Full Brain Blood-Oxygen-Level-Dependent Signal Parameter Estimation Using Particle Filters

Micah C. Chambers

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Chris L. Wyatt, Chair
William T. Baumann
Aloysius. A. Beex
Daniel J. Stilwell

August 30, 2010
Blacksburg, Virginia

Keywords: BOLD Response, FMRI, Nonlinear Systems, Particle Filter, Bayesian Statistics, System Identification

Full Brain Blood-Oxygen-Level-Dependent Signal Parameter Estimation Using Particle Filters

Micah C. Chambers

(ABSTRACT)

Traditional methods of analyzing functional Magnetic Resonance Images use a linear combination of just a few static regressors. This work demonstrates an alternative approach using a physiologically inspired nonlinear model. By using a particle filter to optimize the model parameters, the computation time is kept below a minute per voxel without requiring a linearization of the noise in the state variables. The activation results show regions similar to those found in Statistical Parametric Mapping; however, there are some notable regions not detected by that technique. Though the parameters selected by the particle filter based approach are more than sufficient to predict the Blood-Oxygen-Level-Dependent signal response, more model constraints are needed to uniquely identify a single set of parameters. This ill-posed nature explains the large discrepancies found in other research that attempted to characterize the model parameters. For this reason the final distribution of parameters is more medically relevant than a single estimate. Because the output of the particle filter is a full posterior probability, the reliance on the mean to estimate parameters is unnecessary. This work presents not just a viable alternative to the traditional method of detecting activation, but an extensible technique of estimating the joint probability distribution function of the Blood-Oxygen-Level-Dependent Signal parameters.

Acknowledgments

This thesis would not have been possible without the guidance, advice and patience of my advisor, Chris Wyatt. I knew nothing about being a researcher or about medical imaging when I came to him, and he taught me to read broadly and to always dig deeper. Thank you Dr. Wyatt.

I would also like to thank William Baumann for his help and inspiration. I have learned an incredible amount reading papers with you and Dr. Wyatt.

The Neuroscientists at Wake Forest have also been extremely helpful. In particular I would like to thank Paul Laurienti for help gathering fMRI data and using SPM.

To my parents, for all the psychological (and financial) help along the way, thank you so much. You were always there for me, I couldn't ask for better parents. Love you guys.

Finally I would like to thank my friends, for the support, for the prayer, and, yes, for the occasional stress-relieving game of Halo, SOASE, and every type of SC. You guys are the best.

Contents

List of Figures

List of Tables

List of Acronyms

BOLD Blood-Oxygen-Level-Dependent . . . 2
CBF Cerebral Blood Flow . . . 4
CBV Cerebral Blood Volume . . . 4
CMRO2 Cerebral Metabolic Rate of Oxygen . . . 3
DC Direct Current . . . 16
dHb Deoxygenated Hemoglobin . . . 2
EM Expectation-Maximization . . . 12
EPI Echo Planar Imaging . . . 3
fMRI Functional Magnetic Resonance Imaging . . . 2
FSL FMRIB Software Library . . . 65
FWHM Full-Width Half-Maximum . . . 72
GA Genetic Algorithms . . . 13
GLM General Linear Model . . . 7
HRF Hemodynamic Response Function . . . 2
Hb Hemoglobin . . . 4
MAD Median Absolute Deviation . . . 36
MI Mutual Information . . . 63
MR Magnetic Resonance . . . 3
MRI Magnetic Resonance Imaging . . . 2
MSE Mean Squared Error . . . 27
O2Hb Oxygenated Hemoglobin . . . 2
ODE Ordinary Differential Equation . . . 91
PDF Probability Density Function . . . 22
POSSUM Physics-Oriented Simulated Scanner for Understanding MRI . . . 41
RF Radio Frequency . . . 4
RMSE Root Mean Squared Error . . . 49
RMSR Root Mean Squared Residual . . . 46
SA Simulated Annealing . . . 13
SNR Signal-to-Noise Ratio . . . 4
SPM Statistical Parametric Mapping . . . 5

T1 Longitudinal 4
T2 Spin-Spin 2
TR Repetition Time 2
UKF Unscented Kalman Filter 13

Chapter 1

Introduction

1.1 Overview

Traditional methods of analyzing timeseries images produced by Functional Magnetic Resonance Imaging (fMRI) perform regression using the linear combination of explanatory variables. Though adding degrees of freedom naturally mandates more computation, in this thesis I will demonstrate the use of the Particle Filters to estimate the governing parameters of the Blood-Oxygen-Level-Dependent (BOLD) model at a computation cost that would still allow real time calculations for multiple voxels. More practically, this method is an alternative method of detecting neural activity from the traditionally used Statistical Parametric Mapping (SPM). Though more computationally intense, this method is capable of modeling nonlinear effects, is conceptually simpler and provides more detailed output. Additionally, by using a separate particle filter for each single time series it is possible to estimate parameters and make real-time predictions for small neural regions, a feature which could be useful towards real time fMRI [8]. Future works will also benefit from the ability to apply conditions to the posterior distribution in post-processing without recalculating parameters; for instance, to impose additional physiological constraints. Modeling the BOLD response as a nonlinear system is the best way to determine the correlation of a stimulus sequence with the BOLD response; yet in the past doing this on a large scale has been far too computationally taxing. The solution used here takes approximately 40 seconds for a single 5 minute time series (with Core 2 Duo Q6600).

This thesis is organized as follows. The introduction will introduce BOLD, the method by which neural time changing data is detected. This section will also describe the basic form of the BOLD model - which drives the detectable changes in Magnetic Resonance (MR) signal. Chapter 2 will discuss other methods of analyzing fMRI images as well as other techniques that have been, or could be applied to the nonlinear regression model described here. Chapter 3 derives the particle filter using Bayesian statistics and discusses some practical elements of implementing the particle filter algorithm. Chapter 4 then goes into further detail about the specific particle filter configu-

ration used in this work. This section also describes the pre-processing pipeline. The results are described separately for simulated data and real fMRI data in Chapter 5 and Chapter 6, respectively. In Chapter 7 the results and their implications are further discussed. Future improvements to this technique, as well as applications are then explored in Section 7.3. Finally in Section 7.2 the significant findings and impact of this work are reviewed.

1.2 Historic Context

For the past twenty years, Functional Magnetic Resonance Imaging (fMRI) has been at the forefront of cognitive research. Despite its 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 simple to setup quick and inexpensive. By modeling the governing equations behind the neural response that drives fMRI, it is possible to increase the power of fMRI. The underlying state equations hold important information about how individual brain regions react to stimuli. The model parameters on the other hand, hold important information about the patients individual physiology including existing and future pathologies. In short, the long chain of events driving fMRI signals contain information beyond correlation with stimuli.

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. [30]. The seminal work by Buxton et al. attempted to explain the time evolution of the BOLD signal using a windkessel model to describe the local changes in Deoxygenated Hemoglobin (dHb) content [6]. Incremental improvements were made to this model until Friston et al. brought all the changes together into a single complete set of equations [14]. While there have been numerous adaptations in the model, many of them summarized by Deneux et al., even the basic versions have less bias error than the empirically driven Canonical Hemodynamic Response Function (HRF) [9, 15]. On the other hand BOLD signal models have numbers of parameters ranging from seven [33] to 50 [2] for a signal as short as 100 samples long. This number of parameters presents a significant risk of being underdetermined and having high computation cost. In this work, only the simplest physiologically inspired model will be used (with 7 parameters), and steps will be taken to make the most of computation time.

1.3 Overview

Detecting neural activity using the changes in fMRI images is based on the BOLD signal. The BOLD signal is caused by minute changes in the ratio of dHb to Oxygenated Hemoglobin (O2Hb) in blood vessels throughout the brain [30]. Because dHb is paramagnetic, higher concentrations attenuate the signal detected during Spin-Spin (T2)-weighted Magnetic Resonance Imaging (MRI) techniques. The most common fMRI imaging technique, due to its rapid Repetition Time (TR), is

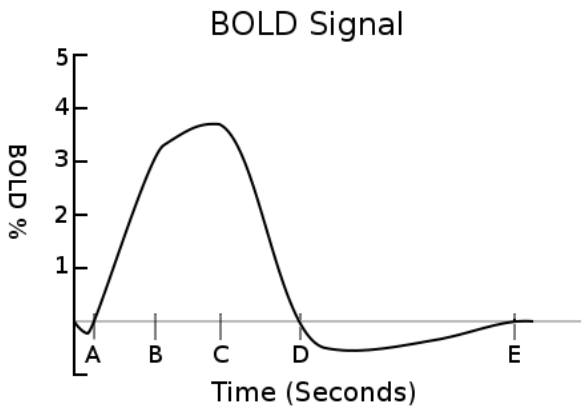


Figure 1.1: A) End of Pre-Stimulus Undershoot, $< 2s$ typical, B) End of Stimulus, C) Input Bloodflow Drops from Supernormal to Subnormal, $5s$ after B typical, D) Beginning of Post-stimulus Undershoot, $12s$ after B typical, E) Full return to normal, between $20 - 30s$ after B. [2]

Echo Planar Imaging (EPI). When axons become active, large amounts 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 significantly active, the action potentials occur more often, resulting in a local increase of the Cerebral Metabolic Rate of Oxygen (CMRO2). Thus, if everything else stays the same, blood vessels in an active area will tend to have less O2Hb, and more dHb (due to the increased rate at which oxygen is being consumed). This would then result in an attenuated fMRI signal. However, to compensate for activation, muscles that control blood vessels relax allowing increased blood flow, which actually overcompensates. This ultimately results in lower than average concentration of dHb. Thus, the BOLD signal consists of a short initial dip in the Magnetic Resonance (MR) signal, followed by a prolonged increase in signal that slowly settles out. It is the overcompensation that provides the primary signal detected with fMRI. This cascade of events is believed to drive a prolonged increase in local metabolism, blood flow, blood volume, and O2Hb. The differences in onsets and recovery times of these variables are what cause the distinguishing characteristics of the BOLD signal. Unfortunately, fMRI signal levels are all relative: within a particular person, scanner and run.

1.4 fMRI

MRI, is a method of building 3D images non-invasively, based on the difference between nuclear spin relaxation times in different molecules. The technique is well established and documented in various works [4]. 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 nuclear spins return their original state at different rates, called the Longitudinal (T1) relaxation time, depending on the material excited. Additionally, the coherence of the spins also decays differently (and roughly an order of magnitude faster than T1 relaxation) based on the properties of the region [4]. 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 is applied, whereas $T2^*$ relaxation times are usually less than 100ms. 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 [29]. Because the entire k-space (spatial-frequency) volume is acquired from a single excitation, the Signal-to-Noise Ratio (SNR) is very low in EPI [36].

Increasing the spatial resolution of EPI necessarily requires more time or faster magnetic field switching. Increasing switching rates can result in more artifacts and lower signal to noise ratios. The result is that at best fMRI is capable of 1 second temporal resolution [21, 32]. The signal is also diluted because each voxel contains a variety of neurons, capillaries and veins [29]. 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 1.3, and explored in depth in Section 1.5, the usefulness of fMRI comes from discerning changes in dHb/O2Hb [30, 29]. Therefore, it is necessary to assume that the only chemical changes will be in capillary beds feeding neurons. In practice this may not be the case, for instance near large veins, and it may explain some of the noise seen in fMRI imaging (see Section 4.2.1). Because MRI is unitless 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.

1.5 BOLD Physiology

It is well known that the two types of Hemoglobin (Hb) act as contrast agents in EPI imaging [6, 42, 30], however the connection between dHb/O2Hb and neural activity is non-trivial. Intuitively, increased metabolism will increase dHb, 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. Since the local MR signal depends on the ratio of dHb to O2Hb, increased blood volume affects this ratio if metabolism doesn't exactly match the increased inflow of oxygenated blood. This was the impetus for the ground breaking balloon model [6] and windkessel model [25]. These works derive, from first principals, the changes in dHb ratio and volume of capillaries given an inflow waveform. 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).

Although Buxton et al. demonstrated that a well chosen flow waveform could explain most features of the BOLD signal, it stopped short of proposing a realistic waveform for the CBF and CMRO2 [6]. Friston et al. gave a reasonable and simple expression for CBF input based on a flow inducing signal and in the same work proposed a simple method of estimating metabolic rate: as a direct function of the inward blood flow [14]. By combining these equations with the balloon model from Buxton et al., it is possible to predict the BOLD signal directly from a stimulus time course:

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

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

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

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

where s is a flow inducing signal, f is the input CBF, v is normalized CBV, and q is the normalized local dHb content. The parameters controlling blood flow are ϵ, which is a neuronal efficiency term, $u(t)$ which is the stimulus, and τ_f, τ_s which are time constants. The parameters for the evolution of blood volume are E_0 which the resting metabolic rate and α which is Grubb's parameter controlling the balloon model. τ_0 is a single time constant controlling the speed of v and q.

This completed balloon model was assembled and analyzed by Riera et al. [34]. Obata refined the readout equation of the BOLD signal based on the dHb content (q) and local blood volume (v), resulting in the final BOLD equation [29].

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

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

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

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

Where $\nu_0 = 40.3s^{-1}$ is the frequency offset in Hz for fully de-oxygenated blood (at 1.5T), $r_0 = 25s^{-1}$ is the slope relating change in relaxation rate with change in blood oxygenation, TE is the echo time used by the EPI sequence and $\epsilon_0 = 1.43$ is the ratio of signal MR from intravascular to extravascular regions at rest. Although, these constants change with experiment (TE, ν_0, r_0), patient, and brain region (E_0, r_0), often the estimated values by Obata are taken as the constants $a_1 = k_1 + k_2 = 3.4$, and $a_2 = k_2 + k_3 = 1$ in studies using 1.5 Tesla scanners [29]. While this model is more accurate than the static hemodynamic model used in Statistical Parametric Mapping (SPM), there are other additions which give it more degrees of freedom.

1.6 Post Stimulus Undershoot

Although the most widely used, the BOLD model described in Equation 1.4 and Equation 1.8 has been extended in various fashions. The most significant feature missing from the original model

is the post-stimulus undershoot. The post-stimulus undershoot is the term used for a prolonged subnormal BOLD response for a period of 10 to 60 seconds after stimulus has ceased [7, 24].

Because Equation 1.4 is not capable of producing such a prolonged undershoot, additional factors must be present. Two prominent theories exist to explain the post stimulus undershoot. Recall that a lower than base signal means that there is an increased dHb 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 studies that show CBV and CBF returning to the baseline before the BOLD signal [11, 10, 5, 23, 35]. Unfortunately, because of limitations on fMRI and in vivo CBV/CBF measurement techniques it is difficult to isolate whether CBF and CBV truly have returned to their baseline. Other studies indicate that there can be a prolonged supernormal CBV, although none of these papers completely rule out the possibility of increased CMRO2 [24, 2, 7]. The discrepancies may in part be explained by a spatial dependence in the post-stimulus undershoot; described by Yacoub et al. [44]. In Chen et al. a compelling case is made that most of the post stimulus undershoot can be explained by combination of a prolonged CBV increase, and a prolonged CBF undershoot [7]. Additionally it was proposed that the previous measurements showing a quick recovery of CBV were in fact showing a return to baseline by arterial CBV (which has little effect on the BOLD signal).

Regardless of the probability that CMRO2 and CBF are detached, research into the post-stimulus undershoot has led to the creation of much more in depth models. In Zhen et al. additional state variables model oxygen transport, whereas Buxton et al. models CMRO2 from a higher level, and somewhat more simply; though it still adds 9 new parameters [46, 5]. Behzadi et al. introduced nonlinearities into the CBF equations as a method to explain the post-stimulus undershoot, which falls in line with a prolonged increase in CBF observed by Chen et al. [2, 7]. Similarly Zheng et al. added additional compartments for venous and arterial blood [45]. Deneux et. al. compared these models and though that work did not deal extensively with the post-stimulus undershoot, it did show incremental improvements in quality from the additional parameters [9]. Importantly, Deneux et al. did show that by simply adding viscoelastic terms from Buxton et al. a slowed return to baseline is possible to model, without greatly increasing complexity [9, 5]. Regardless, because these models are more complex, and the parameters are not well characterized, in this work the simple Balloon model is used.

In summary, there have been extensive refinements to the Balloon model; however, the increased complexity and lack of known priors make these models difficult to work with. Additional degrees of freedom could also make parameter estimation intractable.

1.7 Properties of the BOLD Model

Since the first complete BOLD model was proposed by Friston et al. [14], several studies have analyzed its properties. The most important property is that the system is dissipative, and given enough time will converge to a constant value. This is found simply by analyzing the eigenvalues

Parameter	Friston et al. [14]	Jonston et al. [20]	Vakorin et al. [40]	Deneux et al. [9]
τ_0	$N(.98, .25^2)$	8.38 ± 1.5	.94	.27
α	$N(.33, .045^2)$	$.189 \pm .004$	.4 (NC)	.63
E_0	$N(.34, .1^2)$	$.635 \pm .072$	.6 (NC)	.33
V_0	.03 (NC)	$.0149 \pm .006$	(NC)	.16
τ_s	$N(1.54, .25^2)$	4.98 ± 1.07	2.2	2.04
τ_f	$N(2.46, .25^2)$	8.31 ± 1.51	.45	5.26
ϵ	$N(.54, .1^2)$	$.069 \pm .014$	(NC)	.89

Table 1.1: Parameters found in various studies. (NC) indicates that the value wasn't calculated. [40] made use of the values from [14] where not explicitly stated

of the state equation Jacobian, [9, 17]. The steady state of the Balloon model equations gives:

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

where the parameters are all the same as in Equation 1.4

In real fMRI data, there is a significant nonlinearity in response; with short sequences responding disproportionately strongly [3, 41, 9]. This nonlinearity is accounted for in the Balloon model, although Deneux et al. showed that when duration of stimuli varies greatly, modeling Neural Habituation is necessary to fully capture the range of responses [9]. Both Birn et al. and Deneux et al. found that stimuli lasting longer than 4 seconds tend to be more linear, which is why block designs are so well accounted for by the General Linear Model (GLM) (see Section 2.1.3) [3, 9].

Another interesting result from Deneux et al. was the sensitivity analysis [9]. That work found that the parameters are far from perpendicular, and that very different parameters could give nearly identical BOLD output. The means that that without constraining parameter values, they may not be precisely ascertainable. This could explain discrepancies of parameter estimates in previous studies (Table 1.1).

Chapter 2

Alternate Methods

Currently, fMRI is used to determine the location of responses to stimuli. The method of finding activation is discussed in Section 2.1. The goal of this work is to move away from simple localization, and move toward characterizing the response curve. Doing so necessitates more complex models and requires more computation. Already there have been several other attempts to model the BOLD response; these works will be discussed in this chapter.

2.1 Statistical Parametric Mapping

Although not strictly the same as parameter calculation from fMRI, activation detection is similar and worth discussing. Estimation of parameters is a generalization of the idea of activation detection. Given the popularity of SPM it is important to draw a distinction between it and the methods proposed in this work.

2.1.1 Random Field Theory

SPM methods make significant use of t-Tests across large regions; however, such t-tests work slightly differently than a single individual test. A t-test with a p-value of 0.05 over 10,000 voxels, will on average generate 500 false positives. This is called the multiple comparison problem. Traditional Bonferroni Correction deals with this by requiring each test to pass with P value of $\frac{0.05}{10000}$. The probability of a single false positive would then be 0.05. Unfortunately this leads to unrealistically low p-values; so low that it would be impossible for any biological system to satisfy. To compensate, a Gaussian kernel is applied to smooth the image. This has the benefit of reducing the noise variance and decreasing the effective number of independent measurements. Because the number of *independent* measurements is smaller, Bonferroni correction can theoretically be applied with a lower scaling factor than the original voxel count [43]. As a side effect of this

extreme smoothing, a single active voxel is very difficult to detect.

2.1.2 Classical Activation Detection

The most basic method of analyzing fMRI data is by standard t-test between resting state and active state samples. Simply put, the mean is calculated separately for non-stimulus and stimulus time intervals. A classic t-test may then be applied, giving the probability that the distributions actually have the same mean. Because of the correlated noise present in fMRI (Section 4.2.1), it is necessary to apply a high-pass filter to the data. Without applying such a filter, T-values must be set extraordinarily high to prevent false positives [36]. Because the BOLD response is known not to fit a square wave this method is rarely used (Section 1.5). For this reason other methods are often more used, as discussed in Section 2.1.3.

2.1.3 General Linear Model

The most common fMRI analysis tool is SPM, which generates statistical parametric map using more advanced methods than a simple square wave. Hierarchal Models are one important improvement that allows researchers to combine data across multiple runs, patients and stimuli (see [16] for more on Hierarchical Modeling). Hierarchical Models concatenate all the data into a single dataset, then perform a linear fit between a design matrix and the data. The design matrix encapsulates all known experimental factors such as stimuli, young/old, etc. The general linear model is defined as:

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

where $Y(t)$ is the smoothed or de-trended time course of measurements, $X(t)$ is the design matrix, 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.

As mentioned previously, a square wave stimulus does not result in a square wave in the activation of brain regions. The BOLD signal is in fact a smoothed version of the stimuli. As such, when fitting an fMRI time course to the input, the input (X's columns) is usually smoothed to reduce bias error. The best method, that maintains a linear fit, is convolving the input with a HRF. The HRF mimics the basic shape of BOLD activation, including a delay due to rise time and fall time. The fitting process is then a least squares fit over the space of the vector β. Therefore $Y(t)$ is estimated as a linear combination of the columns of X.

Smoothing the input with a single HRF poses certain problems. It is well known that different Hemodynamic Response Functions are necessary for different regions of the brain [15]. The Canonical HRF that is most often used, has been optimized for the visual cortex. As discussed

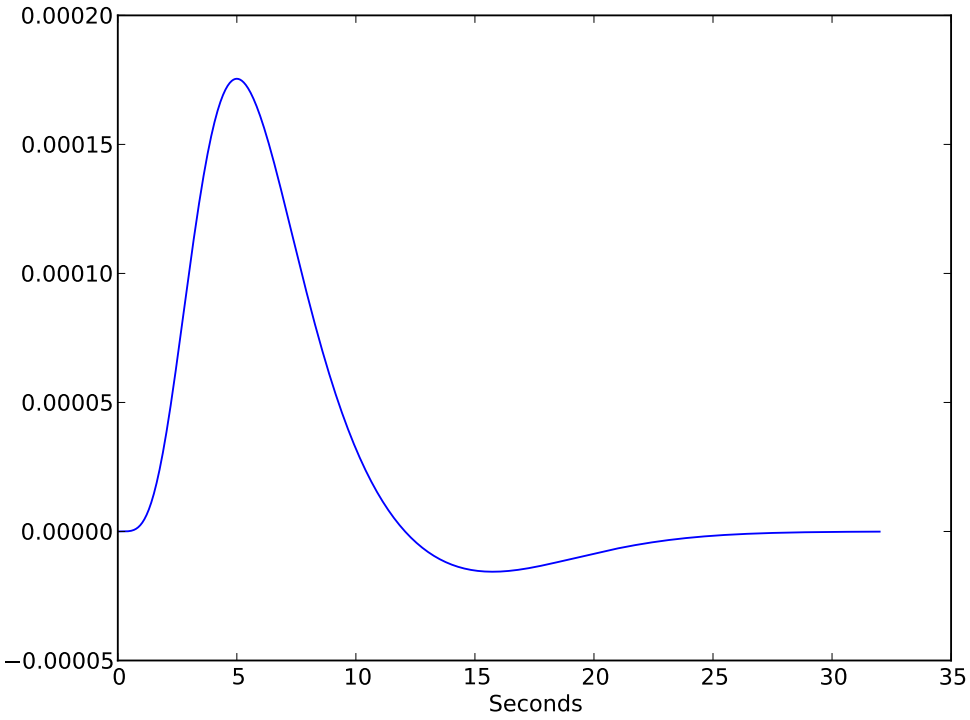


Figure 2.1: Canonical Hemodynamic Response Function, normalization makes y units arbitrary

in Section 1.6, there are definitely variations in the shape of the BOLD response, over both brain regions and patients. Handwerker et. al. discussed the implications of choosing an incorrect HRF, the most important of which is a definite increase in false negatives [15]. While an atlas of Hemodynamic Response Functions for each region could definitely mitigate this risk, it does not account for variation between patients.

2.1.4 Hierarchical Linear Models

As mentioned previously and discussed extensively in Friston et al. [13] and Hoffman et al. [13], hierarchical models may be applied to account for systematic differences between subjects. For instance, if the study mixes young and old subjects, incorporating that into the model is wise, regardless of whether it is the goal of the test. The reason to do this is to account for additional variance that may not be otherwise explainable. The Hierarchical form used by Friston et al. [13] the GLM which is shown in Equation 2.2 .

$$\begin{aligned} Y(t) &= X_1(t)\theta_1 + \epsilon_1(t) \\ \theta_1(t) &= X_2(t)\theta_2 + \epsilon_2(t) \\ &\dots \\ \theta_{n-1}(t) &= X_n(t)\theta_n + \epsilon_n(t) \end{aligned} \tag{2.2}$$

The Empirical Bayes algorithm was used by both Friston et al. [13] and Hofmann et al. [16]. Note that in Empirical Bayes, point estimators are used for each θ, rather than full distributions or the first two moments.

2.1.5 Discussion

In all, the GLM is useful for determining linear dependence of a set of regressors on the output. Unfortunately, as discussed in Section 1.7 there are significant nonlinearities that almost certainly cause false negatives in the Statistical Parametric Maps. Unfortunately nonlinear analyses have only recently become feasible, so the scale of the problem is still unknown. The problem is highlighted by the common scenario where no significant activation can be found in a single run [34, 20].

The static nature of the linear model also limits its potential use. Besides not allowing HRF differences between patients, there is no reasonable way to incorporate other forms of physiological data. Combined fMRI CBF or CBV imaging methods are improving, as seen in Chen et al. [7]. These techniques could shed light on neural activation by providing extra measurements, yet a physiologically reasonable model is necessary to incorporate this extra data. Activation detection methods also don't have the ability to identify pathologies based on state variables or parameters.

For example, decreased compliance of blood vessels could indicate, or even cause, a neurological condition that is not easily seen in other imaging modalities.

2.2 Approaches to the Balloon Model

Unlike Statistical Parametric Mapping, the techniques described in this section are all attempts to regress some version of the BOLD model. Although Buxton et al. [6] and Friston et al. [14] both proposed physiologically reasonable values for the model parameters, Friston et al. [12] was the first paper to calculate the parameters based on actual fMRI data. However, in that case, the voxels were chosen from regions that were detected as active by the GLM. It is therefore possible that the parameters are biased toward parameters that fit the linear model.

2.2.1 Polynomial Approximation

In Friston et al. [12], a novel combination of linear and nonlinear modeling proposed to generate parameter estimates. Because there is no closed form solution to the balloon model, it is impossible to calculate the Jacobian matrix $\frac{\partial Y}{\partial \theta}$. Thus Friston et al. [12] approximated the partial using a low-order Volterra Kernel. At each step of the Expectation-Maximization (EM) algorithm, the differential equation was integrated and the residuals calculated. Then, for each θ_i surrounding the estimate of θ, a Volterra Kernel expansion of the output y was generated. Generation of the Volterra Kernel is quick, and, since there is an analytical solution, calculating partial derivatives is easy. Unfortunately, it was found in that work that estimating a Volterra kernel took longer than simply integrating the state variables. The full E-M algorithm for estimating the states is notation-heavy and can be found in Friston et al. [12].

There are a few caveats with this method of optimization. First, the partials are based on approximate values (using Volterra-Kernels) of y. Importantly, the Volterra-Expansion of y is not able to model interactions between state variables; which certainly increases error [12]. Additionally, to date no extensive review of the accuracy of the Volterra expansion has been performed. Finally all the tests performed by Friston et al. were on regions found to be active by the General Linear Model [12]. For this reason, the reliability of the approximation is unknown for regions that are active but sufficiently nonlinear to avoid detection by conventional tests.

2.2.2 Nonlinear Least Squares

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 linear techniques of gradient descent are off limits; since the model is a true nonlinear dynamical system with no closed form solution (although Section 2.2.1 circumvented this by calculating a Volterra Series

approximation). Without a Jacobian for residuals, the Gauss-Newton method is impossible. Additionally, a gradient descent is slow without the ability to form partials of the output with respect to all the parameters.

Still, there are other heuristic algorithms that could estimate the BOLD response. Simulated Annealing (SA) is a common method of optimizing high dimensional regression problems. First the program selects a random starting point, then at each iteration it selects a random point in the neighborhood. If the new 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. 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. Since every step requires a re-integration of the balloon model, it can be extremely time consuming, which is why I did not use it here.

Genetic Algorithms (GA) 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 weight. New parameter sets are generated by randomly combining pieces of the old parameter sets. The pieces are chosen at a rate proportional to the fitness of the donor; thus good parameter sets tend to pass on their properties. In addition to this, random mutations are introduced that come from no existing parent. The fitness function is then used to weight the new generation, 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. As with simulated annealing (or any generic non-linear optimization), there is no guarantee that a global minimum has been reached.

Although both these methods can be highly effective, they have the downside of requiring high computation time. In the 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 I'll discuss in Chapter 3, the Particle Filter method is able to circumvent this re-calculation to some degree. Its worth noting though, that to beat the particle filter algorithm discussed in Chapter 3 these would have to converge in less than 1000 simulations (which is far less than 1000 generations in GA).

2.2.3 Unscented Kalman Filter

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 UKF generalizes the Extended Kalman Filter by allowing the state update and output functions to be arbitrary functions.

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, .09)
Q_t	N(1, .09)
S_t	N(1, .09)
F_t	N(1, .09)

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

$$X(t) = g(u(t), X(t-1)) \tag{2.3}$$
$$Y(t) = h(X(t)) \tag{2.4}$$

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 estimate of a posterior Gaussian. Hu et. al. used the UKF to perform a similar type of analysis to the one performed in this work [17].

The difficulty of using a Kalman Filter, however, is that it assumes a multivariate Gaussian for the state variables, $X(t-1)$. The more nonlinear the system gets the more likely that the Gaussian will be insufficient to describe the distribution, $X(t)$. When this occurs, every step from $X(t-1)$ to $X(t)$ will introduce additional error in the posterior 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. The assumption of Gaussianity is what allows the UKF to estimate the posterior distribution using only the first and second moments; however, if this assumption is violated significant error will result. On the other hand, for small variances and short time steps the Gaussian distribution is a good fit, and so in some cases the Unscented Kalman Filter could work quite well.

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 a four dimensional Gaussian. The states were then propagated through two seconds of simulation (a typical TR in fMRI) and plotted the resulting marginal distributions against a Gaussian distribution. This also demonstrates the degree of nonlinearity present in the system. The parameters used are shown in Table 2.1

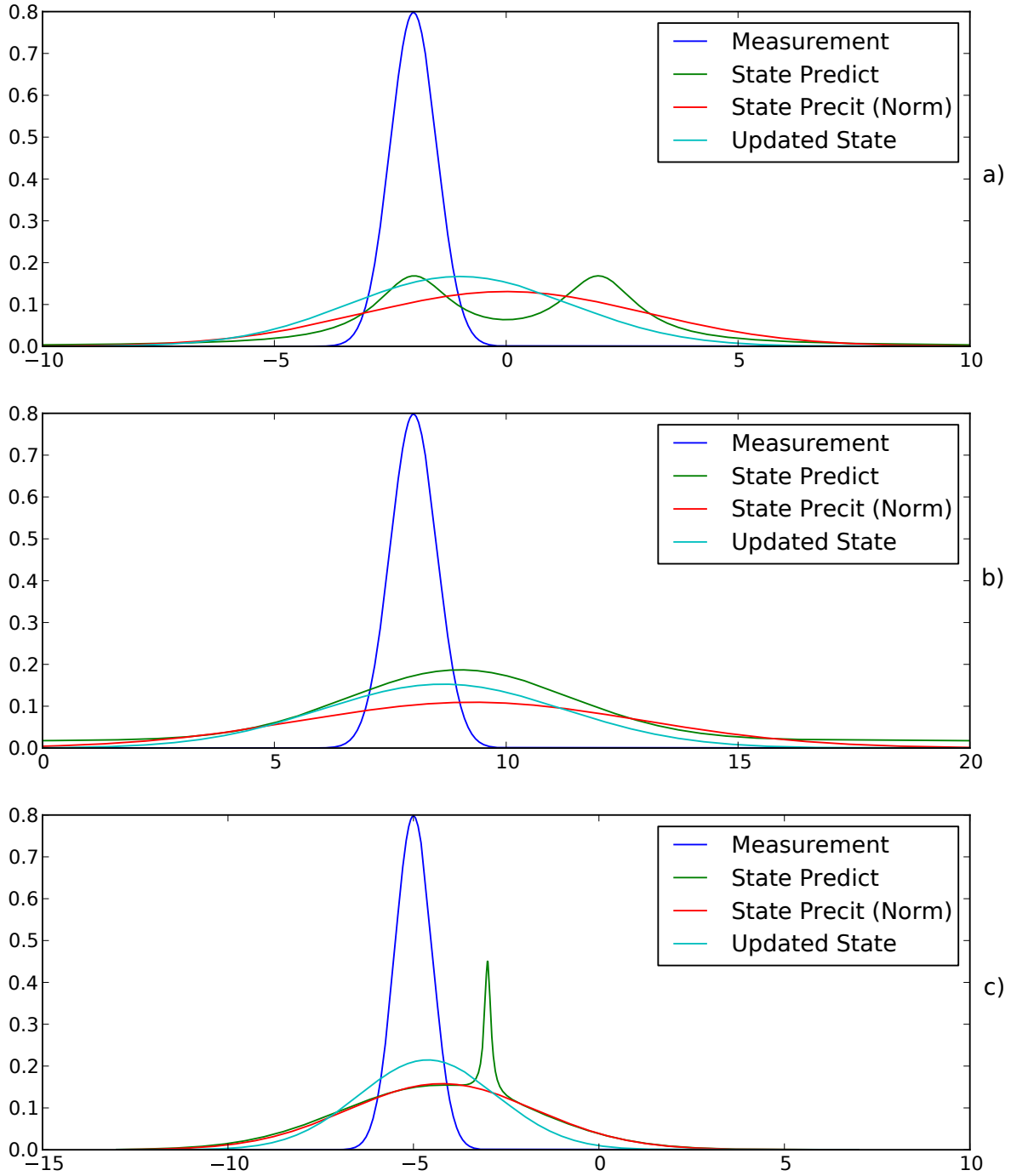


Figure 2.2: a) Sigma Points are passed through the state/measurement equations to get a state prediction (green line). The first two moments of the state prediction are then used to estimate the distribution with a Gaussian (red line). The distribution centered at the new measurement, estimating measurement error (dark blue), is then combined with the existing state prediction (red line) to get the new estimated state (light blue). The same process is applied in b), and c) with different measurements and initial state estimates.

In order to drive the system without input, the initial state was set to a non-steady but physically plausable s_t state value. The value of u was left at zero the entire time, so the system would decay naturally (see Section 1.5) Figure 2.3 shows the results after the system ran for 100 milliseconds. The Q-Q plots fit will with a Gaussian, demonstrating that at this short time interval nonlinearities have not yet begun to effect the distribution. However, Figure 2.4 shows 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 started to deviate from the Gaussian distribution. As a result the uncertainty in y deviated 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, within a period less than typical measurement rates the distribution deviated from a Gaussian substantially.

Thus, even without introducing variation in the model parameters, a distinct nonlinearity and non-Gaussianity was present. More advanced tests where parameters (especially α) are varied would certainly introduce more error into the Gaussian estimate. For this reason, estimating the posterior distribution using only the first two moments, which is the basis for the UKF, is not wise.

2.2.4 Hybrid Methods

In Riera et al. [34], a maximum likelihood method for innovation processes was used, as described by Ozakai [31]. Ozaki [31] used a similar construction to a Kalman filter on nonlinear signals [31]. The method used by Riera et al.[34] used this method to perform maximum likelihood on the innovations rather than the absolute signal levels. By using a Local Linearization Filter, innovation noise such as noise in $\dot{f}$ is turned into simple Gaussian noise. This approach is useful when the Direct Current (DC) portion of the signal is of no use; however, it depends greatly on the type of signal. I found that this method was not very effective for the BOLD signal, when I used it with the particle filter approach discussed in the next chapter; although its effectiveness depended greatly on the stimulus.

In Johnston et al. [20], a hybrid particle filter/gradient descent algorithm was used to simultaneously derive the static and dynamic parameters, (classically known as parameters and state variables, respectively) . A particle filter was used to calculate the state variables at each time; 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. This process was repeated until the parameters converged. Interestingly Johnston et al. [20] came to a very different set of parameter estimates as compared to the original Friston et al. [14] estimates (Table 1.1). In fact the results are significantly different from virtually every other study. The most obvious discrepancy is the larger time constants, τ_f, τ_s and τ_0. While of course this could be poor convergence of the algorithm, there is another other possibility. Unlike all the other methods mentioned, excepting the methods in Section 2.2.2, the algorithm described in Johnston et al. [20] does not depend on prior distributions. It is possible then that the bias toward the prior in other methods skewed their results. While Johnston et al. [20] is certainly in the minority; further exhaustive studies of the parameters, using unbiased techniques

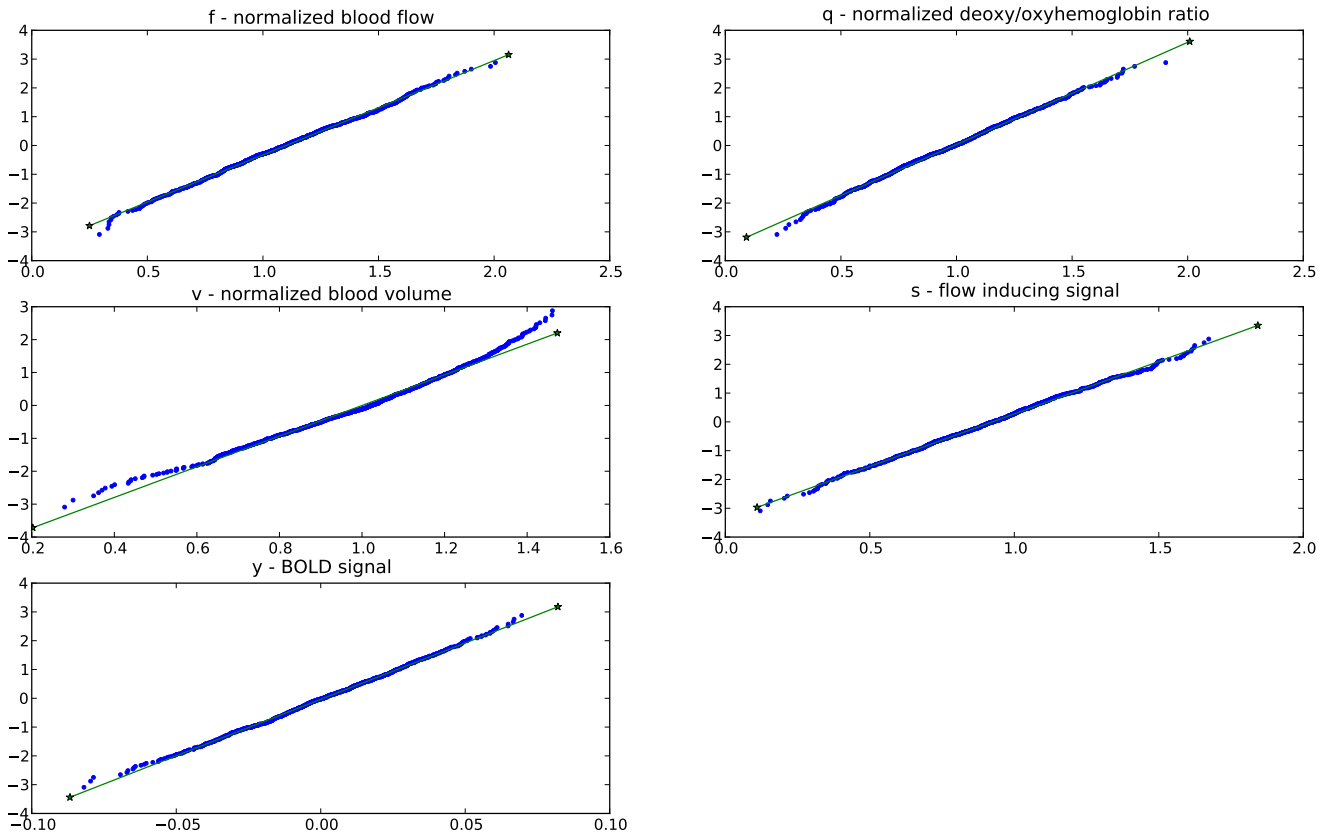


Figure 2.3: Q-Q Plot Of Standard Normal and Result of 0.1 s BOLD Simulation Output. A perfect Gaussian would fall on the green line.

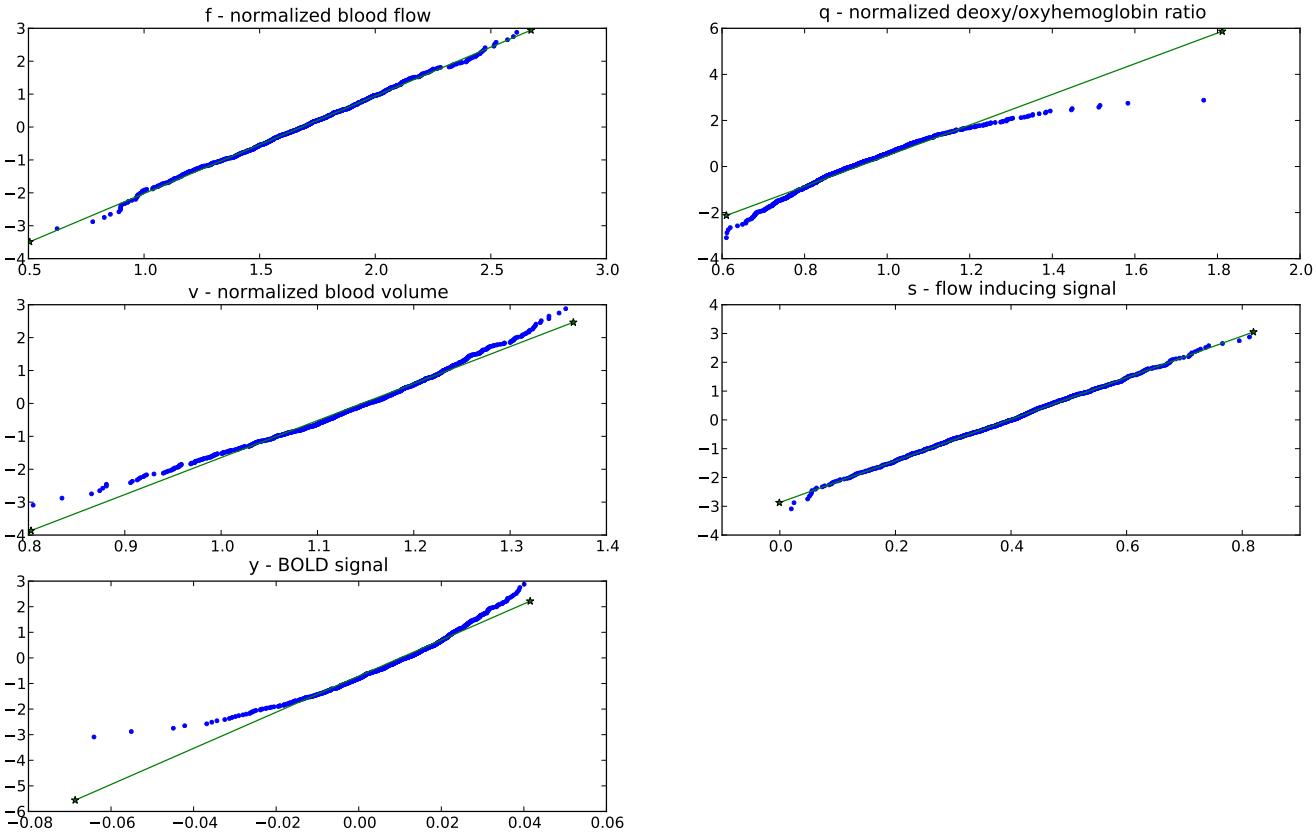


Figure 2.4: Q-Q Plot Of Standard Normal and Result of 1 s BOLD Simulation Output. A perfect Gaussian would fall on the green line.

may be called for. A further comparison between the distributions found in Johnston et al. [20] and Friston et al. [14] will be discussed in Section 4.3.1. Although Johnston et al. [20] did not explicity list computation times, from this work, a particle filter with 100 particles integrating the BOLD model should take around 5 seconds, repeated 10 times should take approximately 50 seconds for the particle filter alone. For the least squares searches performed at each iteration, computation likely takes around a minute, giving the algorithm a 10 minute computation time per voxel.

In Vakorin et al. [40], a combination of Genetic Algorithms and Simulated Annealing was used to estimate not only the parameters, but the true stimuli driving the BOLD signal. This addresses the inherent uncertainty of exactly where and when stimuli actually get applied. Unfortunately this algorithm took in more than 16 hours per voxel.

2.2.5 Previous Particle Filter Approaches

In his PhD thesis, Murray [27] used a particle filter based approach to integrate the BOLD equations. The method used in that work focused primarily on estimating the BOLD output and state equations as a nonlinear stochastic differential equation. The primary difference between that work and this is that Murray [27] took the parameters as a given. Thus, differences in the BOLD output were taken to be primarily driven by stochastic changes in the underlying state equations. Because the parameters were not allowed to change, the estimate of the BOLD signal was not very good. The fact that the differences in BOLD response cannot be explained solely by stochastic elements is important, however. The filtering framework created in Murray [27], dysii, forms the basis for the particle filter used in this work, and was well designed. The work also clearly presents the particle filter; both its derivation and use.

2.3 Conclusion

Currently there is no ideal solution to solving this system of nonlinear equations. Exhaustive search type methods such as those employed by Johnston et al. [20] and Vakorin et al. [40] have greater than 10 minute run times for a single voxel. While Volterra models are an interesting solution,there have not yet been exhaustive tests to determine whether such approximations work well throughout state space. The most promising method of those reviewed here is the Kalman filter based method. It is able to maintain a fast runtime while still approaching the solution. The reliance on a Gaussian estimate to the true posterior distribution could cause problems however. The particle filter method proposed in the next section bears a strong resemblance to the unscented Kalman filter; albeit without the Gaussian assumptions.

Chapter 3

Particle Filters

3.1 Introduction

Particle filters, a type of Sequential Monte Carlo (SMC) method, are a powerful method for estimating the posterior probability distribution of parameters given a timeseries of measurements. Unlike Markov Chain Monte Carlo (MCMC) estimation, particle filters are designed for time-varying random variables. The idea behind particle filters is similar to Kalman Filters; however, unlike Kalman Filters, distributions are stored as an empircal distribution rather than as the first two moments of a Gaussian. Thus, particle filters are preferred when the model is nonlinear, and non-gaussian. This section is largely based on Arulampalam et al. and Thrun et al. [1, 39].

3.2 Model

The idea of the particle filter is to build an empirical distribution out of a large number of parameter sets, called particles. Each particle contains all the parameters and states needed to propagate the model forward. The particle filter begins with a wide distribution (called the Prior Distribution) of possible particles and then, as measurements come in, weights particles based on the quality of their output estimates. Thus parameter sets that tend to give good estimations of the measurements get weighted higher than parameter sets that give poor estimates. Although the reliance on a prior distribution could be problematic, when the system being modeled has physical meaning, establishing reasonable ranges for parameters may be quite easy. Optimizing the prior distribution can be more difficult, unless the system has been extensively studied.

Suppose a set or stream of measurements at discrete times are given, $\{Y_k, k = 1, 2, 3, ...K\}$, where K may be infinite. Let k denote a time index and t_k define the continuous time of index k. Suppose also that there is a hidden set of state variables, $X(t)$ that drives the value of $Y(t)$. Throughout this section X_k will identify $X(t_k)$. The goal of the particle filter is to estimate the distribution of

the true parameters Θ that dictates the movement of $X(t)$. The model also permits random motion in $X(t)$, so the particle filter also estimates the distribution of $X(t)$. The only difference between the members of parameter vector Θ and those of $X(t)$ is that the members of Θ have no known update equation. Members of both vectors are permitted to have some noise, although this may not be explicitly stated in the model. The generic, continuous, nonlinear system definition is shown in Equation 3.1.

$$\begin{aligned} \dot{X}(t) &= f(t, X(t), u(t), \theta, \nu_x) \\ Y(t) &= g(t, X(t), u(t), \theta, \nu_y) \end{aligned} \tag{3.1}$$

$X(t)$ is vector of state variables, Θ is a vector of system constants, $u(t)$ is an input, $Y(t)$ the observation, and ν_x and ν_y are random variates. Although any of these variables could be a vector, for the sake of simplicity only Θ and $X(t)$ will be considered as such.

I will also make a few simplifying assumptions for this work. First, the systems are assumed to be time invariant. This assumption is based on the idea that if you paused the system for Δt seconds, when unfrozen the system would continue as if nothing happened. Few biological systems are predictable enough for them to be summarized by a time varying function, least of all the brain. While heart beats are certainly periodic and have an effect on the BOLD signal, the period varies too much for the system to be considered varying with time. Next, it was assumed that input cannot directly influence the output, which in the case of the BOLD signal is a good assumption. Finally, because the only difference between the members of $X(t)$ and Θ is an update function, from now on X will encapsulate Θ. The assumptions allow for a simplified version of the state space equations:

$$\dot{X}_k = f(X_{k-1}, u_k, \nu_x) \tag{3.2}$$

$$Y_k = g(X_k, \nu_y) \tag{3.3}$$

3.3 Sequential Importance Sampling

The goal of the particle filter is to evolve an empirical distribution $P(x_k|u_{0:k}, Y_{0:k})$, that asymptotically approaches the true probability distribution $P(X_k|u_{0:k})$. Note that capital X will be used as the actual realizations of the state variable, whereas x will denote estimates of X. Additionally, the notation $a : b$ indicates the set $[a, b]$, as in $u_{a:b}$, which would indicate all the inputs from time a to time b. Considering the noise present in X, $P(X_k|u_{0:k})$ is not a single true value but probability distribution.

To begin, the particle filter must be given a prior distribution, from which the initial N_p particles are drawn. A particle contains a weight as well as an estimate of X_k, which as previously men-

tioned, contains every variable needed to run the model. Then the prior is generated from a given distribution, $\alpha(X)$, by:

$$\{[x_0^i, w^i] : x_0^i \sim \alpha(X), w^i = \frac{1}{N_p}, i \in \{1, 2, ..., N_p\}\} \tag{3.4}$$

where N_p is the number of particles or points used to describe the prior using a Mixture Probability Density Function (PDF). Note that any exponents will be explicitly labeled as such, to avoid confusion with the particle numbering scheme.

Therefore, after the particles have been generated they should approximate $\alpha(X)$:

$$\alpha(X) \approx P(x_0) = \int_{-\infty}^{\infty} \sum_{i=0}^{N_p} w^i \delta(X - x_0^i) dx \tag{3.5}$$

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

If a flat prior is preferred, then each particle's weight could be scaled to the reciprocal of the density at the particle:

$$w^i = \frac{1}{\alpha(x_0^i)} \tag{3.6}$$

Whether or not to flatten the prior is a design decision. The reason this might be preferred over a direct uniform distribution is that the distribution width will inherently scale for increased particle counts although some distributions flatten out better than others. Either way, $\alpha(X)$ must be wide enough to incorporate any posterior that arises. If the prior is not sufficiently dense, the particle filter may be able to compensate, if it is not sufficiently wide the particle filter won't converge.

3.3.1 Weighting

For all the following areas, the probabilities implicitly depend on $u_{0:k}$, so those terms are left off for simplicity.

Whenever a measurement becomes available it permits refinement of the posterior density. This process of incorporating new data is called sequential importance sampling, and eventually allows convergence. The weight is defined as

$$w_k^i \propto \frac{P(x_{0:k}^i | y_{0:k})}{q(x_{0:k}^i | y_{0:k})} \tag{3.7}$$

where q is called an *importance density*. The importance density is the density of the points, thus by dividing by this value, the weight should not depend on the location of the estimation points, but rather only on $P(x_{0:k}^i | y_{0:k})$, the probability of that particle being correct given all the measurements up to time k. Note that if there is a distant peak in the posterior that q does not have support points

in, there will be quantization errors, and that part of the density cannot be modeled. This is why it is absolutely necessary that q fully covers $P(x^i_{0:k}|y_{0:k})$.

It is helpful to consider how the importance density affects the initial distribution. In the initial distribution, the weights are all the same; and for the sake of argument, let them all be scaled up to 1. Then

$$w^i_k q(x^i_{0:k}|y_{0:k}) = q(x^i_{0:k}|y_{0:k}) = P(x^i_{0:k}|y_{0:k}) \tag{3.8}$$

the estimated probability, $P(x^i_{0:k}|y_{0:k})$ depends only on the way the particles are distributed. As new measurements are incorporated, the weight will accumulate probabilities through time, which will be discussed next.

3.3.2 Calculating Weights

To calculate the weight of a particular particle, it is necessary to calculate both $q(x^i_{0:k}|y_{0:k})$ and $P(x^i_{0:k}|y_{0:k})$. Note that $q(x^i_{0:k}|y_{0:k})$ may be simplified by assuming that y_k doesn't contain any information about x_{k-1}. Technically this could be false; since later measurements may shed light on currently hidden changes in x, however in this work it is a good assumption because the input does not immediately effect the BOLD signal.

$$q(x^i_{0:k}|y_{0:k}) = q(x^i_{0:k}|y_{0:k-1}) \tag{3.9}$$

The choice of the importance density is another design decision; however it is common to use the integrated state equations. Although other importance density functions exist; for the particle filter used here, the standard importance density will be used: the model prior.

$$q(x_k|x_{k-1}, y_{0:k}) = P(x_k|x_{k-1}) \tag{3.10}$$

This choice for importance density is conventient since its simply the set of particles propagated forward in time using the state equations. Additionally it makes updating weights simple, as seen later in Equation 3.14.

The $q(x^i_{0:k}|y_{0:k})$ may then be simplified:

$$\begin{aligned} q(x_{0:k}|y_{0:k}) &= q(x_k|x_{0:k-1}, y_{0:k})q(x_{0:k-1}|y_{0:k}) \\ &= q(x_k|x_{0:k-1}, y_{0:k})q(x_{0:k-1}|y_{0:k-1}) && \text{[Equation 3.9]} \\ &= q(x_k|x_{k-1}, y_{0:k})q(x_{0:k-1}|y_{0:k-1}) && \text{[Markov Property]} \\ &= P(x_k|x_{k-1})q(x_{0:k-1}|y_{0:k-1}) && \text{[Equation 3.10]} \end{aligned} \tag{3.11}$$

Calculating $P(x_{0:k}|y_{0:k})$ is a bit more involved. First, using the assumption that the distribution of y_k is fully constrained by x_k, and that x_k is similarly fully constrained by x_{k-1}, I make the assumptions that:

$$\begin{aligned} P(y_k|x_{0:k}, y_{0:k-1}) &= P(y_k|x_k) \\ P(x_k|x_{0:k}, y_{0:k-1}) &= P(x_k|x_{k-1}) \end{aligned} \tag{3.12}$$

These are of course just re-statements of the state equations assumed by Equation 3.2 and Equation 3.3.

Additionally, for the particle filter y_k and $y_{0:k-1}$ are constant across all particles, thus $P(y_k|y_{0:k-1})$ can be dropped when the equality is changed to a proportion. Using these properties, $P(x^i_{0:k}|y_{0:k})$ may be broken up as follows (primarily using Bayes' Theorem):

$$\begin{aligned}
P(x_{0:k}|y_{0:k}) &= \frac{P(y_{0:k},x_{0:k})}{P(y_{0:k})} \\
&= \frac{P(y_k,x_{0:k}|y_{0:k-1})\cancel{P(y_{0:k-1})}}{P(y_k|y_{0:k-1})\cancel{P(y_{0:k-1})}} \\
&= \frac{P(y_k|x_{0:k},y_{0:k-1})P(x_{0:k}|y_{0:k-1})}{P(y_k|y_{0:k-1})} \\
&= \frac{P(y_k|x_{0:k},y_{0:k-1})P(x_k|x_{0:k-1},y_{0:k-1})P(x_{0:k-1}|y_{0:k-1})}{P(y_k|y_{0:k-1})} \\
&= \frac{P(y_k|x_k)P(x_k|x_{k-1})P(x_{0:k-1}|y_{0:k-1})}{P(y_k|y_{0:k-1})} && \text{[Equation 3.12]} \\
&\propto P(y_k|x_k)P(x_k|x_{k-1})P(x_{0:k-1}|y_{0:k-1}) && [P(y_k|y_{0:k-1}) \text{ is constant}]
\end{aligned} \tag{3.13}$$

Inserting Equation 3.10 and the result of Equation 3.13 into Equation 3.7 leads to:

$$\begin{aligned}
w^i_k &\propto \frac{P(y_k|x^i_k)\cancel{P(x^i_k|x^i_{k-1})}P(x^i_{0:k-1}|y_{0:k-1})}{\cancel{P(x^i_k|x^i_{k-1})}q(x^i_{0:k-1}|y_{0:k-1})} \\
&\propto w^i_{k-1}P(y_k|x_k)
\end{aligned} \tag{3.14}$$

Thus, by making the following simple assumptions, evolving a posterior density requires no knowledge of the noise distribution.

1. $f(t,x(t),u(t)) = f(x(t),u(t))$ and $g(t,x(t),u(t)) = g(x(t))$
2. The prior distribution PDF, $q(x_i(0))$, covers $P(x_i(0))$
3. Markov Property: $P(x_k|x_{0:k-1}) = Pr(x_k|x_{k-1})$
4. $q(x_{0:k-1}|y_{0:k}) = q(x_{0:k-1}|y_{0:k-1})$

From the definition of w_i, the algorithm to calculate an approximation of $P(X(t_k)|Y_{0:k})$ or $P(X(t_k+\delta t)|Y_{0:k})$ is simple. This basic form of the particle filter is given in algorithm 3.1.

3.4 Sequential Importance Resampling

As a consequence of the wide prior distribution (required for a proper discretization of a continuous distribution), a significant proportion of particles will quickly develop insignificant weights. While this does help describe the tails of the distribution, it means a great deal of computation will be wasted. Instead, it would be preferable if most of the computation is spent on the most probable

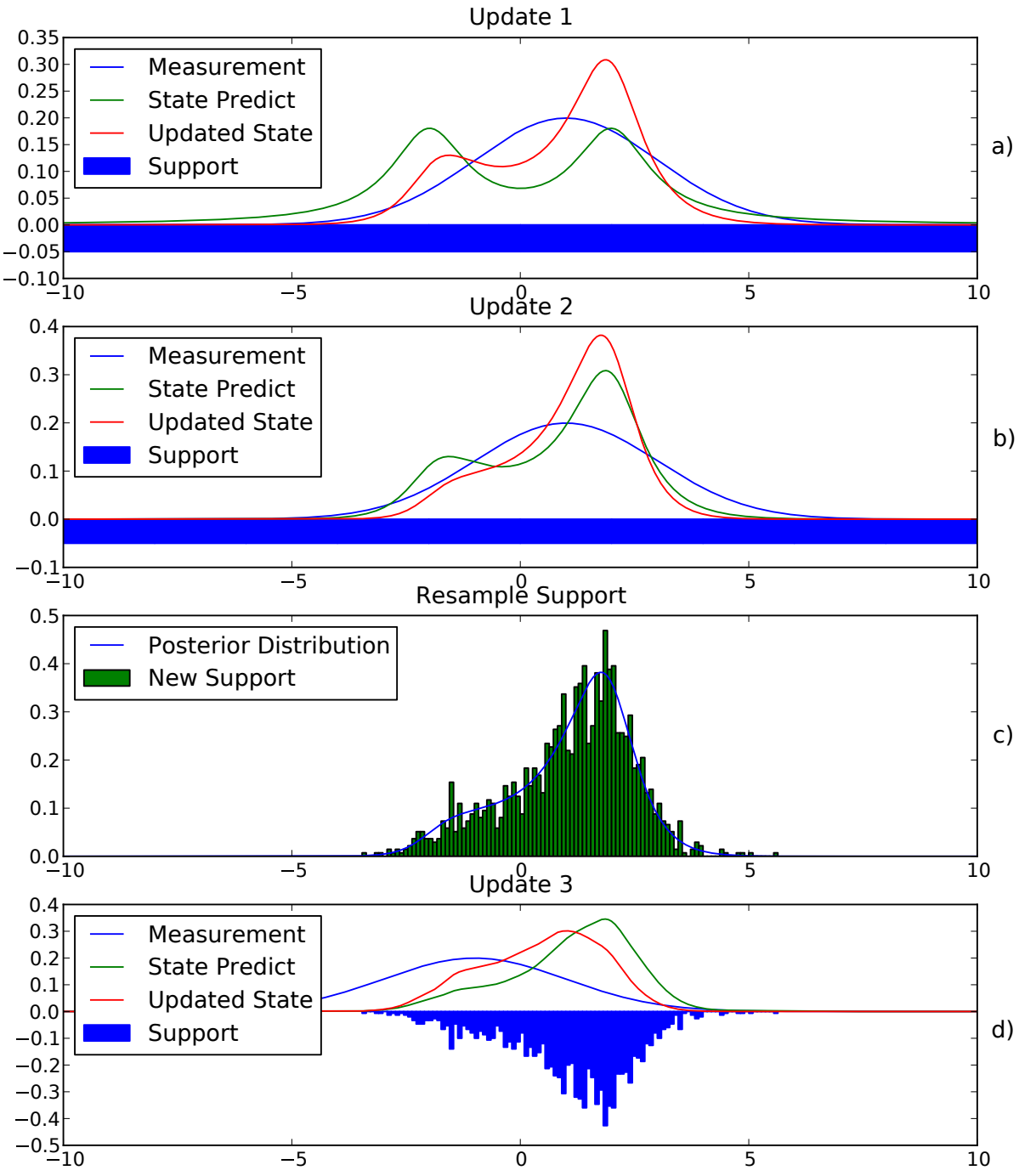


Figure 3.1: a) Initial particle locations uniformly distributed in [-10, 10], with weights set to give the State Prediction (green). When a measurement is received it converted to a distribution approximating uncertainty. This distribution is then used to alter the weights of the existing particles. b) Another measurement is applied similar to a), c) Resampling draws new particles in locations according to the Mixture PDF approximated by the particle weights. d) Another measurement is applied as before, but now with non-uniform particle locations.

Algorithm 3.1 Sequential Importance Sampling

Initialize Particles:
for i : each of N_p particles **do**
  $x_0^i \sim \alpha(X)$
  $w_0^i = \frac{1}{N_p}$
end for
for k : each measurement **do**
  for i : each particle **do**
    $x_k^i = x_{k-1}^i + \int_{t-1}^{t} f(x(\tau), u(\tau))d\tau$
    $w_k^i = w_{k-1}^i P(y_k|x_k)$
  end for
end for
$P(x(t_k + \Delta t)) \approx \sum_{i=0}^{N_p} w_k^i \delta \left(x - (x_k^i + \int_{t_k}^{t_k+\Delta t} f(x(\tau), u(\tau))d\tau) \right)$

regions. Ideally the computation time spent on tails would be proportional to the actual size of the tails. In this case particle locations would match the true posterior and all weights would be equal. The case where a large number of the weights have become extremely small is called particle degeneracy. In Lui et al. an ideal calculation of the effective number of particles is found based on the particles' true weight [22]. However, given that only an approximation for the true weight exists, they also provide a simple heuristic calculation of N_{eff}.

$$N_{eff} \approx \frac{\sum_{i=0}^{N_p} w_i}{\sum_{i=0}^{N_p} w_i^2} \tag{3.15}$$

Unless the prior is particularly accurate, N_{eff} drops very quickly. To alleviate particle degeneracy a common technique known as resampling is often applied. The idea of resampling is to draw from the approximate posterior, thus generating a replica of the posterior with a better support. Therefore, a new set of particles may be drawn from the empirical distribution as follows:

$$\hat{x}_j \sim \left(\sum_{i=0}^{N_p} w_k^i \delta(x - x_k^i) \right) \tag{3.16}$$

For infinite particles this new distribution will match the old. Unfortunately, this isn't the truth in practice: since the support is still limited to the original particles, the number of *unique* particles can only go down. This effect, 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 particles are drawn from that distribution. Because it is continuous, particle impoverishment cannot occur. The easiest way to sample from the continuous distribution is to break the re-sampling down into two steps. After calculating an estimate of the scale of the original distribution, algorithm 3.2 is performed. Next, a distribution is generated based on the variance of the original distributions. Finally, for each particle in the

Algorithm 3.2 Resampling Algorithm

Calculate total weight, $W_t = \sum_{i=0}^{N_p} w^i$
for all $0 < i < N_p$ **do**
  Draw V from uniform range $[0, W]$
  $C = W_t$
  for all $0 < j < N_p$ and $C < V$ **do**
    $C = C - w^j$
  end for
  Add $[x^j, \frac{1}{N_p}]$ to the new distribution
end for

discretely re-sampled distribution, a sample is drawn from the smoothing distribution and added to the particle. The regularization process is defined as:

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

Where h is an optional bandwidth, σ is the standard deviation such that $\sigma\sigma^T = cov(x)$ and ϵ is drawn from the chosen kernel. The choice of a kernel is complex, although it has been proven that the optimal kernel for reducing Mean Squared Error (MSE) between the original and resampled distributions is is the Epanechnikov Kernel [28]. However, Musso et al. also espoused the usefulness of the Gaussian Kernel, due to the ease drawing samples from it, which for this work was more important [28].

Algorithm 3.3 Regularized Resampling Algorithm

Calculate Covariance, C, of empirical distribution, $\hat{x}$
Find D such that $DD^T = C$
Resample $\hat{x}$ using algorithm 3.2
for $0 < i < N_p$ **do**
  Draw ϵ from the standard normal, same dimensionality as X
  $x^i = x^i + hD\epsilon$
end for

Hurzeler et al. demonstrated that if the underlying distribution is non-Gaussian, then using the original bandwidth may over-smooth [18]. In reality, over smoothing will only become an issue if resampling is performed often. Thus if resampling is performed at every step then this would certainly cause problems. If the distribution is over-smoothed then the algorithm may not converge as rapidly, or at all. However, because the bandwidth is still based on particle variance, which should decay as particles are ruled out, the particle filter is still able to converge. In fact, over-smoothing is preferable to under smoothing, since over-smoothing simply slows convergence while under-smoothing could leave gaps in the distribution. Moreover, because of the high dimensionality of

the BOLD model, and limited measurements, it is helpful to have a broader bandwidth to explore the distribution.

Algorithm 3.4 Regularized Particle Filter

Initialize Particles:
for i : each of N_p particles **do**
  $x_0^i \sim \alpha(X)$
  $w_0^i = \frac{1}{N_p}$
end for
for k : each measurement **do**
  for i : each particle **do**
    $x_k^i = x_{k-1}^i + \int_{t-1}^{t} f(x(\tau), u(\tau))d\tau$
    $w_k^i = w_{k-1}^i P(y_k|x_k)$
  end for
  Calculate N_{eff} with Equation 3.15
  if $N_{eff} < N_R$ (recommend $N_R = min(50, .1N_p)$) **then**
    Resample using algorithm 3.3
  end if
end for
At $t + \Delta t$, $t \in T$, $P(x(t + \Delta t)) \approx \sum_{i=1}^{N_p} w_i(t)\delta\left(x - (x_i(t) + \int_t^{t+\Delta t} f(x(\tau), u(\tau))d\tau)\right)$

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. Therefore in this work regularized resampling was only performed when N_{eff} dropped below 50. Other than the periodic regularized resampling, the regularized particle filter is identical to the basic sampling importance sampling filter (SIS).

With regularized resampling, it is possible to prevent both particle degeneracy as well as particle impoverishment. Note that resampling carries certain risks. If for some reason the solution is not covered by the new support, the algorithm may not be able to reach the true value. 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.

3.5 Weighting Function

Because the distribution of ν_y in Equation 3.3 is unknown, it is necessary to choose a distribution for this. This distribution is important because $\nu_y \sim P(y_k|x(T))$, which is used for updating

weights. Ideally this weighting function would exactly match the measurement error in the output. While a Gaussian function is the traditional choice, there are other reasonable distributions, given the nature of the noise present in FMRI. The choice of this function will be discussed further in Section 4.3.4.

Chapter 4

Methods

Although the particle filter is a standard Regularized Particle filter, as described by Arulampalam et al. [1], optimizing the particle filter for use with fMRI data is non-trivial.

4.1 Model

As originally written in Section 1.5 the state variables for the BOLD model are as follows:

$$\dot{s} = \epsilon u(t) - \frac{s}{\tau_s} - \frac{f-1}{\tau_f} \tag{4.1}$$

$$\dot{f} = s \tag{4.2}$$

$$\dot{v} = \frac{1}{\tau_0}(f - v^{\alpha}) \tag{4.3}$$

$$\dot{q} = \frac{1}{\tau_0}(\frac{f(1-(1-E_0)^f)}{E_0} - \frac{q}{v^{1-1/\alpha}}) \tag{4.4}$$

The original assumption regarding particle filter models (Section 3.2) included noise in the update of x, however that is not included here. The reason for the difference is that the cloud of particles is, to some extent, able to account for that noise. It is common, however, to model that noise in particle filters by adding a random value to each updated state variable. Because the purpose of this particle filter is to learn the underlying distribution of the static parameters, rather than precisely modeling the time course of the in the dynamic state variables $\{s, f, v, q\}$ this noise is left out. It also helps that detrending is applied before the particle filter and that the BOLD model is dissipative. When no stimuli are applied, all the particles decay to $\{0, 1, 1, 1\}$. Typical particle filters also use this state noise as an exploratory measure; however, this method is less necessary when good priors are available, and when regularized particle filtering is being used.

For all the analysis in this work, 1400 integration points per second were used. Typically a step

size of 0.001 was sufficient, however, from time to time 0.001 can still result in problems for the BOLD model.

4.2 Preprocessing

The normal pipeline for analyzing fMRI involves a several preprocessing steps to condition the signal. The first and most important task is motion correction. To do this, a single volume in time is chosen, and volumes at every other time point are realigned to best match the target volume. This corrects for motion by the patient as well as small changes in the magnetic fields that cause the image to shift. In conventional statistical parametric mapping, a Gaussian smoothing filter is applied across the image as discussed in Section 2.1.1. After this, detrending is performed, which is discussed in Section 4.2.2. Recall that fMRI signal levels are unit-less and though detrending is not always necessary, the data must always be converted into % difference from baseline. The generally accepted method is to use a high pass filter, although the cutoff frequency is application dependent and often applied haphazardly. Before going into the detrending used in this work, it is necessary to discuss the type of noise present in fMRI.

4.2.1 BOLD Noise

As demonstrated in Section 1.5 the BOLD response has been extensively studied and despite minor discrepancies, the cause of the BOLD signal is well known. However, as fMRI detects an aggregate signal over the space of cubic centimeters, there are plenty of noise sources. 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 and structured noise.

A particularly difficult form of noise present in fMRI is a low frequency drift, often characterized as a Wiener process [33]. Though not present in all regions, as many as ten to fifteen percent of voxels can be affected [38], thus it is prevalent enough to cause significant inference problems [36]. It is still not clear what exactly causes this noise, although one possibility is the temperature difference in scanner magnetic coils [36]. It is clear that this drift signal is not solely due to physiological effects, given its presence in cadavers and phantoms [37]. Interestingly, it is usually spatially correlated, and more prevalent at interfaces between regions. Though one potential source could be slight movement, co-registration is standard, making this unlikely. Regardless, the problem mandates the use of a high pass filter [36].

In order to characterize the noise, resting state data was analyzed. During resting state, the patient is shown no images, and he is asked to avoid movement and complex thought. Overall there should be very little activation, and thus the signal consists entirely of noise. Therefore resting state data is perfect for analyzing noise. The locations were chosen from points all around the brain, all in

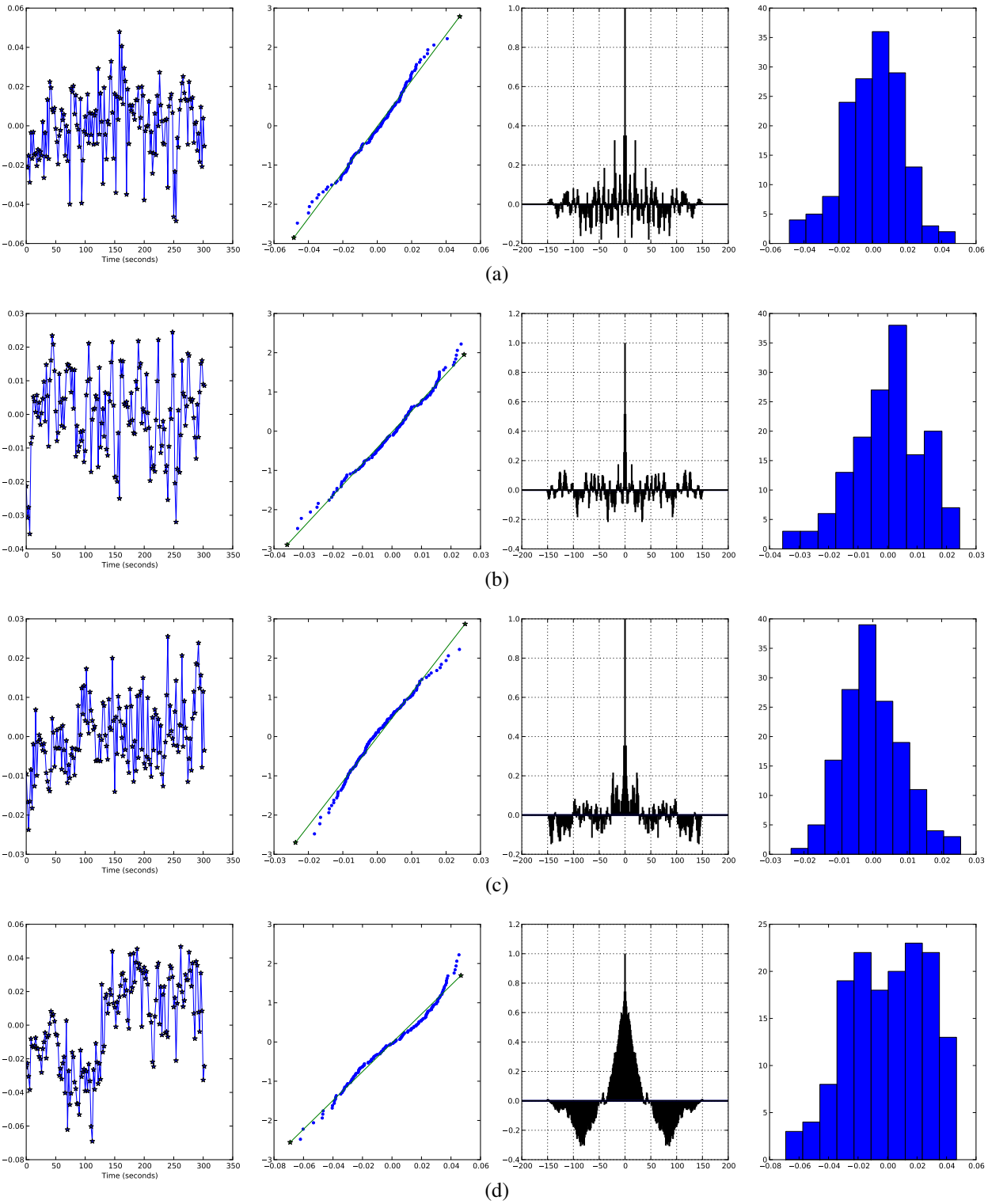


Figure 4.1: Resting state noise analysis. From Left to Right: (resting state) signal, Q-Q Plot of measurements with the standard Normal, Signal auto-regression, Histogram of the Points

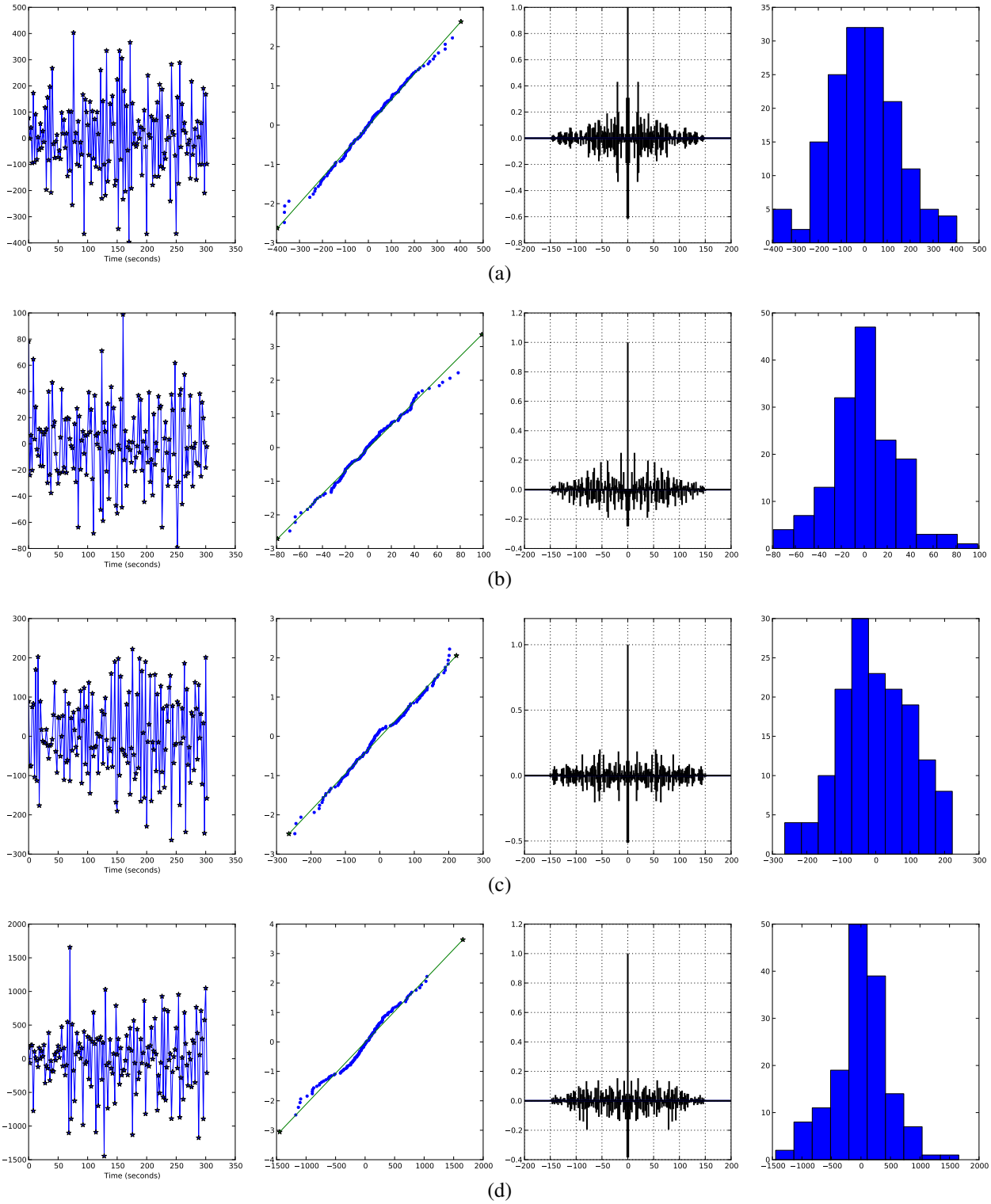


Figure 4.2: Resting state noise analysis, where signal levels are the difference between adjacent signals in Figure 4.1. From Left to Right: (resting state) signal, Q-Q Plot of measurements with the standard Normal, Signal auto-regression, Histogram of the Points

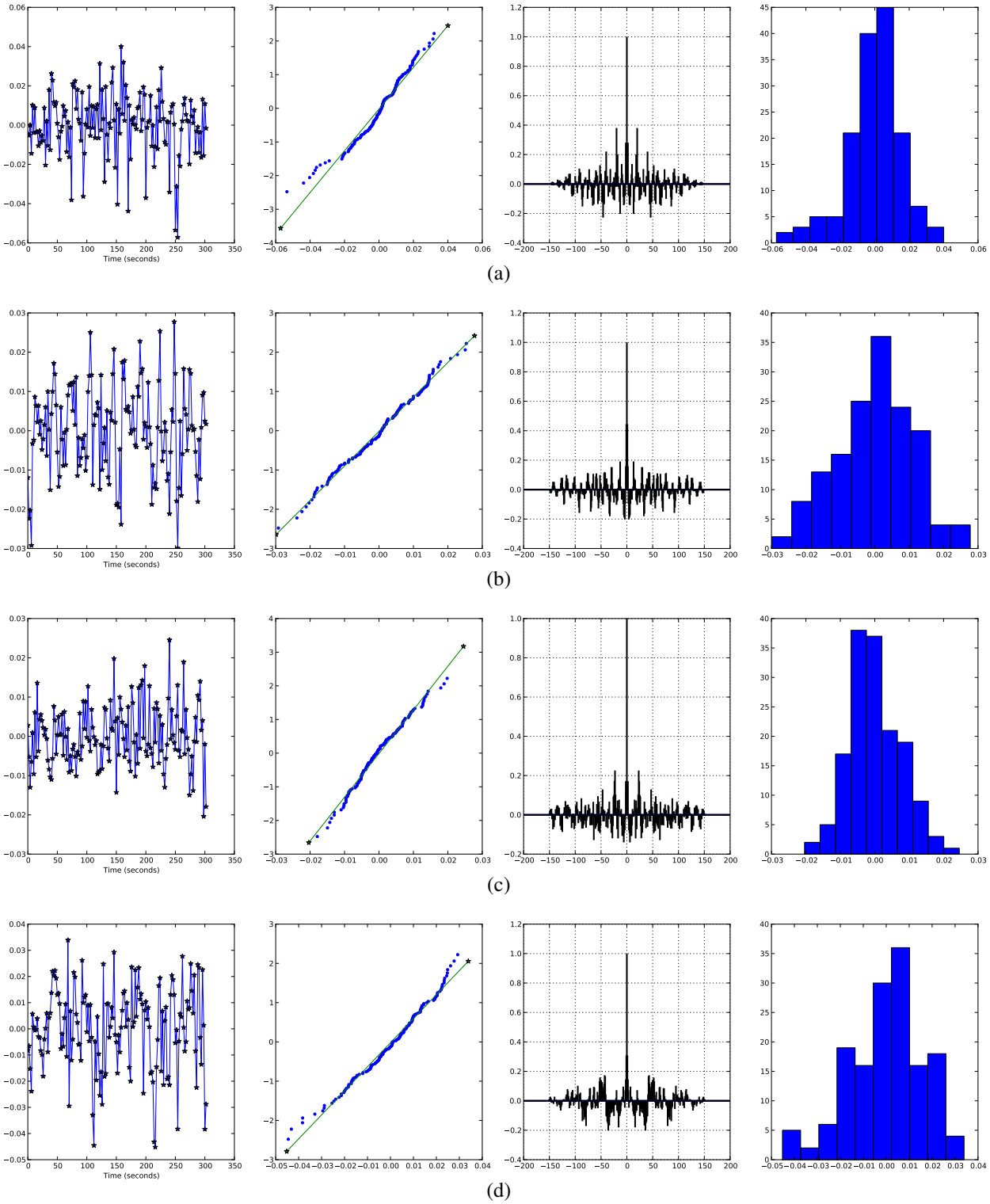


Figure 4.3: Resting State Noise Analysis, where signal levels are the result of subtracting a spline from the signal in Figure 4.1. From Left to Right: (resting state) signal, Q-Q Plot of measurements with the standard Normal, Signal auto-regression, Histogram of the Points

grey matter voxels. These time series were chosen because they were representative of different types of noise found in the resting state data.

The resting state was gathered in the exact same way as the data in Section 6.1, except without the stimuli.

Because most methods (including the one used in this paper) assume the noise realizations are independent of each other, the autocorrelation is of particular interest (which is a necessary but not sufficient condition for independence). Gaussianity is also a common assumption made in studies of fMRI data, though that assumption is not needed in this work. Regardless, comparing the distribution to a Gaussian is informative, so Q-Q plots are used to compare example data with the Normal distribution. Additionally, in fMRI data, the noise is often considered to be Wiener [34]. Recall that a Wiener random process is characterized by steps that are independent and Gaussian. The simulations discussed in Section 5.1 make use of this, by adding a Wiener random process to the overall signal. To determine whether the noise is in fact Wiener, the distribution of the steps were also plotted against a Gaussian.

Finally, removal of the drift is often performed with a high pass filter, so I also analyzed the distribution after subtracting a spline, (see Section 4.2).

Figure 4.1 shows the results with a regression line fit to the points. Recall that in a Quartile-Quartile (Q-Q) plot, if the points plotted on the x and y axes come from the same type of distribution, then all the points will be collinear. 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. Note that the points have all been normalized (changed to percent difference).

4.1(a) and 4.1(b) are well described by a Gaussian process with a small autocorrelation, but 4.1(c) and 4.1(d) are not. In particular the tails of 4.1(c) do not seem to fit the Gaussian well. Also note the significant autocorrelation in 4.1(c) and 4.1(d). As expected, the noise is not strictly Gaussian white noise. On the other hand, the steps do conform rather closely to the normal distribution. As expected, most of the autocorrelation disappears for the step data (Figure 4.2). Given that the steps seem to fit the Normal distribution, the low autocorrelation indicates that the steps could be Independently Distributed. Therefore, the noise is consistent with a Wiener process.

Detrending the time-series by subtracting a spline fit to the distribution (Figure 4.3) removed much of the autocorrelation present in 4.1(c) and 4.1(d), though not perfectly. Though the distributions still do not exactly fit the Normal, 4.3(d) is much improved compared to 4.1(d). In all, the detrending is effectively removing Wiener noise.

4.2.2 Detrending

The non-stationary aspect of a Wiener process, presumably the result of integrating some ν_x, is difficult to compensate for, and so many methods have been developed to compensate for it. Tanabe

et al. [38] and Smith et al. [37] demonstrated that this component is prevalent, and may in fact be an inherent characteristic of fMRI. It has been reported that in some studies as many as half the voxels benefited from detrending [36]. In comparison of high pass filters, Tanabe et al. [38] showed that in most cases subtracting off a spline worked the best. Unfortunately no method will perfectly remove noise, and no method will leave the signal untouched.

The method used to calculate the spline was one knot for every 20 measurements in an image. Thus a 10 minute session at a repetition time of 2.1 seconds would have 19 knots. The first and last knots were each given half the number of samples as the rest of the knots; which were all located at the center of their sample group. The median of each sample group was then taken and used as the magnitude for the group. Taking the median versus the mean seemed to work better, given the presence of outliers. There is potential to optimize the spline further using a canonical HRF to find resting points; however, the experiment would have to be designed with this in mind.

Problematically, after removing the DC component of the signal, by definition the signal will have a median near zero. Unfortunately this is not the natural state of the 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. Because most of the BOLD signal is above baseline, after removing the spline the resting state will be below 0%. One method of accounting for this is to simply add a DC gain model parameter. Like all the other model parameters, with enough measurements, the particle filter would be able to settle on a good estimate. Yet adding another dimension increases the complexity of the model, for a parameter that is relatively easy to estimate by visual inspection. In this work a simpler approach was used. To determine the DC gain a robust estimator of scale was used. The Median Absolute Deviation (MAD) proved to be accurate in determining how much to shift the signal up by. Both methods analysis were tested, and it was found that the increase in model complexity far outweighed the slight increase in flexibility. Other methods may work better, however the MAD worked well, as Figure 5.4 and Figure 5.7 show. The DC gain was then set to:

$$y_{\text{gain},0:K} = 1.4826 \underset{i=0:K}{\text{median}}(y_i - \text{median}(y_{0:K})) \tag{4.5}$$

For more information on the MAD, it is discussed in great detail in Iglewicz's text on robust estimation [19]. A serious concern when adding constant values to real data is whether this will create false positives. This is a legitimate concern; however, a boosted response does not affect how well the BOLD model predicts the actual measurements; and as mentioned before, the DC signal of fMRI is never used.

4.3 Particle Filter Settings

There are quite a few options when using a particle filter; therefore this section will discuss the design choices made in this work.

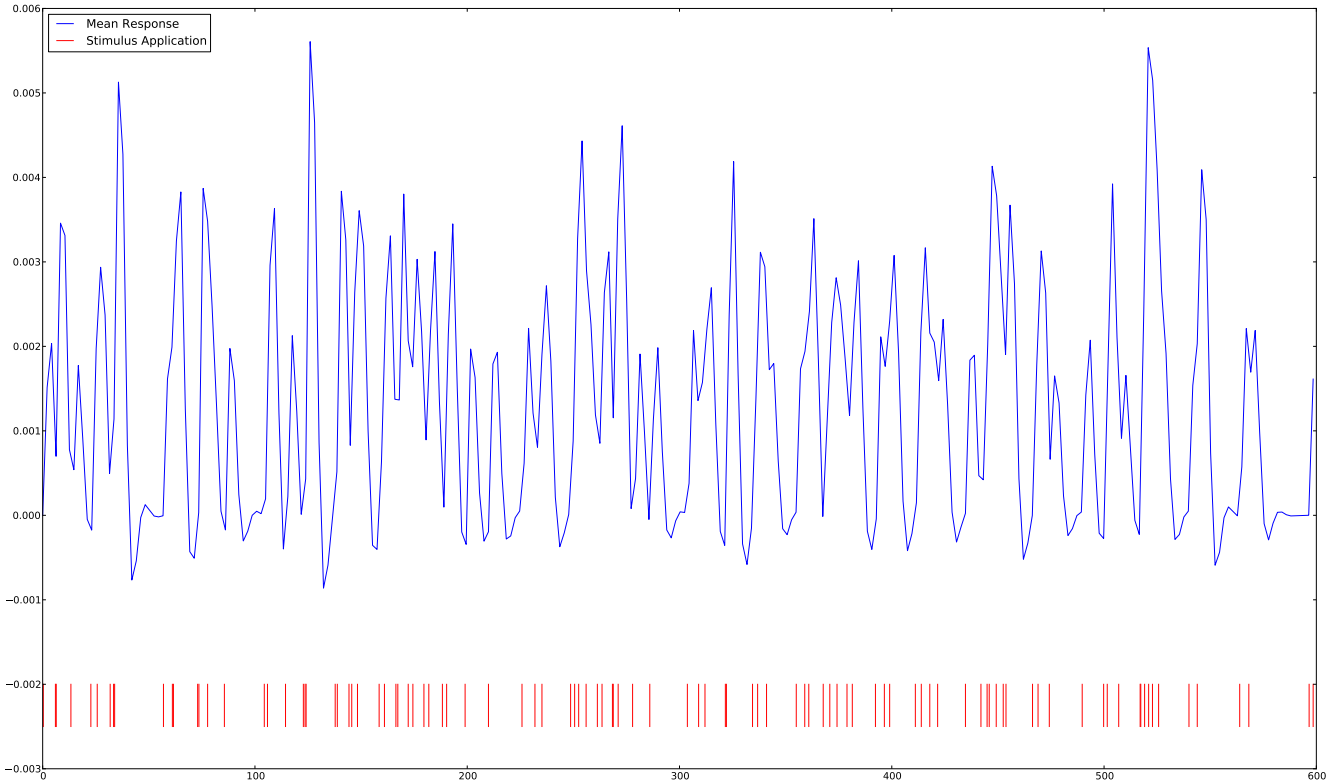


Figure 4.4: BOLD Response to 0.1 s impulses with the mean parameters from Friston et al. [12]

4.3.1 Prior Distribution

For the BOLD model described in Section 1.5, several different studies have endeavored to calculate parameters. The results of these studies may be found in Table 1.1, and the methods used for each may be found in Chapter 2. Unfortunately, Friston et al. [12] only studied regions deemed active by the General Linear Model; and most other research used these results as the source for their priors. The one exception is to this is Johnston et al. [20] which came to extremely different distribution. For a particle filter, the choice of a prior is the single most important design decision. A very wide prior will require more particles to be sufficiently dense, a very thin (low variance) prior may miss the true parameters. Consequently, for this work it was natural to use priors that would give results consistent with previous works [12]. This constrains the usefulness of the model to areas that fall within the prior distribution, yet allowed results to be comparable to other works. There is a significant need for better estimates of the physiological parameters; and, while physical experiments may not be possible, it would not be unreasonable to do a study with exhaustive simulated annealing or hill climbing tests for multiple regions and multiple patients. That said, the parameters may not be strictly identifiable, which is discussed in Section 5.1

The differences between the parameter estimates of Johnston et al. [20] and Friston et al. [12] are clearly visible in Figure 4.5 and Figure 4.4. Therefore, to account for these discrepancies, tests on real data was used to adjust the distributions from those arrived at in Friston et al. [12]. In tests

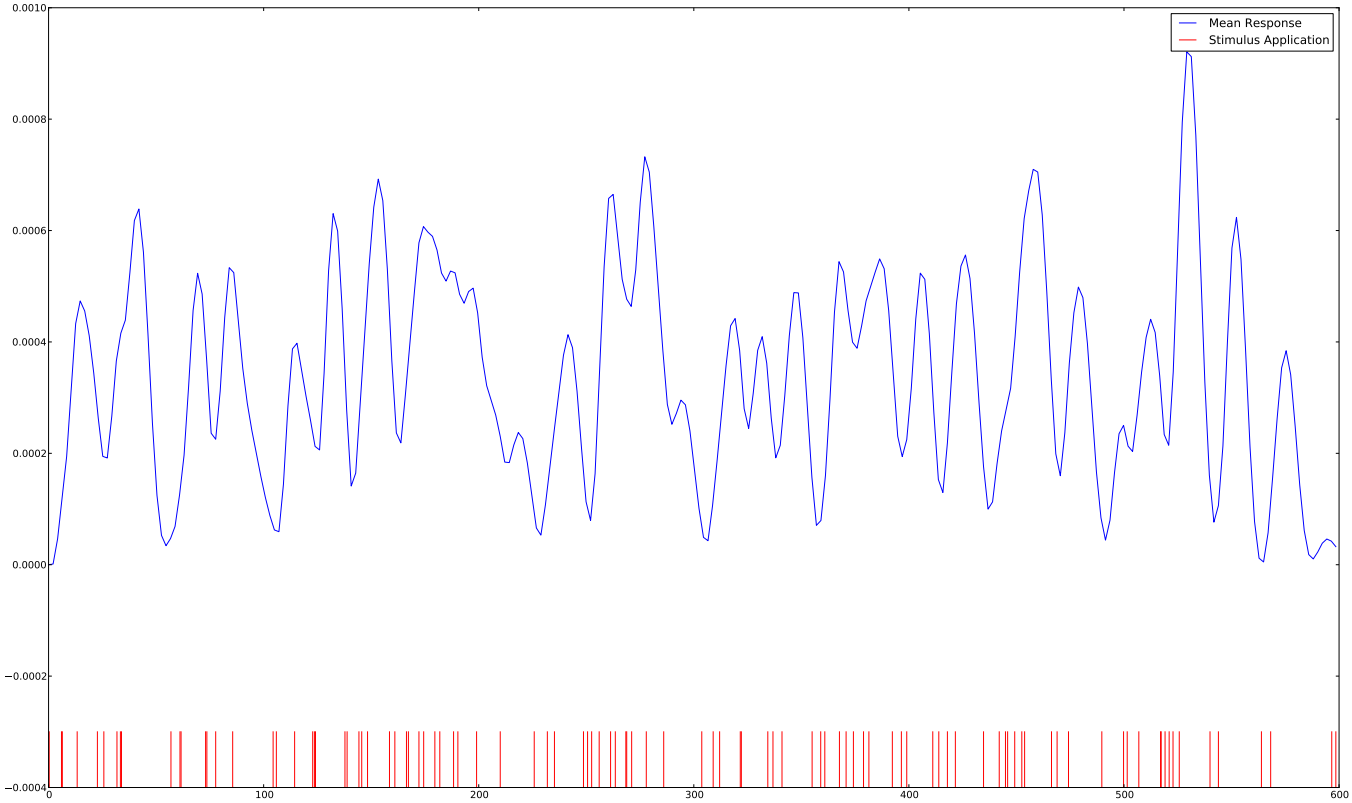


Figure 4.5: BOLD Response to 0.1 s impulses with the mean parameters from Johnston et al. [20]

on real data, the Johnston et al. [20] distributions never converged to meaningful BOLD estimates. The priors used in the particle filter may be found in Table 4.1.

Note that although the mean remains the same for all the parameters other than ϵ, the standard deviations were adjusted based on tests with real data. Also, the state variables were all assumed to be at rest at the beginning of the simulation.

It is also important to use enough particles to obtain a sufficiently dense approximation of the prior. For 7 dimensions, getting a dense prior is difficult. Insufficiently dense particles will result in inconsistent results. Of course the processing time will scale up directly with the number of particles. A dense initial estimate is important to ensure that some particles land near the solution; but as the variance decreases the number of particles needed decreases as well. Thus, as a heuristic, initially the number of particles was set to 16,000, but after resampling, the number of particles was dropped to 1,000. Typically during the first few measurements the variance drops precipitously because most particles are far from a solution. The particles that are left are in a much more compact location, allowing them to be estimated with significantly fewer particles. These numbers aren't set in stone, and depending on the complexity of the system or desired accuracy they could be changed; however, they seem to be the minimum that will give consistent results.

Parameter	Distribution	μ	σ
τ_0	Gamma	.98	.25
α	Gamma	.33	.045
E_0	Gamma	.34	.03
V_0	Gamma	.04	.03
τ_s	Gamma	1.54	.25
τ_f	Gamma	2.46	.25
ϵ	Gamma	.7	.6

Table 4.1: Prior distributions used in the particle filter.

4.3.2 Resampling

The algorithm for resampling is described in Section 3.4. When regularizing, the Gaussian kernel is convenient, because it is simple to sample from and long tailed. As discussed in Section 3.4, as long as resampling is kept as a last resort, some over-smoothing doesn't impair convergence. Therefore, for this work a Gaussian kernel of bandwidth equal to the original distribution's covariance was used. This applies a large amount of smoothing to the distribution; however, on average resampling was only applied every 20 to 30 measurements. At this rate, regularized resampling still gives the filter some ability to explore state space, without excessively revering convergence.

Resampling is not strictly necessary, but it increases the effectiveness of the particle filter by adjusting the support to emphasize areas of higher probability. Resampling is slow because it requires redrawing all the particles. It also closes off avenues of investigation, but is designed to over-smooth to prevent overly thinning the support. For all these reasons, resampling was only performed when the N_{eff} dropped below 50 (for 1000 particles). As a measure against sharp drops in the N_{eff} caused by a large spike in error, resampling was only performed when two consecutive low (< 50) N_{eff}'s were found.

4.3.3 Particle Deprivation

To prevent particle deprivation (all weights dropping to 0), which may happen even with a good estimate of the prior distribution, a method of rescuing the particle filter from this state was implemented. When particle deprivation was detected, resampling was performed with a saved version of the covariance matrix (from the previous time step). This allowed for the particles to be rescattered without having to go all the way back to the prior distribution. Particle deprivation was defined by N_{eff} dropping to 1.

4.3.4 Choosing $P(y_k|x_k)$

The choice of $P(y_k|x_k)$ is the second most important design decision, behind the prior. While the conventional choice for an unknown distribution is the Gaussian, there are reasons why it may not be the best in this case. As noted in Section 4.2.1, the noise is not strictly Gaussian, nor is it strictly Wiener. As with any unknown noise however, it is necessary to make some assumption. If the weighting function ($P(y_k|x_k)$) exactly matches the measurement error, then the ideal particle filter will result. Particles with x_k's that repeatedly estimate y_k with large residual will quickly have weights near 0. Thus, a weighting function that exactly matches $P(Y(t)|X(t))$ will easily and correctly throw out invalid particles. The cost of choosing an overly broad distribution for this function is slow convergence. On the other hand, an overly thin distribution will lead to particle deprivation (all particles being zero-weighted). Three weighting functions were tested. In addition to the Gaussian I also tested the Laplace and Cauchy distributions, both of which have heavy tales. Wider tailed distributions don't down-weight particles as fast; and thus converge more slowly. The Laplace distribution also has the benefit of a non-zero slope at the origin; meaning it differentiates between particles even near the origin.

After trial and error, for this thesis the Gaussian with a standard deviation of 0.005 was used. Although using an adaptive weighting function might be better, in tests this often led to unpredictable results. With some work it may be possible to set the standard deviation by taking a small sample from resting data and using the sample standard deviation. In this work, however, the standard deviation was manually set at 0.005, because it gave the best consistency.

4.3.5 Runtime

The run-time for a single voxel depended on several factors. First, the overall length of the signal being analyzed directly scaled the run-time. For 1000 measurements it took approximately 6 minutes. On the other hand, in real fMRI the length tends to be around 150 measurements which took around 40 seconds (for 1000 particles, 1400 integration points and a Quad Core CPU). The number of integration steps was also crucial. Using step sizes above 0.001 seconds is not recommended. In most cases millisecond resolution was fine; however, when generating simulated data at times this was still not enough. This could cause problems in the actual particle filter since, given the large number of simultaneous integrations taking place, its probable that a few particles would fail and be unfairly thrown away. To prevent such events, 1400 integration points per second were used throughout the tests.

Because the prior is initially represented with significantly more particles, if for some reason the effective number of particles stays high, resampling could take a long time to occur. For this reason, rather than allowing the particle filter to continue with this large number of particles, after 20 seconds have passed the algorithm forces resampling. The choice of 20 seconds is arbitrary, but at the very least it gives the particles time to be weighted.

Chapter 5

Simulation

Two levels of simulation are discussed in this section; first of single voxels, and second of a single slice (64x64 voxels). The single time series tests investigate the ability of particle filters to estimate parameters of the BOLD model in a noisy environment. Single voxel tests were performed eleven different times, each with a new noise realization. Repeating tests with different noise realizations demonstrates the particle filter's resilience to noise and explores the variance of the model.

The slice simulation was performed using Physics-Oriented Simulated Scanner for Understanding MRI (POSSUM) which models noise realistically. POSSUM demonstrates the particle filter modeling the BOLD signal on a large scale. In the POSSUM simulation there were four discrete parameter sets driving the time series, although each voxel had a different tissue composition and a different noise realization. Therefore the POSSUM simulation was a good test of the applicability of particle filters for performing whole brain analysis.

Note that except for Section 5.1.1 the same stimulus sequence that was used in real data (Chapter 6) was used in the simulations. This sequence may be found in Section 6.1, but because of the availability of ground truth in simulations is not displayed in this section.

5.1 Single Time Series

Given the state-space equations for the BOLD signal, simulating a single time series was straightforward. After generating a true signal, identically and independently distributed (I.I.D.) Gaussian noise and a Wiener process with Gaussian I.I.D. steps were added to the true signal. Finally a carrier level was added, since BOLD is typically measured as a % difference from the baseline. The particle filter algorithm immediately removes this by calculating the % difference, but adding a carrier level meant that the exact same algorithm used for simulated data could be used for the real data.

Once this noisy simulated time series was generated, the particle filter algorithm was run on the

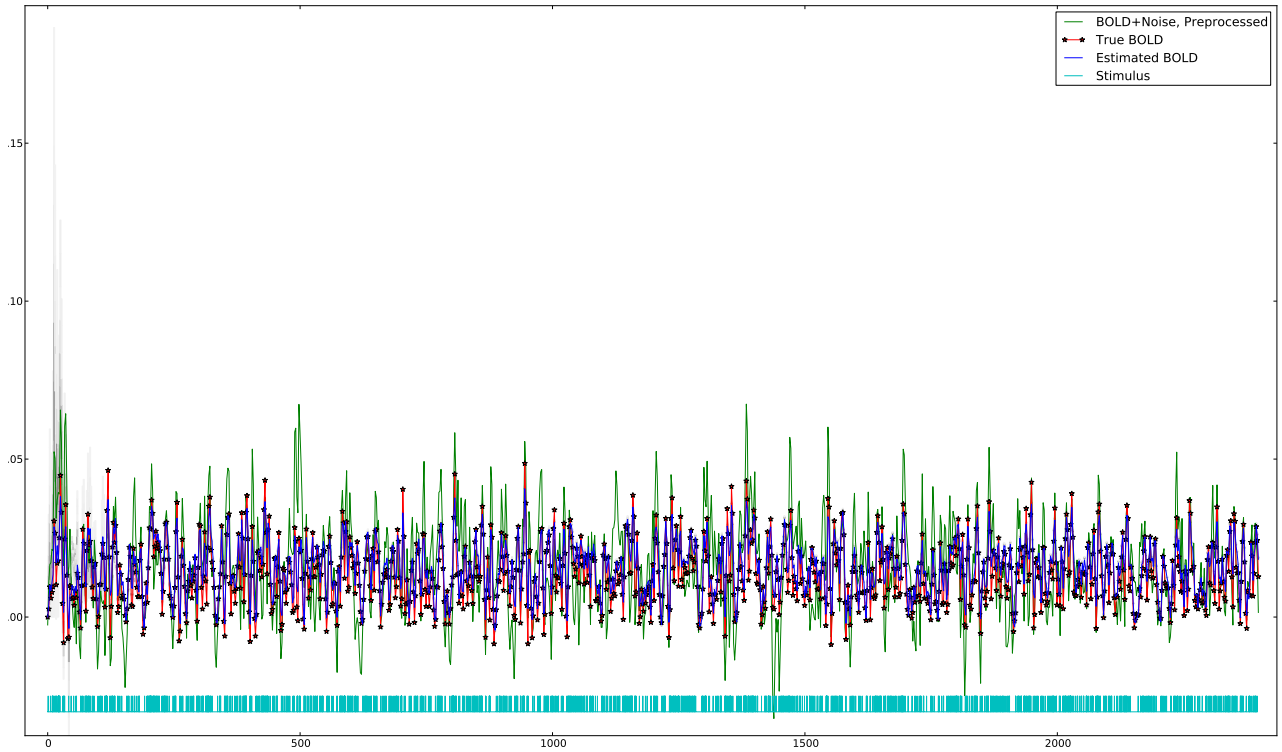


Figure 5.1: BOLD estimate converging for a 2400 second fMRI run. Darker bars indicate bins with more particles with BOLD estimate in that range. ($\sigma_x = 0.0005$, $\sigma_y = 0.001$)

single time series image. Five sets of simulation tests were performed on the particle filter. The first test demonstrates the particle filter on a very long fMRI run, and discusses the inherent learn-ability of the BOLD model (Section 5.1.1). The next two (Section 5.1.2 and Section 5.1.3) demonstrate the ability of the particle filter to find the most likely set of parameters/state variables over the course of a run identical to the real run in Chapter 6. The last two tests (Section 5.1.4 and Section 5.1.5) investigate the problem of false-positives. As the first round of tests show, given the presence of an underlying BOLD signal, the particle filter is excellent at finding the most probable cause of the observed signal. As discussed in Section 5.1.4 and Section 5.1.5, even when there is no underlying signal, the particle filter may still converge. Because of this problem, it was necessary to investigate methods of identifying false positives.

5.1.1 Ideal Analysis

To begin the single voxel simulation; a signal using the following parameters was generated: $\{\tau_0 = 1.45, \alpha = 0.3, E_0 = 0.47, V_0 = 0.044, \tau_s = 1.94, \tau_f = 1.99, \epsilon = 1.8\}$. These same parameters were used throughout this chapter. Noise was generated based on measurement noise (σ_y) of 0.001 and drift standard deviation (σ_x) of 0.0005. The measurement noise as well as the steps of the

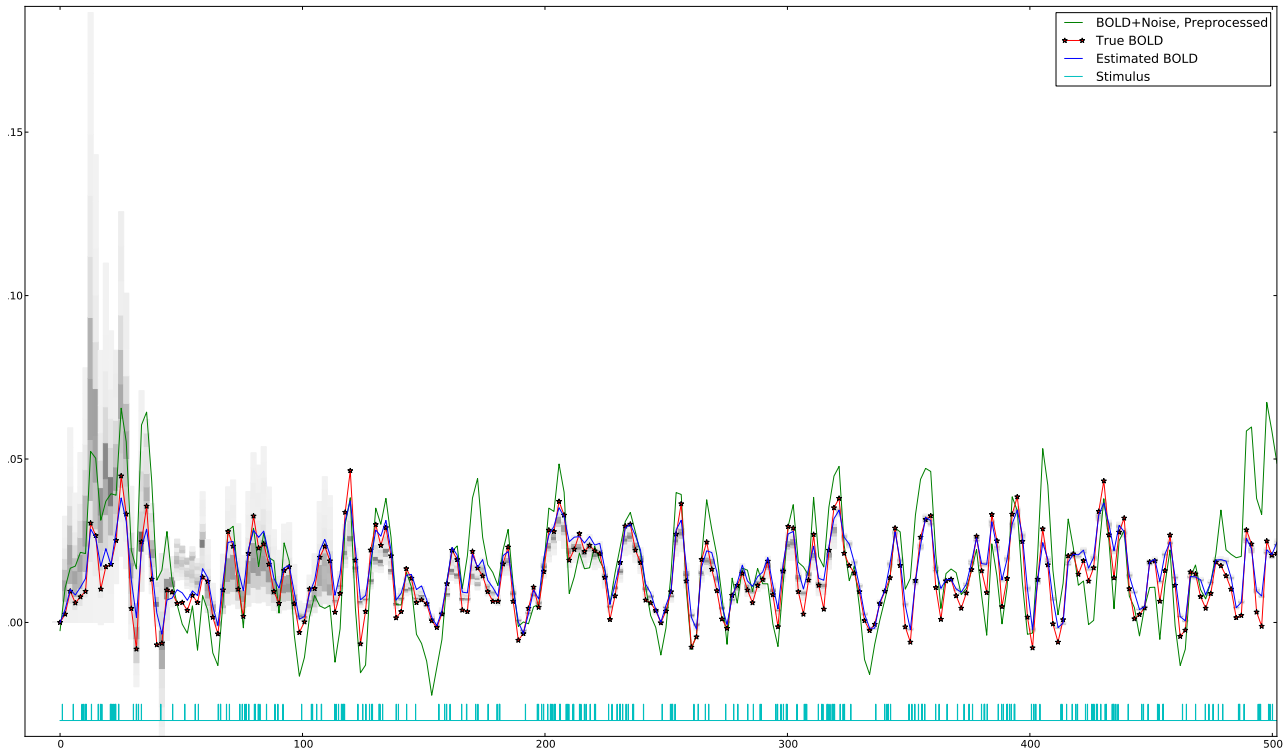


Figure 5.2: First 500 seconds of BOLD estimate converging for a 2400 second fMRI run. Darker bars indicate bins with more particles with BOLD estimate in that range. ($\sigma_x = 0.0005$, $\sigma_y = 0.001$)

	τ_0	α	E_0	V_0	τ_s	τ_f	ϵ
τ_0	0.0004334	5.2e-05	-6.95e-05	3.3e-06	0.0001628	-2e-07	0.0001798
α	5.2e-05	7.9e-06	-6.4e-06	3e-07	1.04e-05	-1.92e-05	2.58e-05
E_0	-6.95e-05	-6.4e-06	1.9e-05	-9e-07	-4.11e-05	-3.24e-05	-3.92e-05
V_0	3.3e-06	3e-07	-9e-07	1e-07	1.1e-06	9e-07	1e-06
τ_s	0.0001628	1.04e-05	-4.11e-05	1.1e-06	0.0001589	0.0001518	7.88e-05
τ_f	-2e-07	-1.92e-05	-3.24e-05	9e-07	0.0001518	0.0002966	-2.34e-05
ϵ	0.0001798	2.58e-05	-3.92e-05	1e-06	7.88e-05	-2.34e-05	0.0001966

Table 5.1: Covariance matrix of the parameters at the end of Figure 5.1.

	τ_0	α	E_0	V_0	τ_s	τ_f	ϵ
τ_0							
α	0.889884						
E_0	-0.7661395	-0.5230723					
V_0	0.6244049	0.4239271	-0.7964774				
τ_s	0.6204843	0.295425	-0.7481253	0.3440421			
τ_f	-0.0004259	-0.3966881	-0.4314174	0.1962954	0.6990775		
ϵ	0.6158116	0.6558179	-0.641348	0.2846632	0.4458142	-0.097079	

Table 5.2: Correlation of parameter estimates at the end of Figure 5.1.

drift were taken to be Gaussian. The actual signal delivered into the particle filter was the result of preprocessing to remove drift, as described in Section 4.2.

To test how well the particle filter would do with plenty of data, and determine the inherent variance in the model parameters, a very long simulation with randomly generated impulse stimuli was created. The preprocessed time series and the final estimate are shown in Figure 5.1; the final covariance matrix of the parameters is in Table 5.1. Note that even though the BOLD response converged (Figure 5.1, Figure 5.2), the parameters still have significant correlation (Table 5.2). Based on the histograms in the first 500 seconds, the parameters converged to their final values well before the end. Although this is only a single test, the correlation (Table 5.2) of the parameters increases the probability that the parameters are ill-defined. When the input consists entirely of impulses, the best parameters are not one particular set, but a joint distribution. Note that the correlation is in parameters whose priors are completely independent. It is possible that varying the type of input could give improved results, although informal tests did not show significant difference. In spite of the noisy input (green line in Figure 5.1), the estimates of the BOLD were actually very close to the true (noise-free) BOLD signal.

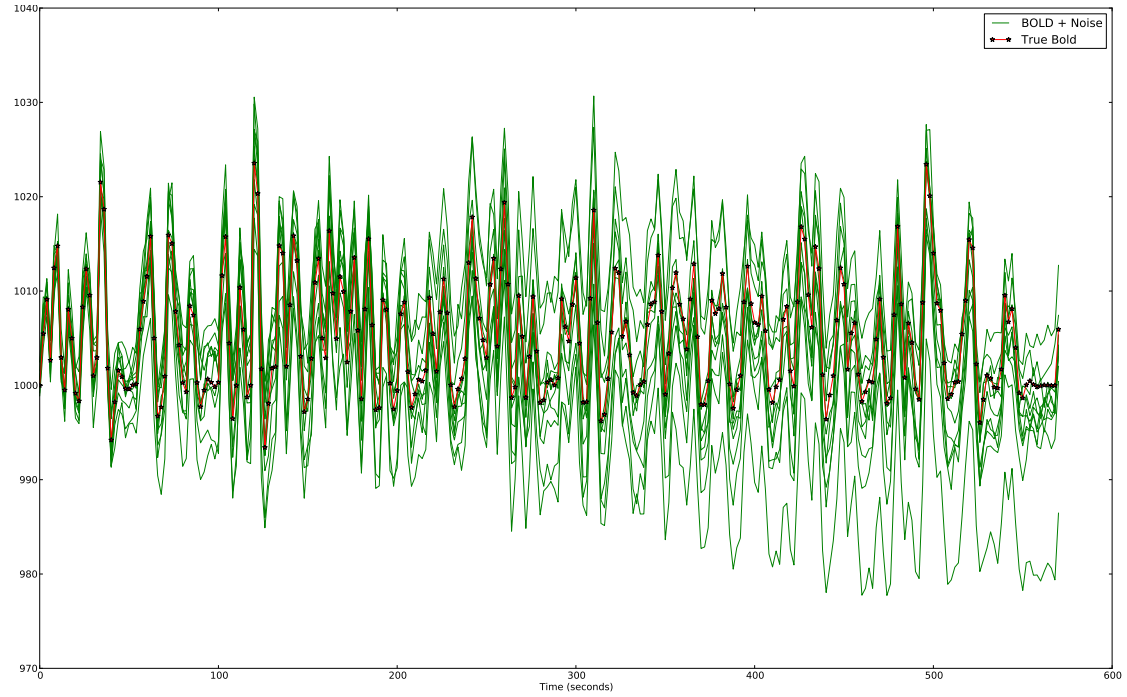


Figure 5.3: Simulated signals with different noise realizations compared to true signal. ($\sigma_x = 0.0005$, $\sigma_y = 0.001$)

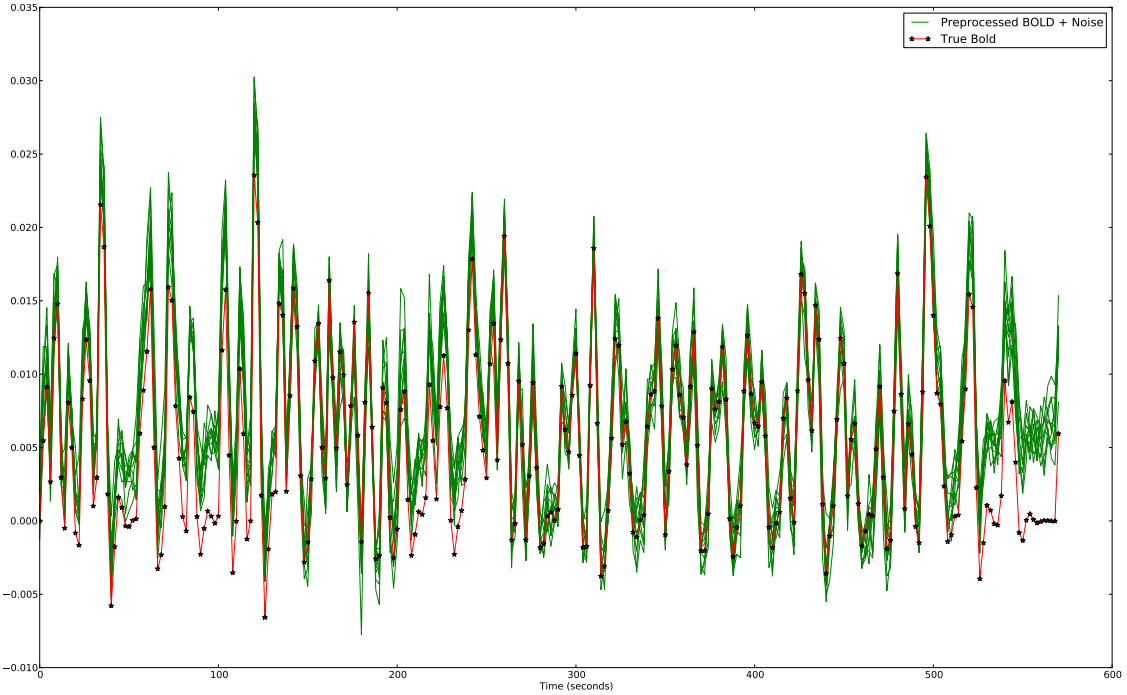


Figure 5.4: Results of preprocessing compared to the true signal. Knots of spline placed every 20 measurements. ($\sigma_x = 0.0005$, $\sigma_y = 0.001$)

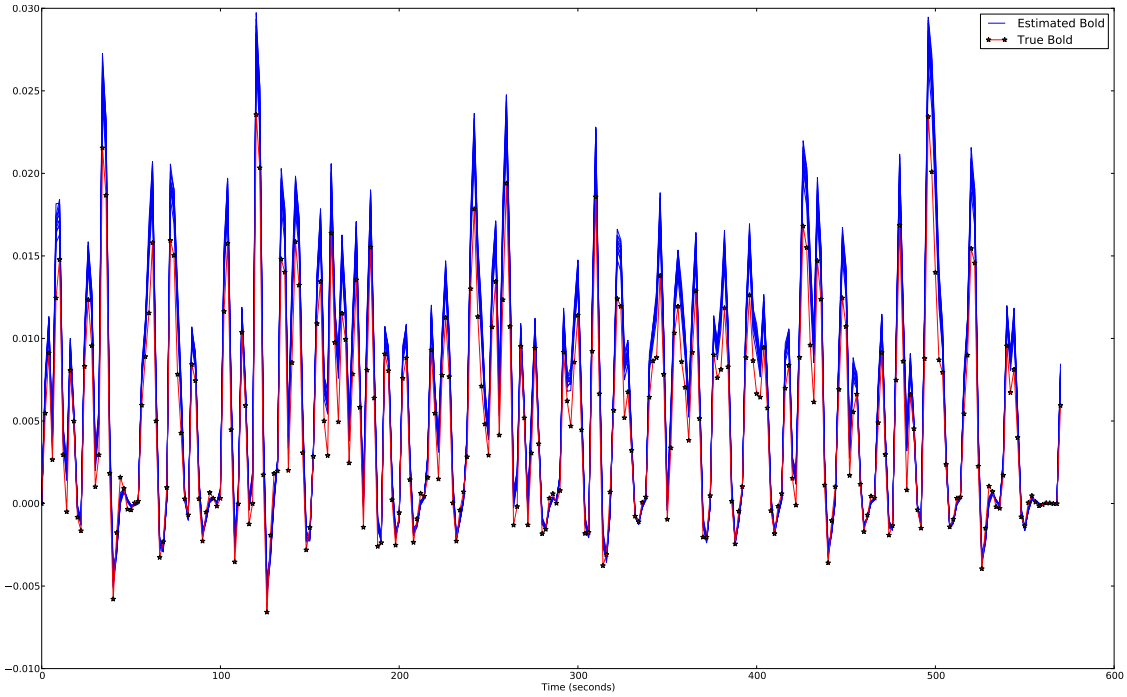


Figure 5.5: Results of the particles filter with preprocessed signals from Figure 5.4 as input.

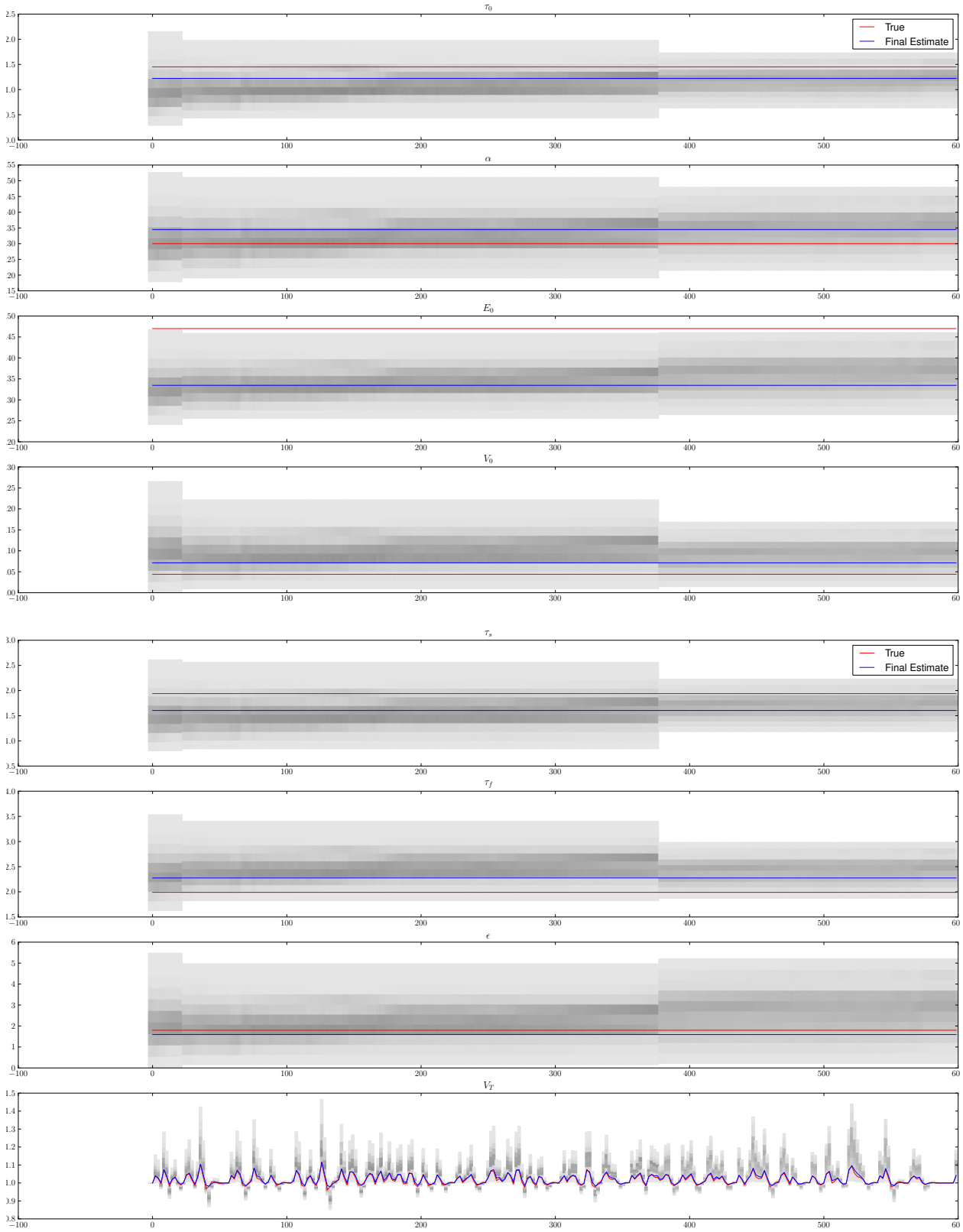
5.1.2 Simulation with Low Noise

The following tests all use less data (shorter sequences), on par with the average length of an fMRI session. For this low noise case ($\sigma_y = 0.001,\ \sigma_x = .0005$), the eleven realizations are shown in Figure 5.3. The bias introduced into the signal by preprocessing could have some effect on the resulting fit; thus the preprocessed signal is compared to the true BOLD signal in Figure 5.4. Overall, Figure 5.4 shows that the preprocessing was effective at removing trends, although the spline did leave some noise structure toward the end (fig:LowNoiseRealization). This slight drift effect was caused by the final median being well above the base level. At several time points the particle filter successfully filtered the input (Figure 5.5). For instance, in the last 30 seconds the estimates stay flat in spite of the preprocessed data drifting off. By this point, the algorithm had converged sufficiently to prevent such inexplicable movement. A similar circumstance occurs at around 100 seconds in. A combination of noise and preprocessing biased the results toward a peak signal above the true peaks. The final parameter sets are shown in Table 5.3.

For the purpose of quantifying the quality of fit, the Root Mean Squared Residual (RMSR) was used:

$$\text{RMSR} = \sqrt{\frac{1}{N}\sum(\hat{y}_k - y_k)^2} \tag{5.1}$$

where $\hat{y}_k$ is the estimated output at time k and y_k is the preprocessed output sampled at time k.


τ_0
True
Final Estimate
α
E_0
V_0
τ_s
True
Final Estimate
τ_f
ϵ
V_T

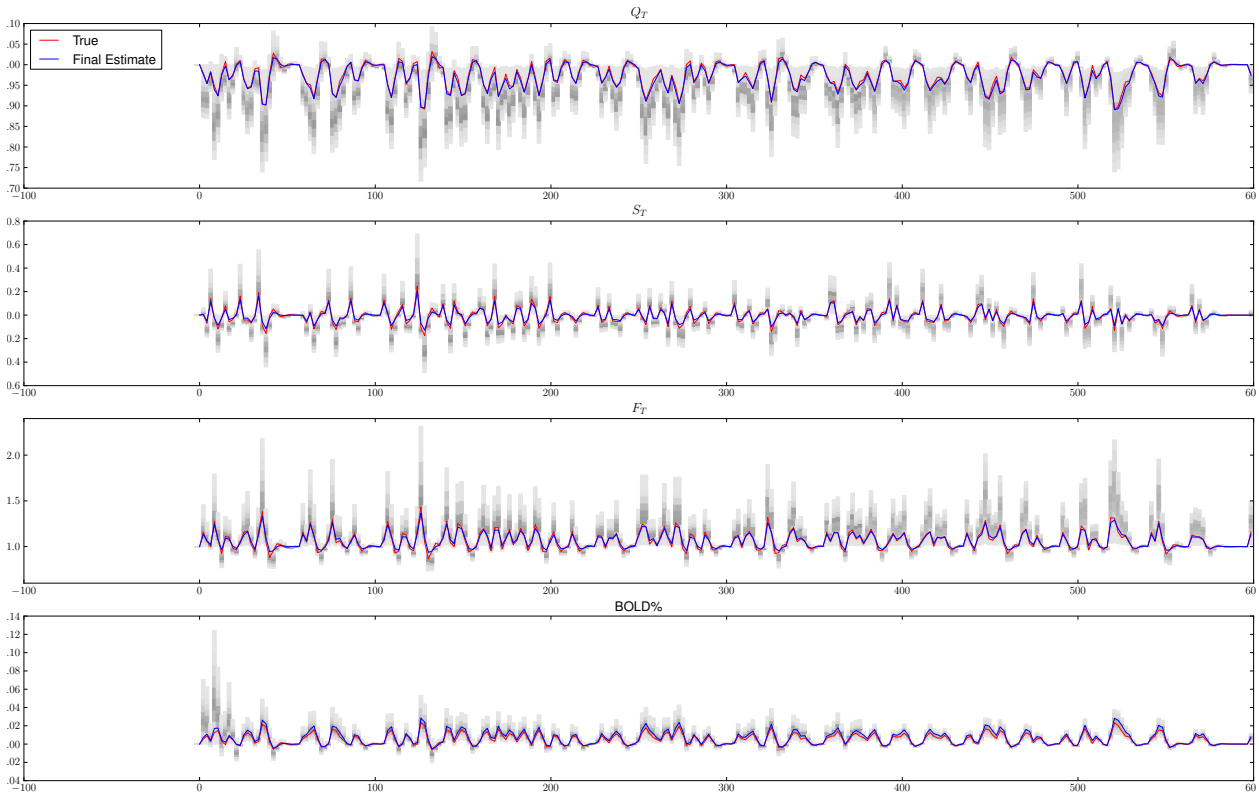


Figure 5.6: Convergence of the parameters from the first run of Table 5.3. Order of estimates: $\tau_0, \alpha, E_0, V_0, \tau_s, \tau_f, \epsilon, v, q, s, f, BOLD$. The bars represent a histogram, where darker bars indicate more particles with parameters in that bin. The red line is the parameter used to generate the true signal, blue line is the final mean of the particles.

τ_0	α	E_0	V_0	τ_s	τ_f	ϵ	$\sum \tau$	RMSR	RMSE
1.45	0.3	0.47	0.044	1.94	1.99	1.8	5.38		
1.2221	0.3449	0.3346	0.0714	1.6045	2.2753	1.5945	5.1019	0.003211	0.00224
1.3749	0.3318	0.3630	0.0733	1.6408	2.1030	1.5763	5.1187	0.003055	0.00223
1.1660	0.3221	0.3406	0.0822	1.6477	2.3535	1.2452	5.1672	0.003289	0.00205
1.2318	0.3271	0.3403	0.0796	1.6270	2.1852	1.3033	5.0439	0.002847	0.00147
1.1832	0.3179	0.3472	0.0821	1.5496	2.2912	1.2782	5.0240	0.003006	0.00213
1.1424	0.334	0.3473	0.0737	1.6221	2.2908	1.4025	5.0553	0.002833	0.00184
1.3004	0.3596	0.3564	0.0768	1.5641	2.1323	1.6034	4.9968	0.003028	0.00255
1.2401	0.3460	0.3398	0.0891	1.6499	2.2366	1.2900	5.1265	0.003044	0.00238
1.1709	0.3274	0.3464	0.0826	1.5373	2.2826	1.3783	4.9909	0.003345	0.0027
1.1897	0.3434	0.3355	0.0798	1.5358	2.3075	1.4277	5.0330	0.003175	0.00244
1.184	0.3405	0.3502	0.0892	1.6103	2.2793	1.1645	5.0735	0.002889	0.00188
1.2187	0.3359	0.3456	0.0800	1.599	2.2488	1.3876	5.0665	0.003066	0.00217

Table 5.3: Estimated Parameters on 11 different runs with low noise. First row is the true value, last is the average. $\sum \tau$ is the sum of the estimated time constants.

Note the subtle distinction between this and the Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum(\hat{y}_k - Y_k)^2} \tag{5.2}$$

where $\hat{y}_k$ is the estimated output, and Y_k is the underlying (free of noise) signal.

There are a few important results in the final parameter estimates (which are the mean of the particle filter's posterior distribution). First, the time constants vary greatly across runs, yet the sum of the individual time constants (τ_f, τ_s and τ_0) was more consistent. On average the time constants fell short of the true time constant. This could be a limitation based on the prior distribution (which notably had an initial mean below the true values) or it could be that other parameters compensated. It is also possible that the output is insensitive to small differences in the time constants. The convergence of the first run in Table 5.3 demonstrates the migration of parameters through the run. In spite of the significant differences in parameter estimates, the estimated BOLD consistently performed well (Figure 5.5).

5.1.3 Simulation with High Noise

For the high noise simulation, the exact same procedure was followed as in Section 5.1.2 except that σ_y and σ_x were set to 0.01 and 0.005, respectively. This is an order of magnitude higher than the previous tests, and indeed the noise appears to dominate the output, as Figure 5.7 shows. The results of the particle filter for each of the eleven runs are shown in Figure 5.8. The noise and preprocessing again led the estimates to higher peak activation levels, and the subtleties of different time constants were lost in the noise.

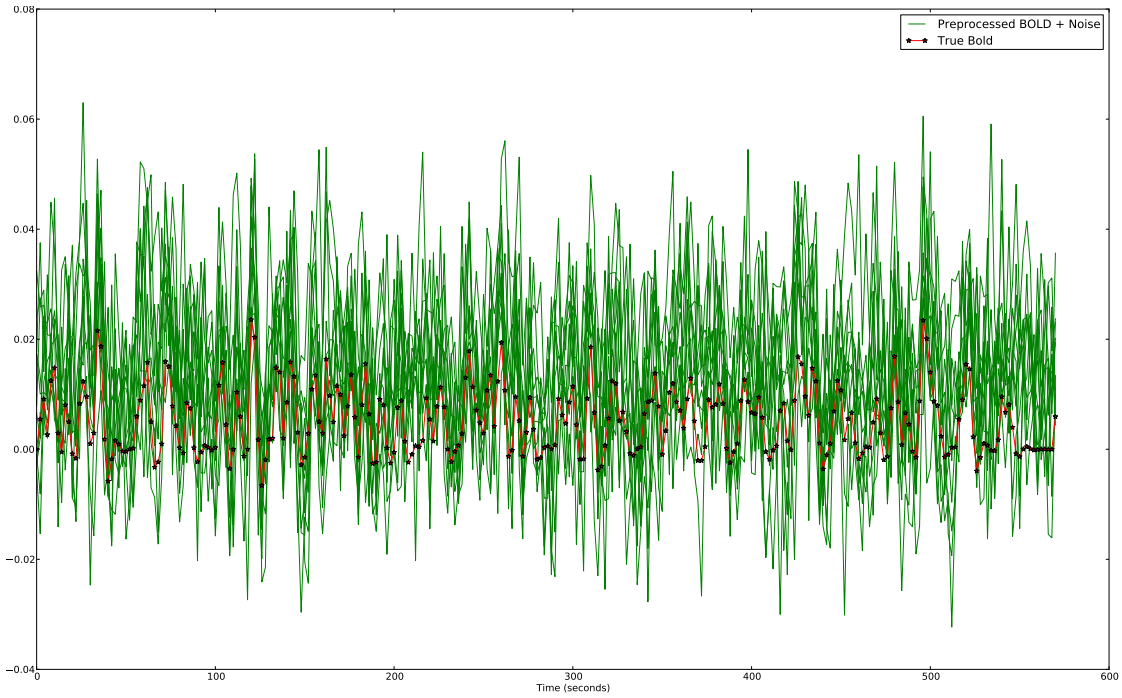


Figure 5.7: Results of preprocessing compared to the true signal. Knots of spline placed every 20 measurements. ($\sigma_x = 0.005$, $\sigma_y = 0.01$)

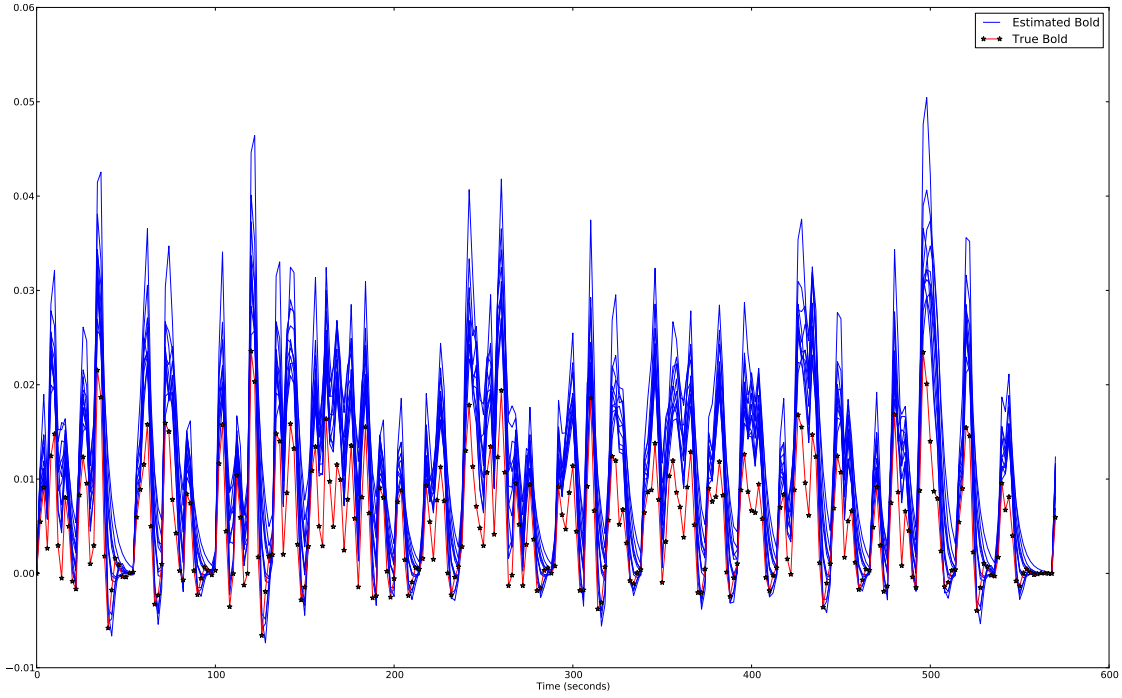


Figure 5.8: Results of the particles filter with preprocessed signals from Figure 5.7 as input.

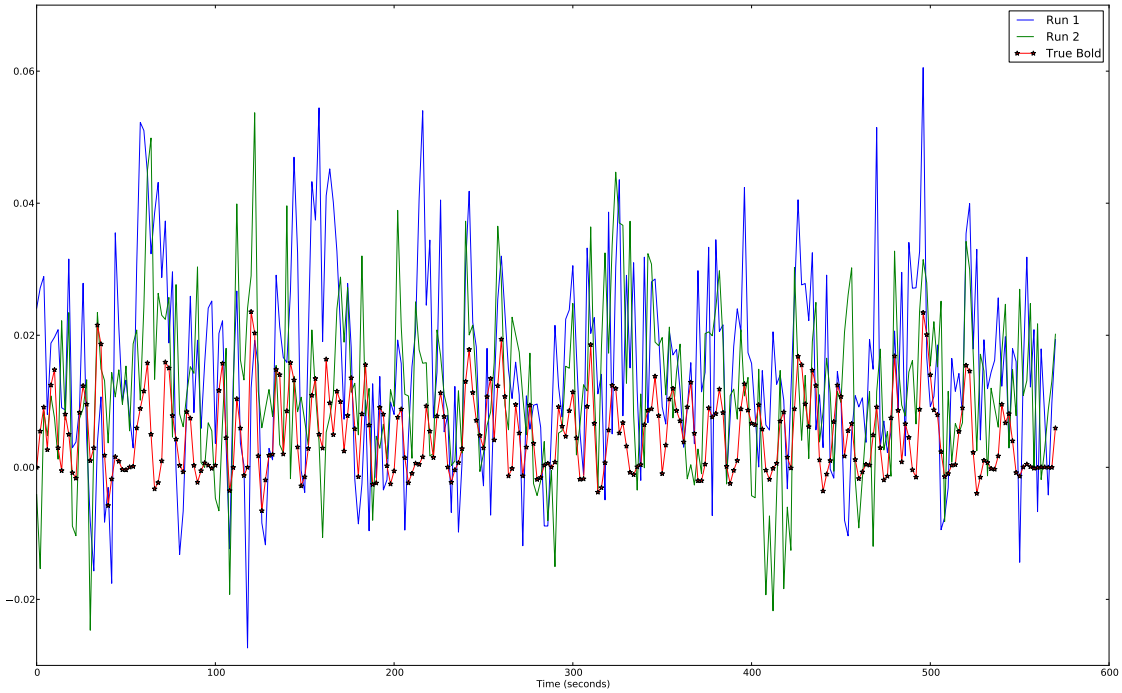


Figure 5.9: Two particular preprocessed noise realizations for the high noise case.

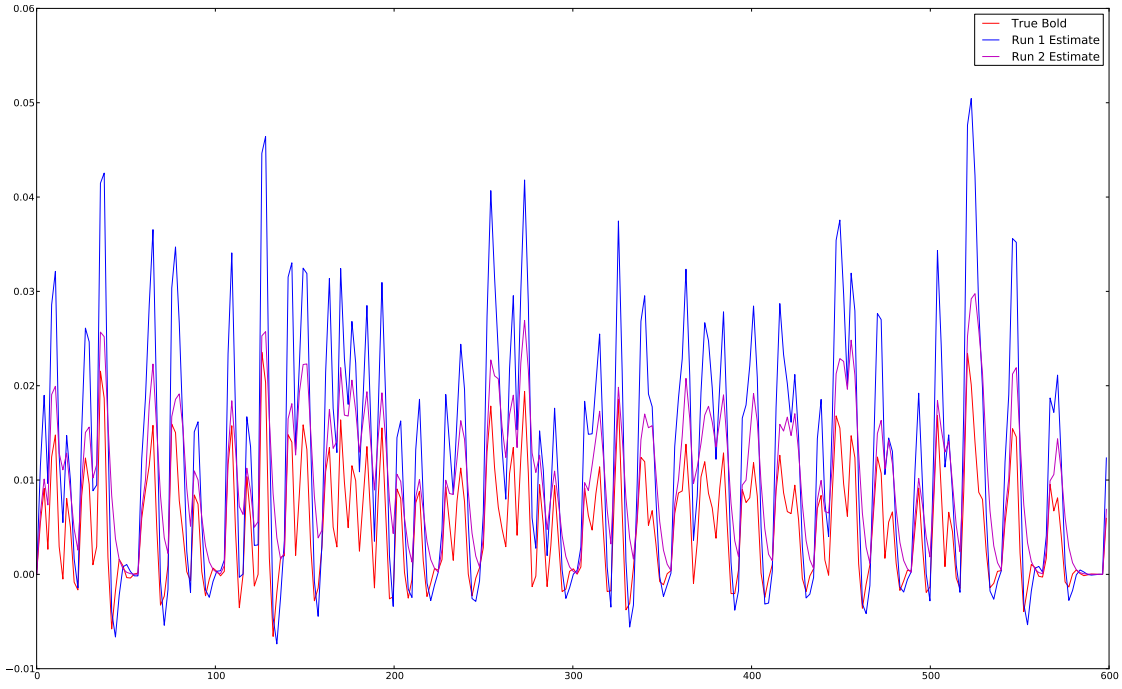
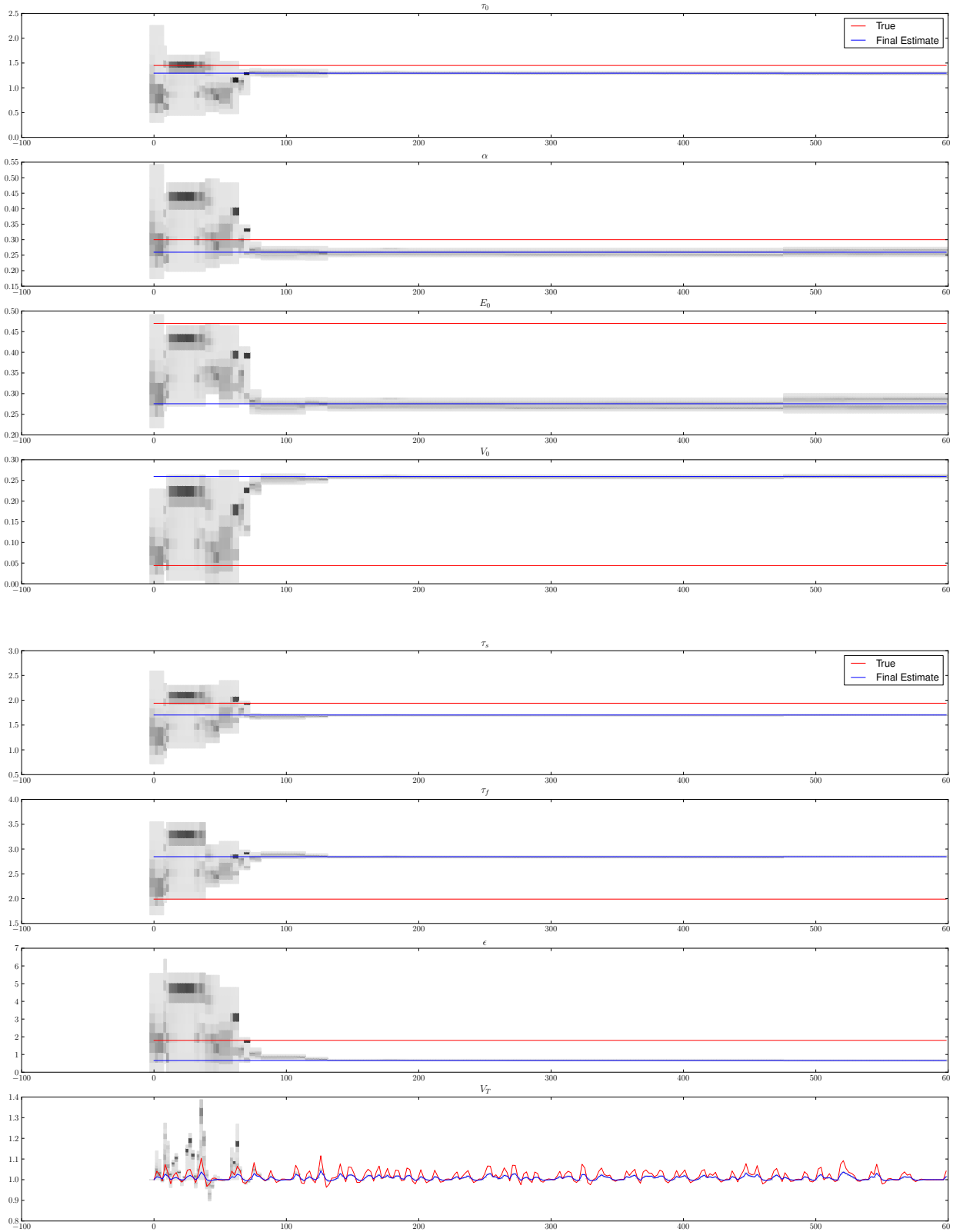


Figure 5.10: The results for the noise realizations shown in Figure 5.9.

τ_0	α	E_0	V_0	τ_s	τ_f	ϵ	$\sum\tau$	RMSR	RMSE
1.45	0.3	0.47	0.044	1.94	1.99	1.8	5.38		
1.1900	0.2349	0.4223	0.128	1.0147	2.4779	1.1168	4.6826	0.01406	0.00859
0.9721	0.2190	0.3051	0.061	0.5780	1.9960	3.4613	3.5461	0.01373	0.00735
1.5795	0.1415	0.3380	0.1089	0.5843	2.1247	1.7834	4.2885	0.01275	0.00951
1.1094	0.2374	0.5349	0.0351	1.2186	3.0736	2.3504	5.4016	0.01673	0.00479
1.1071	0.2753	0.3365	0.0316	1.5057	2.6518	4.1910	5.2646	0.01370	0.00475
0.5803	0.4793	0.4135	0.1189	0.9756	3.6902	1.0008	5.2461	0.01150	0.00672
1.2952	0.2596	0.2756	0.2595	1.7026	2.8458	0.6617	5.8436	0.01555	0.01039
1.5185	0.2199	0.2835	0.0742	0.8882	3.0771	1.7393	5.4838	0.01205	0.00655
0.6874	0.3283	0.3979	0.1561	1.0778	3.1158	0.6643	4.8810	0.01510	0.0057
1.0170	0.285	0.3474	0.0567	1.5877	2.6516	2.2852	5.2563	0.01249	0.00582
0.9925	0.298	0.3221	0.2094	0.4276	2.2108	1.0167	3.6308	0.01217	0.00916
1.0954	0.2708	0.3615	0.1126	1.0510	2.7196	1.8428	4.8659	0.01362	0.00721

Table 5.4: Estimated Parameters on 11 different runs with high noise. First row contains the true parameters, last row contains the mean. The red row is Run 1 and the blue row is Run 2 from Figure 5.9 and Figure 5.10, respectively.

Figure 5.9 shows two runs in more detail; these results show that more drift was present that 20 measurements per knot could fit. Insufficient flexibility of the spline explains the prolonged increase at 170 seconds in Run 1; although such areas permeate the preprocessed signals. Interestingly, Run 1 and Run 2 emphasize different aspects of the signal. Run 2 had a much better match to the peaks, when compared to the true signal, yet Run 1 matched the post-stimulus undershoot better. Table 5.4 shows the RMSE for all eleven runs, and highlights the two runs analyzed in Figure 5.11 and Figure 5.12.


τ_0
True
Final Estimate
α
E_0
V_0
τ_s
True
Final Estimate
τ_f
ϵ
V_T

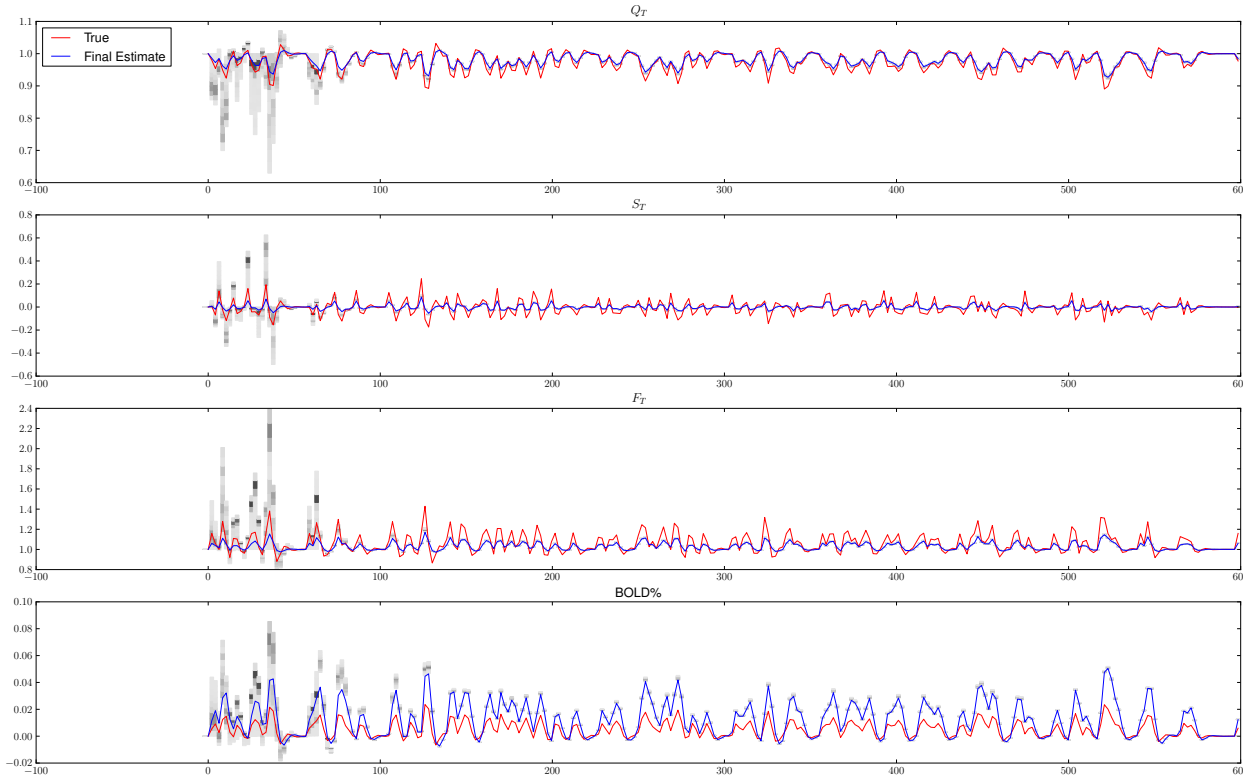
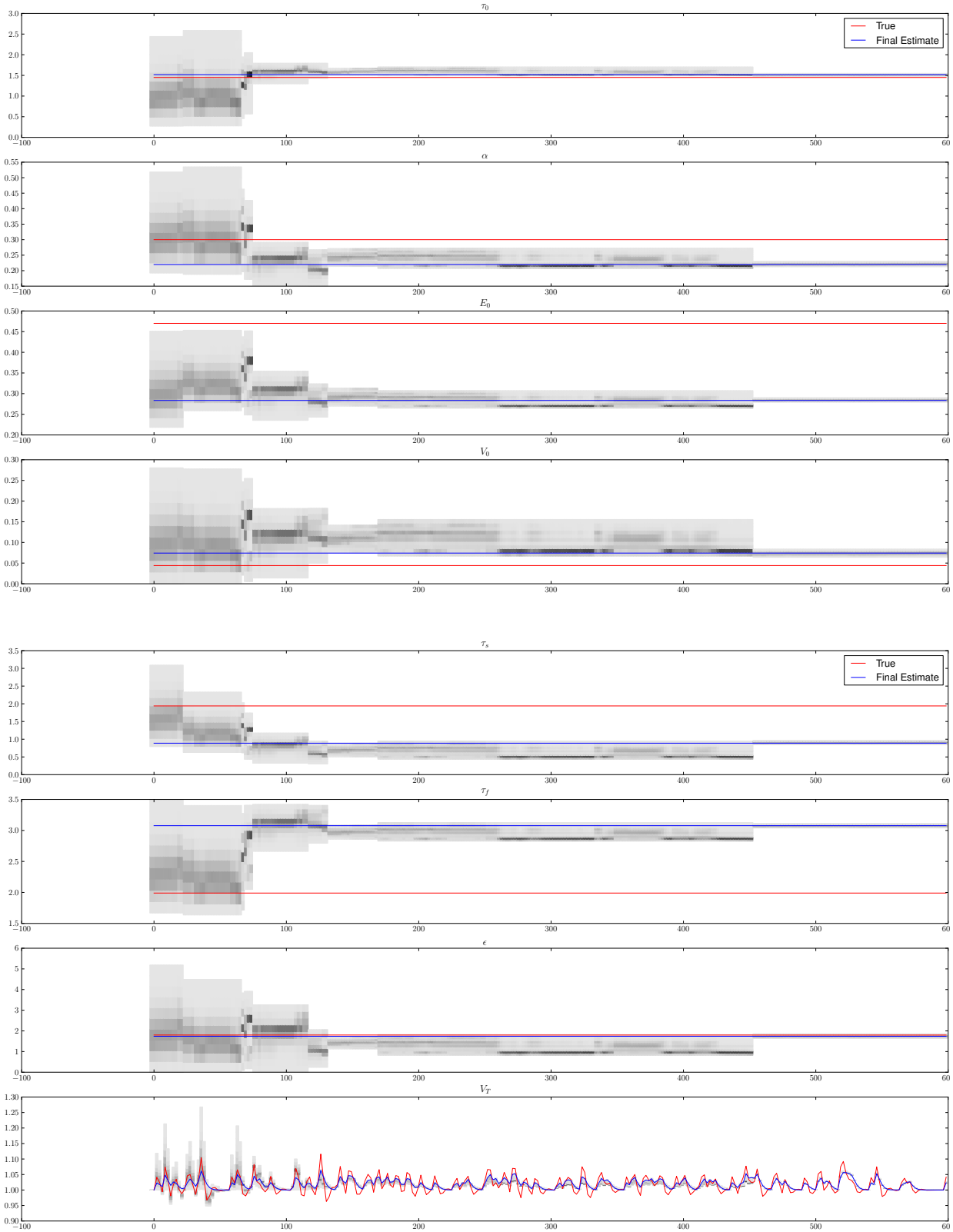


Figure 5.11: Convergence of the parameters during run 1 of Figure 5.9. Order of estimates: $\tau_0, \alpha, E_0, V_0, \tau_s, \tau_f, \epsilon, v, q, s, f, BOLD$. The bars represent a histogram, where darker bars indicate more particles with parameters in that bin. The red line is the parameter used to generate the true signal, blue line is the final mean of the particles.


τ_0
True
Final Estimate
α
E_0
V_0
τ_s
True
Final Estimate
τ_f
ϵ
V_T

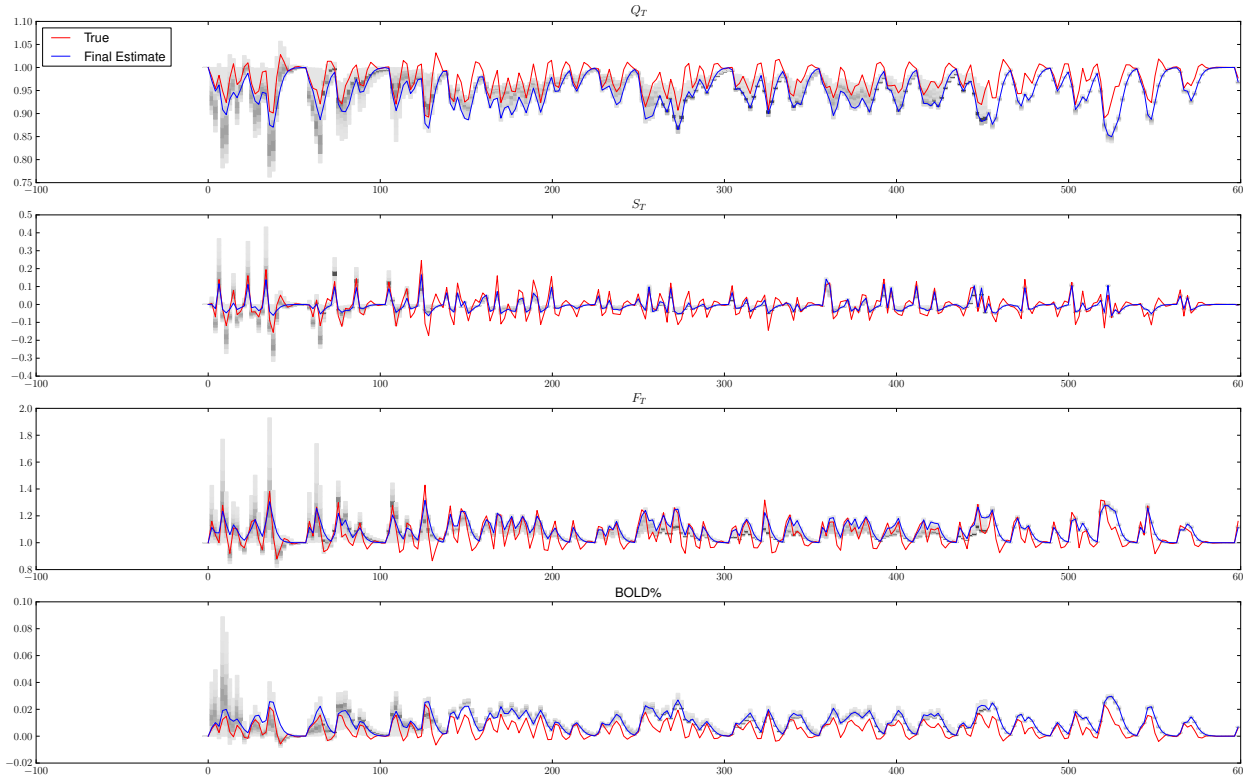


Figure 5.12: Convergence of the parameters during run 2 of Figure 5.9. Order of estimates: $\tau_0, \alpha, E_0, V_0, \tau_s, \tau_f, \epsilon, v, q, s, f, BOLD$. The bars represent a histogram, where darker bars indicate more particles with parameters in that bin. The red line is the parameter used to generate the true signal, blue line is the final mean of the particles.

The particles converged much faster when more noise was present (Figure 5.6 vs. Figure 5.11, Figure 5.12). This also caused significantly more resampling which is the explanation for the perceived jumps in the histograms. Clearly the additional noise resulted in less consistent results (Table 5.4). This is often the result when the particle filter converges too fast, in this case the result of the weighting function's variance being smaller than the measurement noise (0.005 vs. 0.01). The RMSE clearly suffers due to this effect (Table 5.4). Note that neither Run 1 nor Run 2 estimated the underlying state well (5.11(c) and 5.12(c)), whereas in the previous test the particle filter was extremely successful in this area (5.6(c)).

5.1.4 Pure Noise, Low magnitude

The next two single-voxel tests forced the particle filter to attempt to learn a noise-only time series. In this test the noise was the same as that from the Section 5.1.3, $\sigma_x = 0.01, \sigma_y = 0.005$. The stimulus neuronal efficiency (ϵ) was set to 0, in effect simulating a brain region with no response to

the stimuli. This test was used to determine how the output of a pure noise time series is different from that of a simple noisy signal (as in the previous two sections). The preprocessed signals are shown in Figure 5.13 and line fit for each run is shown in Figure 5.14.

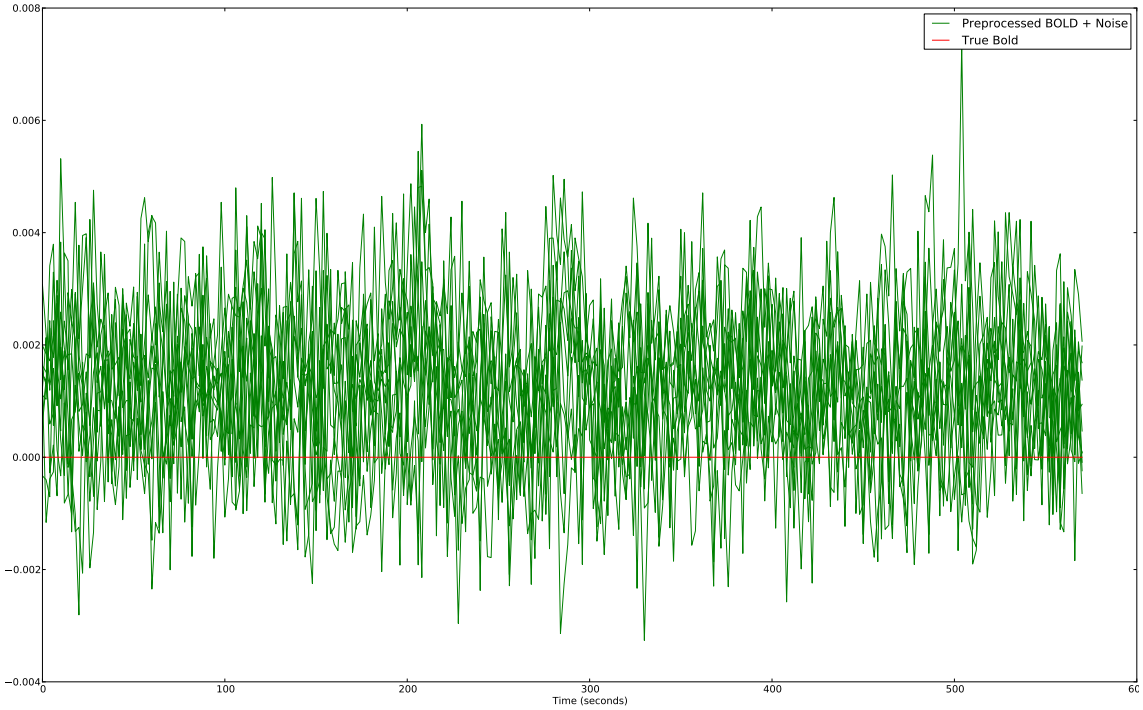


Figure 5.13: Comparison of the preprocessed signals for the low noise signal-free case. ($\sigma_x = 0.005$, $\sigma_y = 0.01$)

τ_0	α	E_0	V_0	τ_s	τ_f	ϵ	RMSR
1.0324	0.33211	0.34058	0.03012	1.40665	2.52079	0.5311	0.00167
0.98189	0.33047	0.3386	0.03014	1.45707	2.47232	0.45049	0.00159
1.0429	0.33224	0.34124	0.02946	1.4618	2.49245	0.43012	0.00165
1.02054	0.3321	0.33484	0.02586	1.45848	2.48741	0.4193	0.00151
1.0565	0.33405	0.33758	0.02791	1.43784	2.52545	0.47517	0.00152
1.01867	0.33528	0.33918	0.02782	1.48345	2.49605	0.44209	0.00156
1.051	0.33038	0.33837	0.02985	1.47651	2.48621	0.42719	0.00159
1.00281	0.32929	0.33988	0.0298	1.43519	2.49256	0.48899	0.00164
1.00893	0.33273	0.33982	0.0289	1.42903	2.49754	0.45688	0.00168
1.01289	0.33275	0.3376	0.02997	1.41188	2.49881	0.50628	0.00183
1.10247	0.33371	0.3419	0.02939	1.43774	2.53384	0.44079	0.00195
1.03009	0.33228	0.33905	0.02902	1.44506	2.50031	0.46076	0.00165

Table 5.5: Estimated Parameters on 11 different runs with low noise and no signal present. ($\sigma_x = 0.005$, $\sigma_y = 0.01$)

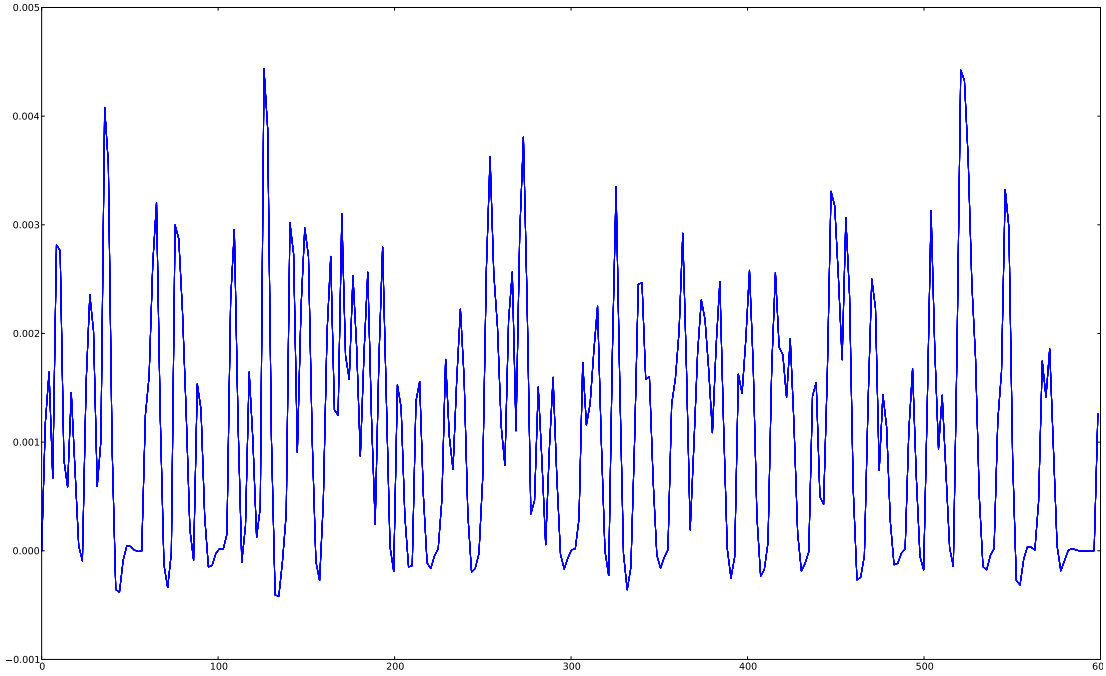


Figure 5.14: Comparison of the estimated BOLD signal for the low noise signal-free case. ($\sigma_x = 0.005$, $\sigma_y = 0.01$). Note the line thickness is caused by all the estimates overlapping.

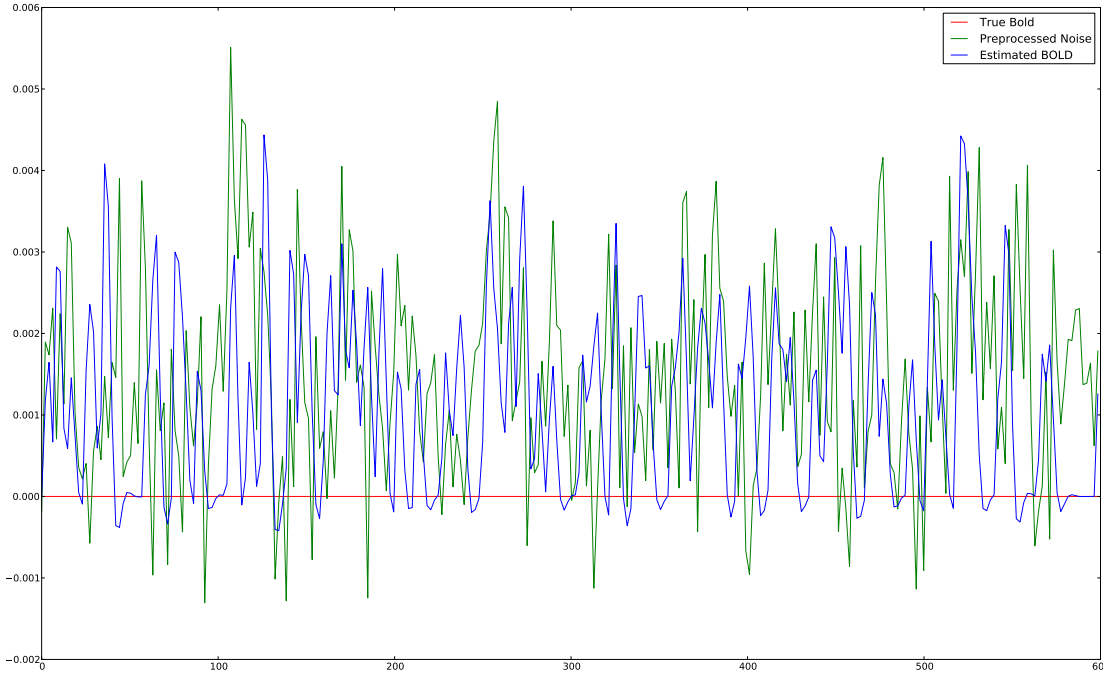
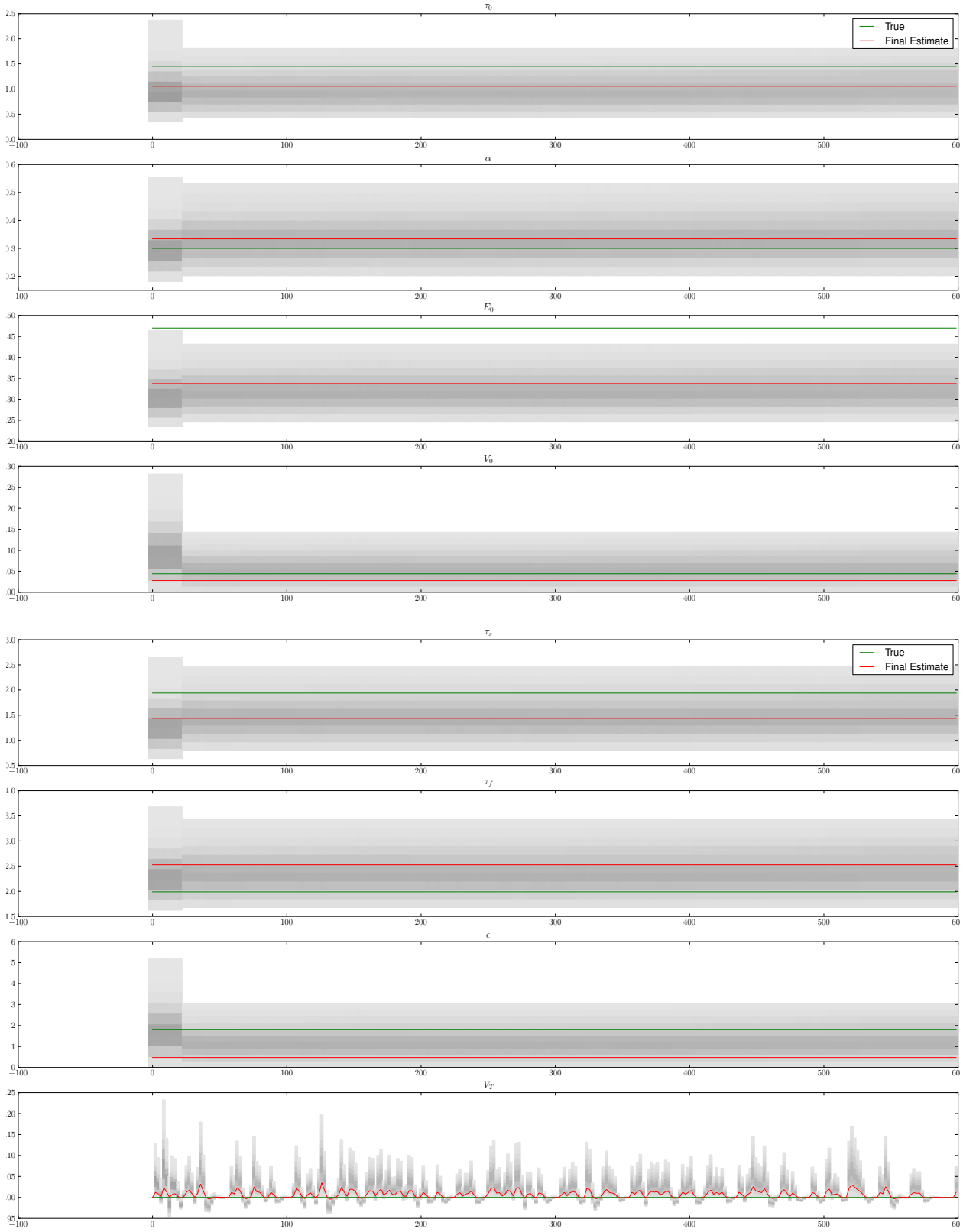


Figure 5.15: Single Fit Results for non-active, low noise signal ($\sigma_x = 0.005$, $\sigma_y = 0.01$).


τ_0
True
Final Estimate
α
E_0
V_0
τ_s
True
Final Estimate
τ_f
ϵ
V_T

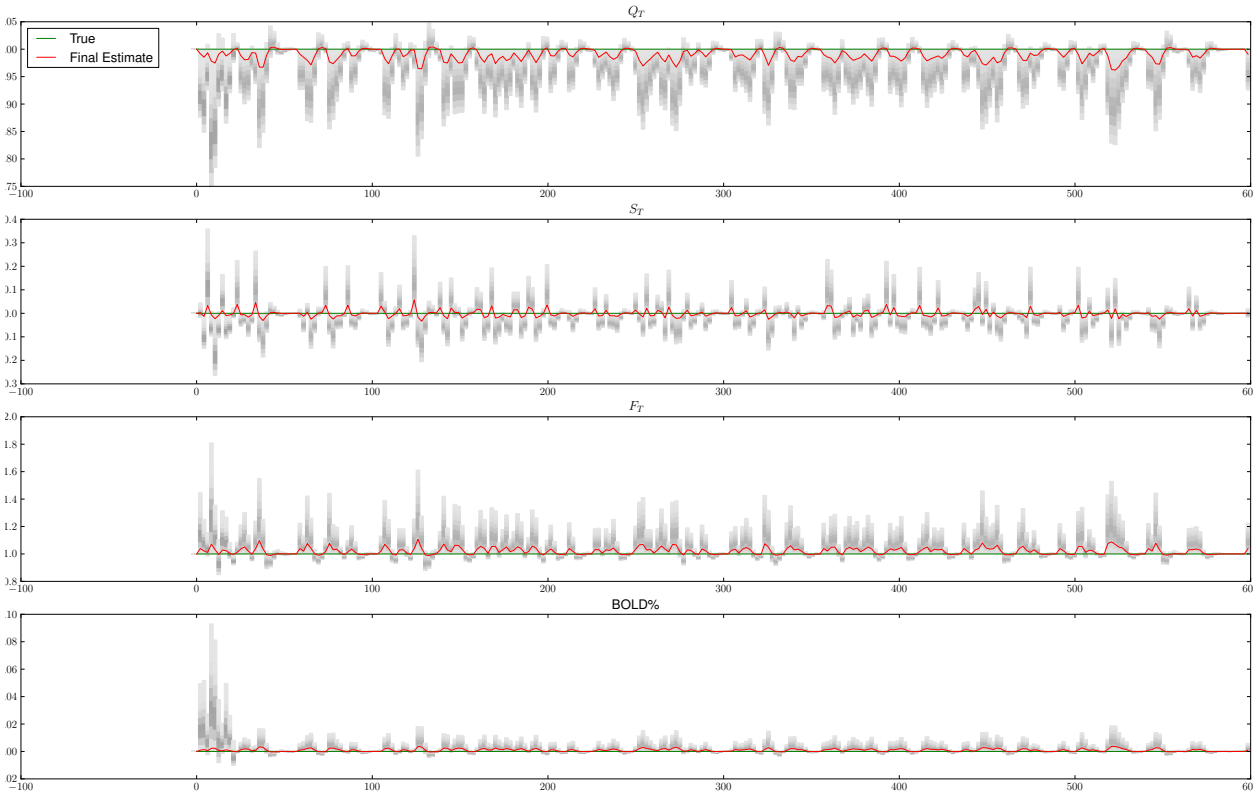


Figure 5.16: Convergence of the parameters when the signal consisted purely of low level noise ($\sigma_x = 0.01$, $\sigma_y = 0.005$). Order of estimates: $\tau_0, \alpha, E_0, V_0, \tau_s, \tau_f, \epsilon, v, q, s, f, BOLD$. The bars represent a histogram, where darker bars indicate more particles with parameters in that bin. The red line is the parameter used to generate the true signal, blue line is the final mean of the particles. See Figure 5.15

The data shows that the parameters did not converge (Figure 5.16). The peaks never even reached 1% difference (Figure 5.13) so the signal stayed well within the range of 0.005, the standard deviation of the weighting function. Note that the RMSRs were actually lower than the RMSRs in the low noise simulation from Section 5.1.2 and the parameter estimates were more consistent across 11 runs. The low RMSR was caused by the overall signal being significantly smaller than any previous simulation. This run would benefit greatly from an adaptive weight function, since such a much tighter weighting function would have forced the particle filter to fail.

5.1.5 Pure Noise, High Magnitude

To determine how the particle filter responds to active, yet unrelated portions of the brain, this section repeats the test of Section 5.1.4 with much higher noise peaks. To simulate this case another pure noise signal was generated using σ_x of 0.1 and σ_y of 0.05.

As before, the convergence all follows a similar path, leaving almost no variance in the estimated time series (Figure 5.17). Interestingly the algorithm suffered from almost constant particle deprivation, meaning that the heuristic for rescuing the particle filter from particle deprivation, discussed in Section 4.3.3 was used when it shouldn't have been. When this mechanism was removed, all 11 runs stopped due to particle deprivation (all weights hit zero). The problem with allowing particle deprivation to occur is that it can *rarely* occur in otherwise good data if resampling is performed at the wrong moment.

Because of the preprocessing steps, the pure noise signal can look markedly like a real signal (Figure 5.18). The preprocessing causes the particle filter to converge to a non-zero response in spite of the fact that the input does not correlate with the stimuli in any way (Figure 5.17).

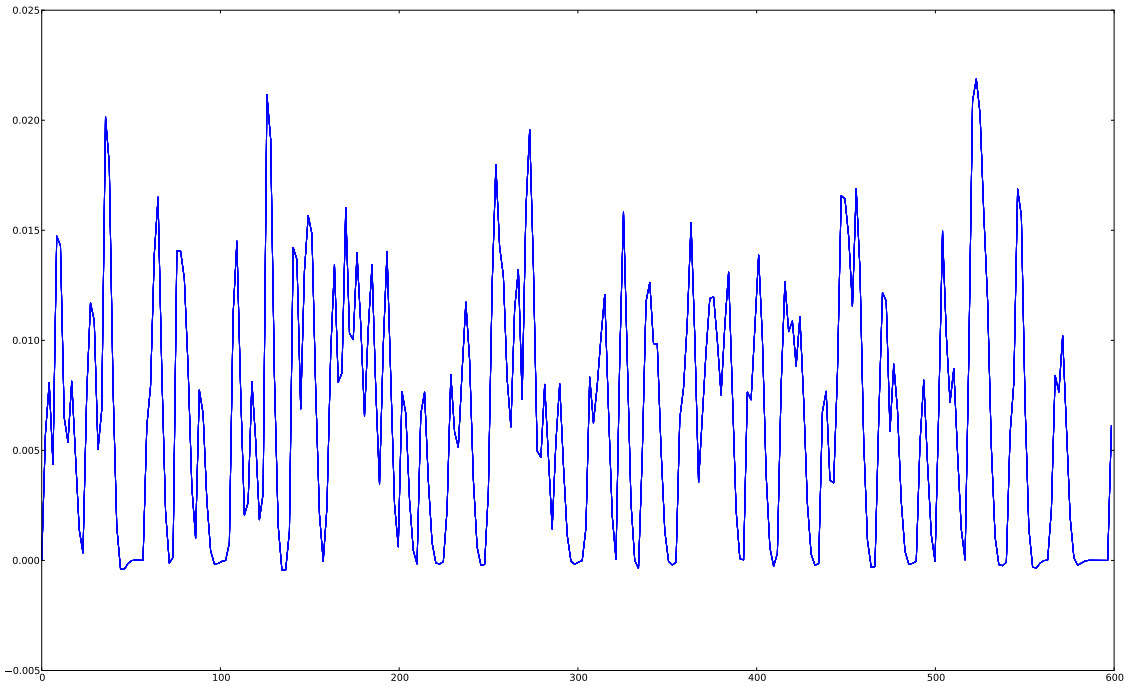


Figure 5.17: BOLD estimates for the non-active, high noise signal ($\sigma_x = 0.1$, $\sigma_y = 0.05$). Note the line thickness is caused by all the estimates overlapping.

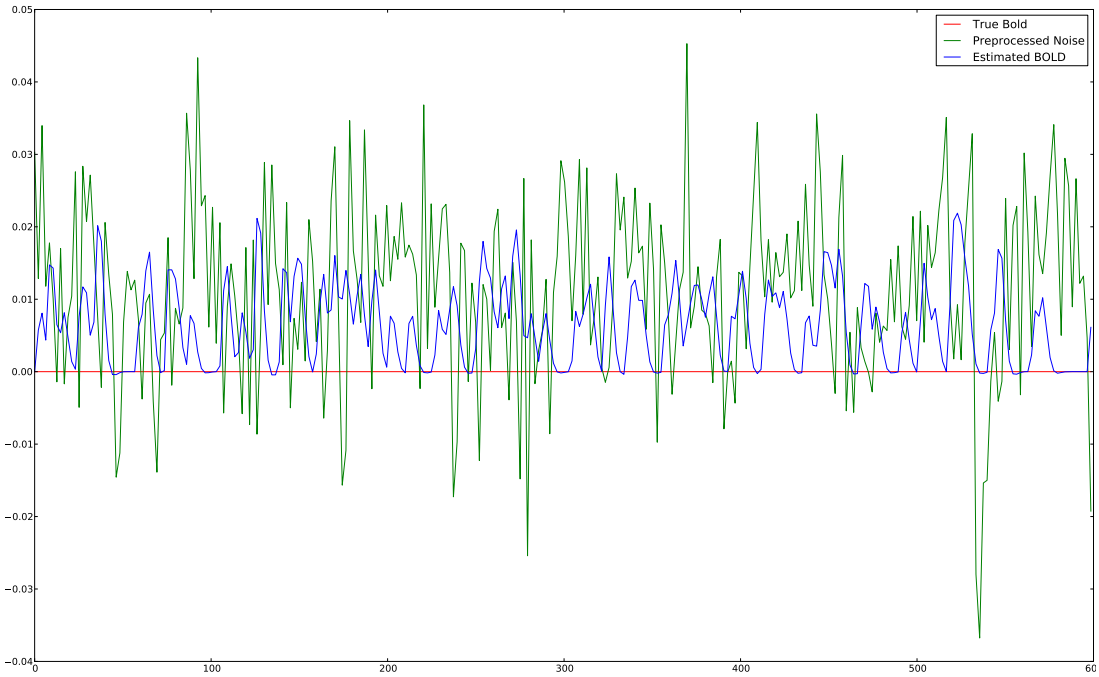


Figure 5.18: Single Fit Results for non-active, high noise signal ($\sigma_x = 0.005$, $\sigma_y = 0.01$).

5.1.6 Single Voxel Summary

Because of the variability in the signal levels, the raw RMSR cannot be used to rate the fit. As demonstrated by Table 5.5, a low RMSR does not necessarily indicate a good fit. Therefore, a normalized version of the RMSR was used. To normalize the residual, both the estimated and preprocessed BOLD signals were divided by an estimator of scale and then the new RMSR was calculated. Considering the tendency of fMRI to have large unexplainable peaks and troughs, the MAD was used (Equation 4.5). This is an estimator of the standard deviation, and thus a good estimator of the scale of the input signal. The normalized RMSR values are shown in Table 5.6. A second potential method of gauging performance is Mutual Information (MI). Mutual Information is a method of measuring the interdependence of two random variables. If two signals are truly independent, then the MI will be zero. Although ideally suited to discrete distributions, by using histograms it is possible to derive a joint distribution of two signals. The algorithm for MI is based on that joint distribution:

$$\sum_{x,y} p(x,y) \log_2 \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{5.3}$$

Unfortunately the number of bins causes bias in the output, thus to correct for this, I subtracted the estimated bias:

$$\text{bias} = \frac{N_{bins}}{2Nlog(2)} \tag{5.4}$$

	Signal				No Signal			
	Low Noise		High Noise		$\sigma_y = 0.001, \sigma_x = 0.0005$		$\sigma_y = 0.01, \sigma_x = 0.005$	
	MI	N. Res.	MI	N. Res.	MI	N. Res.	MI	N. Res.
1	0.86687	0.47801	0.09077	1.03894	0.06326	1.29501	0.03024	1.33641
2	0.93975	0.53177	0.13767	0.95165	-0.01075	1.30175	-0.02677	1.33667
3	0.82382	0.5458	0.13505	0.99539	0.02345	1.26287	-0.0111	1.15957
4	0.94661	0.49824	0.04341	1.16129	-0.00906	1.43196	0.00147	1.09988
5	0.94281	0.46805	0.13718	1.03972	0.00663	1.25664	-0.00204	1.20107
6	0.92539	0.459	0.12337	1.00214	-0.00816	1.2708	0.01775	1.04589
7	0.98892	0.46096	0.15381	1.08847	0.02664	1.15441	0.03163	1.20543
8	0.98796	0.51838	0.11325	1.05962	0.03285	1.27456	0.01951	1.1225
9	0.8804	0.5253	0.09669	1.0157	0.01628	1.32024	0.01039	1.08637
10	0.88721	0.49211	0.18339	1.18996	0.00407	1.34456	0.00508	1.22135
11	0.96644	0.49092	0.10949	0.95368	0.03323	1.32522	-0.01284	1.11737
mean	0.92329	0.49714	0.12037	1.04514	0.01622	1.29437	0.00576	1.17568
min	0.82382	0.459	0.04341	0.95165	-0.01075	1.15441	-0.02677	1.04589
max	0.98892	0.5458	0.18339	1.18996	0.06326	1.43196	0.03163	1.33667

Table 5.6: MI and the normalized RMSE, for each of the previous sections.

where N is the number of samples and N_{bins} is the number of bins. For all the MI estimates in this work 6 bins were used for the marginal distribution of each signal. Additionally, throughout log base 2 will be used. This leads to 36 total bins in the joint, so the bias is:

$$\text{bias} = \frac{18}{N} \tag{5.5}$$

Note that subtracting the bias can result in negative MI, which should not technically be possible; so any negative Mutual Information was taken as 0.

Comparing the results of Section 5.1.3 and Section 5.1.5 in Table 5.6, distinguishing between these cases with either normalized RMSR or MI is not clear cut. While the average MI is more than 10 times the average MI in the two non-signal cases, the maximum MI of the low noise/no signal case exceeds the minimum MI of the high noise/signal case. The Low Noise/signal determination is easier to make; given the minimum MI is above .8 and the maximum normalized RMSR is below .6. However, it is worth noting that the worst case scenario for MI (maximum) in the low noise/no signal case does not coincide with the worse case (minimum) normalized residual. There is no reason why this has to be the case, but it could be beneficial. In other words, if it were necessary to make a statement that a particular voxel were active or inactive; the accuracy would improve if both techniques were used with loose restrictions. In all, both MI and the normalized residual provide a good measure of performance.

There were two primary purposes of these tests. First, given the nature of Monte Carlo techniques it is important to ensure consistency of results. Although the parameter sets were inconsistent,

Region	τ_0	α	E_0	V_0	τ_s	τ_f	ϵ
1	1.454	0.321	0.369	0.036	0.994	2.774	1.348
2	1.151	0.353	0.380	0.026	1.98	2.333	1.645
3	1.951	0.317	0.348	0.027	1.657	3.719	0.757
4	1.203	0.310	0.326	0.036	2.168	2.272	0.086

Table 5.7: Actual parameters for each regions in the simulated slice.

the quality of the fit for the BOLD output was often accurate even when noise was drastically increased. The second purpose was to validate MI and the normalized RMSR as measurements of output quality. Although not perfect, both methods do provide a decent metric.

5.2 Multi-voxel Simulation

To test the usefulness of the particle filter on a larger scale, a modified version of the FMRIB Software Library (FSL) tool POSSUM was used to generate an entire fMRI image from a parameter map. The parameter map was generated by taking an existing activation map and assigning discrete parameter sets to each region. The result was a four dimensional (length x width x height x parameter) image with spatially varying parameters. Possum was then modified to take a parameter map and generate activation levels depending on the parameters at that point. The patch for POSSUM will be made available. Note that the noise level was set to an SNR of 20, but due to changes in the program the true signal-to-noise ratio was much lower, as seen in the in 5.19(b). The mean SNR for each region was calculated only for the voxels with a Signal-To-Noise ratio above 0.1.

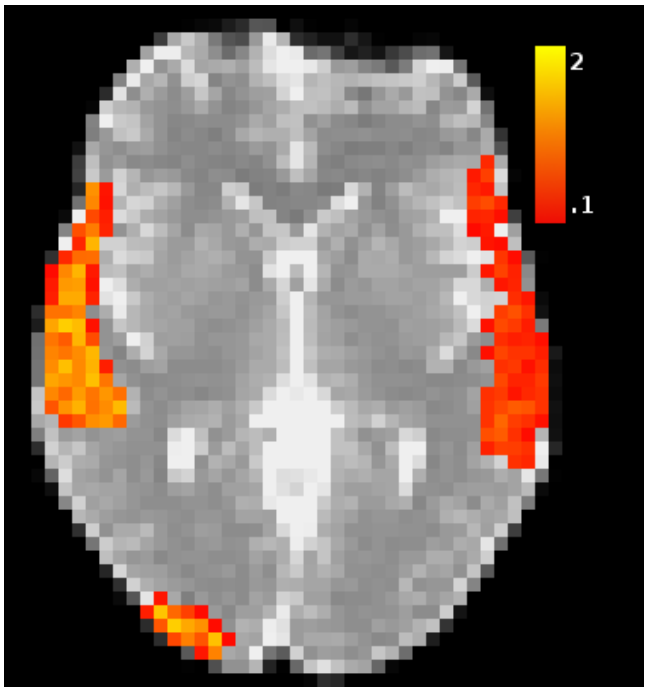
For each time series in the simulated fMRI image, the final parameters were saved into a parameter map. This parameter map could then be compared to the map used to generate the simulated data. Additionally a new simulation using the calculated parameters could also be generated to test the difference in BOLD levels between the real parameters and the estimated ones. Since the parameters were far from orthogonal, this provided a quantitative difference between the two parameter sets [9].

The regions are numbered according to 5.19(a); the parameters for each region may be found in Table 5.7.

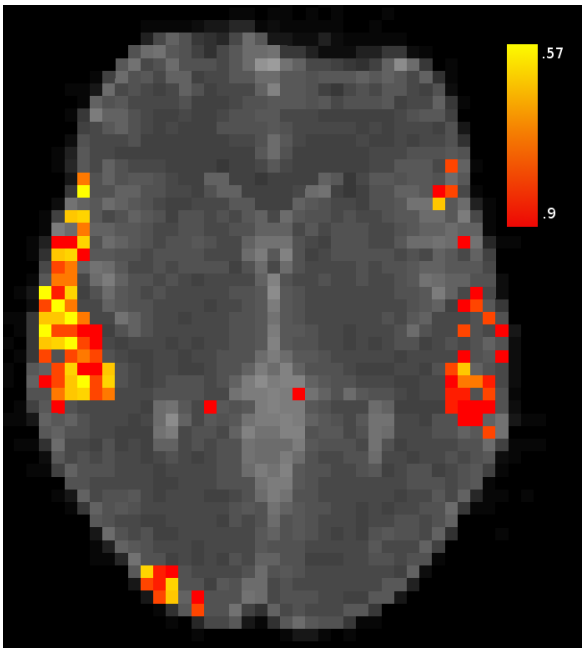
Note that region 4 had a very low ϵ, putting it below the noise threshold. For this reason, the only areas with significant estimates of the BOLD time series were 1,2 and 3. Notice that the regions 1, 2 an 3 stick out in both the RMSR and the MI map, indicating that the particle filter was successful in matching those regions. MI was an extremely successful metric, with the exception of a few false positives (5.19(c)). Figure 5.20 shows what the heat map looks like when the threshold is raised from 0.1 to 0.15.

If the thresholds are left at the values of Figure 5.19, then the normalized RMSR gave $2/1479$ false

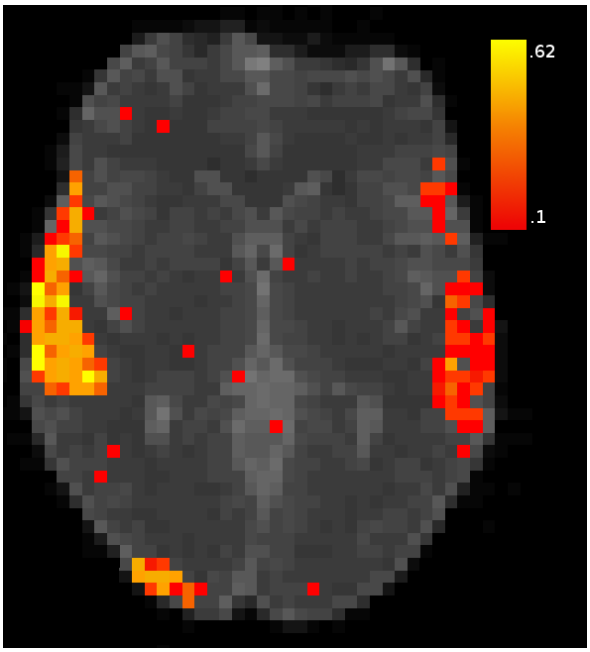
(a) Region labels for simulated slice.



(b) SNR Map of POSSUM simulated data. Mean SNR per region, Region 1: 0.8, Region 2: 0.97, and Region 3: 0.39.



(c) Normalized RMSR Map. Scale is Normalized RMSE. Lower (yellow) is better.



(d) MI Map. Scale is bits. Higher (yellow) indicates better fit.

Figure 5.19: Comparison of activation regions, a) grey matter and regions, b) SNR of active regions, c) RMSR Map, d) MI Map.

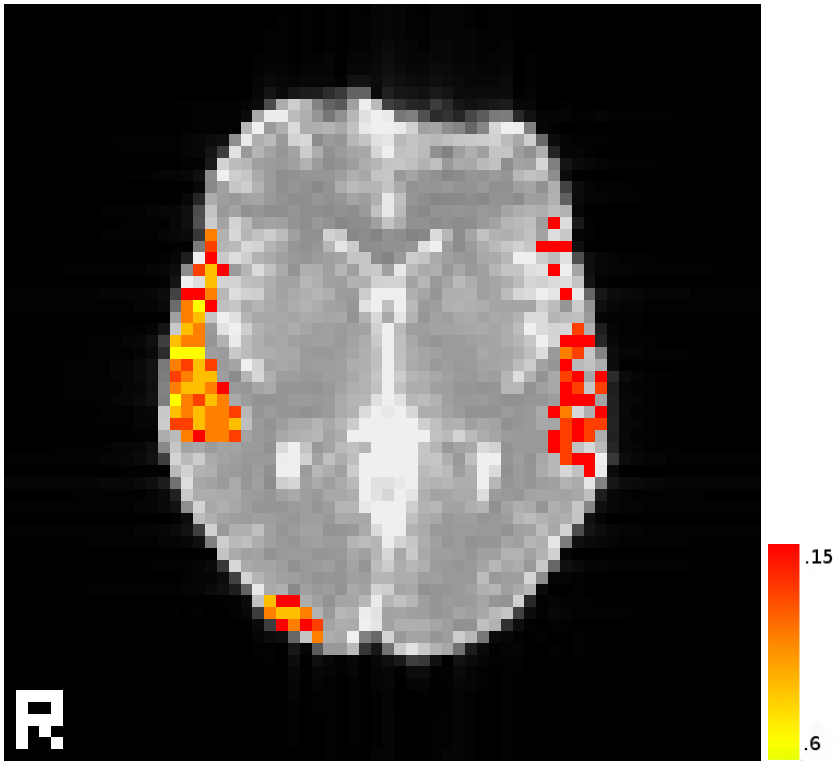


Figure 5.20: More stringent MI heatmap. Higher (yellow) is better.

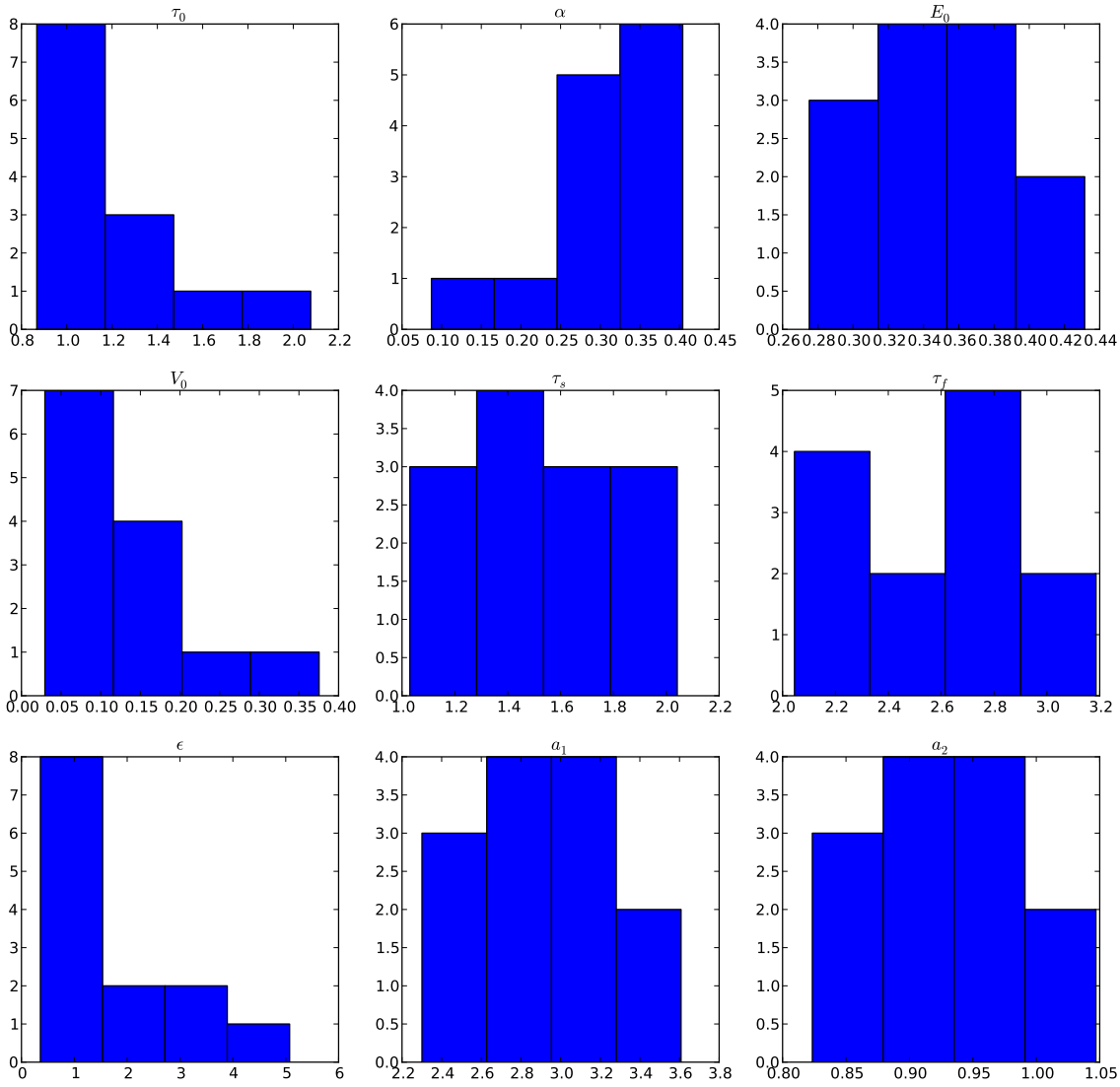


Figure 5.21: Histogram of estimated parameters in section 1 in voxels with MI greater than 0.15

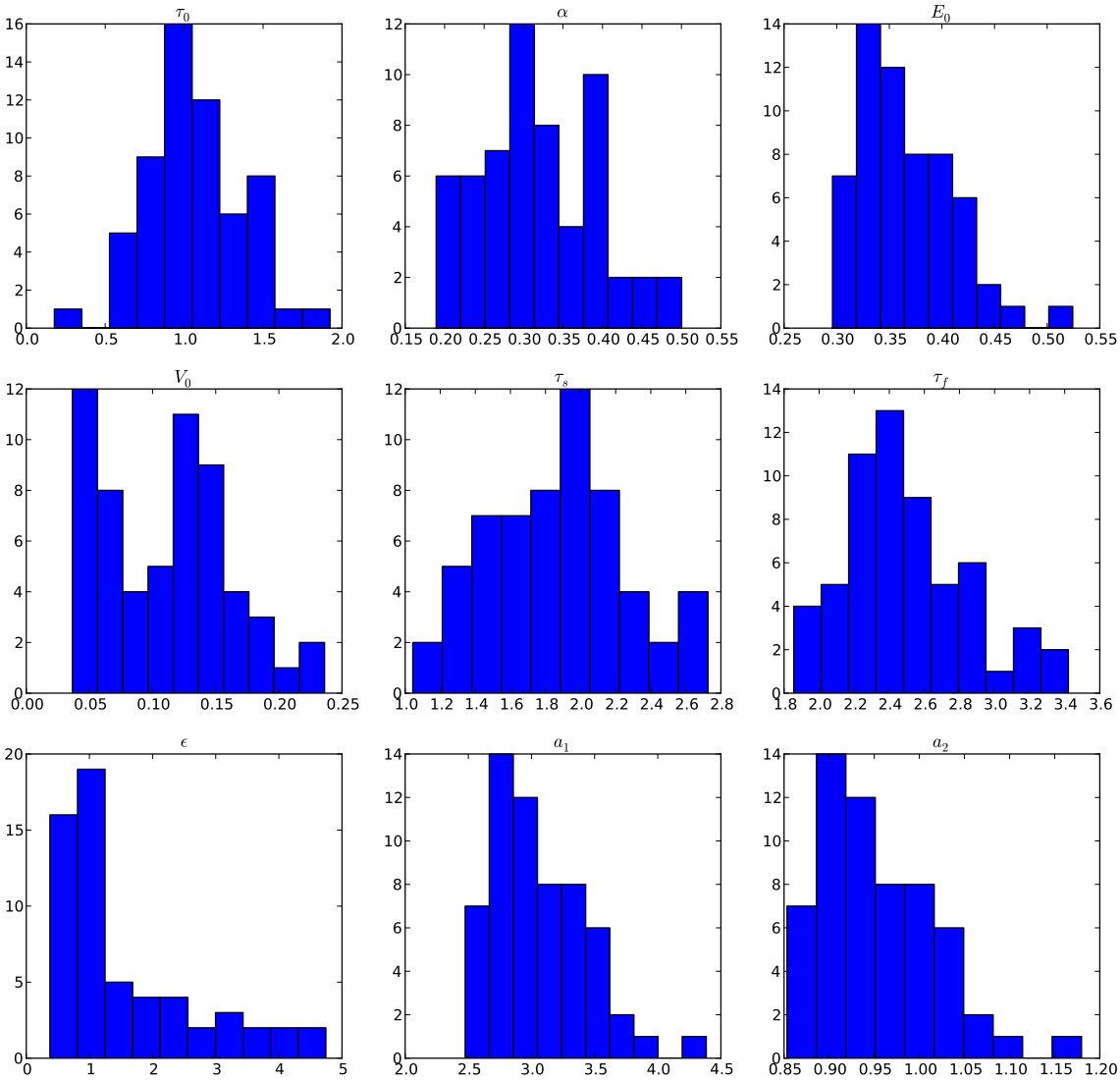


Figure 5.22: Histogram of estimated parameters in section 2 in voxels with MI greater than 0.15

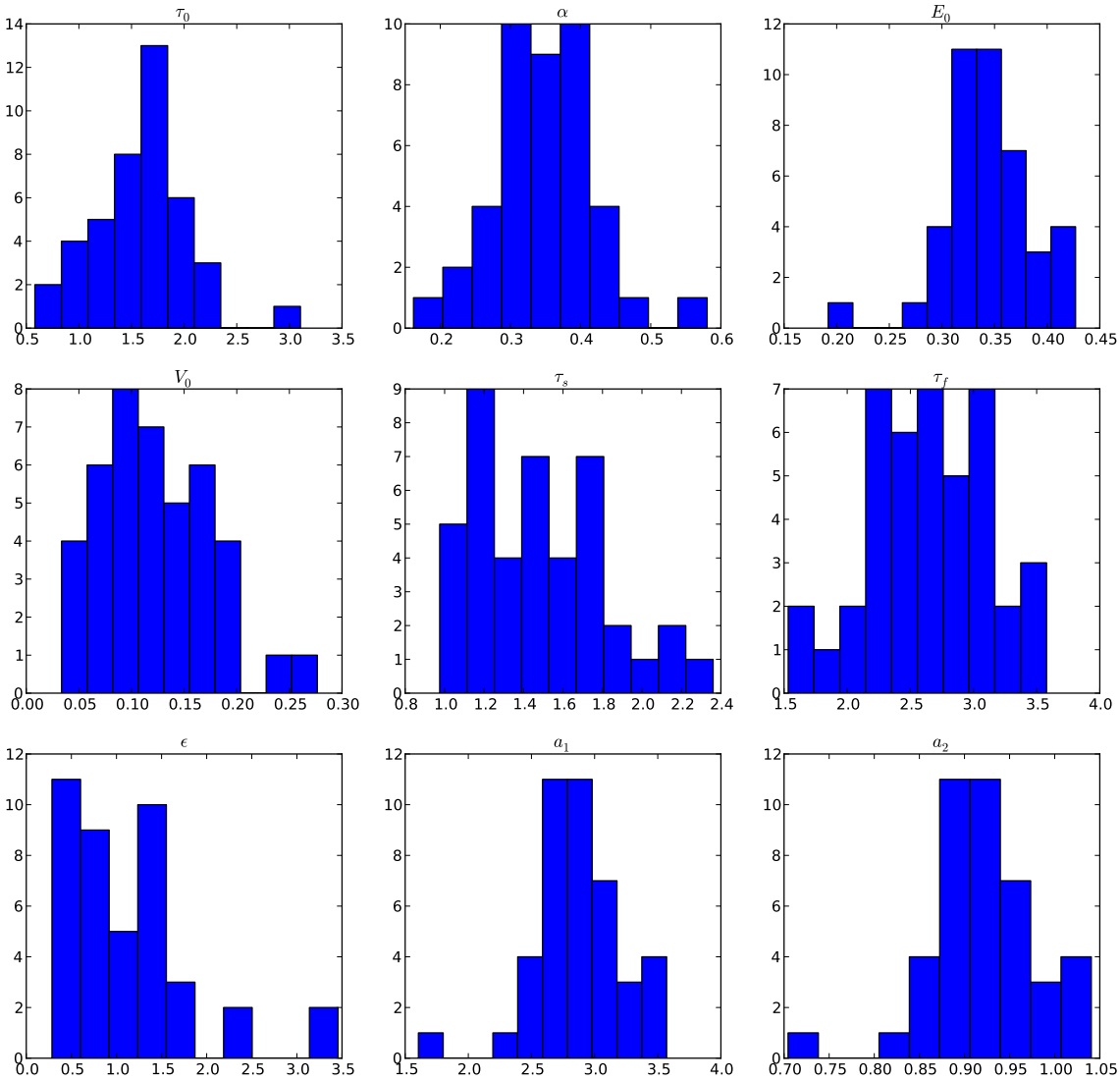


Figure 5.23: Histogram of estimated parameters in section 3 in voxels with MI greater than 0.15

positives and MI gave $11/1479$. However, normalized RMSR also clearly missed some active voxels, although false negatives are harder to quantify because of the presence of white matter. Raising the threshold to 0.15 eliminated the false positives in the MI map, although some active voxels were also removed.

The histograms again demonstrate that a single point estimate of the parameters is elusive for this set of parameters. However the data clearly show the power of the particle filter at identifying regions of activation. The thresholds applied to this slice, both for MI and RMSRs are arbitrary. Applying a threshold is helpful for visualization but is rarely useful for further analysis. As noted in Section 5.1.6, the false positives present in the MI map are different from those in the RMSR map. This furthers the argument for combining the two metrics to increase power. Although at first glance it would appear that there are false negatives in 5.19(d); this is not actually the case. POSSUM simulates different tissues, and white matter does not typically have a BOLD response. This explains some of the holes in regions 2 and 3. These results certainly indicate that the particle filter is effective at regressing against a noisy signal.

Chapter 6

Real Data

The ultimate goal of modeling the BOLD response is to estimate parameters in real fMRI data. The most basic use of modeling the BOLD signal is to locate activation. A voxel is considered active when the stimulus is the primary drive for the BOLD response. This is in contrast to most voxels which are controlled by intermediate or completely unrelated factors. Inactive regions cannot be modeled because their input is unknown, therefore parameter estimates in such regions are impossible. Because SPM is the de facto standard for localizing activation, this section compares its output with that of the particle filter.

Note that SPM must pre-process the image with a spatial smoothing filter. For this work SPM8 was smoothed with a $8\text{mm} \times 8\text{mm} \times 8\text{mm}$ Full-Width Half-Maximum (FWHM) Gaussian kernel. Additionally, SPM8 applied a high pass filter (with a cut off based on a globally estimated autocorrelation). Thus the preprocessing pipeline of SPM is very different from that of the particle filter. SPM8 also outputs a t-statistic for each voxel, whereas the particle filter's primary output is a posterior probability distribution of the parameters at every voxel. To validate the quality of the particle filter, the results were compared with SPM, both in terms of the location and the fit.

6.1 Experiment Configuration

For the fMRI data discussed in Chapter 6, tests were performed on a right handed volunteer using a GE SIGNA HDx 1.5 Tesla scanner with a single echo EPI sequence. Slice spacing was 5 mm, and pixel sizes were 3.75 mm. Repetition Time was 2.1 s, Echo Time was 40 ms and the imaging frequency was set to 63.854 MHz. The image resolution was $64 \times 64 \times 28$. The subject was presented with either a single or double flash and was asked to respond with a right handed or left handed finger tap, respectively. The fMRI began 18.9 seconds before the beginning of the experiment, to allow for the Gradient Echo spins to to settle out. The timing of the flashes is shown below with time 0 corresponding to the beginning of the 10th TR (so 9 images were dropped).

The timing of the single flashes were:

1.706, 11.944, 17.063, 18.769, 34.125, 39.244, 44.231, 47.644, 49.350, 61.294, 64.706, 66.413, 69.825, 71.531, 73.238, 76.650, 80.063, 90.169, 96.994, 110.644, 117.469, 120.881, 130.988, 132.694, 154.875, 158.288, 161.700, 165.113, 168.525, 176.925, 178.631, 183.750, 190.575, 204.225, 205.931, 211.050, 216.038, 222.863, 226.275, 236.513, 248.456, 255.281, 258.563, 263.681, 273.919, 277.331, 287.569, 292.688, 294.394, 299.381

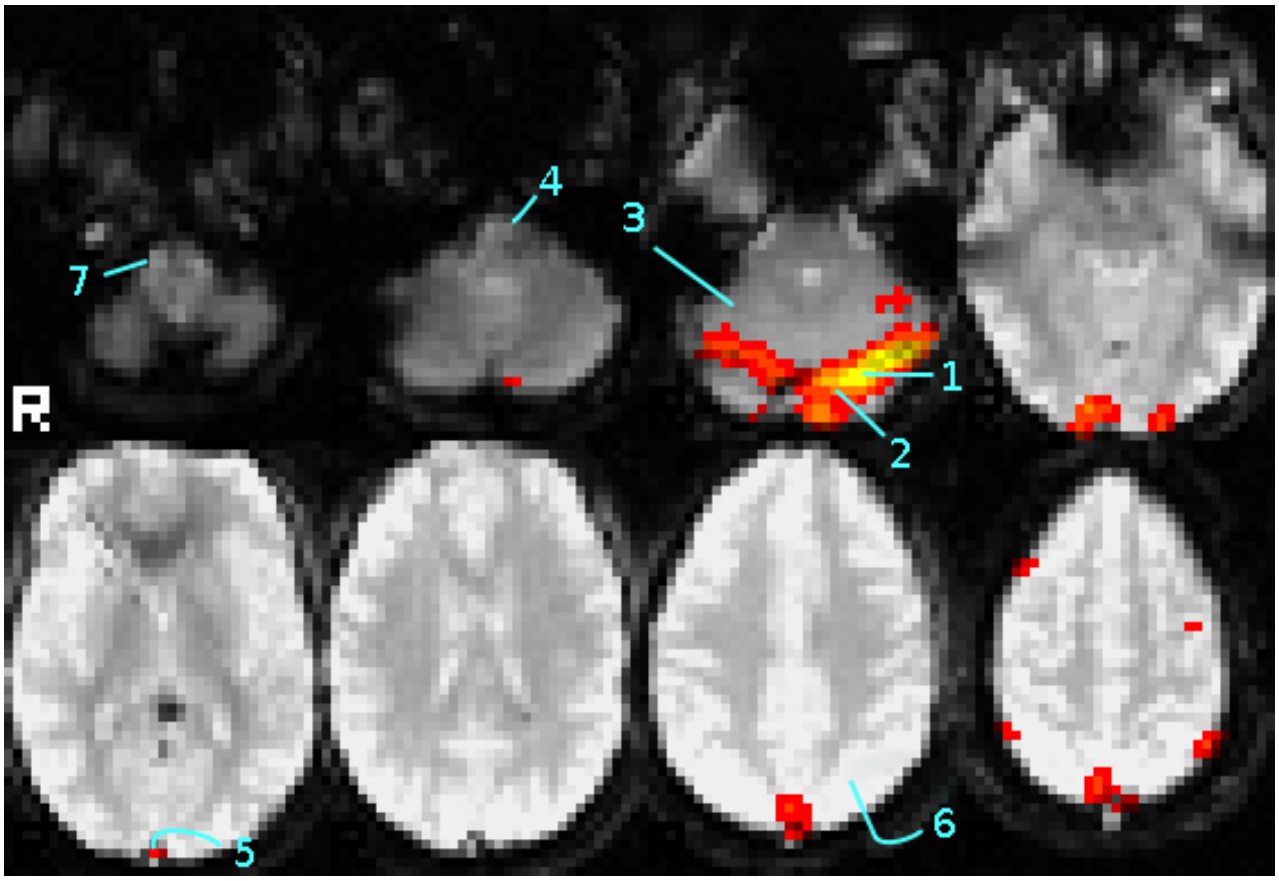
and the timing of the double flashes were:

0.131, 6.825, 20.475, 27.300, 35.831, 52.763, 54.469, 59.588, 86.756, 91.875, 107.231, 108.938, 112.350, 114.056, 115.763, 119.175, 126.000, 134.400, 136.106, 141.225, 144.638, 156.581, 159.994, 166.819, 171.806, 175.219, 185.456, 188.869, 202.519, 212.756, 217.744, 221.156, 227.981, 229.688, 233.100, 243.338, 245.044, 246.750, 250.163, 261.975, 270.506, 272.213, 280.744, 282.450, 289.275, 296.100, 301.088, 304.500

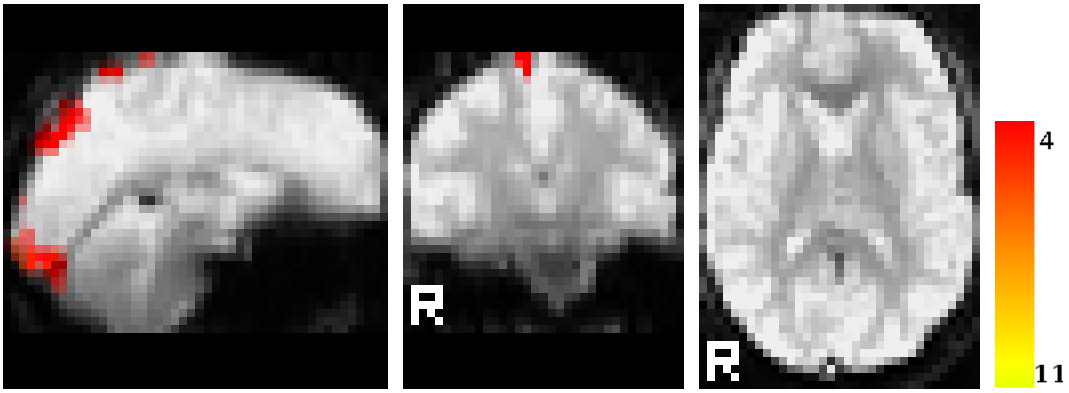
After dropping the first 9 volumes, each of the remaining volumes was co-registered with the new first volume. At this point the SPM method diverges from the experimental method. For the particle filter, detrending was then applied as discussed in Section 4.2.2 and the resulting data was processed with the particle filter. To generate the SPM output in this section, the co-registered data was spatially smoothed, and then filtered with an adaptive cut off (which is built into the SPM analysis). For the SPM analysis, the Canonical HRF was used and model time derivatives were included although not in the contrast vector. All other settings used in SPM8 were left at the default for analyzing fMRI data.

6.2 Results

The t-values from SPM8 are shown in Figure 6.1 (threshold of 4), and the results from the particle filter are shown in Figure 6.2 and Figure 6.3. Note that the scales for all three images are different, because the metrics are different. SPM measures using t-tests to determine the likelihood of a false positive. Figure 6.2 uses simple normalized residuals, meaning that lower indicates less error. Figure 6.3 measures in terms of mutual dependence between the measured signal and the estimated signal; thus higher indicates a better fit. The particle filter data shows a large number of false positives, however application of a threshold of 0.85 on the residual map removes these false positives. Similarly, in the Mutual Information map, the false positives may be eliminated by upping the threshold to 0.15. However, just because the results disagree with SPM does not necessarily mean they are false positives. SPM operates on smoothed data (8 mm x 8 mm x 8 mm), so there are certainly active areas that have been missed because of the smoothing. Note that throughout this section yellow indicates a better fit.

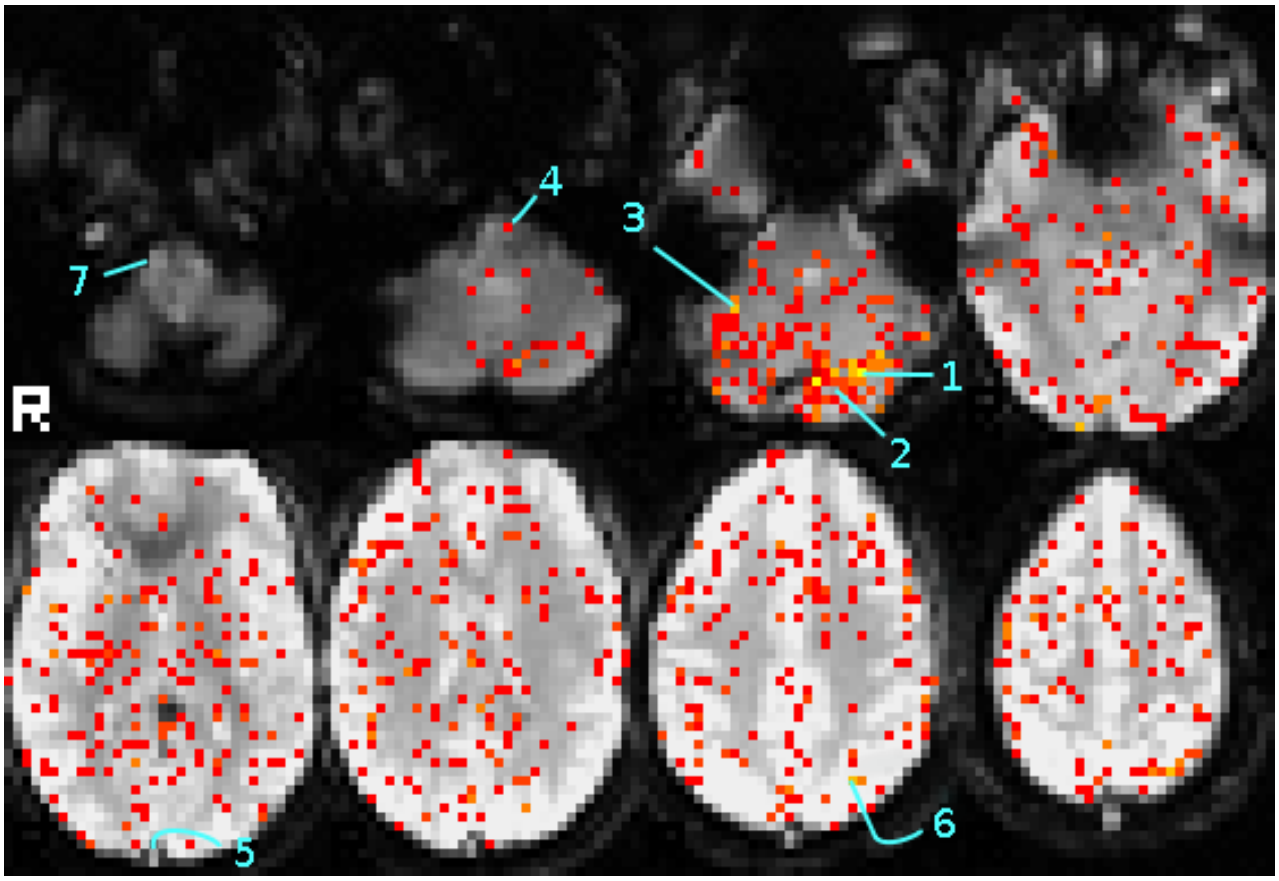


(a) Axial Slices

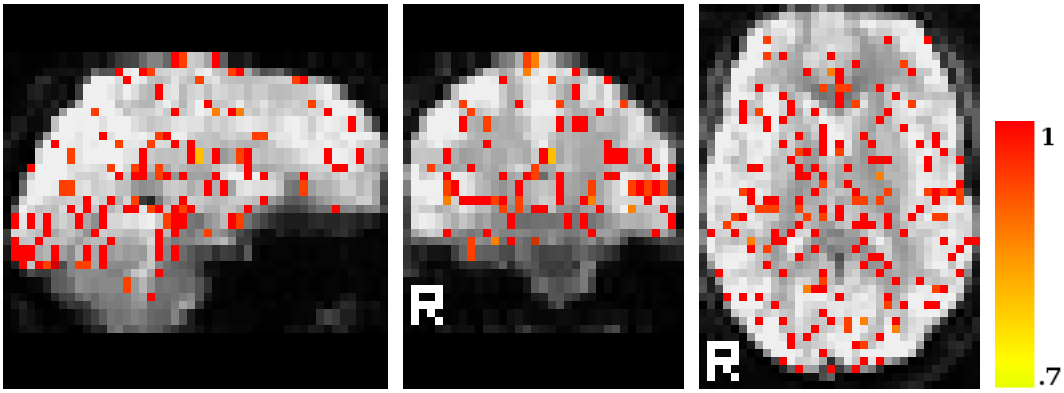


(b) Sagittal Slice (c) Coronal Slice (d) Axial Slice

Figure 6.1: SPM results. Units of activation are in Student's t-scores; higher indicates higher assurance that the signal cannot have occurred through noise alone.

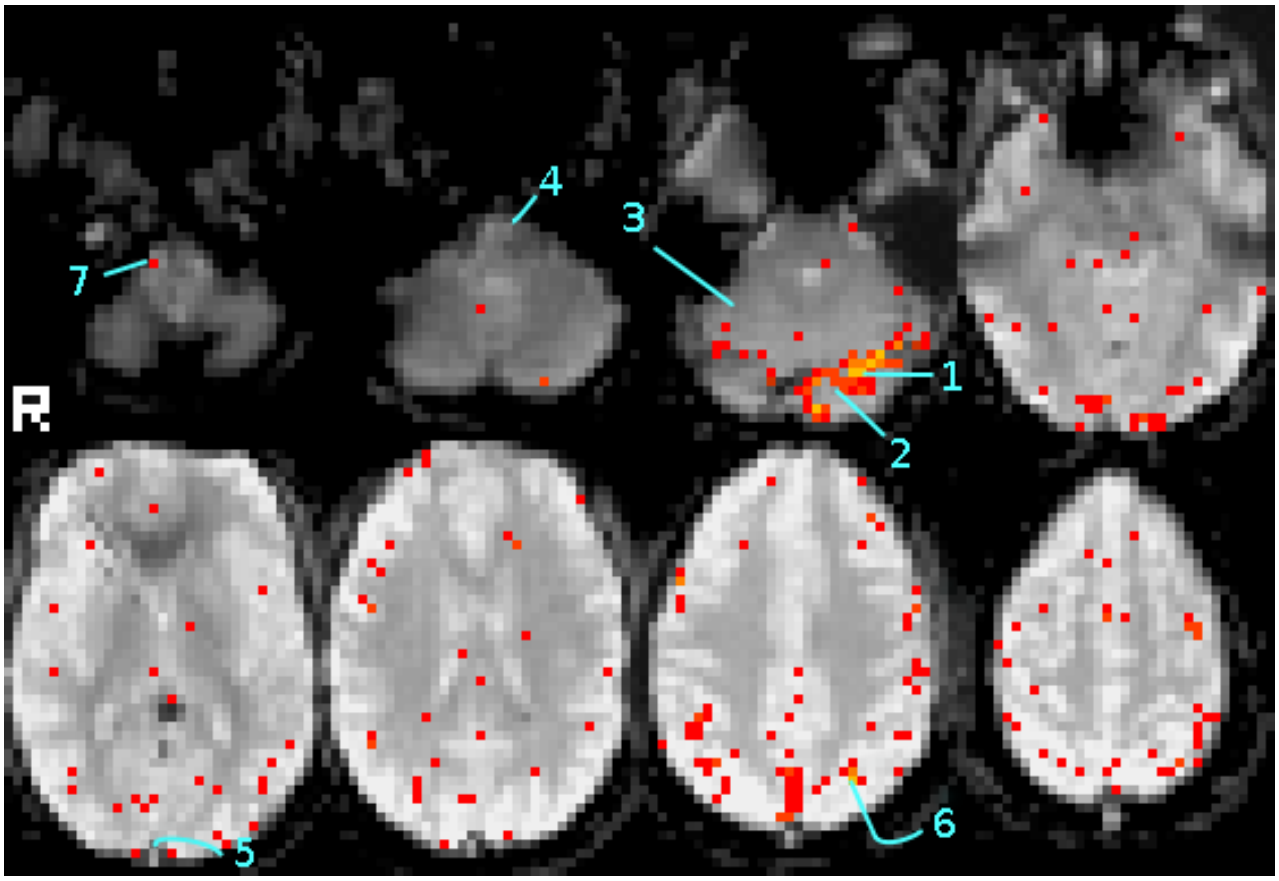


(a) Axial Slices

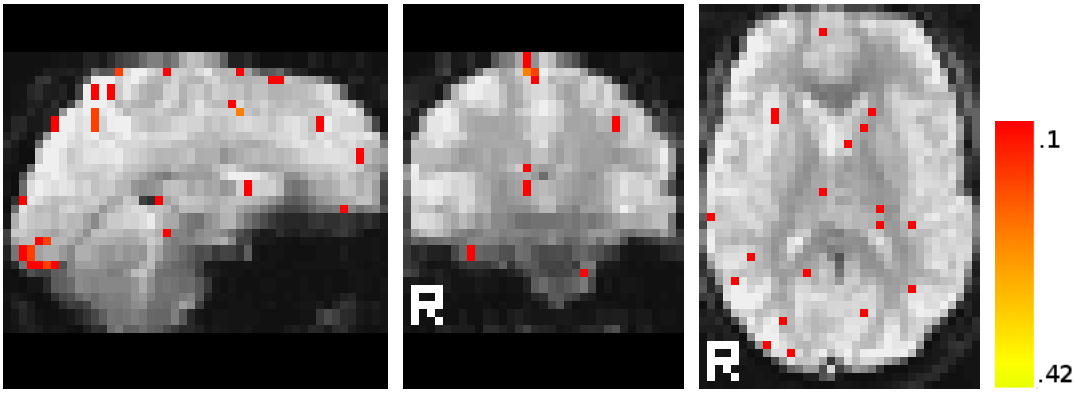


(b) Sagittal Slice (c) Coronal Slice (d) Axial Slice

Figure 6.2: Particle Filter results measured in normalized error. Units of match is normalized residual where the lowest (best) levels shown are 0.7 and the highest error shown (threshold) is 1. Sagittal, coronal and axial slices are shwon in 6.2(b).

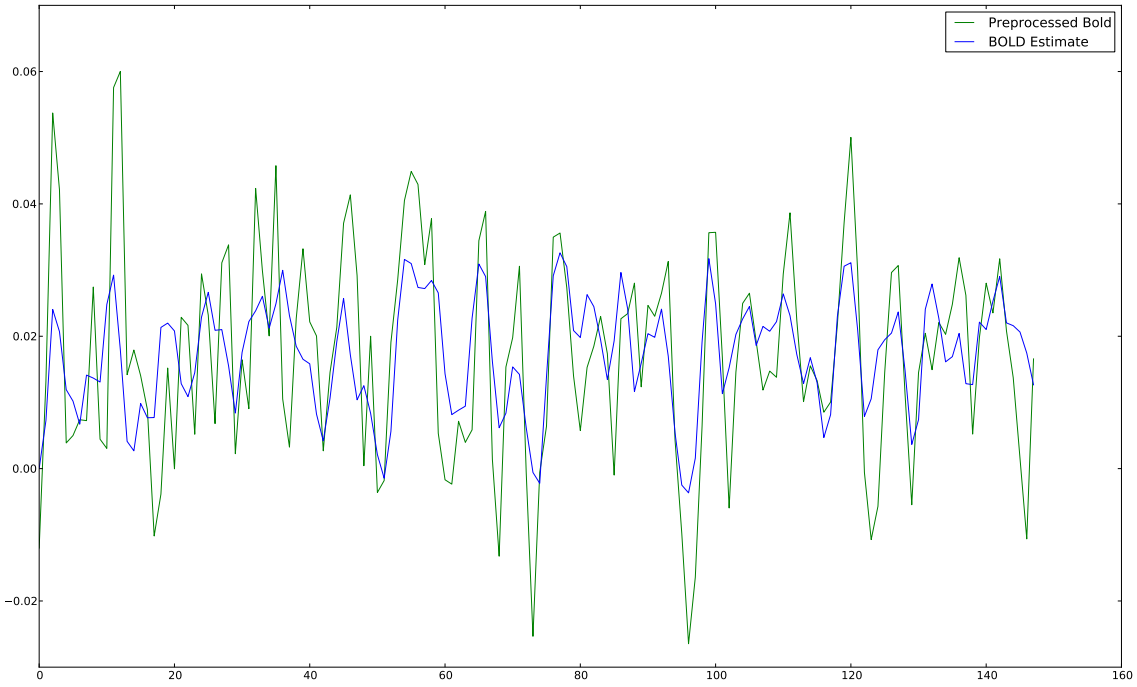


(a) Axial Slices

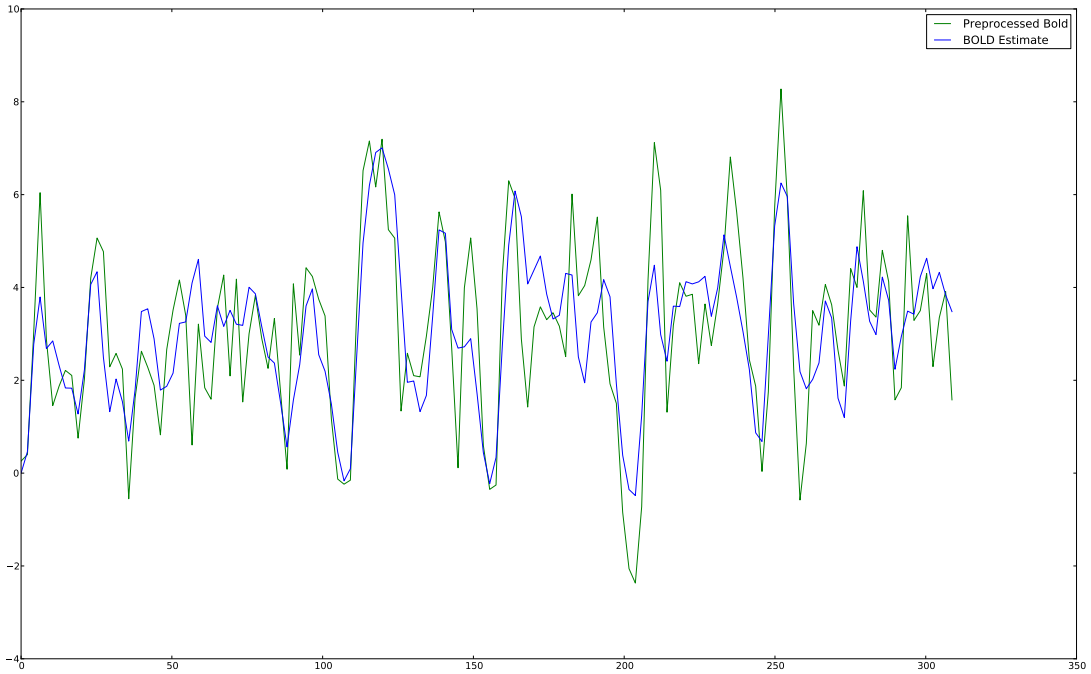


(b) Sagittal Slice (c) Coronal Slice (d) Axial Slice

Figure 6.3: Particle Filter results measured in MI. Units of match is bits (standard for base-2 MI). The highest (best) levels are 0.42 and the worst shown (threshold) is 0.1.

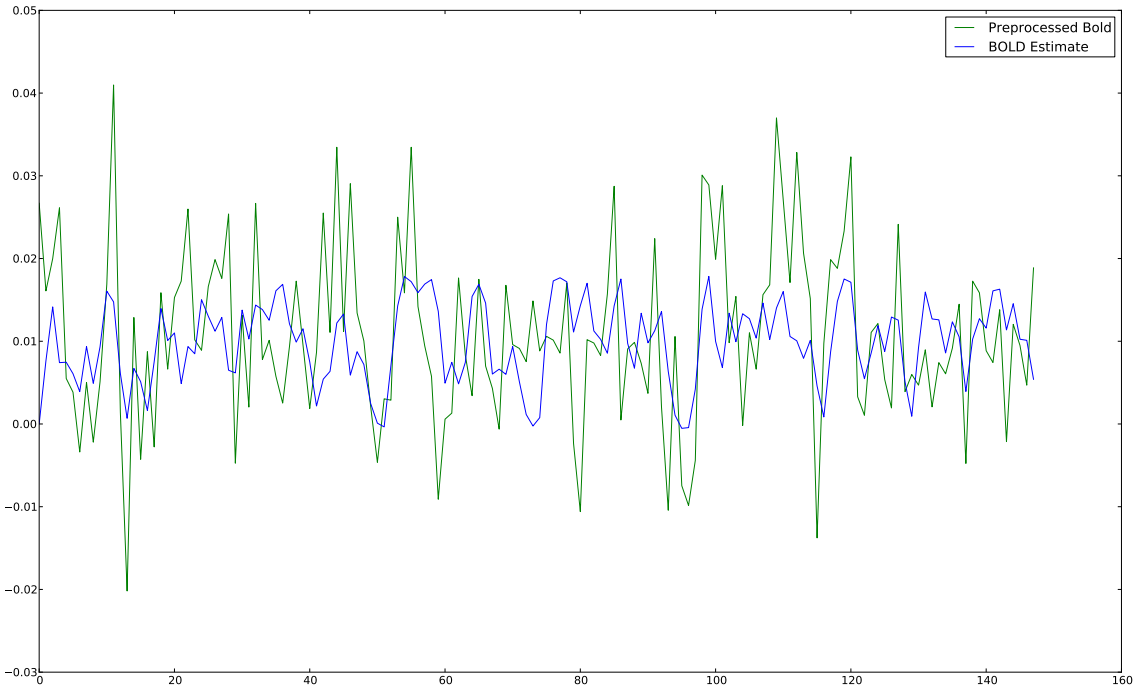


(a) Particle Filter

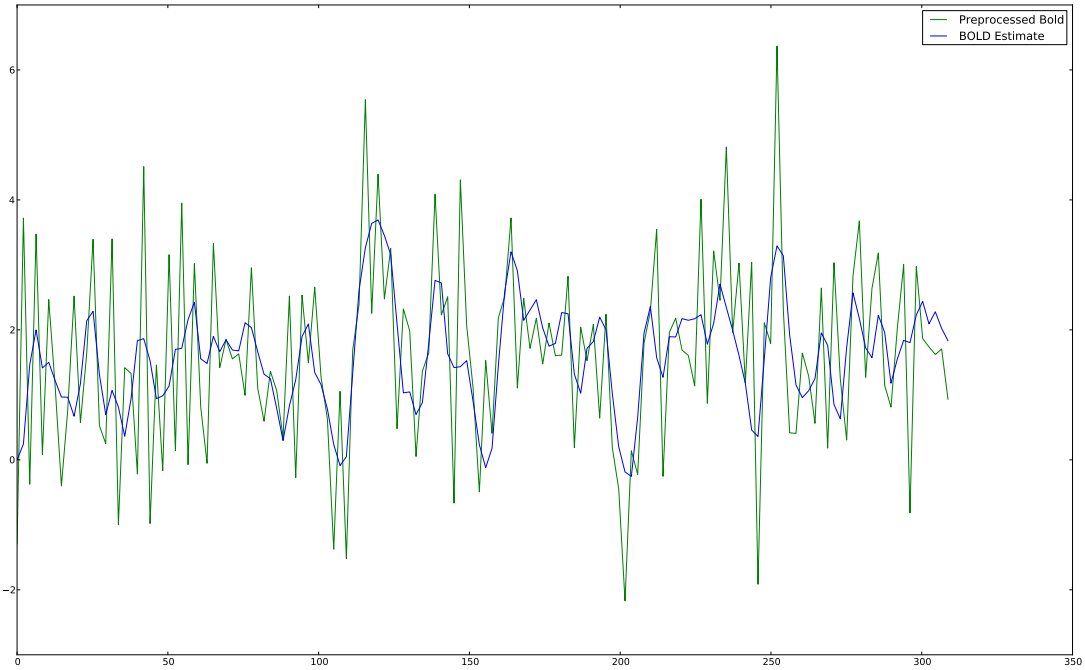


(b) SPM

Figure 6.4: Section 1, Estimated vs. Actual BOLD response. t-Score: 10.71, MI: 0.33, Residual: 0.72.

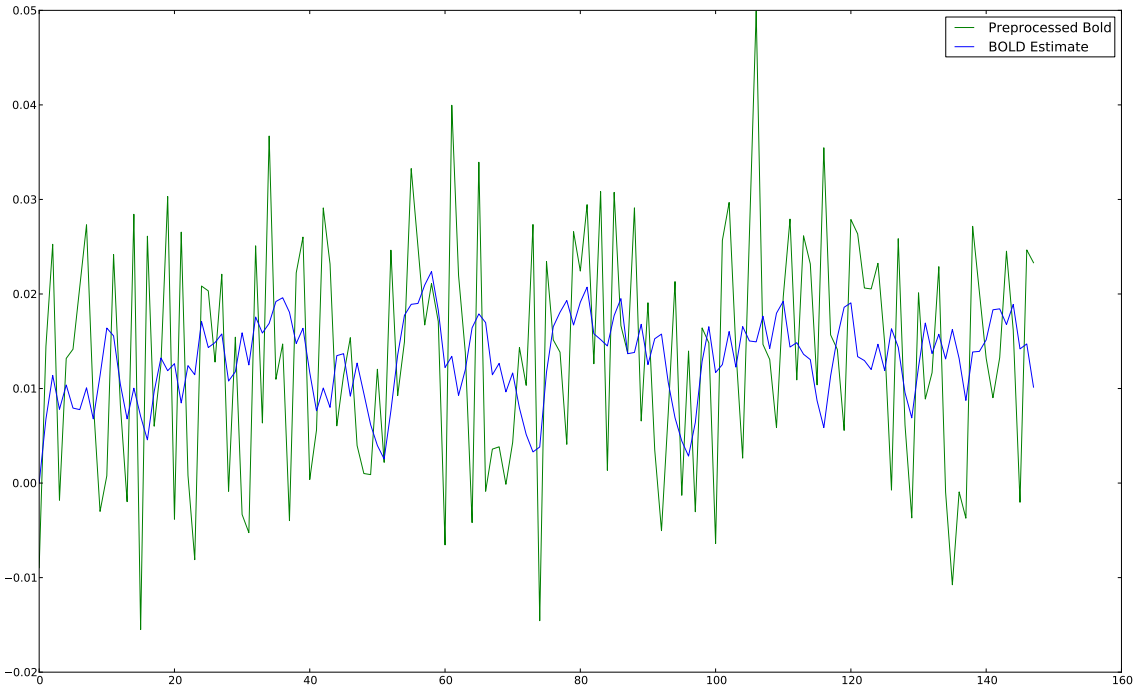


(a) Particle Filter

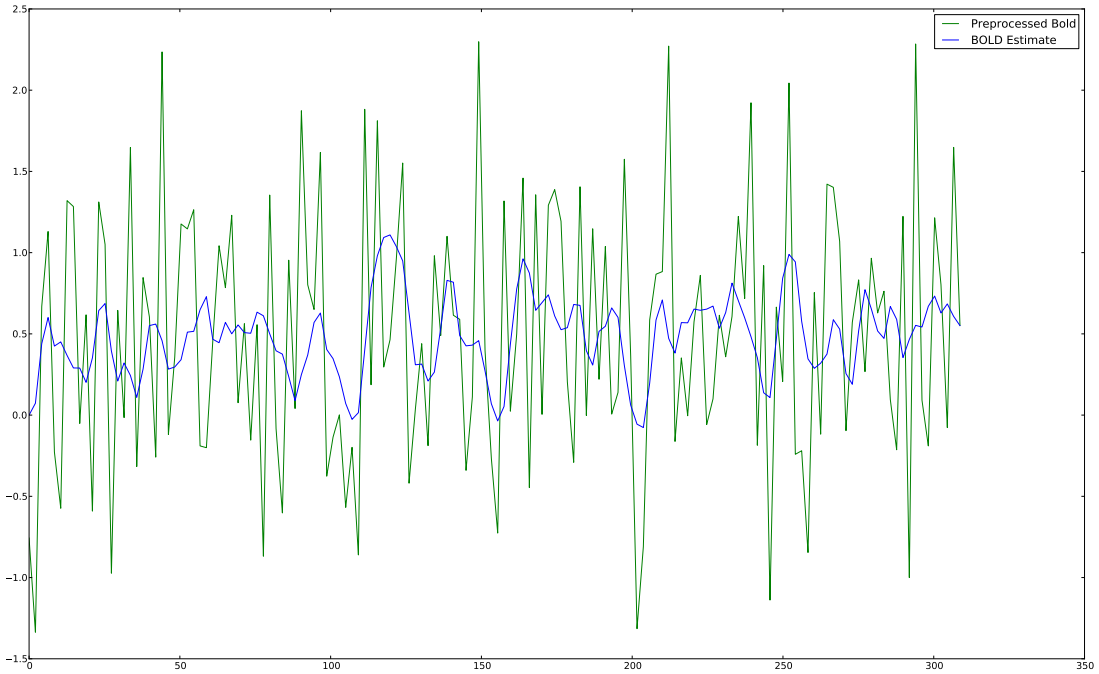


(b) SPM

Figure 6.5: Section 2, Estimated vs. Actual BOLD response. t-Score: 6.97, MI: 0.04, Residual: 1.02.

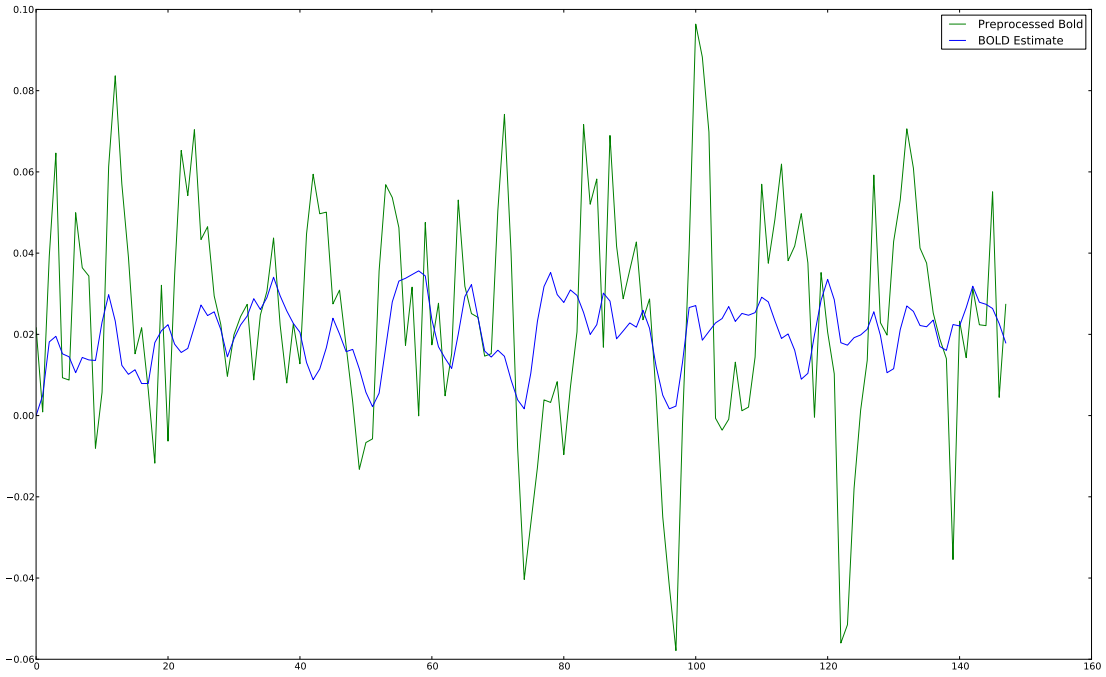


(a) Particle Filter

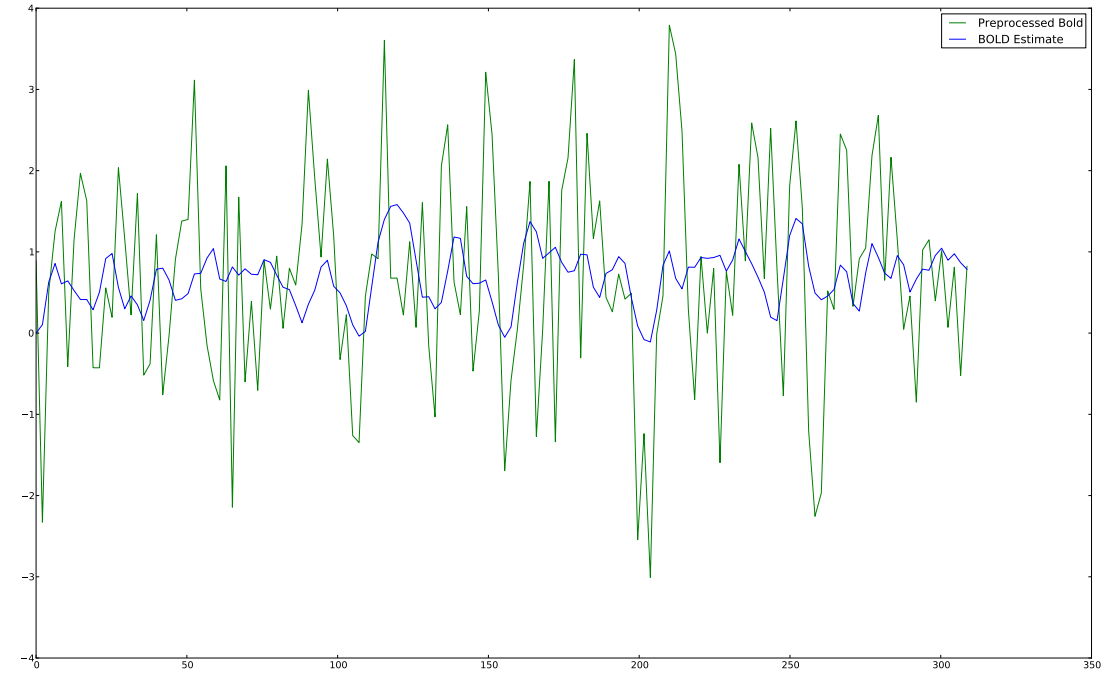


(b) SPM

Figure 6.6: Section 3, Estimated vs. Actual BOLD response. t-Score: 2.85, MI: -0.03, Residual: 0.81.

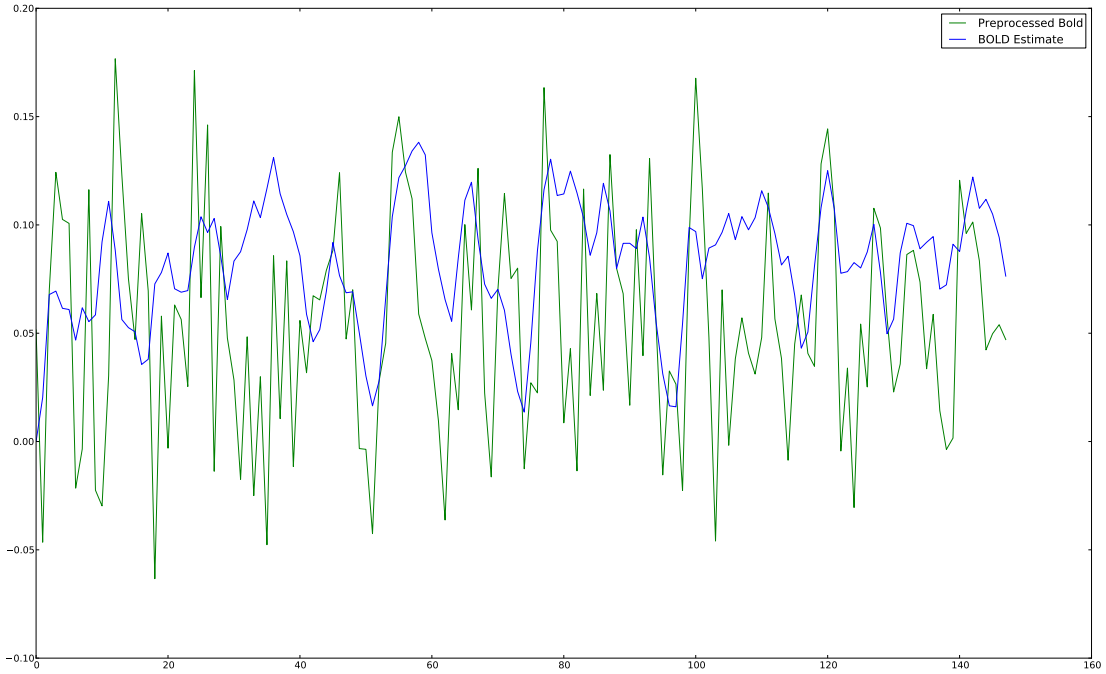


(a) Particle Filter

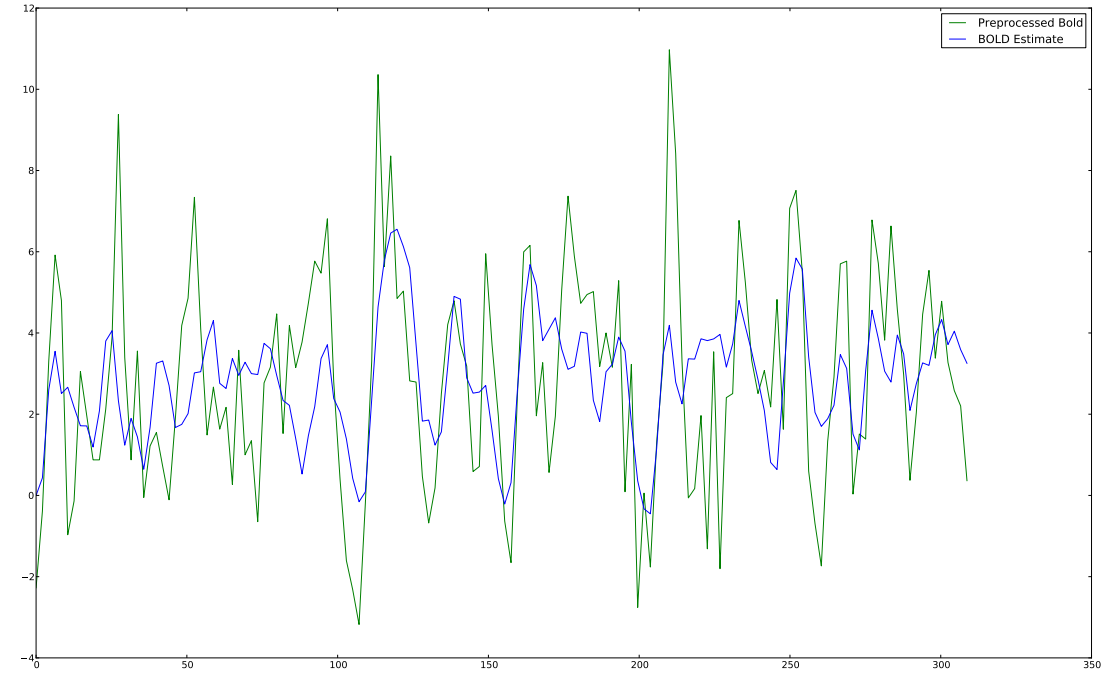


(b) SPM

Figure 6.7: Section 4, Estimated vs. Actual BOLD response. t-Score: 0.50, MI: 0.06, Residual: 0.95.

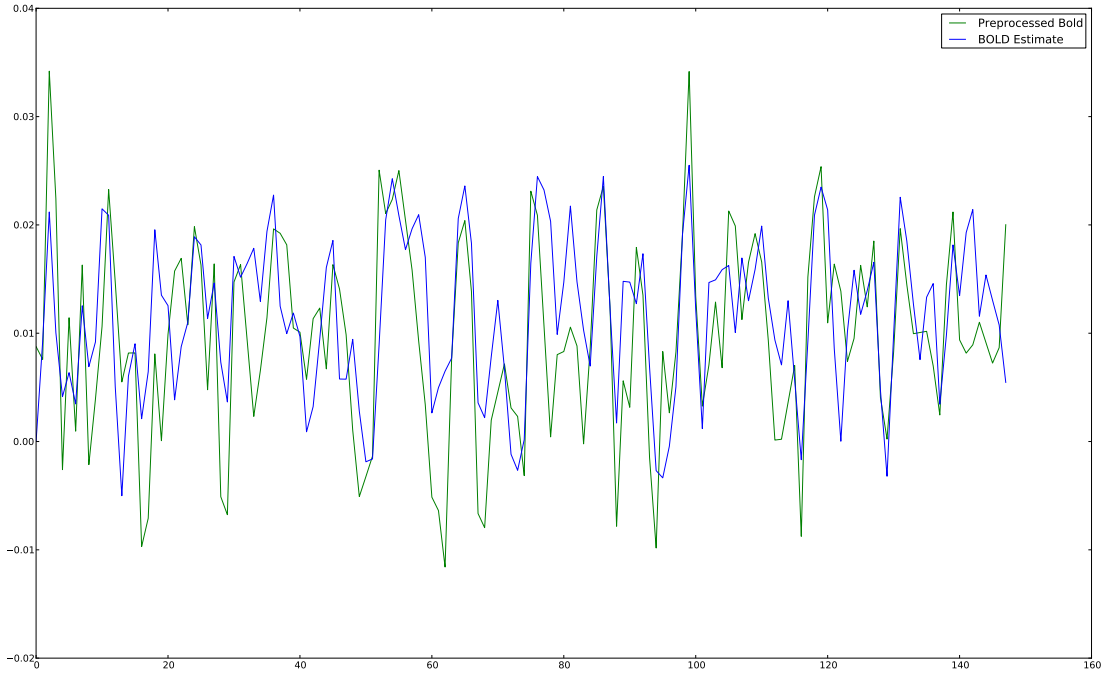


(a) Particle Filter

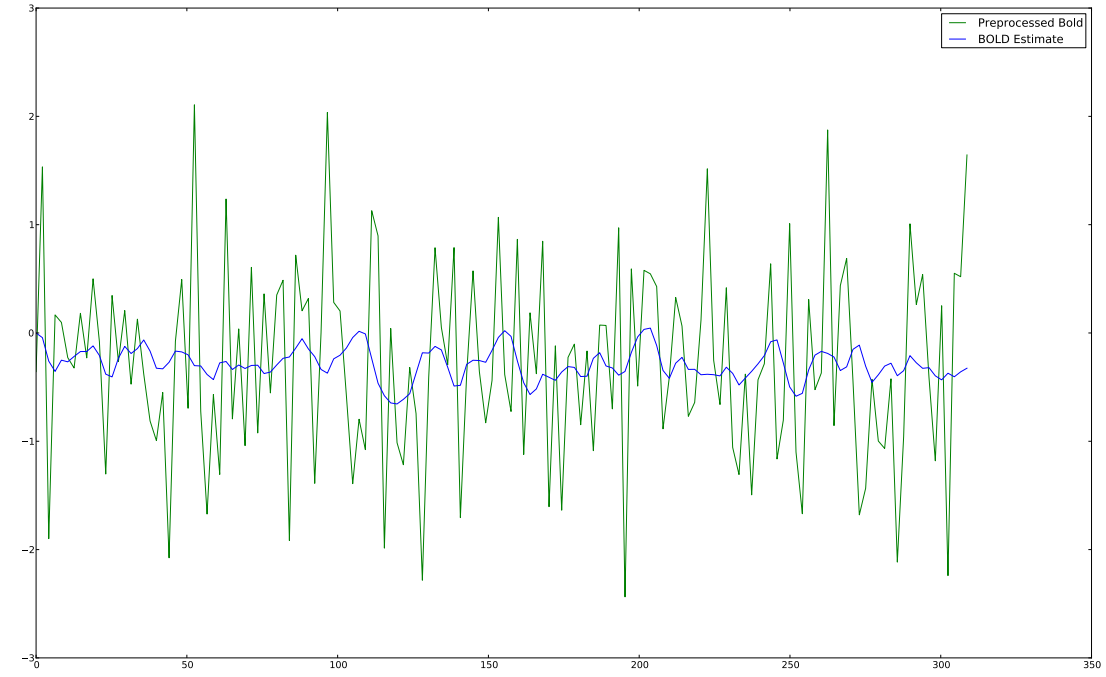


(b) SPM

Figure 6.8: Section 5, Estimated vs. Actual BOLD Response. t-Score: 4.17, MI: 0.02, Residual: 1.14.

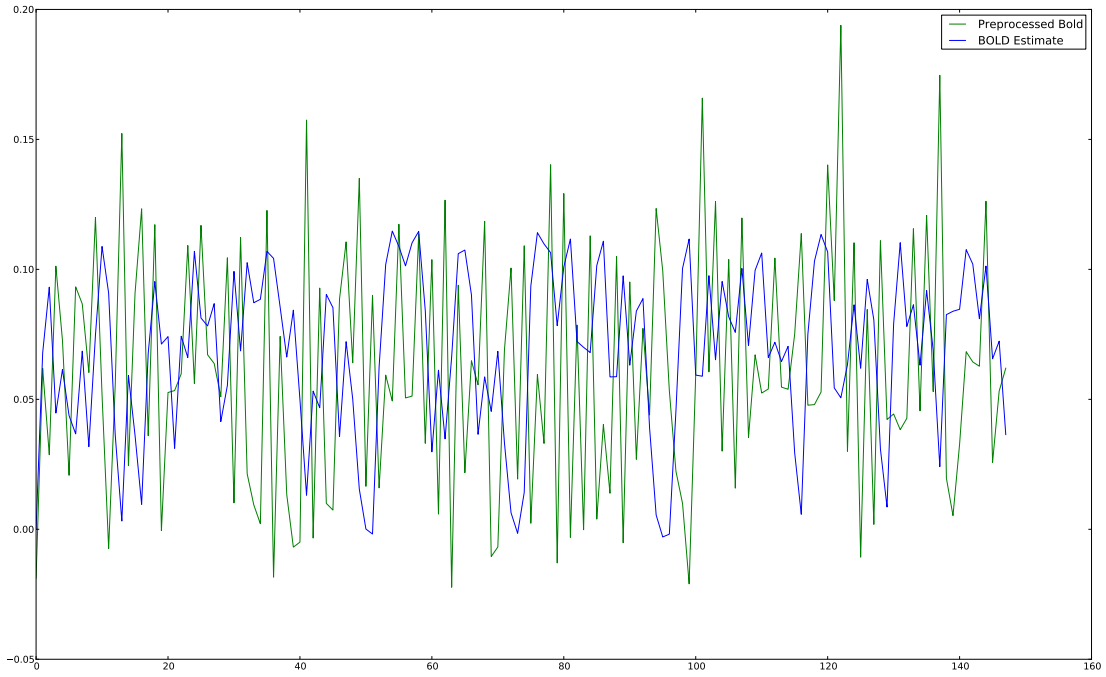


(a) Particle Filter



(b) SPM

Figure 6.9: Section 6, Estimated vs. Actual BOLD Response. t-Score: 2.49, MI: .34, Residual: 0.78.



(a) Particle Filter

Figure 6.10: Section 7, Estimated vs. Actual BOLD Response. t-Score: 1.32, MI: 0.11, Residual: 1.10.

Several voxels were chosen for further investigation from Figure 6.2 and Figure 6.1.

Section 1 (Figure 6.4) had a very high t-score, high MI and a low error. Visual inspection makes it obvious that this voxel fits the BOLD model. A region such as this would be called strong activation by neurologists. The reason for the cleaner looking signal in SPM is the additional Gaussian spatial smoothing applied in preprocessing. This has allowed for an extremely good fit in SPM, although the particle filter handles the noise well.

Section 2 is a difficult voxel to assess (Figure 6.5). While certain peaks seem to correlate, overall the particle filter input signal is extremely noisy. This would appear to be a false positive on SPM's part because of the Gaussian smoothing. This voxel is not being driven directly by the input, although it could be gated or driven through intermediate region.

Section 3 and 4 are both examples of nearly pure-noise signals, at least from the standpoint of the given stimulus (Figure 6.6, Figure 6.7). These two voxels are false positives by the normalized residual. Note that in both cases the MI was extremely low, meaning that according to the MI these were not good fits. Both section 3 and 4 have significant peaks, raising the question of whether they may be active, but not correlated directly with the input.

It is questionable whether the fifth voxel should be considered inside the brain; however, like section 2, SPM deemed it active (Figure 6.8). Also like section 2, the most likely cause of the problem is the smoothing applied for SPM. There are certain peaks that certainly match the BOLD

signal, yet that is only a small part of the whole. Take for instance the measurements between 30 and 40 seconds. In that interval there is a significant spike in the BOLD signal, yet the actual measured signal declines. This is an indication that the BOLD signal is not directly being driven by the input. The SPM fit is not particularly good either, with the t-score barely above the threshold of 4.

Section 6 (Figure 6.9) is a clear example of a false negative from SPM. The BOLD model fit is extremely good, yet the SPM signal looks completely like noise. One possible reason for this is that the low peak signal (< 0.03), made it more susceptible to being smoothed out.

Section 7 is an ambiguous time series. While the thresholds applied here are empirical, this voxel is on the borderline for both MI and the normalized residual. This case is reminiscent of the pure-noise tests performed in the previous chapter, and is a perfect example of the false positive problem. The signal oscillates far faster than the BOLD estimate, yet somehow the MI reaches $.1052$ and the normalized residual is just below 1.1.

6.3 Parameter Estimates

Although the parameters are not uniquely identifiable by a single time-series, that does not mean the estimates are useless . The parameters still contain useful information about the system. Additionally, as an aggregate they form a distribution of the feasible parameters parameters for a particular patient. Therefore Figure 6.11 through Figure 6.17 contain the parameter maps for the system and Figure 6.18 is a histogram across all voxels for which the MI was greater than 0.15. As before, the threshold is not scientifically derived, yet in tests this threshold provided a decent balance to remove most of the questionably active voxels in the system.

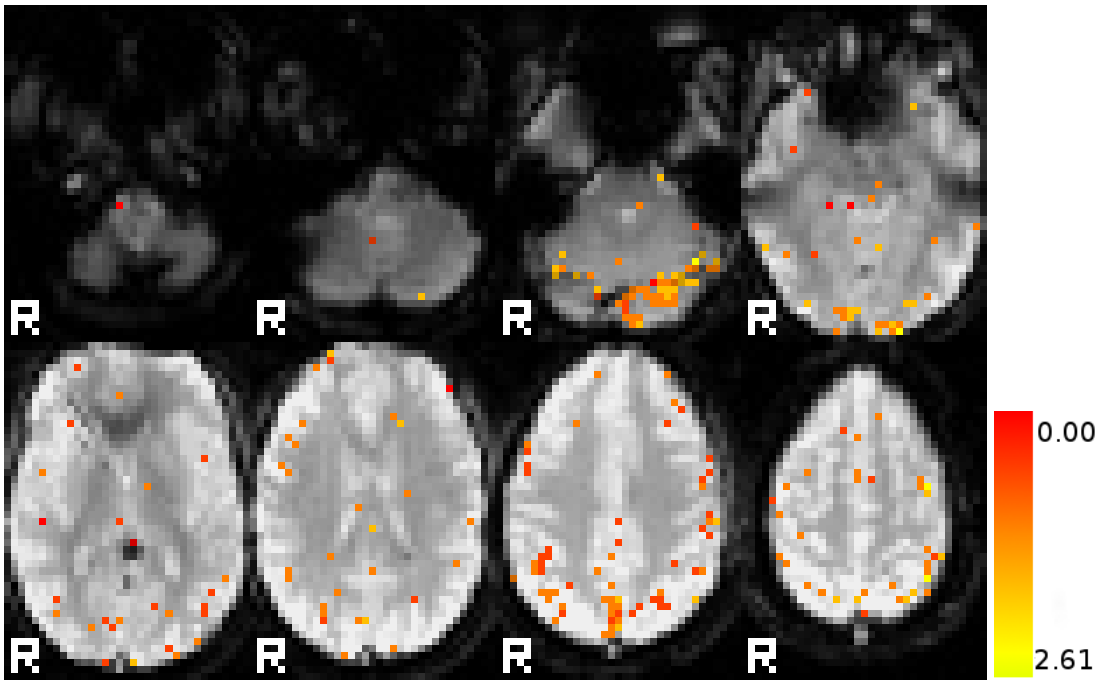


Figure 6.11: Regional τ_0 Estimates

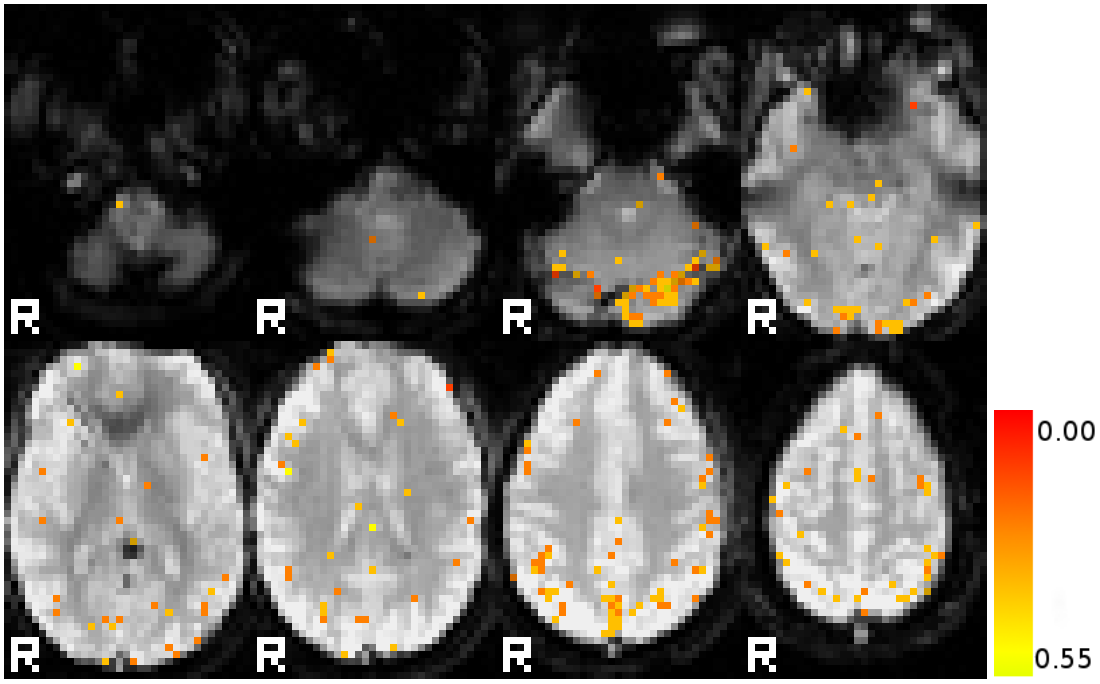


Figure 6.12: Regional α Estimates

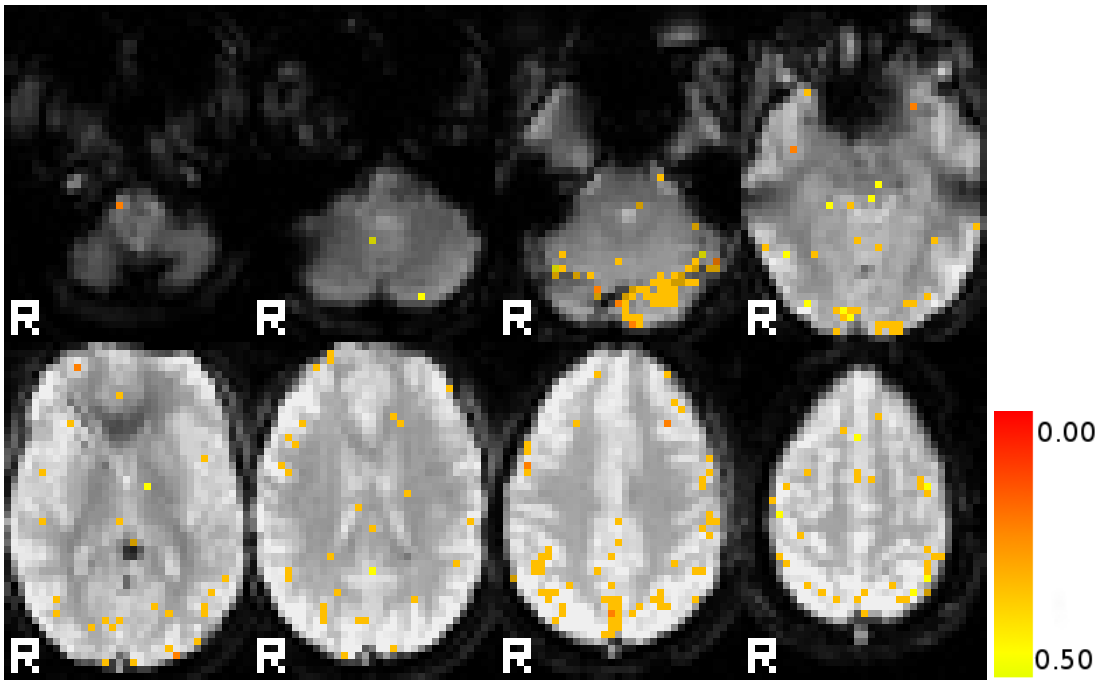


Figure 6.13: Regional E_0 Estimates

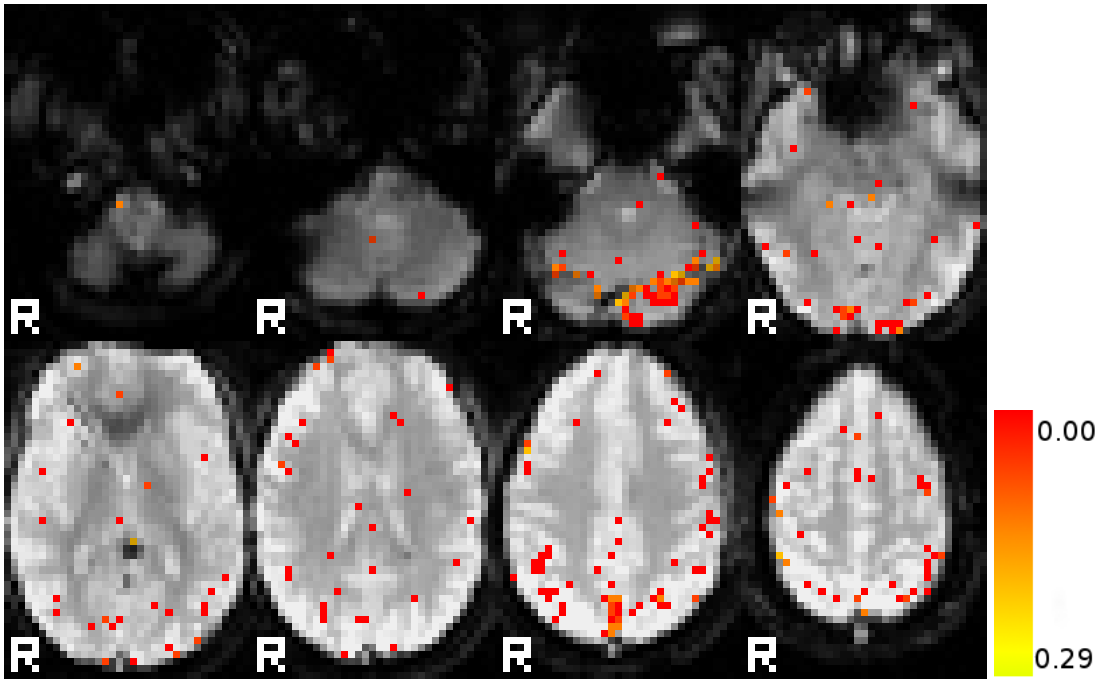


Figure 6.14: Regional V_0 Estimates

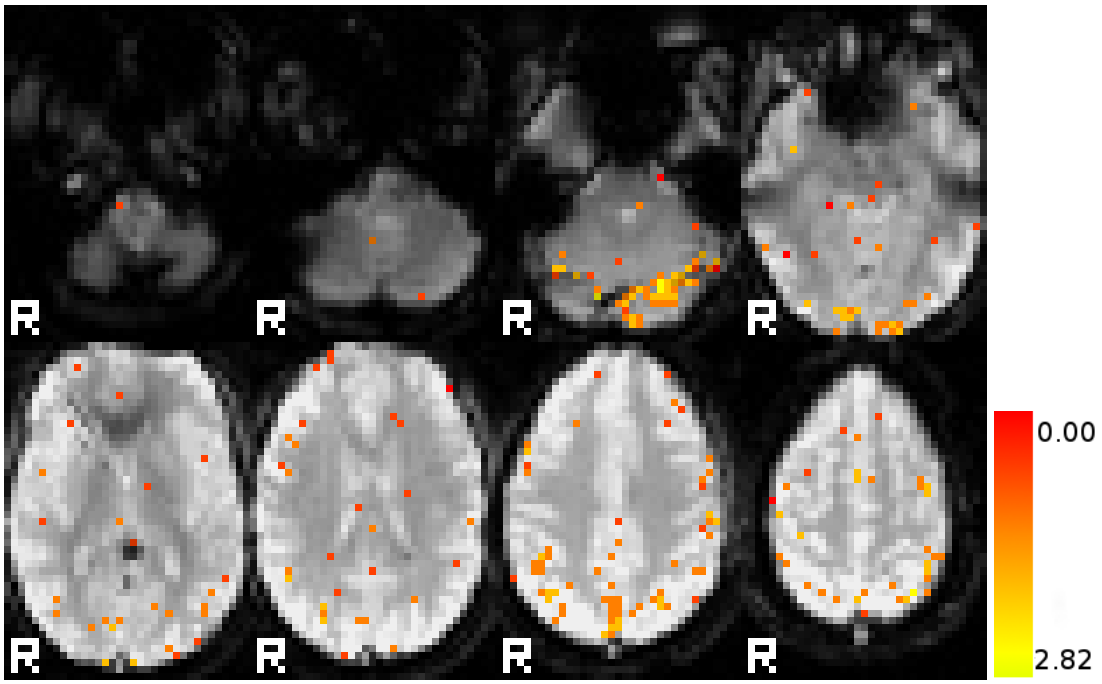


Figure 6.15: Regional τ_f Estimates

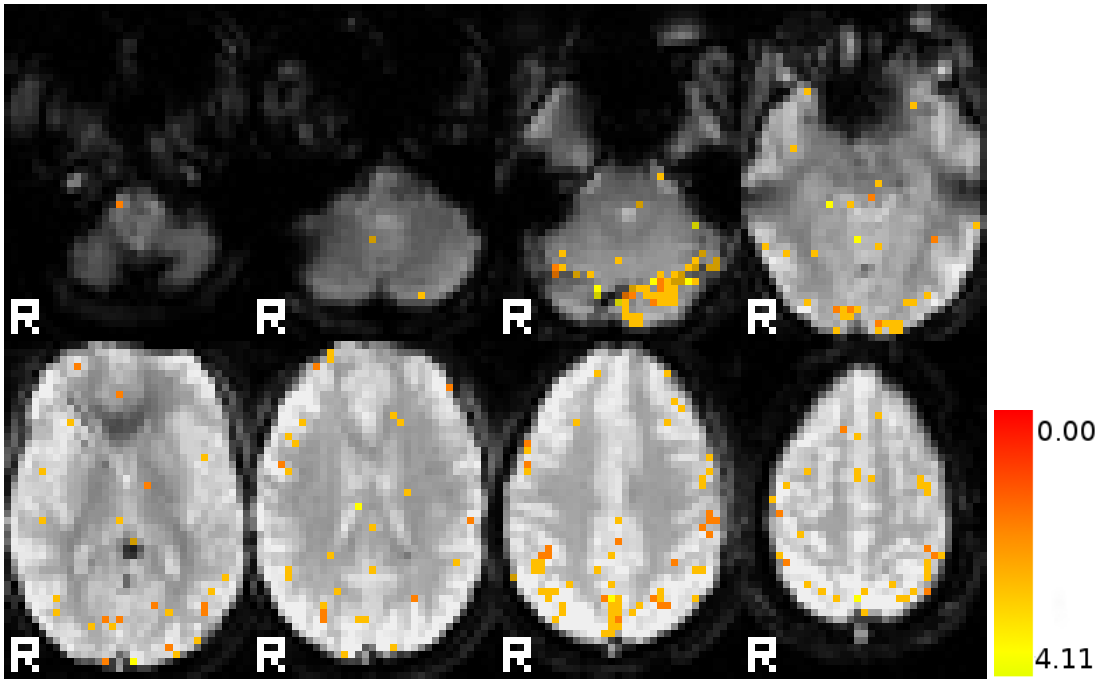


Figure 6.16: Regional τ_s Estimates

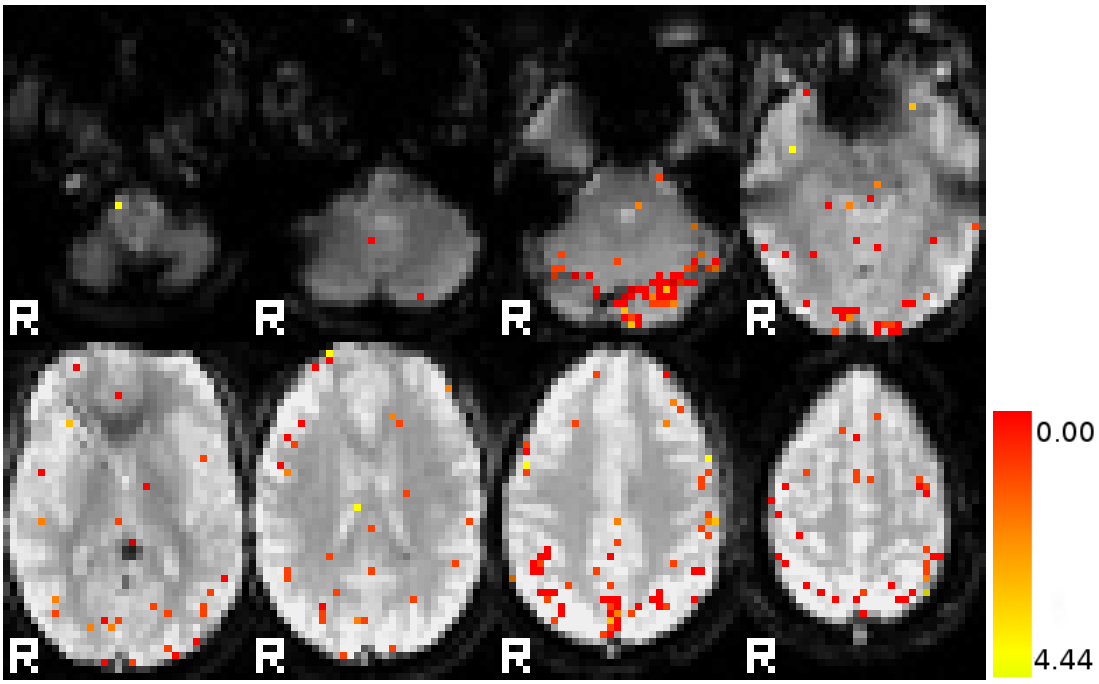


Figure 6.17: Regional ϵ Estimates

Regions with poor fit cannot have reliable parameter estimates because the input did not meaningfully correlate input with output, which is why a threshold was applied. MI was chosen over the residual because the maps had more coherency, and had less error in slice simulations (Figure 5.20 vs. 5.19(c)).

The histograms show variance higher than the prior, indicating that parameters are migrating quite a bit during particle filter convergence. It can be said that all of these regions had relatively accurate BOLD measurements, and yet the final parameter estimates vary widely.

6.4 Discussion

The activation maps generated by the particle filter are very similar to those produced by SPM. This indicates that the particle filter is successfully estimating the BOLD output, in spite of the lack of spatial smoothing. Comparing the Mutual Information metric with the normalized residual, MI gives much better clusters, although that doesn't necessarily mean it is better. Although many of the differences between SPM and the particle filter were driven by differences in preprocessing, it is worth noting that the preprocessing applied by SPM is necessary for SPM to give decent results. Without spatial smoothing SPM is unable to cope with additional noise, and gives significant numbers of false positives. Although the parameter maps have some spatial coherence, the histograms of the parameters still indicate that the parameters are under-determined.

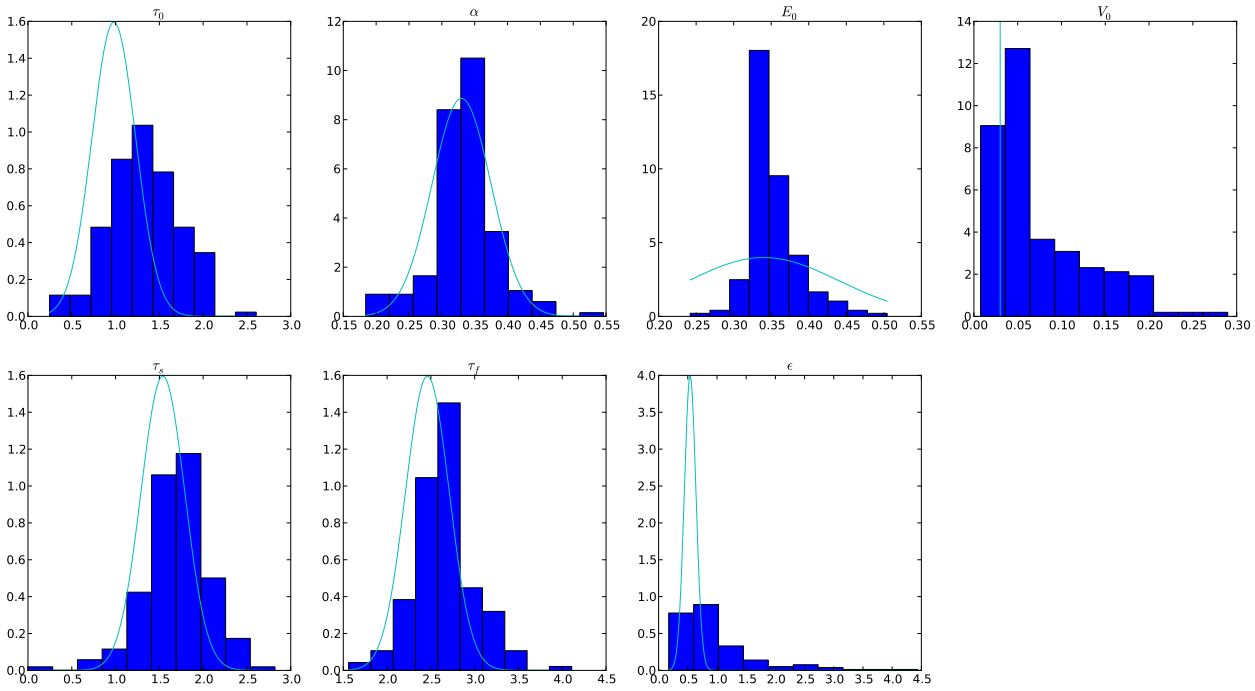


Figure 6.18: Histogram of parameters in active regions (MI $> .15$) overlayed with the parameter estimates from Friston et al. [12]

Chapter 7

Discussion and Conclusions

7.1 Overview of Results

7.1.1 Parameter Estimation

There are two possible causes for the large variance in parameter estimates. First, it could be that the particle filter is incapable of learning the model. However, given the high quality of BOLD estimates, this is almost certainly not the case. The other possible explanation is that the given stimulus is not sufficiently varied to bring out all the properties of the system. While it is possible that this is the case, it is unlikely that any stimulus sequence could differentiate all the parameters. It is also extremely interesting that the histograms of parameters' mean approach the width of the prior, even when the particle filter converges properly to far thinner distributions. It is possible the prior distribution is far wider than it needs to be. Finally, the fact that Section 5.1.1, which used randomly spaced impulses, still had a high correlation further indicates that the parameters are ill-defined.

Thus there is little doubt that the parameters of the BOLD model are under-constrained. While unsurprising given the sensitivity analysis by Deneux et al. [9], it is a notable conclusion. Other methods that calculate a point estimate of the parameters, such as least squares or Kalman filters (which uses the first two moments) are limited in their capacity to estimate such parameters. While estimates of the BOLD signal may still be correct, the underlying parameters and state variables cannot be described without using a joint probability distribution function. In this sense, particle filters represent an important step forward in BOLD parameter estimation. Representing uncertainty with a mean and variance is insufficient; so using a particle filter or Bayesian estimate of the posterior is not simply an enhancement, but a necessary precaution.

Because of the ill-defined nature of the parameters, the results of Friston et al. [12] may have been wider than true parameter distribution . Future works that attempt to calculate parameters of the BOLD model should be run multiple times to ensure consistency, which most likely cannot

be attained. This result is particularly troublesome given the number of papers that use those parameters for a prior distribution.

As the results in Chapter 6 show, the heat maps, especially those of mutual information, closely resembled the results of SPM. While the activation tests were more sensitive, there were some additional false positives, though the problem is difficult to quantify. Regions with M.I. above 0.15 consistently fit the fMRI data well, and simulations showed that the particle filter performed extremely well in the face of significant noise. In sum, the estimated BOLD output remained consistent despite large swings in parameters.

7.1.2 Particle Filter Review

The Particle Filter algorithm was originally designed for on-line parameter estimation. For this reason, there is no guarantee of optimality or even convergence for finite measurements. However, for the BOLD nonlinear Ordinary Differential Equation (ODE) this is less of a concern than it might first appear to be. For this particular problem there can be no guarantee of a global minimum, and although other techniques guarantee a local minimum, tests show that the particle filter did converge relatively quickly Section 5.1.1.

One difficulty with the use of a particle filter when given a finite number of measurements is finding a good weight function $P(y_k|x_k)$. This is more important for a finite number of measurements because $P(y_k|x_k)$ needs to converge in finite time. In spite of this potential problem, Section 5.1.1 managed to converge in less than 500 seconds. Given sufficient measurements it is better to let the algorithm take longer to converge, because the convergence will be less prone to particle deprivation. The particle filter takes longer to run than Volterra approximation method from Section 2.2.1; however, it is free from the uncertainty of whether a quadratic approximation is sufficient for the BOLD model.

The particle filter also has advantages over other estimation procedures discussed in Chapter 2. The most important advantage is that it provides an estimate of the posterior probability, rather than a single estimate. While researchers often want a simple estimate of parameters, such an estimate is impossible with this particular model. The fact that the final distribution does not need to conform to a parametric distribution is also advantageous, given the nonlinearities in the system. While the particle filter took a day to calculate for full brain calculations, its speed was sufficient on a quad core machine to perform real time calculations of small regions (approximate run time .27 seconds per voxel-measurement). Today it would be possible to perform real time analysis of 10 voxels on an average quad core. The algorithm also scales well and does not require burdensome amounts of memory (approximately 11 megabytes).

7.2 Conclusions

This work has demonstrated the use of the particle filter to learn parameters of the BOLD model. Since the inception of the BOLD equations, many attempts have been made use fMRI data to to learn the model parameters. These attempts have typically either been extremely slow or relied on extensive assumptions. While the particle filter method is not quick, 40 seconds to analyze each voxel is well within the capabilities of the typical research lab. Previous attempts have also treated the problem as if there were a single solution.

One significant finding of this work, that would not be clear without calculating a full posterior distribution, is the interplay between the parameters. The results of simulations clearly demonstrate that identifying a single set of parameters is not possible with this model. Although sensitivity tests in Deneux et al. [9] certainly hinted at this, the current work clearly demonstrates this fact. Therefore, any single estimate of parameters is insufficient for analysis. As such, identification of the BOLD model that does not treat the parameters as distributions will not be able to overcome the inherent ill-posed nature of the parameters. Besides the Unscented Kalman Filter, this is the only approach that accomplishes this task without repeatedly calculating the parameters to build a distribution. The Unscented Kalman Filter though, is restricted to the Gaussian distribution which limits its ability to estimate the posterior.

The primary reason this method has not been used before is the high dimensionality of the system. In Murray, 2008 the idea of learning the parameters is floated as the best method, yet learning 7 parameters with a Monte Carlo method was deemed intractable [27]. The concern was that seven parameters creates a prohibitively large search space, which would require too many particles to learn properly. To overcome this difficulty, instead of starting the algorithm with 1000 particles, the initial number of particles was set to 16,000. This meant that when the search space was the largest, the number of particles was sufficiently dense to represent the prior distribution. The result is a particle filter algorithm that spends computing resources only where it is really needed.

The particle filter is a Bayesian non-parametric algorithm that, given enough measurements will approach the true probability distribution of the parameters. The conclusion that the parameters are not uniquely identifiable is disappointing in light of the plethora of research papers that have attempted to learn them. However, the fact that the parameters are not identifiable further demonstrates the need to estimate a probability distribution rather than a single parameter estimate. At the same time, the output of the particle filter is still able to give good estimates of the BOLD output, and is not dependent on heavy Gaussian smoothing. Therefore, all the current experimental paradigms to determine regional activity are still viable. To borrow a computer term, the particle filter algorithm is backward compatible with the current methods. Indeed, the particle filter excelled at determining activation in very low SNR simulations (Section 5.2). Not only that, but the particle filter algorithm identified areas of activation that were completely missed by SPM (Figure 6.9).

The physiological plausibility is also of great value to future research. Unlike traditional methods, use of the BOLD model allows the incorporation of outside, physiological knowledge in regression.

If, for instance, it becomes possible to measure v in vivo that data could instantly be added to the model to improve regression. Data can also be retrieved from the model. BOLD parameter distributions could for the first time be used as diagnosis tools. In the past the fact that the BOLD model is ill-posed would have prevented such research. However, now that a full distribution is available, comparing parameter estimates is actually feasible.

Concluding, the particle filter provides comparable performance with conventional tests, but also provides a platform for a wide range of future uses beyond what was possible with the methods that have been used before.

7.3 Future Work

7.3.1 Algorithm Improvements

There are a few areas where future research may improve upon the current work. The first is modifying the prior distribution. The prior distribution used in this work is listed in Table 4.1, and is based on the findings of Friston et al. [12]. Unfortunately, that result depended on a quadratic approximation (Volterra Series) which has not been extensively tested (at least not in a published work). This is in addition to the fact that the parameters are ill-defined, and thus the parameters found in Friston et al. [12] are most likely far too wide. The prior is also based completely on the BOLD output, which is why it is good at generating estimates of the BOLD signal, but weak at estimating parameters. The best solution would be to have in-vivo estimates of the actual parameters, although this is unlikely to happen. Instead, better estimates could be found using population studies with additional measurements as discussed in Section 7.3.2. Better knowledge of the prior distribution would make it possible to decrease variance of certain parameters, reducing the mobility of the parameters.

Another major area for improvement is removing drift. Several variations of spline detrending were tested, but only the simple case spline described in Section 4.2.2 gave consistent results. One potentially viable method that could be utilized, given the right experimental design, is detrending based on intervals of low activity. This has the advantage that it wouldn't require an arbitrary constant to be added to the pre-processed signal for the BOLD model to fit properly. It would also minimizes the actual removed. The disadvantage of this approach is that it could hide long fall times by normalizing them out. It also requires periodic breaks in the stimulus, which could limit persistent excitation.

Linearizing is another possible method of dealing with drift. This would use the slope between measurements for fitting, rather than the direct value. This has the advantage of not requiring detrending and thus gives nearly raw data to the particle filter for processing. The effectiveness of this method depends directly on the type of stimulus. In a test with rapid impulse stimuli, this could be extremely effective because the DC signal carries less information; conversely prolonged flat levels make this less effective. In tests I found that large drift/low white noise type signals

performed much better with a linearization approach. For the stimulus sequence used in Chapter 6, the results tended to be worse than using spline detrending.

As discussed in Section 4.3.4, choosing a weighting function is difficult, and the optimal solution varies based on the input. Automatically estimating measurement error could improve the quality of the particle filter results. Although I made some attempts to do this, finding a generic, consistent solution is complex, and often depends on the experimental design. Part of the problem is that SNR varies greatly across the brain, and there is no way to actively measure noise without also knowing the underlying signal. One possible solution is having the particle filter automatically set the weight based on the total particle weight, or on the prevalence of resampling.

7.3.2 Experimental Design Improvements

One definite way of improving the results of the particle filter is through additional measurements. While increasing the sample rate of fMRI scanners may not be possible, simultaneous measurements of volume and flow is possible, albeit at 9T in a cat [17]. Just one of those measurements would be extremely powerful when incorporated into the BOLD model. By adding another measurement, the variance in the parameter estimates would significantly drop. The fact that such a measurement would be closer to the true stimulus would make it all the more useful. A more conventional method of adding measurements is to simply perform longer fMRI tests. Although this wouldn't be groundbreaking, it would certainly increase the ability of the particle filter to develop the posterior distribution. It may also be possible to link fMRI tests together by using the parameter estimates from previously acquired data as the prior for later experiments.

Given the non-linearities in the system, differences in stimuli could make a large difference in the observability of parameters. The mentality for using a physiologically based nonlinear model for BOLD signal is to model those nonlinearities. It is logical then that certain nonlinear parameters may not be identifiable when the input is primarily an impulse response. It has been reported that short responses are disproportionately large in fMRI data [26, 9]. Therefore, a wide range of activation hold time may shed further light on the parameters' distribution as well as the validity of the BOLD model.

7.3.3 Future Applications

There are a number of advantages to the particle filter approach presented here. In the past, fMRI data has been analyzed strictly for determining correlation between a stimulus and response. With this new method the correlation is simply a means to determining the joint posterior distribution of the parameters. While only regions that correlate with the input will be calculable, this method exploits that correlation to constrain the prior distribution of the parameters. The resulting distribution, while difficult to visualize because of the high-dimensionality, could nevertheless be correlated with neural pathologies. In spite of the fact that the parameters are under-determined,

the final distribution still reduces the uncertainty of the parameters. The availability of a full posterior distribution opens up many avenues for further inquiry, for instance to compare normal vs. symptomatic populations. This would be especially useful in patients whose symptoms are caused by functional differences in the brain, rather than structural differences. This would effectively be a way to differentiate the way the brains operate.

Because of limitations present in every imaging modality, it is becoming increasingly clear that combining data from multiple sources will be necessary to push neurology forward. In order to do so however, the output of each source needs to fully represent what information that source can provide. Combining the sources using Bayesian statistics is promising yet often difficult because full probability distributions are hard to come by. However, in this case, the particle filter provides a full posterior which is extremely versatile. Therefore, future works will easily be able to plug in data from multiple sources if they all output Bayesian posteriors. For instance, if two different modalities have calculated the probability distribution of neural efficiency, those two beliefs (distributions) may be combined into one conditional belief for a new conditional probability of neural efficiency.

An advantageous aspect of using a physiological model such as this, is that it permits estimates of otherwise hidden parameters. In particular the BOLD model gives an estimate of the value of the flow inducing signal, s. Having this value available opens up new avenues for determining inter-regional dependencies. The values of s could serve as a proxy for activation in that region and thus could be used to drive other inputs regions' $u(t)$. Thus, the particle filter could be re-run with the time-series of a particular voxel's s value as an additional stimulus. In this way, it could be possible to determine chains of events. This is just one possible benefit of being able to determine the time course of the hidden state variables present in the BOLD equations; the potential benefits of being able to determine this information is limitless. Of course the quality of these estimates will continue to improve with the model priors.

Bibliography

[1] Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

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

[3] Rasmus M. Birn, Ziad S. Saad, and Peter A. Bandettini. Spatial heterogeneity of the nonlinear dynamics in the FMRI BOLD response. *NeuroImage*, 14(4):817–26, October 2001.

[4] Jerrold T. Bushberg, J. Anthony Seibert, Edwin M. Jr. Leidholdt, and John M. Boone. *The Essential Physics of Medical Imaging*. Lippincott Williams & Wilkins, Philadelphia, PA, 2 edition, 2002.

[5] R. B. Buxton, Kamil Uludag, David J. Dubowitz, and Thomas Lui. Modeling the hemodynamic response to brain activation. *NeuroImage*, 23 Suppl 1:S220–33, 2004.

[6] R. B. Buxton, E. C. Wong, and L. R. Frank. Dynamics of blood flow and oxygenation changes during brain activation: the balloon model. *Magnetic Resonance in Medicine*, 39:855–864, 1998.

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

[8] R. Christopher DeCharms, Fumiko Maeda, Gary H. Glover, David Ludlow, John M. Pauly, Deepak Soneji, John D. E. Gabrieli, and Sean C. Mackey. Control over brain activation and pain learned by using real-time functional MRI. *Proceedings of the National Academy of Sciences of the United States of America*, 102(51):18626–31, December 2005.

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

[10] Manus J. Donahue, Robert D. Stevens, Michiel de Boorder, James J. Pekar, Jeroen Hendrikse, and Peter C. M. van Zijl. 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*, 29(1):176–85, 2009.

[11] Jens Frahm, Jürgen Baudewig, Kai Kallenberg, Andreas Kastrup, K. Dietmar Merboldt, and Peter Dechent. The post-stimulation undershoot in BOLD fMRI of human brain is not caused by elevated cerebral blood volume. *NeuroImage*, 40(2):473–81, April 2008.

[12] K. J. Friston. Bayesian estimation of dynamical systems: an application to fMRI. *NeuroImage*, 16:513–530, 2002.

[13] K. J. Friston, D. E. Glaser, R. N. A. Henson, S. Kiebel, C. Phillips, and J. Ashburner. Classical and Bayesian inference in neuroimaging: applications. *NeuroImage*, 16(2):484–512, June 2002.

[14] K. J. Friston, A. Mechelli, R. Turner, and C. J. Price. Nonlinear Responses in fMRI: the Balloon Model, Volterra kernels, and other Hemodynamics. *NeuroImage*, 12:466–477, 2000.

[15] Daniel A. Handwerker, John M. Ollinger, and Mark D'Esposito. Variation of BOLD hemodynamic responses across subjects and brain regions and their effects on statistical analyses. *NeuroImage*, 21(4):1639–51, April 2004.

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

[17] Zhenghui Hu, Xiaohu Zhao, Huafeng Liu, and Pengcheng Shi. Nonlinear Analysis of the BOLD Signal. *EURASIP Journal on Advances in Signal Processing*, 2009:1–14, 2009.

[18] Markus Hürzeler, Hans R. Künsch, Markus Hurzeler, and Hans R. Kunsch. Monte Carlo Approximations for General State-Space Models. *Journal of Computational and Graphical Statistics*, 7(2):175, June 1998.

[19] B. Iglewicz. Robust scale estimators and confidence intervals for location. *Understanding robust and exploratory data analysis*, pages 405–431, 1983.

[20] Leigh A. Johnston, Eugene Duff, Iven Mareels, and Gary F. Egan. Nonlinear estimation of the BOLD signal. *NeuroImage*, 40(2):504–14, 2007.

[21] M. E. Ladd, D. Timmann, and E. R. Gizewski. Ultra-High-Field MRI in Neurology. *Aktuelle Neurologie*, 37(5):219–230, June 2010.

[22] Arthur K. Liu, John W. Belliveau, and Anders M. Dale. Spatiotemporal imaging of human brain activity using functional MRI constrained magnetoencephalography data: Monte Carlo simulations. *Proceedings of the National Academy of Sciences of the United States of America*, 95(15):8945–50, July 1998.

[23] Hanzhang Lu, Xavier Golay, James J. Pekar, and Peter C. M. van Zijl. Sustained Poststimulus Elevation in Cerebral Oxygen Utilization after Vascular Recovery. *Journal of Cerebral Blood Flow and Metabolism*, 24:764–770, 2004.

[24] Joseph B. Mandeville, John J. A. Marota, C. Ayata, Michael A. Moskowitz, Robert M. Weisskoff, and Bruce R. Rosen. MRI Measurement of the Temporal Evolution of Relative CMRO 2 During Rat Forepaw Stimulation. *Magnetic Resonance in Medicine*, 951:944–951, 1999.

[25] Joseph B. Mandeville, John J. A. Marota, C. Ayata, Greg Zaharchuk, Michael A. Moskowitz, Bruce R. Rosen, and Robert M. Weisskoff. Evidence of a cerebrovascular postarteriole Windkessel with delayed compliance. *Journal of Cerebral Blood Flow and Metabolism*, 19(6):679–689, 1999.

[26] K. L. Miller, W. M. Luh, T. T. Liu, A. Martinez, T. Obata, E. C. Wong, L. R. Frank, and R. B. Buxton. Nonlinear temporal dynamics of the cerebral blood flow response. *Human Brain Mapping*, 13(1):1–12, May 2001.

[27] Lawrence Murray and Amos Storkey. Continuous Time Particle Filtering for fMRI. *Advances in Neural Information Processing Systems*, 20:1049–1068, 2008.

[28] C. Musso, N. Oudjane, and F. LeGland. Improving regularised particle filters. *Sequential Monte Carlo methods in practice*, 2001.

[29] T. Obata. 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, 2004.

[30] S Ogawa, R S Menon, D W Tank, S Kim, H Merkle, J M Ellermann, and K Ugurbilt. 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, 1993.

[31] Tohru Ozaki. 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, 1994. Kluwer Academic Publishers.

[32] B. A. Poser, P. J. Koopmans, T. Witzel, L. L. Wald, and M. Barth. Three dimensional echo-planar imaging at 7 Tesla. *NeuroImage*, 51(1):261–6, May 2010.

[33] J. Riera, J. Bosch, Okito Yamashita, R. Kawashima, Norihiro Sadato, T. Okada, and T. Ozaki. fMRI activation maps based on the NN-ARx model. *NeuroImage*, 23:680–697, 2004.

[34] Jorge J. Riera, Jobu Watanabe, Iwata Kazuki, Miura Naoki, Eduardo Aubert, Tohru Ozaki, and Ryuta Kawashima. A state-space model of the hemodynamic approach: nonlinear filtering of BOLD signals. *NeuroImage*, 21:547–567, 2004.

[35] Qiang Shen, Hongxia Ren, and Timothy Q Duong. CBF, BOLD, CBV, and CMRO(2) fMRI signal temporal dynamics at 500-msec resolution. *Journal of Magnetic Resonance Imaging*, 27(3):599–606, 2008.

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

[37] Anne M. Smith, Bobbi K. Lewis, Urs E. Ruttimann, Frank Q. Ye, Teresa M. Sinnwell, Yihong Yang, Jeff H. Duyn, and Joseph A. Frank. Investigation of Low Frequency Drift in fMRI Signal. *NeuroImage*, 533(5):526–533, 1999.

[38] Jody Tanabe, David Miller, Jason Tregellas, Robert Freedman, and Francois G. Meyer. Comparison of detrending methods for optimal fMRI preprocessing. *NeuroImage*, 15(4):902–907, 2002.

[39] S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). *cognition*, 96, 2008.

[40] Vasily A. Vakorin, Olga O. Krakovska, Ron Borowsky, and Gordon E. Sarty. Inferring neural activity from BOLD signals through nonlinear optimization. *NeuroImage*, 38(2):248–60, November 2007.

[41] Tor D. Wager, Alberto Vazquez, Luis Hernandez, and Douglas C. Noll. Accounting for nonlinear BOLD effects in fMRI: parameter estimates and a model for prediction in rapid event-related studies. *NeuroImage*, 25(1):206–18, March 2005.

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

[43] Keith J. Worsley, Jonathan E. Taylor, Francesco Tomaiuolo, and Jason Lerch. Unified univariate and multivariate random field theory. *NeuroImage*, 23 Suppl 1:S189–95, 2004.

[44] Essa Yacoub, Kamil Ugurbil, and Noam Harel. The spatial dependence of the poststimulus undershoot as revealed by high-resolution BOLD- and CBV-weighted fMRI. *Journal of Cerebral Blood Flow and Metabolism*, 26:634–644, 2006.

[45] Ying Zheng, David Johnston, Jason Berwick, Danmei Chen, Steve Billings, and John Mayhew. A three-compartment model of the hemodynamic response and oxygen delivery to brain. *NeuroImage*, 28(4):925–39, 2005.

[46] Ying Zheng, John Martindale, David Johnston, Myles Jones, Jason Berwick, and John Mayhew. A model of the hemodynamic response and oxygen delivery to brain. *Neuroimage*, 16:617–637, 2002.