be distributed the same as the previous time but moved forward according to the integration of $f(x(t), u(t))$.

In addition to $q(x_i(T)|x_i[0:T-1], y[0:T])$, the weight is also based on $Pr(x_i[0:K]|y[0:K])$, which

may be broken up as follows.

$$\begin{aligned}
\hat{Pr}(x[0:T]|y[0:T]) &= \frac{\hat{Pr}(y[0:T],x[0:T])}{\hat{Pr}(y[0:T])} \\
&= \frac{\hat{Pr}(y(T),x[0:T]|y[0:T-1])\cancel{\hat{Pr}(y[0:T-1])}}{\hat{Pr}(y(T)|y[0:T-1])\cancel{\hat{Pr}(y[0:T-1])}} \\
&= \frac{\hat{Pr}(y(T)|x[0:T],y[0:T-1])\hat{Pr}(x[0:T]|y[0:T-1])}{\hat{Pr}(y(T)|y[0:T-1])} \\
&= \frac{\hat{Pr}(y(T)|x[0:T],y[0:T-1])\hat{Pr}(x(T)|x[0:T-1],y[0:T-1])\hat{Pr}(x[0:T-1]|y[0:T-1])}{\hat{Pr}(y(T)|y[0:T-1])}
\end{aligned} \tag{3.10}$$

Using the assumption that $y(t)$ is fully constrained by $x(t)$ Equation 3.4, and that $x(t)$ is fully constrained by $x(t-1)$ Equation 3.3, we are able to make the reasonably good assumptions that:

$$\hat{Pr}(y(T)|x[0:T],y[0:T-1]) = \hat{Pr}(y(T)|x(T)) \tag{3.11}$$

$$\hat{Pr}(x(T)|x[0:T],y[0:T-1]) = \hat{Pr}(x(T)|x(T-1)) \tag{3.12}$$

Additionally, for the particle filter $y(T)$ and $y[0:T-1]$ are given, and therefore constant across all particles. Thus $\hat{Pr}(x[0:T]|y[0:T])$ may be simplified to:

$$\begin{aligned}
\hat{Pr}(x[0:T]|y[0:T]) &= \frac{\hat{Pr}(y(T)|x[0:T],y[0:T-1])\hat{Pr}(x(T)|x[0:T-1],y[0:T-1])\hat{Pr}(x[0:T-1]|y[0:T-1])}{\hat{Pr}(y(T)|y[0:T-1])} \\
&= \frac{\hat{Pr}(y(T)|x(T))\hat{Pr}(x(T)|x(T-1))\hat{Pr}(x[0:T-1]|y[0:T-1])}{\hat{Pr}(y(T)|y[0:T-1])} \\
&\propto \hat{Pr}(y(T)|x(T))\hat{Pr}(x(T)|x(T-1))\hat{Pr}(x[0:T-1]|y[0:T-1])
\end{aligned} \tag{3.13}$$

Plugging these simplifications into Equation 3.8 leads to:

$$\begin{aligned}
w_i(T) &\propto \frac{\hat{Pr}(y(T)|x(T))\cancel{\hat{Pr}(x(T)|x(T-1))}\hat{Pr}(x[0:T-1]|y[0:T-1])}{\cancel{\hat{Pr}(x_i(T)|x_i(T-1))}q(x[0:T-1]|y[0:T-1])} \\
&\propto w_i(T-1)\hat{Pr}(y(T)|x(T))
\end{aligned} \tag{3.14}$$

Thus, by making the following relatively weak assumptions, evolving a posterior density is easy and requires almost no knowledge of noise distribution.

1. $f(t,x(t),u(t)) = f(x(t),u(t))$ and $g(t,x(t),u(t)) = g(x(t))$ provide a sufficiently flexible model to

encapsulate the true time series.

2. $E[\nu_d] = 0$ and $E[\nu_y] = 0$, and $\nu_x = d\nu_d$, ν_y are stationary
3. The PDF $q(x_i(0))$ (the prior) fully covers $Pr(x_i(0))$
4. Markov Assumption: $Pr(x(T)|x[0:T]) = Pr(x(T)|x(T-1))$
5. $q(x[0:T-1]|y[0:T]) = q(x[0:T-1]|y[0:T-1])$

3.1.4 Basic Particle Filter Algorithm

From the definition of w_i, the algorithm sequential importance sampling (SIS) is relatively simple.

Initialize N_p Particles: $\{x_i(0), w_i(0) : x_i(0) \sim \alpha(x), w_i(0) = \frac{1}{N_p}, i \in \{1, 2, ..., N_p\}\}$

T = {Set of Measurement Times}

for t in T **do**

  for i in N_p **do**

    $x_i(t) = x_i(t-1) + \int_{t-1}^{t} f(x(\tau), u(\tau))d\tau$

    $w_i(t) = w_i(t-1)\hat{Pr}(y(t)|x(t))$

  end for

end for

At $t + \Delta t, t \in T$, $\hat{Pr}(x(t + \Delta t)) \approx \sum_{i=1}^{N_p} w_i(t)\delta\left(x - (x_i(t) + \int_t^{t+\Delta t} f(x(\tau), u(\tau))d\tau)\right)$

The result is then a discrete approximation of the posterior distribution.

3.1.5 Resampling

As a consequence of the wide prior distribution (required for a proper discretization of a continuous distribution), there will be many particles with insignificant weights. While this does help describe the tails of the distribution very well, it means that only a small portion of the computation will be spent describing the most probable region. Ideally every particle would equally decrease the entropy of the distribution, thus the lower the variance of the weights, the more efficiently the discrete distribution is in describing the continuous distribution. A common measure of "Particle Degeneracy" is the effective number of particles, described in (Bergman "Navigation and Tracking Applications", 1999, J S Liu and R Chen "Sequential Monte Carlo Methods for Dynamical Systems", 1998), which is based on the "true weight" of each particle. Of course the true weight is unknown, so a heuristic approximating N_{eff} is used:

$$\hat{N}_{eff} \approx \frac{N_p}{\sum_{i=1}^{N_p} w_i^2} \tag{3.15}$$

Any quick run of a particle filter will reveal that unless the prior is particularly accurate, N_{eff} drops precipitously. To alleviate this problem a common technique known as resampling must be applied. The idea of re-sampling is to draw from the approximate posterior, thus generating a replica of the posterior with a support more suited to the distribution. Thus, if weights are all set to $1/N_p$, and N_p points are drawn from the posterior,

$$\hat{\chi}_j \sim \left(\sum_{i=1}^{N_p} w_i(t)\delta(x - x_i(t)) \right), j \in \{1, ..., N_p\} \tag{3.16}$$

then $\hat{\chi} \sim \hat{x}$ should hold. Unfortunately, this isn't necessarily the truth: since the support is still limited to the original particles, the number of unique particles can only go down. This effect, often dubbed "particle impoverishment" can result in excessive quantization errors in the final distribution. However, there is a solution. Instead of sampling from the discrete distribution, a smoothing kernel is applied, and $\hat{\chi}_j$ are drawn from that distribution. Because the distribution is continuous, there is no way for a collapse of the particles to occur. The question then, is how to decide on the smoothing kernel. Often times the easiest way to sample from the continous distribution is to break the re-sampling down into two steps. First a member of the discrete distribution is randomly selected based on the weights, and then based on the smoothing a nearby state variable is selected. The process of the selection will be defined as:

$$\chi_i = x_i + h\sigma\epsilon \tag{3.17}$$

Where h is the bandwidth, σ is the standard deviation such that $\sigma\sigma^T = cov(x)$ and ϵ is drawn from the chosen kernel. It has been proven that when all the elements of the mixture have the same weight, as is the case after basic resampling, the kernel that minimizes the MSE between the estimated and true posterior is the Epanechnikov Kernel (cite Improving Regularised Particle Filters, C Musso, N Oudjane and F LeGrand).

$$K = \begin{cases} \frac{n_x+2}{2c_{n_x}}(1 - \|x\|^2) & if\ \|x\| < 1 \\ 0 & otherwise \end{cases} \tag{3.18}$$

3.1.6 Weighting Function

Because $\hat{Pr}(y(T)|x(T))$, what I will call the weighting function, is based on an unknown distribution, it is necessary to decide on a function that will approximate $\hat{Pr}(y(t)|x(T))$. Obviously the function, $\omega(y(t), f(x(t))$ needs to be centered at zero and have a scale comparable to the signal levels. Obviously if the actual noise present in $y(t)$ were to be known, then that would be the best distribution for Ω. In that case, particles that fell far out on that distribution would be statistically impossible representations of the system, and it would be

completely reasonable to throw such particles away. While a Gaussian function is the natural choice, because this distribution and weight are unknown we wanted to try distributions with wider tails, so that outliers don't completely destroy particle's weights (and thus convergence proceeds more slowly).

Another natural choice might be one of the robust estimator weight functions, for example the Huber or bi-square. For the purpose of this work we will stick with long tailed distributions, however it is worth noting that long tails may not be the optimal choice in all, or even this situation. The justification for long tailed distributions is that we believe the noise to be long tailed, and the variance of the noise is not well known.

Therefore, we tried three weighting functions based on three distributions: Gaussian, Laplace and the Cauchy. The standard deviation of the distribution is extremely important to the convergence of particle filter. A standard deviation that is too large will not allow the distribution to converge in any reasonable number of measurements. A standard deviation below the standard deviation of the noise will cause the algorithm to throw out perfectly acceptable particles. The weighting function ultimately will shape the output distribution, $P[y]$, into that distribution, however the distribution of x will still approach a reasonable estimate of its true distribution. Even an overly wide weighting function, will allow the Gaussian Mixture estimate of X to converge to the correct location parameters of the "real" posterior distribution, though the scale parameters may be overly large.

A reasonable method of setting the standard deviation of Ω may be by taking a small sample from "resting" data and using the sample standard deviation. Since this is the first attempt at using particle filters for modeling the BOLD model, in this work we set the standard deviation manually at ¡weight standard dev¿, because it gives a more consistency and control. Of course this could be taken further, by testing the sample data against a set of stock distributions and choosing the best fit. This of course depends on having enough samples to make a reasonable inference, which may not always exist.

3.2 Simple, Nonlinear Example

A typical half wave rectifier takes a AC voltage circuit and removes one half (say the negative half) of the signal. The resulting waveform is still not DC, however it is then possible to use a capacitor to smooth the signal into something similar to DC, as shown in Figure 3.2. There are other, more complex circuits that convert the negative portion into positive and waste less energy but but here we will keep the system simple. Thus, let us consider a simple half wave rectifier circuit, shown in Figure 3.1.

The half wave rectifier circuit smoothes the gaps between high voltage with a capacitor. Thus, when $u(t)G$ is less than v_t, the circuit will discharge the capacitor and maintain a non-zero voltage, but when $u(t)G$ is greater than v_t, the output voltage will be set by $u(t)G$ and the capacitor will charge up. We will assume a very

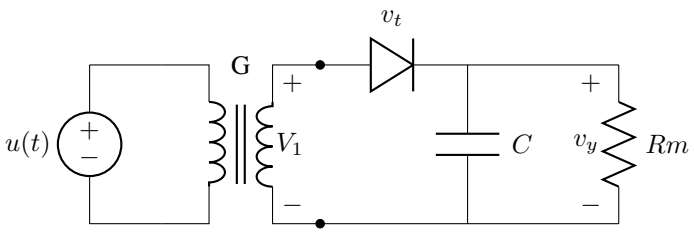


Figure 3.1: An Example Half Wave Rectifier Circuit, where G is the transformer gain, v_t is the activation voltage of the diode, $u(t)$ is the input at time t, C is the capacitance, R is the load resistance and v_y is the output voltage

Figure 3.2: Example Input/Output of the Half Wave Rectifier

simple model for all the components, ignoring the complex nonlinear behavior that can occur in a diode.

Although rectifiers are typically thought of as receiving an AC circuit 60 Hz, we will ignore such specifics and assume the voltage across the output of the transformer is simply a scalar multiple of the input voltage. As discussed in the subsection 3.1.1 any variable with uncertainty must be part of the state variable. Therefore the state variable will be: $X(t) = \{G, v_t, C, R_m, v_y\}$. Of course, $u(t)$ cannot be allowed to be a square wave in such a system, since that such a signal would never get across the transformer and regardless it would necessitate an unrealistic infinite current across the capacitor. The state equations would then be

$$v_y(t) = f(v_y(t-1, u(t)) = \begin{cases} u(t)G & \text{if } u(t)G - v_y \geq v_t \\ v_y(t-1)\left(1 - \frac{\delta t}{R_m C}\right) & \text{if } u(t)G - v_y < v_t \end{cases} \tag{3.19}$$

To run the particle filter is relatively easy then, since there exists a recursive definition of the dynamic state variable, v_y. To start with, an initial distribution must be assumed and while at first a Gaussian seems like a good idea, all the static state variables are strictly positive and thus not well suited to the Gaussian. In this case then, it would be wise to start with a Gamma distribution, and just be wary of any standard deviation that gets larger than the prior mean. We will define the gamma distribution as follows:

$$X \sim Gamma(k, \theta) \rightarrow f(x) = x^{k-1} \frac{e^{-x/\theta}}{\theta^k \Gamma(k)} \tag{3.20}$$

where Γ is the gamma function. The in some ways the margin for error is decided by the weighting function, which here will define as $W(V_y, v_{yi})$, where V_y is the actual measurement, v_y is the estimate based on all the particles, and v_{yi} is the estimate by a particular (i^{th}) particle. The choice of this function is difficult, and although the Gaussian is typically used, we found the exponential helpful in dealing with particle deprivation. The algorithm will then look like the following,

Initialize N_p Particles:

for i in N_p **do**

$G \sim Gamma(\frac{\mu_G^2}{\sigma_G^2}, \frac{\sigma_G^2}{\mu_G})$

$v_t \sim Gamma(\frac{\mu_{v_t}^2}{\sigma_{v_t}^2}, \frac{\sigma_{v_t}^2}{\mu_{v_t}})$

$C \sim Gamma(\frac{\mu_C^2}{\sigma_C^2}, \frac{\sigma_C^2}{\mu_C})$

$R_m \sim Gamma(\frac{\mu_R^2}{\sigma_R^2}, \frac{\sigma_R^2}{\mu_R})$

$v_y = 0$, (Assume the system has been off for a long time)

let $X_i(0) = \{G, v_t, C, R_m, v_y\}$

let $w_i(0) = 1$ or to make a flat prior, $w_i(0) = \frac{1}{Pr(X_i(0))}$

end for

Run the Filter:

for t in Set of Measurement Times **do**

for i in N_p **do**

$v_{yi}(t) = f(v_{yi}(t-1), u(t))$

(All other members of $X_i(t)$ remain the same)

$w_i(t) = w_i(t-1)W(V_y(t), v_y(t))$

end for

end for

Initially the particles will have the same output, 0, however, as $u(t)$ changes, the response of each particle to that input will result in different outputs. Particles that have a v_{yi} near V_y will be weighted higher, and others farther away will be weighted lower. As the particle filter runs, weights will compound resulting in a distribution that asymptotically approaches the true joint distribution of the $X(t)$. Of course, as we mentioned in subsection 3.1.5, particles weighted zero do not significantly contribute to the empirical distribution, so re-sampling may be necessary. If the noise is assumed to be Gaussian then it is possible to further optimize. Thus we let h be defines as:

$$h = [N_s 8 c_{n_x}^{-1}(n_x + 4)(2\sqrt{\pi})^{n_x}]^{\frac{1}{n_x+4}} \tag{3.21}$$

and although it is very possible the underlying noise is non-gaussian, the Gaussian may work, but sub-optimally. It has been proposed that (Monte Carlo Approximations for General State-Space Models, markus Hurzeler and Hans R. Kunsch) if the underlying distribution is non-Gaussian, then using this bandwidth will oversmooth. In reality over smoothing should not be too great an issue because the smoothing is only being applied to find new particles. If the distribution is over smoothed then the algorithm may not converge as rapidly; however, because

the bandwidth is still based on particle variance, which will decay as particles are ruled out, it is still able to converge. In fact over smoothing is preferrable to under smoothing, since the latter would result in false negatives, but the previous only results in a slower decay of the variance. At the same time, as n_x, the number of dimensions in x, goes to infinity, the standard deviation based approximation becomes less effective (cite a Tutorial on Particle Filters for on-line non-linear non-gaussian bayesian tracking, sanjeev arulampalam, simon maskell, neil gordon...). Because of the high dimensionality of our system, and limited measurements, it is helpful to have a broader bandwidth to explore the distribution. Nevertheless, because of the potentially wide smoothing factor applied by regularized resampling, performing this step at every measurement would allow particles a great deal of mobility. This mobility is the enemy of convergence, which is why regularized resampling should only be done when $\hat{N}_{eff}$ drops very low (say less than 50). Other than the periodic regularized resampling then, the regularized particle filter is nearly identical to the basic sampling importance sampling filter (SIS).

Initialize N_p Particles: $\{x_i(0), w_i(0) : x_i(0) \sim \alpha(x), w_i(0) = \frac{1}{N_p}, i \in \{1, 2, ..., N_p\}\}$

T = {Set of Measurement Times}

for t in T **do**

  for i in N_p **do**

    $x_i(t) = x_i(t-1) + \int_{t-1}^{t} f(x(\tau), u(\tau))d\tau$

    $w_i(t) = w_i(t-1)\hat{Pr}(y(t)|x(t))$

  end for

  Calculate N_{eff} with Equation 3.15

  if $N_{eff} < N_R$ (recommend $N_R = min(50, .1N_p)$) **then**

    Calculate empirical σ

    $h = [N_s 8 c_{n_x}^{-1}(n_x + 4)(2\sqrt{\pi})^{n_x}]^{\frac{1}{n_x+4}}$

    Redraw particles using (stratified) basic resampling

    for i in N_p **do**

      Draw $\epsilon \sim K$

      $x_i = x_i + h\sigma\epsilon$

    end for

  end if

end for

At $t + \Delta t, t \in T$, $\hat{Pr}(x(t + \Delta t)) \approx \sum_{i=1}^{N_p} w_i(t)\delta\left(x - (x_i(t) + \int_t^{t+\Delta t} f(x(\tau), u(\tau))d\tau)\right)$

The ultimate effect of this regularized resampling is a convergence similar to simulated annealing or a genetic algorithm. Versions of x that are "fit" (give good measurements) spawn more children nearby which

allow for more accurate estimation near points of high likelihood. As the variance of the estimated x's decrease, the radius in which children are spawned also decreases. Eventually the radius will approach the width of the underlying uncertainty, ν_x and ν_y.

Chapter 4

Methods

This paper describes two types of experiments; first we will cover simulations, which of course have the benefit of a ground truth, second we will cover the methods for real data.

4.1 Choosing $\hat{Pr}(y(T)|x(T))$

Choosing a representation of an unknown distribution is certainly tricky, and so the fact that $\hat{Pr}(y(T)|x(T)) = \nu_d + \nu_y$ means that there is a significant piece of the algorithm that is based primarily conjecture. Studies of the noise in FMRI typically attribute noise to a Gaussian random variable or an additive noise process with Gaussian steps.

4.1.1 Classical De-trending

The non-stationary aspect of a Weiner process as with ν_d is difficult to compensate for, and so various methods have been developed to compensate for it. [Tanabe et al., 2002] and [Smith et al., 1999] have demonstrated that this component is prevalent, and may in fact be a characteristic of FMRI. In some studies, as many as half the voxels benefit from detrending, meaning that this is certainly a serious barrier to inference. All the existing methods are performed during the preprocessing stage, rather than as an integral part of analyzing the BOLD signal. There is no shortage of theories on the "best" method of detrending; however a head to head comparison, [Tanabe et al., 2002], showed that in most cases subtracting off a spline works the best. The benefit of the spline versus wavelets, high pass filtering or other DC removal techniques is that the frequency response is not set. Rather, the spline is adaptive to the input, having a low cut off if the signal's median stays constant but a high cut off frequency if the signal's median tends to shift heavily over the course of time. In spite of this, the spline

Figure 4.1: image of de-spline'd lines with "true" lines

will still remove some amount of signal, just like all of these methods, which is why the the method proposed in subsection 4.1.2 was considered.

When using the median-based spline techniques, the resulting signal is will almost certainly end with a median of zero; in other words half the signal will be above zero, half below. The problem with this is this is not the "natural" BOLD signal. More specifically, when the signal is inactive, the BOLD response *should* be at 0% change from the base level; activation may then increase, or for short periods decrease from this base. After removing the spline, the BOLD resting state will be below 0%. This is troublesome because it reduces the power the BOLD model. One obvious solution is to add an arbitrary constant to each BOLD response. Of course this won't scale well to whole brain analysis. A more effective technique is adding a DC gain to the BOLD model. Essentially every measurement would then have a constant level added to it. Like all the other model parameters, enough measurements should allow the correct value to be found. On the downside adding another dimension increases the complexity of the model, for a variable that is relatively obvious to a human.

Thus, a more conventional approach was used. A robust estimator of scale, was used to determine roughly where the base level was. The Median Absolute Deviation (MAD) proved to be extremely accurate in determining how much to shift the signal up by. We tested both methods during the course of analysis, and found that the increase in model complexity far outweighted the slight increase in flexibility. Its possible that a more accurate method may exist; however, for this case the MAD seemed to work perfectly, as Figure 4.1 shows.

4.1.2 Delta Based Inference

I also propose and test a different method of dealing with the so called "drift". Instead of comparing the direct output of the particle filter with the direct measurement, the algorith compares the change in signal over a single TR. In most signal processing cases this would foolish, but that is because the general assumption that all noise is high frequency is not the case here. In fact, every pipeline for the analysis of BOLD signal uses a high pass filter, but low poss filters are rarely applied, because it is a well known fact that most of the signal is in the high frequency range and most of the noise is actually in the low frequency range. The particle filter is an extremely robust method of inference, and so I would assert that the particle filter ought to be given as *raw* data as possible. While taking direct measurements without de-trending would give awful results, using the difference removes the DC component and turns a Weiner process into a Gaussian random variable.

$$\Delta y = y(t) - y(t-1) = g(x(t)) - g(x(t-1)) + \nu_y(t) - \nu_y(t-1) + \nu_d(t) - \nu(t-1) \tag{4.1}$$

Figure 4.2: frequency response graphs, highlighting noise frequency range and signal frequency range

Because ν_d is a Weiner process, then $\nu_d(t) - \nu_d(t-1)$ is simply a Gaussian step. If ν_d is some other additive process, the difference will still be one of a few stable distributions. If ν_y is i.i.d. then the resulting distribution will still be zero mean with a maximum variance of twice the original variance. All the assumptions made originally for the particle filter hold, and all of the parameters may be distinguished based on the step sizes, thus it is not unreasonable to attempt to match the string of step sizes rather than string of direct readings.

4.2 Preprocessing

As discussed in the section on de-trending, the normal pipeline for analyzing FMRI involves a great deal of preprocessing. In this paper we make an effort to minimize any type of preprocessing that will degrade the signal. After FMRI data has been acquired it is always necessary to modify the data in some way to make different runs comparable. Because FMRI signal levels are unit-less, at the very least it is necessary to convert the data into % difference from the baseline. This process removes no data from signal since it merely subtracting then dividing by a constant. This is the signal that was input into the delta based particle filter. Of course there are much more advanced ways of performing this task. The generally accepted standard is actually to use a high pass filter, although the cutoff frequency is application dependent and often applied haphazardly. The high pass filter thus removes the DC component of the signal, and some amount of the so called "drift". The problem with this method is that it is not adaptive to the input. Huge variations in drift frequencies can exist in a single time-series. Thus, a single cutoff frequency could miss a significant drift component, or it could remove *actual* signal, if the cutoff frequency is set too high. This is why, as I mentioned in the De-trending section, a spline based detrending method will generally give better results.

For simulated and real images (tests with multiple time-series), tests were also run with and without Gaussian filtering with sigma of were run, since it is standard practice to apply a Gaussian spatial filter to the images at each timestep. Obviously a spatial filter such as Gaussian filtering increased SNR but can also lead to less precision in the output maps.

4.3 Simulation

We performed two types of simulations. First, we simulated a single BOLD time-series based on a random chosen set of model parameters. This process was relatively straight forward given the state-space equations for the BOLD signal. After a "true" signal was generated, we then added a carrier level, since BOLD is typically

measured as a % difference from the base level. Finally, we added Gaussian noise, and a Weiner Process to the clean signal. The variance of the Gaussian noise may be expressed in terms of the desired noise SNR, R as:

$$var(y_{noisy}) = var(y)/R \tag{4.2}$$

Since SNR doesn't have quite the same meaning for a Wiener process based noise, the variance of the Gaussian steps was set to be:

$$var(y_{noisy}) = var(y)/(4R) \tag{4.3}$$

Once this noisy simulated time series was generated, the exact same particle filter algorithm that would later be run on full sized images, was run on this single voxel image. We ran a series of tests to determine the convergence rate of the particle filter, the number of particles that were required, how weighting functions compared, how different de-trending methods compared with each other and, finally the variance of the result. By running the exact time-series with different noise realizations, it was possible to determine the model variance. As the reader may know, the error of an estimator may be calculated as:

$$MSE(\Theta) = Var(\Theta) + Bias(\Theta)^2 \tag{4.4}$$

The variance is an expression of how much the result would change for different noise realizations, whereas the bias is an expression of how well the model matches the true underlying model. In this case, because the same model is being used in the particle filter and underlying simulation, the bias is actually zero. Obviously when this is calculated using *real* data with an unknown underlying state space equation, there will be some amount of bias error, but assuming that the noise is similar to the noise used in these tests, the model variance will actually be about the same. Thus calculating the model variance is extremely helpful in calculating how well determined our model is, and how consistent it will be for real data. A single timeseries, as opposed to the thousands present in a real image, makes it easier to compare the output with the ground truth, with various parameters.

Second we used a modified version of the FSL tool POSSUM to generate an entire FMRI image from a parameter map. The parameter map was generated by creating a random image, smoothing it with a large Gaussian kernel, then thresholding the results. Finally connected regions were each given a set of parameters from a finite list of randomly chosen parameter sets. The result was a four dimensional (length x width x height x parameter) image with spatially varying parameters. Time-series of activation level was generated for each set of parameters, then activation levels were fed into POSSUM's function for generating frequency domain data. The patche for POSSUM will be made available. For each time-series in the simulate FMRI image, the

final *static* parameters are saved into a parameter map. This parameter map may then be compared to the map used to generate the simulated data; additionally a new simulation using the calculated parameters may also be generated to test the functional difference between the two maps. This would give an absolute quantitative difference between the two parameter sets irrespective to parameter slopiness. So for instance, if V_0 is halved, ϵ doubling may very well give a similar result. In this case the % difference between the parameters will be large in each case, but the functional difference between the parameters will not be great. This is obviously a bad situation, which is why we wanted to test for it.

4.4 Real Data

Finally, we also performed inference based on real FMRI data. The scanner we used...

The final result from calculating parameters with the real data was similar to that from the results from the POSSUM simulated data. The difference being that there was no ground truth the check it with.

Chapter 5

Results

5.1 Single Time-Series Simulation

Graphs:

For simulated data, single timeseries:

For {delta, DC/Spline}, {exponential, gaussian, cauchy}, {biased, unbiased initial}, {100, 500, 1000} particles

1. Ground truth vs. Estimated signal during particle filter run
2. Ground truth vs. Estimated signal with final parameter set
3. True Parameters vs. Final Parameter Sets
4. Variance of final parameters when faced with same ground truth, different noise
5. MSE of (a new timeseries based on X(t) vs. ground truth) for all t
6. Estimator Variance based on different noise runs
7. Final Particle Distribution

For Simulated Data, Full Volume:

5.2 Simulated Volume

1. Parameter Map
2. Error map of parameters

3. Histogram of %errors between parameters

4. Activation Map based on a single region with high ϵ, compared with linear

5.3 FMRI Data

....

image comparing epsilon-map with GLM activation map

Chapter 6

Conclusion

Bibliography

[Behzadi and Liu, 2005] Behzadi, Y. and Liu, T. T. (2005). An arteriolar compliance model of the cerebral blood flow response to neural stimulus. *NeuroImage*, 25:1100–1111.

[Buxton et al., 2004] Buxton, R. B., Uludag, K., Dubowitz, D. J., and Lui, T. (2004). Modeling the hemodynamic response to brain activation. *NeuroImage*, 23 Suppl 1:S220–33.

[Buxton et al., 1998] Buxton, R. B., Wong, E. C., and Frank, L. R. (1998). Dynamics of blood flow and oxygenation changes during brain activation: the balloon model. *Magn. Reson. Med.*, 39:855–864.

[Chen and Pike, 2009a] Chen, J. J. and Pike, G. B. (2009a). Origins of the BOLD post-stimulus undershoot. *NeuroImage*, 46(3):559–68.

[Chen and Pike, 2009b] Chen, J. J. and Pike, G. B. (2009b). Origins of the BOLD post-stimulus undershoot. *NeuroImage*, 46(3):559–68.

[Deneux and Faugeras, 2006] Deneux, T. and Faugeras, O. (2006). Using nonlinear models in fMRI data analysis: model selection and activation detection. *NeuroImage*, 32(4):1669–1689.

[Donahue et al., 2009] Donahue, M. J., Stevens, R. D., de Boorder, M., Pekar, J. J., Hendrikse, J., and van Zijl, P. C. M. (2009). Hemodynamic changes after visual stimulation and breath holding provide evidence for an uncoupling of cerebral blood flow and volume from oxygen metabolism. *Journal of cerebral blood flow and metabolism : official journal of the International Society of Cerebral Blood Flow and Metabolism*, 29(1):176–85.

[Frahm et al., 2008] Frahm, J., Baudewig, J., Kallenberg, K., Kastrup, A., Merboldt, K. D., and Dechent, P. (2008). The post-stimulation undershoot in BOLD fMRI of human brain is not caused by elevated cerebral blood volume. *NeuroImage*, 40(2):473–81.

[Friston, 2008] Friston, K. J. (2008). Variational filtering. *NeuroImage*, 41:747–766.

[Friston et al., 2000] Friston, K. J., Mechelli, A., Turner, R., and Price, C. J. (2000). Nonlinear responses in fMRI: the Balloon model, Volterra kernels, and other hemodynamics. *NeuroImage*, 12:466–477.

[Friston et al., 2002] Friston, K. J., Penny, W., Phillips, C., Kiebel, S., Hinton, G., and Ashburner, J. (2002). Classical and Bayesian inference in neuroimaging: theory.

[Friston et al., 2008] Friston, K. J., Trujillo-Barreto, N., and Daunizeau, J. (2008). DEM: a variational treatment of dynamic systems. *NeuroImage*, 41(3):849–85.

[Handwerker et al., 2004] Handwerker, D. a., Ollinger, J. M., and D'Esposito, M. (2004). Variation of BOLD hemodynamic responses across subjects and brain regions and their effects on statistical analyses. *NeuroImage*, 21(4):1639–51.

[Hofmann, 1997] Hofmann, D. A. (1997). An Overview of the Logic and Rationale of Hierarchical Linear Models. *Journal of Management*, 23(6).

[Hu et al., 2009] Hu, Z., Zhao, X., Liu, H., and Shi, P. (2009). Nonlinear Analysis of the BOLD Signal. *EURASIP Journal on Advances in Signal Processing*, 2009:1–14.

[Johnston et al., 2007] Johnston, L. a., Duff, E., Mareels, I., and Egan, G. F. (2007). Nonlinear estimation of the BOLD signal. *NeuroImage*, 40(2):504–14.

[Johnston et al., 2008] Johnston, L. a., Duff, E., Mareels, I., and Egan, G. F. (2008). Nonlinear estimation of the BOLD signal. *NeuroImage*, 40(2):504–14.

[Lu et al., 2004] Lu, H., Golay, X., Pekar, J. J., Zijl, V., and P.c.m (2004). Sustained poststimulus elevation in cerebral oxygen utilization after vascular recovery. *J. Cereb. Blood Flow Metab.*, 24:764–770.

[Mandeville et al., 1999a] Mandeville, J., Marota, J., Ayata, C., Zaharchuk, G., Moskowitz, M., Rosen, B., and Weisskoff, R. (1999a). Evidence of a cerebrovascular postarteriole Windkessel with delayed compliance. *Journal of cerebral blood flow and metabolism : official journal of the International Society of Cerebral Blood Flow and Metabolism*, 19(6):679–689.

[Mandeville et al., 1999b] Mandeville, J. B., Marota, J. J. A., Ayata, C., Moskowitz, M. A., Weisskoff, R. M., and Rosen, B. R. (1999b). MRI Measurement of the Temporal Evolution of Relative CMRO 2 During Rat Forepaw Stimulation. *Magnetic Resonance in Medicine*, 951:944–951.

[Obata, 2004] Obata, T. (2004). Discrepancies between BOLD and flow dynamics in primary and supplementary motor areas: application of the balloon model to the interpretation of BOLD transients. *NeuroImage*, 21(1):144–153.

[Ogawa et al., 1993] Ogawa, S., Menon, R. S., Tank, D. W., Kim, S., Merkle, H., Ellermann, J. M., and Ugurbilt, K. (1993). Functional brain mapping by blood oxygenation level-dependent contrast magnetic resonance imaging A comparison of signal characteristics with a biophysical model. *Biophysical Journal*, 64:803–812.

[Ozaki, 1994] Ozaki, T. (1994). The Local Linearization Filter with Application to Nonlinear System Identifications. In *Proceedings of the first US/Japan Conference on the Frontiers of Statistical Modeling: An Informational Approach*, pages 217–240, Dordrecht. Kluwer Academic Publishers.

[Riera et al., 2004] Riera, J. J., Bosch, J., Yamashita, O., Kawashima, R., Sadato, N., Okada, T., and Ozaki, T. (2004). fMRI activation maps based on the NN-ARx model. *NeuroImage*, 23:680–697.

[Riera et al., 2003] Riera, J. J., Watanabe, J., Kazuki, I., Naoki, M., Aubert, E., Ozaki, T., and Kawashima, R. (2003). A state-space model of the hemodynamic approach: nonlinear filtering of BOLD signals. *NeuroImage*, 21:547–567.

[Shen et al., 2008] Shen, Q., Ren, H., and Duong, T. Q. (2008). CBF, BOLD, CBV, and CMRO(2) fMRI signal temporal dynamics at 500-msec resolution. *Journal of magnetic resonance imaging : JMRI*, 27(3):599–606.

[Smith et al., 1999] Smith, A. M., Lewis, B. K., Ruttimann, U. E., Ye, F. Q., Sinnwell, T. M., Yang, Y., Duyn, J. H., and Frank, J. A. (1999). Investigation of Low Frequency Drift in fMRI Signal. 533:526–533.

[Smith et al., 2007] Smith, A. T., Singh, K. D., and Balsters, J. H. (2007). A comment on the severity of the effects of non-white noise in fMRI time-series. *NeuroImage*, 36(2):282–8.

[Tanabe et al., 2002] Tanabe, J., Miller, D., Tregellas, J., Freedman, R., and Meyer, F. G. (2002). Comparison of detrending methods for optimal fMRI preprocessing. *NeuroImage*, vol:15no4pp902–907.

[Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press, Cambridge, MA.

[Vakorin et al., 2007] Vakorin, V. a., Krakovska, O. O., Borowsky, R., and Sarty, G. E. (2007). Inferring neural activity from BOLD signals through nonlinear optimization. *NeuroImage*, 38(2):248–60.

[Weisskoff et al., 1994] Weisskoff, R. M., Zuo, C. S., Boxerman, J. L., and Rosen, B. R. (1994). Microscopic susceptibility variation and transverse relaxation : theory and experiment. *Magnetic resonance in medicine*, 31(6):601–610.

[Worsley et al., 2004] Worsley, K. J., Taylor, J. E., Tomaiuolo, F., and Lerch, J. (2004). Unified univariate and multivariate random field theory. *NeuroImage*, 23 Suppl 1:S189–95.

[Yacoub et al., 2006] Yacoub, E., Ugurbil, K., and Harel, N. (2006). The spatial dependence of the poststimulus undershoot as revealed by high-resolution BOLD- and CBV-weighted fMRI. *J. Cereb. Blood Flow Metab.*, 26:634–644.

[Zheng et al., 2005] Zheng, Y., Johnston, D., Berwick, J., Chen, D., Billings, S., and Mayhew, J. (2005). A three-compartment model of the hemodynamic response and oxygen delivery to brain. *NeuroImage*, 28(4):925–39.