# 1 Introduction

Functional Magnetic Resonance Imaging (FMRI) is a powerful tool in the analysis of neural activity. Despite its rather limited temporal resolution, FMRI is still the best way of measuring neural activity deep in the brain. Whereas other methods of analyzing neural signals can be invasive or difficult to acquire, FMRI is relatively quick and cheap, and its analysis is straight forward. Because of these benefits, FMRI continues to be crucial to the study of human cognition. Despite the fact that FMRI is so widely used, the choice of analysis tools is still relatively limited. Every widely available analysis tool is based on parametric modeling, which requires prior knowledge of the noise distribution. Indeed, the linear methods usually applied are known not to be robust to non-gaussian, non-white noise, which is why so many pre-processing steps are necessary. These limitations cast a long shadow over any experiment that claims to be able to reject the null hypothesis, no matter the P-value. While obviously there is a correlation between the BOLD signal and actual activity, its possible the current parametric tests are over-estimating the actual amount of activity. Its also possible that the current methods underestimate actual activity. In this paper we propose a robust model-based approach to activation detection that makes no assumptions about the underlying noise model.

fact, typical activation detection does little to account for variations in the shape of the signal, but instead is a direct T-test between the measured signal and the expected shape of the signal. Because of these limitations, multiple comparison tests are imperative (ref fish), and active regions must be very large to be detected with any confidence. This paper will introduce a more robust method of analyzing FMRI signals for activation, by means of estimating the BOLD parameters.

## 1.1 FMRI

Before going further, it is necessary to discuss Functional MRI, and what it measures. All MRI works by exciting nuclear spin away from a steady state controlled by a large electromagnet. Different nuclei respond to RF impulses based on their natural frequency of oscillation. Therefore contrast in MRI is based on the aggregate nuclear composition of a given region. Thus, regions with more oxygen will respond differently from regions with proportionately less oxygen. This exact difference is what causes the changes over time in the MR signal in functional MRI. Unfortunately, it takes a significant amount of time for nuclear spin to return to equilibrium. Thus instead of waiting, a type of imaging called Echo Planar Imaging (EPI) is used, wherein a single excitation is used, and then as quickly as possible every voxel (3 Dimensional Pixel) in a particular region is read. Because ”reading” a particular region requires the switching on of a gradient magnetic field, it takes a decent amount of time to perform measurements. Problematically, the signal from the original excitation drops exponentially, thus the signal to noise ratio can be very low even at modest resolutions. Eventually the signal becomes unreadable, and a new excitation must be applied. As more measurements are made, and thus as the spatial resolution increases, the rate at which entire volumes can be acquired drops. Typically volumes can be acquired every two to three seconds, at resolutions of about 64 by 64 by 32, but their signal to noise ratio is often poor. Other imaging modalities such as EEG and MEG are capable of much faster measurements, but are unable to isolate measurement locations so precisely as FMRI. As with all biological systems, noise is much higher than in human designed systems, and yet these biological systems are capable of performing amazing feats.

## 1.2 BOLD Physiology

The physiology that leads to the BOLD model has been well studied over the past decade and a half. While some uncertainty remains (for instance in relation to the BOLD post-stimulus undershoot ¡papers relating to back and forth on post-stimulus undershoot¿) generally the model fits reality very well. As with every piece of tissue in the body, the brain requires oxygen to extract energy from glucose in the blood. This process of removing oxygen from red blood cells sets into motion a chain of events that locally alters the composition of the blood. Inactive neurons can be thought of as a slingshot cocked and ready to fire. As soon as a signal moves up the axon and causes the neuron to fire (and thus changes the state of a membrane at that location), ions quickly move across the altered membrane to compensate for a high charge imbalance between the interior and exterior of the cell. This process, similar to allowing a stretched rubber band to contract leaves the system at a lower energy state. This cannot last though, because the neuron needs to be ready to fire again rather quickly. To return the cell to a ready position the membrane becomes impermeable to the ions again, and then begins pumping ions back into the cell. This process takes a large amount of energy, whereas the actual firing takes very little energy. Thus, after firing, glucose is burned, removing oxygen from the blood and thus causing a dip in the ratio of oxygenated hemoglobin to de-oxygenated hemoglobin. This is the first potentially measurable effect that MRI can see, although it usually lasts fewer than 2 seconds making it difficult for FMRI to catch. As a result of the decreased amount of local oxygen, the capillaries compensate by increasing blood flow to that region. Because, of the quickly increased blood flow into local capillaries, the blood volume increases in addition to the local oxygen content. This leads to a Windkessel effect which further lags the normalization of oxygen content. The effect of all this is an overshoot in the oxygen content above the initial level. After the work of recharging is done, there may be a prolonged undershoot lasting as much as 90 seconds [**?**], though the reason for this is debated [**?**].

A large number of models have been proposed for the BOLD signal with varying amounts of complexity. The simplest model is the so called "Balloon Model" which first proposed the windkessel effect as a factor the BOLD response. The model we will use is the model proposed by Buxton Et. Al. and later used in Riera et. al. [**?**]. The model has four state variables, $s$, $f$, $v$, $q$, representing flow inducing signal, cerebral blood flow, cerebral blood volume and deoxyhemaglobin to hemaglobin ratio, respectively. The state variables change over time, given by the state evolution equations:

$$\dot{s} = \epsilon u - \tag{1}$$

$$\dot{f} = s \tag{2}$$

$$\dot{v} = \tag{3}$$

$$\dot{q} = \tag{4}$$

Additional papers have added such things as ....

All these effects are generally accepted as the cause of the BOLD signal, but FMRI doesn't detect this happening in one neuron, but rather as the aggregate over hundreds or millions of cells. Though local neurons act "together" (i.e. around the same time), the density of neurons, the density of capillaries, and slight differences in activation across a particular voxel can all lead to signal attenuation or noise. A particularly insidious type of noise present in FMRI is a low frequncey drift, characterized by a Weiner process. Though not present in all regions, it is prevalent enough to cause problems [**?**]. It is still not clear where exactly this noise comes although it is possible it is the result of magnets heating up, or some distortion in magnetic fields. It is clear that this drift signal is not due to a true physiological effects however, given its presence in cadavers and phantoms[**?**].

# 2 Current Techniques

## 2.1 Basic Statistical Parametric Mapping

The most basic method of analyzing FMRI data is through a basic T-test between "resting state" and "active state" samples. This is done by taking the average and variance of the inactive period, and the active period separately then treating them both as gaussian distributions. If they are in fact Gaussian distributions, then a basic t-test will give the likelihood that the samples came from the same distribution (the p-value). Of course, this test is fraught with problems; even if the drift mentioned earlier has been removed, there is little reason to believe that the noise is Gaussian, or even stable. Additionally, even if the noise were Gaussian, a t-test with a p-value of .05 over 5000 or more samples is on average going to generate $.05 * 5000$ false positives. To compensate for this, bonferoni correction, also known as multiple comparison tests are performed; essentially p-values are divided by the number of independent tests being run. This, however, leads to extremely low p-values, so low that it would be impossible for any biological system to satisfy. To compensate, a Gaussian kernel is applied to the image, thus reducing variance (and thus separating the active and inactive distributions) as well as decreasing the effective number of voxels. Since t-tests are now no longer being applied to n ¡I need to define n¿ independent voxels, the factor by which the p-value must be divided by can be decreased. ¡Do I need to mathematically define all this?¿ The derivation and application of random field theory, and its use can be found in various papers [**?**].

## 2.2 General Linear Model

The most used form of FMRI analysis is still based on Statistical Parametric Mapping, but is able to account for several different levels or types of stimulus. By adding a General Linear Model to the analysis, the output signal timeseries (what FMRI detects) is regressed over the weighted sum of the various confound's timeseries. The equation for a general linear model is then

$$Y(t) = X(t)\beta + \epsilon(t) \tag{5}$$

where $Y(t)$ is the smoothed or detrended timeseries of measurements, $X(t)$ is a row vector of stimuli, $\beta$ is a column vector of weights, and $\epsilon$ is the error. Thus for every time, the measurement is assumed to be a weighted sum of the inputs plus some error. The calculation of $\beta$ then is merely a gradient descent search to minimize the mean squared error.

¡Image of GLM¿

Of course, a square wave input is not going to result in a square wave in the activation of brain regions. Thus, various methods are used to smooth $X(t)$ through time, and bandlimit the input. The best technique is convolving the stimulus input with a hemodynamic response function, which mimicks the basic shape of BOLD activation, including a delay due to rise time and fall time. This hemodynamic signal is static however, so every region of the brain gets the same design matrices (X(t)), although the weights of various stimulus or confounds are allowed to vary.

¡Image of Hemodynamic Response Function¿

Ultimately, activation due to a particular stimuli is decided by the $\beta$ value corresponding to that stimuli's column of $X(t)$. ¡Need to check this¿ The null hypothesis as to whether the outcome was random is then based on a t-test of the $\epsilon(t)$ timeseries.

### 2.3 Whats wrong with these techniques

There are a few problems with the techniques mentioned in the previous sections. First, they essentially ignore prior knowledge about the system. Although the most advanced form of the general linear model includes a "Hemodynamic Response Function," that hemodynamic response function is static across every region of the brain. It is well known that capillary beds are not uniform and so blood perfusion cannot possible be static across the brain. Thus, if extra information were available a-priori, that information could not be incorporated without modifications to the General Linear Model. Similarly heart rate could not be added either. It would obviously be advantageous to have true physiological parameters as entry points for these various other model parameters. The physiological models for the BOLD signal are quite good and based on realistic physics. While the exact connection between a stimulus and the flow inducing signal is not precisely known, model fits are actually quite good [?]. Regardless, being based on some real physiological parameter would allow for the establishment of reasonable priors and decrease model variance without breaking a sweat. Of course, using real parameters has the additional bonus of potentially providing information about physical pathologies. It is quite possible that physical properties such as decreased compliance of blood vessels could indicate a neurological condition that is not easily seen in a T1 or T2 map. In essence, this could make FMRI a much more useful clinical tool than it is now. The other problem with linear models is that they are a linear fit to a nonlinear signal. It is not uncommon for data to be thrown out in FMRI studies because no significant activation has been seen. However, if, for whatever reason, the BOLD response was acting more nonlinear than in other patients it would be completely possible for SPM to miss that activation.

¡Image with two different $\alpha$s¿

¡image comparing the results of 10% changes in various signals¿

Secondly, these methods are still based on t-tests, which notoriously lack robustness to non-Gaussian noise. While different techniques exist for imposing Gaussianity, those techniques are incapable of discriminating noise from signal. There is no way to know how much signal is removed by various smoothing techniques, or even if entire regions have been smoothed into oblivion. Instead of extensively filtering data to remove noise, the analysis method itself must be robust a wide range of noise, which is why we propose here the use of particle filters.

## 3 Proposed Approach

### 3.1 Goal

The ultimate goal of this project is to provide a new set of tools for analyzing FMRI data. Whereas SPM techniques have been highly successful at finding macroscopic regions of activation, linear modeling can carry significant bias error due to lack of model flexibility. While adding parameters can significantly increase error due to model variance, this effect is mitigated by the fact that we plan to use a model that is based on first principals. The purpose of this paper is thus to evaluate the potential of using a particle filter along with the BOLD model to derive physical parameters. In so doing, we hope to be able to show that neuronal efficacy, $\epsilon$ is a suitable variable for estimating voxel activation from a standard FMRI image. We also hope to show that estimated posterior distribution of the parameters, derived from the particle filter, is able to provide an accurate measure of the confidence interval.

### 3.2 Introduction to Particle Filters

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

### 3.2.1 Model

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

$$\dot{x}(t) = f(t, x(t), u(t), \theta, \nu_d) \tag{6}$$

$$y(t) = g(t, x(t), u(t), \theta, \nu_s) \tag{7}$$

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

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

$$\dot{x}(t) = f(x(t), u(t)) + \nu_x \tag{8}$$

$$y(t) = g(x(t)) + \nu_y \tag{9}$$

$\nu_x$ will result in something akin to adding a Weiner process to $y[0 : T]$, and thus will include low frequency noise. $\nu_y$ on the other hand will cause i.i.d. noise in $y[0 : T]$. For some of the tests, I will use de-trending methods to reduce the effects of $\nu_x$, the remainder of which should be re-factored into $\nu_y$. Both $\nu_x$ and $\nu_y$ have biological and non-biological sources. MR can lead to both types of noise, as demonstrated in [**?**]. Meanwhile changes in metabolism, heart rate, or other biochemical intervention could all lead to either $\nu_x$ or $\nu_y$.

### 3.2.2 Prior

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

$$\{\hat{Pr}x_i(0), w_i] : x_i(0) \sim \alpha(x), w_i = \frac{1}{N_p}, i \in \{1, 2, ..., N_p\}\} \tag{10}$$

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

$$\hat{Pr}(x(0) = \hat{x}) = \sum_{i=1}^{N_p} w_i \delta(\hat{x} - x_i(0))dx \tag{11}$$

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

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

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

$$\hat{Pr}(x(T + \Delta t)) \approx \sum_{i=1}^{N_p} w_i \delta \left( x - (x_i(T) + \int_T^{T+\Delta} \dot{x}_i(t)dt) \right) \tag{12}$$

### 3.2.3 Weighting

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

$$w_i(T) \propto \frac{\hat{Pr}(x_i[0:T]|y[0:T])}{q(x_i[0:T]|y[0:T])} \tag{13}$$

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

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

$$\begin{aligned} q(x[0:T]|y[0:T]) &= q(x(T)|x[0:T-1], y[0:T])q(x[0:T-1]|y[0:T]) \\ &= q(x(T)|x[0:T-1], y[0:T])q(x[0:T-1]|y[0:T-1]) \\ &= q(x(T)|x(T-1), y[0:T])q(x[0:T-1]|y[0:T-1]) \end{aligned} \tag{14}$$

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

In addition to $q(x_i(T)|x_i[0:T-1], y[0:T])$, the weight is also based on $Pr(x_i[0:K]|y[0:K])$, which may be broken up as follows.

$$\begin{aligned}
\hat{P}r(x[0:T]|y[0:T]) &= \frac{\hat{P}r(y[0:T], x[0:T])}{\hat{P}r(y[0:T])} \\
&= \frac{\hat{P}r(y(T), x[0:T]|y[0:T-1])\cancel{\hat{P}r(y[0:T-1])}}{\hat{P}r(y(T)|y[0:T-1])\cancel{\hat{P}r(y[0:T-1])}} \\
&= \frac{\hat{P}r(y(T)|x[0:T], y[0:T-1])\hat{P}r(x[0:T]|y[0:T-1])}{\hat{P}r(y(T)|y[0:T-1])} \\
&= \frac{\hat{P}r(y(T)|x[0:T], y[0:T-1])\hat{P}r(x(T)|x[0:T-1], y[0:T-1])\hat{P}r(x[0:T-1]|y[0:T-1])}{\hat{P}r(y(T)|y[0:T-1])}
\end{aligned} \tag{15}$$

Using the assumption that $y(t)$ is fully constrained by $x(t)$ (9), and that $x(t)$ is fully constrained by $x(t-1)$ (8), we are able to make the reasonably good assumptions that:

$$\hat{P}r(y(T)|x[0:T], y[0:T-1]) = \hat{P}r(y(T)|x(T)) \tag{16}$$

$$\hat{P}r(x(T)|x[0:T], y[0:T-1]) = \hat{P}r(x(T)|x(T-1)) \tag{17}$$

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

$$\begin{aligned}
\hat{P}r(x[0:T]|y[0:T]) &= \frac{\hat{P}r(y(T)|x[0:T], y[0:T-1])\hat{P}r(x(T)|x[0:T-1], y[0:T-1])\hat{P}r(x[0:T-1]|y[0:T-1])}{\hat{P}r(y(T)|y[0:T-1])} \\
&= \frac{\hat{P}r(y(T)|x(T))\hat{P}r(x(T)|x(T-1))\hat{P}r(x[0:T-1]|y[0:T-1])}{\hat{P}r(y(T)|y[0:T-1])} \\
&\propto \hat{P}r(y(T)|x(T))\hat{P}r(x(T)|x(T-1))\hat{P}r(x[0:T-1]|y[0:T-1])
\end{aligned} \tag{18}$$

Plugging these simplifications into (13) leads to:

$$\begin{aligned}
w_i(T) &\propto \frac{\hat{P}r(y(T)|x(T))\cancel{\hat{P}r(x(T)|x(T-1))}\hat{P}r(x[0:T-1]|y[0:T-1])}{\cancel{\hat{P}r(x_i(T)|x_i(T-1))}q(x[0:T-1]|y[0:T-1])} \\
&\propto w_i(T-1)\hat{P}r(y(T)|x(T))
\end{aligned} \tag{19}$$

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

1. $f(t, x(t), u(t)) = f(x(t), u(t))$ and $g(t, x(t), u(t)) = g(x(t))$ provide a sufficiently flexible model to encapsulate the true time series.
2. $E[\nu_x] = 0$ and $E[\nu_y] = 0$, and $\nu_x$ and $\nu_y$ are stationary
3. The PDF $q(x_i(0))$ (the prior) fully covers $Pr(x_i(0))$
4. Markov Assumption: $Pr(x(T)|x[0:T]) = Pr(x(T)|x(T-1))$
5. $q(x[0:T-1]|y[0:T]) = q(x[0:T-1]|y[0:T-1])$

### 3.2.4 Basic Particle Filter Algorithm

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

```
Initialize $N_p$ Particles: $\{x_i(0), w_i(0) : x_i(0) \sim \alpha(x), w_i(0) = \frac{1}{N_p}, i \in \{1, 2, ..., N_p\}\}$
$T = \{\text{Set of Measurement Times}\}$
for $t$ in $T$ do
  for $i$ in $N_p$ do
    $x_i(t) = x_i(t-1) + \int_{t-1}^{t} f(x(\tau), u(\tau))d\tau$
    $w_i(t) = w_i(t-1)\hat{Pr}(y(t)|x(t))$
  end for
end for
At $t + \Delta t$, $t \in T$, $\hat{Pr}(x(t + \Delta t)) \approx \sum_{i=1}^{N_p} w_i(t)\delta\left(x - (x_i(t) + \int_{t}^{t+\Delta t} f(x(\tau), u(\tau))d\tau)\right)$
```

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

### 3.2.5 Resampling

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

$$N_{eff} \approx \frac{N_p}{\sum_{i=1}^{N_p} w_i^2} \tag{20}$$

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

$$\hat{x}_j \sim \left(\sum_{i=1}^{N_p} w_i(t)\delta(x - x_i(t))\right), j \in \{1, ..., N_p\} \tag{21}$$

then $\hat{x} \sim x$.

The ultimate effect of this regularized resampling is a convergence similar to simulated annealing or a genetic algorithm. ”Fit” versions of $x$ 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 $\nu_x$ and $\nu_y$.

This has the added benefit that it is not necessary to re-draw points from a $q$ function at every time step, which is faster. this the definition of the weight, so when weights are updated it is necessary to divide by the proposal distributionto converge to the true distribution it is necessary to divide by the proposal distribution.

Thus $\hat{x}(0)$ is an empirical representation of the prior, and $\hat{x}_i(0)$ is the $i^{th}$ particle, a single possible representation of the system at time 0. Initially, since each $\hat{x}_i$ has been drawn from the prior, and no measurements have been made, the particles are all equally weighted. When a measurement becomes available, that information is then incorporated into the posterior by re-weighting the particles according to how well they estimated this new measurement.

The estimated posterior of $\hat{x}(t)$, for discrete time, $t$ (before incorporating any measurement that might be available at $t$) is then:

$$\Pr\{\hat{x}(t)|y[0 : t-1]\} = \int \Pr\{\hat{x}(t), \hat{x}(t-1)|y[0 : t-1]\}d\hat{x}(t-1) \tag{22}$$

and by bayes theorem $\Pr\{\hat{x}(t), \hat{x}(t-1)|y[0 : t-1]\} = \Pr\{\hat{x}(t)|\hat{x}(t-1), y[0 : t-1]\} \Pr\{\hat{x}(t-1)|y[0 : t-1]\}$, therefore

$$\Pr\{\hat{x}(t)|y[0 : t-1]\} = \int \Pr\{\hat{x}(t)|\hat{x}(t-1), y[0 : t-1]\} \Pr\{\hat{x}(t-1)|y[0 : t-1]\}d\hat{x}(t-1) \tag{23}$$

Since in this work we will only deal with markov processes, we assume that $\Pr\{\hat{x}(t)|\hat{x}(t-1), y[0:t-1]\} = \Pr\{\hat{x}(t)|\hat{x}(t-1)\}$ thus the posterior becomes:

$$\Pr\{\hat{x}(t)|y[0:t-1]\} = \int \Pr\{\hat{x}(t)|\hat{x}(t-1)\}\Pr\{\hat{x}(t-1)|y[0:t-1]\}d\hat{x}(t-1) \tag{24}$$

Therefore in between measurements, there is no need to know previous measurements in order to make predictions about the internal state. This is important because it means makes particle filters significantly more memory efficient, and although it isn't useful in this case, it means a particle filter could theoretically be run indefinitely without the need to store old data.

As stated previously, new measurements may be incorporated by reweighting particles. To find the updated distribution, $\Pr\{\hat{x}(t)|y[0:t]\}$, bayes theorem is again applied:

$$\Pr\{\hat{x}(t)|y[0:t]\} = \Pr\{\hat{x}(t)|y[0:t-1], y(t)\} = \frac{\Pr\{y(t)|\hat{x}(t), y[0:t-1]\}\Pr\{\hat{x}(t)|y[0:t-1]\}}{\Pr\{y(t)|y[0:t-1]\}} \tag{25}$$

$$\Pr\{\hat{x}_i(t)|y[0:t-1]\} = \int \Pr\{\hat{x}_i(t)|\hat{x}_i(t-1), y[0:t-1]\}\Pr[\hat{x}_i(t-1) \tag{26}$$

Since $\hat{x}(t) = \hat{x}(t-1) + \int_{t-1}^{t} f(x(t), u(t))dt + \int_{t-1}^{t} \nu_d dt$,

$$\Pr\{\hat{x}_i(t)|\hat{x}_i(t-1)\} = \Pr\{\int t - 1^t \nu_d\} \tag{27}$$

of course $\int_{t-1}^{t} f(x(t), u(t))dt$ may need to be evaluated numerically. Therefore, to estimate the posterior probability at $t$, it is just necessary to propagate each particle forward in time according to a state space function. To form the posterior probability around the true $x(t)$, each particle is weighted based on

This then results in the expected value of $\hat{y}(t)$:

$$E[\hat{y}_i(t)|\hat{x}_i(t-1)] = E[\hat{x}_i(t)] \tag{28}$$

$$\dot{y} = -ky \tag{29}$$

$\theta_i$ would have an estimate for $k \; and \; y$. Of course in many cases the model is a potentially non-linear differential equation with no closed form solution. Initially every $\theta_i$ will have the same weight, because there isn't yet anything to base a weighting on. Note that this doesn't mean that the particles have been evenly distributed, so the distribution is not necessarily "flat" just because all the particles are equaly weighted. When a new measurement is available, the particles are then reweighted based on how well they predicted the new measurement.

## 3.3 My Specific Particle Filte

Although particle filters are well defined, significant design decisions are application dependent. The most important choice is the weighting function, $w(y_t - y(\theta_t))$. Obviously the function should weight 0 maximally, the drop off rate is extremely important. A very thin gaussian probability distribution function has nice properties, but thin tails. As a result, large outliers in the measurement vector could easily force all the particles to have near 0 weights, thus forcing the particle filter to converge improperly. On the other end of the spectrum, a cauchy PDF may not weight particles in the middle enough, preventing the particles to never converge. Of course, the scale factor, or variance, also plays a signifcant role in the convergeance rate. The importance of the weighting function cannot be overstated, as this is the primary factor in deciding the rate at which the particle filter converges. As part of the simulations, I compare a gaussian based weighting function with an exponential weightin function. In both cases, I scaled the width of the weighting functions automatically based on the signal variance.

Dealing with low frequency noise is also a crucial part of the tests. I tested two different methods of dealing with this noise. Various methods are used to deal with this problem in other toolsets. The most common method is to apply a high-pass filter to the measurement vector. Although this makes sense in many applications, in this case it may not be adaptive enough for a random walk type noise. On the other hand, it has been shown that a spline de-trending method is able to better deal with the low frequency noise typically present in FMRI. Thus, my first method was to use a spline fit with knots every 80 or 90 measurements to trance the basic shape of the drift. I then added an extra DC gain parameter to the model to account for the fact that the de-trended signal tended to have a mean of 0, rather than some positive value.

My second method of dealing with low frequency drift was to counteract it by weighting particles based on $(y_t - y_{t-1}) - (y(\theta_t) - y(\theta_{t_1}))$ rather than directly on $y_t - y(\theta_t)$. It is generally believed that the noise present in FMRI has approximately gaussian steps, which means that this method only has to deal with gaussian white noise. Such a technique is usually not considered a good idea because high frequency noise is almost exclusively assumed in systems theory. In this case however; the noise is all in the lower frequency range, so performing a differentiation like operation, which magnifies high frequency signals, is exactly what we want.

... more specifics...

The General Linear Model, and SPM's goal is to find a correlation between input, or what a person is thinking, and which regions are active. This is an important task, but it misses the potential of FMRI. FMRI's greatest weakeness as a tool for studying neural activity is that it is connected to neurons through a long chain of events. But what if, in addition to seeing which brain regions are active for which stimuli, we could also study pathologies such as poor blood flow, or derive physical parameters such as blood vessel compliance? By using a realistic physical model as the basis for discovering the relation between the outside world and FMRI signal, the path to detecting activation gives all those other things for free, precisely *because* FMRI is the result of such a long chain of events. It is actually quite common to use models to derive the values of physiological parameters. For instance, blood volume is often measured by injecting a dye into the blood stream and then assuming that the dye is removed at a rate proportional to the concentration in the blood. By performing a simple regression it is possible to estimate blood volume quite accurately. While for this particular project I don't claim that the results will give physiologically plausible results, with some tuning of the algorithm it would very possible.

While the General Linear Model has historically done well with detecting activation, it is limited to scaling the entire input signal up or down. This leaves out even basic effects such as longer or shorter time constants, as well as nonlinear effects which are known to be present in the BOLD system. **

While the General Linear Model has been extremely effective at gleaning information about which voxels are "active" and which are not, its purpose is only to say which regions are "statistically significant". However, considering that voxels consist of millions of neurons, the reality is that a region could be active, but at some other level or time constant than expected; and be completely missed by the linear model. Studies showing activation in Cadavers of Phantoms highlight the inherant problem with the General Linear Model: a problem that mandates multiple comparison testing with very high thresholds to ensure any sort of face validity [?]. By moving to models that actually "learn" the parameters of the model, model error will be drastically reduced allowing for much more reasonable statistical thresholds.

### 3.4 Particle Filters

## 4 Methods

### 4.1 Preprocessing

After FMRI data has been acquired it is always necessary to modify the data in some way to make different runs comparable. Because FMRI signal levels are essentially unit-less, at the very least it is necessary to convert the data into % difference from the baseline. Finding the baseline is the first hurdle in dealing with FMRI. Various methods exist for doing this, but they are all ad-hoc. It is common to treat each stimulus pulse independently, ignoring the long decay of the BOLD signal. By treating each input separately, the "base" for calculating the percent difference is merely the signal level before the stimulus is applied. The problem with this is that the BOLD signal can be heavily delayed, and fall times can be as long as 10 or 15 seconds, meaning that the assumption is clearly a false one. Another popular method of dealing with this problem, is to put all the time series through a high pass filter. This will of course remove the DC component of the signal, and some amount of so called "drift". The problem with this method is that it is not adaptive to the input. Huge variations in drift frequencies can exist in a single time-series. Thus, a single cutoff frequency could miss a significant drift component, or it could remove *actual* signal, if the cutoff frequency is set too high.

## 5 Introduction to Modeling

Of course, it is well known that many of the assumptions that the GLM makes about BOLD activation do not hold. The General Linear Model ignores the fact that the BOLD response is nonlinear, contains non-gaussian noise and most importantly that brain networks are so called "Small World Networks" [?], [?]. Merely accounting for nonlinearities in the time-series of the BOLD response can result in significantly better estimation as shown in [?]. But even more crucial is the recent movements away from mass- univariate models. Current mass-univariate models assume uniform connectivity from input to every voxel; yet this is

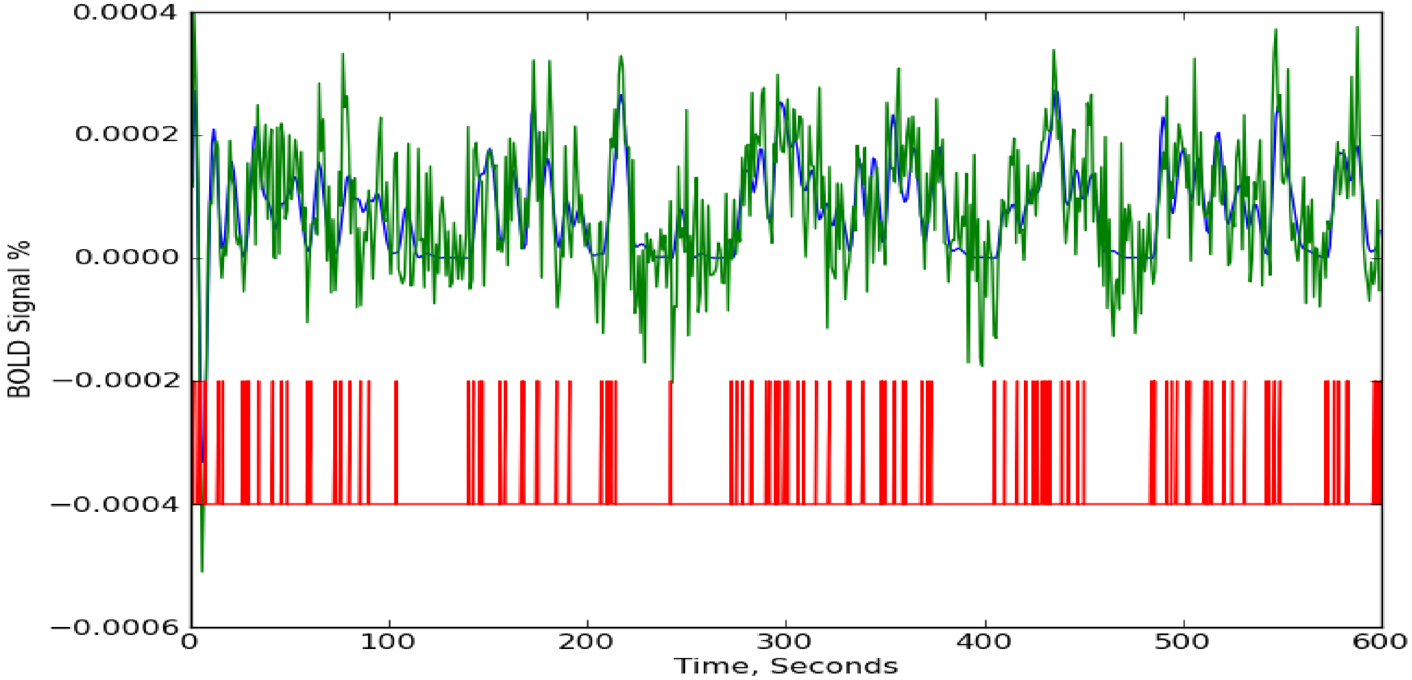

Figure 1: Simulated time-series for a single region. The red line shows the input stimulus, the blue line shows the base BOLD response, and the green line is the BOLD response with Gaussian white noise added at SNR of 2

obviously not the case. While violating linearity assumptions or Gaussian assumptions requires us to be more conservative, the lack of any method to switch brain circuits on and off means that any model we make for brain regions would not be Turing complete; in effect implying that humans are incapable of thought [**?**].

So called Dynamic Causal Modeling is the beginning of the next phase of neurology. DCM is the first brain study to show any significant connection between diffusion tensor imaging and actual function layout in the human brain [**?**]. By incorporating connections between regions and a realistic activation model, DCM corrects two of the largest problems with the GLM. While there are other techniques that may be useful in the future, they either lack physiological analogs (Artificial Neural Networks) or are extremely computationally expensive (mutual information). Given that much of the potential of the GLM has been exhausted, and that DCM is one of the first well defined methods capable of learning complicated brain circuits DCM is crucial to the future of FMRI and our understanding of the human brain.

# 6 Results

The particle filter shows great promise in being able to learn a variety of different regional activation parameters. We have performed tests with a random binary pulse train as input to a simulated region and then tested the particle filter's ability to estimate the signal parameters. Additionally random gaussian noise with an Signal-To-Noise ratio of 2.0 was added. Figure 1 shows the pulse train, the clean signal and the noisy signal, which was the input to the particle filter. A random set of system parameters was chosen to simulate the region, which the particle filter had no knowledge of other than a mean and rough variance. The tests were performed using 10,000 particles and took approximately 4 minutes to compute.

Figure 2 shows the estimated BOLD response versus the simulated data (without noise). It is clear that as the estimated BOLD response converges to the true BOLD as time progresses. This is because more and more potential states are eliminated by system's response to new input sequences. Convergence depends highly on the input signal and the weighting function thus choice of both are extremely important. We have found that an impulse train (emulated with .1 second width square waves) gives a very good response. It is also important to choose an input with some longer hold times, at least as long as than the expected time constants, which in this case was around 8 seconds. The weighting function we have chosen is based on the exponential

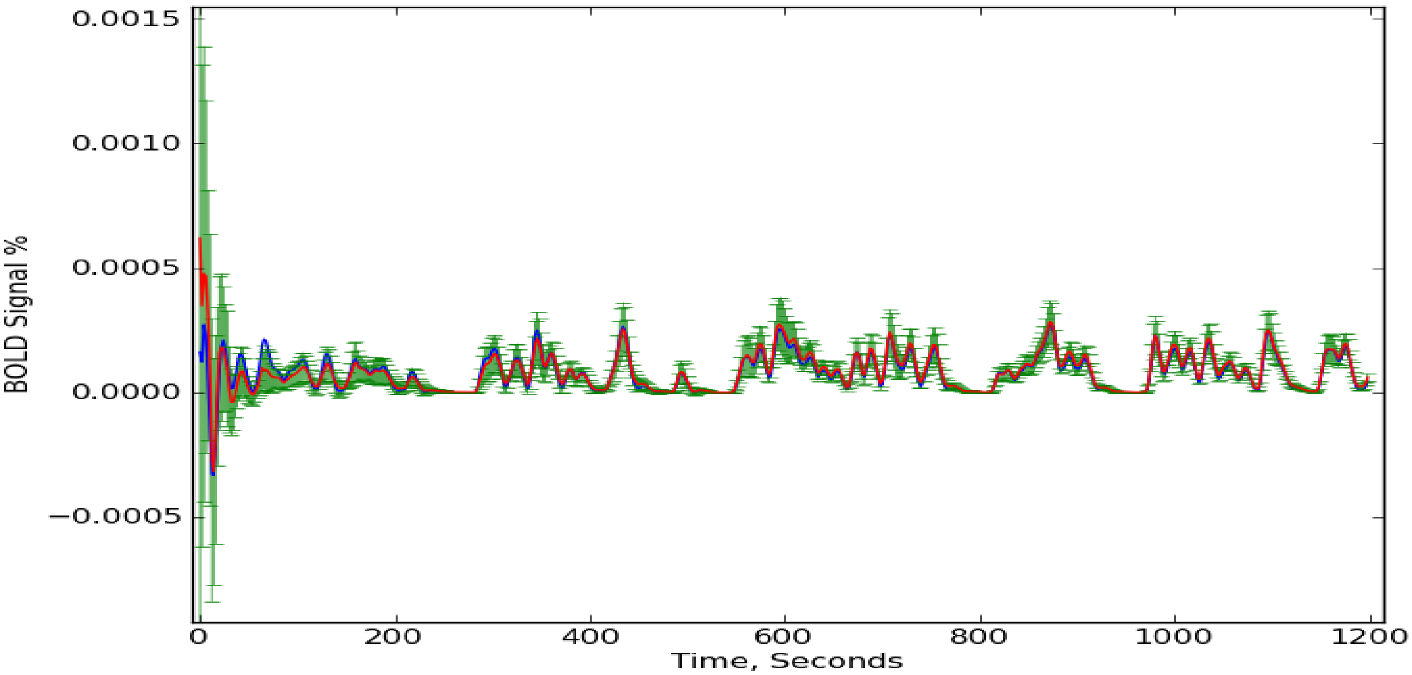

Figure 2: Particle Filter Results. The blue line is the "true" (simulated) BOLD response, the red line is the output of the particle filter with 2 standard deviations in green. Note that the error bar for time 0 is outside the scope of the image and is approximately ±.003.

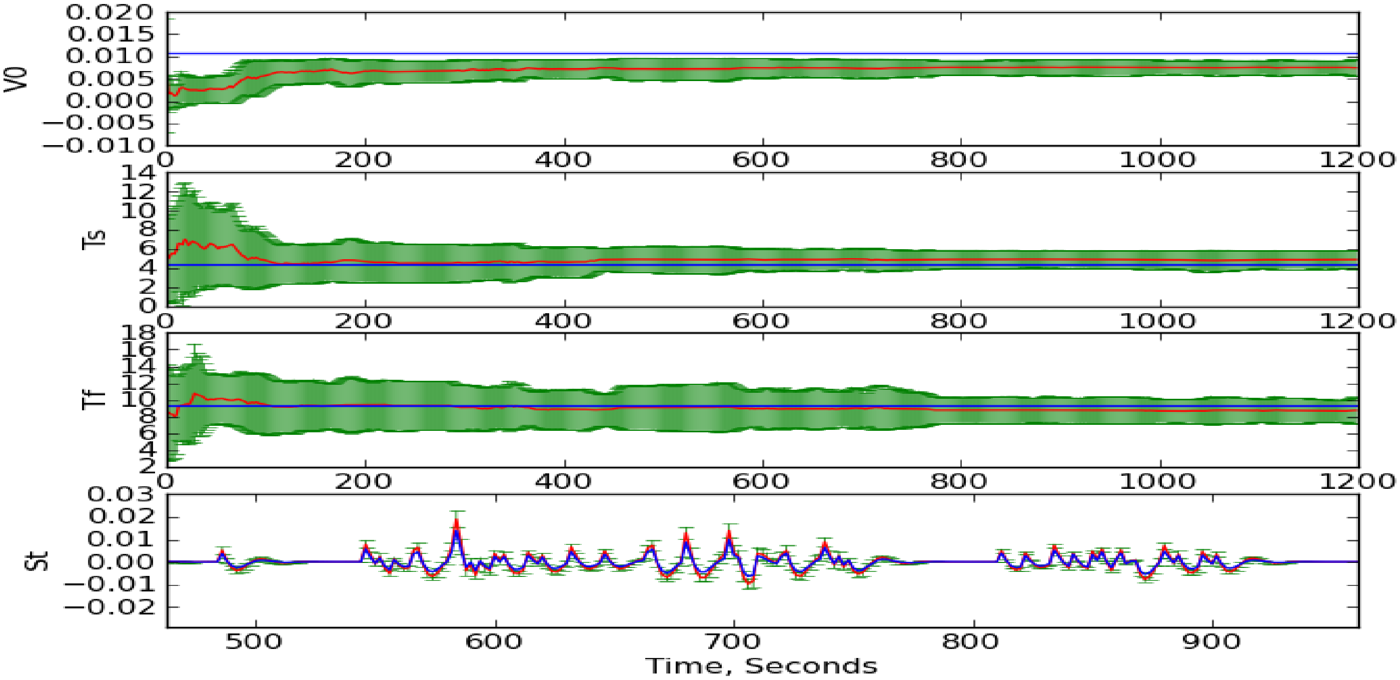

Figure 3: Particle Filter Results. The blue line is the "true" parameter and the green line is the estimated value for the parameter. Note that the timescale for $S_t$ is different to highlight an typical stimulus response. Again 2 standard deviations are shown.

distribution, with a variance equal to the RMS of the signal. While it is important to converge by the end of the time-series, it is often the case that converging too fast will harm the state estimation by over- emphasizing the measurements of single time points. The rate of convergence is well within control of the user of the particle filter based on the weighting function and the frequency of re-sampling. Note that resampling can cause quantization errors due to the nature of estimating a long tailed distribution with a finite number of samples.

Figure 3 shows the estimated versus simulated values for several key parameters. Notice how the variance drops as the particle filter continues. The parameters shown are $V_0$ which is a scaling parameter, $\tau_s$, and $\tau_f$ which are both time constants and $s_t$ which is a hidden state variable, estimating the flow inducing signal. Note that even though some of the parameters don't converge to the exact value, that the estimated $s_t$ still matches the true $s_t$ relatively well. It is often the cast that although a few parameters don't converge to their true value, one parameter may pick up the slack for another. It is of note that $\tau_s$ and $\tau_f$ do converge at least close to their true values, because they cannot be determined from steady state.