NONLINEAR STATE AND PARAMETER ESTIMATION OF THE HEMODYNAMIC MODEL USING fMRI BOLD SIGNAL

by

SERDAR ASLAN

B.S. Electrical and Electronics Engineering, METU, 2001

M.S. Electrical and Electronics Engineering, METU, 2004

Submitted to the Institute of Biomedical Engineering

in partial fulfillment of the requirements

for the degree of

Doctor

of

Philosophy

Boğaziçi University

2015

NONLINEAR STATE AND PARAMETER ESTIMATION OF THE HEMODYNAMIC MODEL USING fMRI BOLD SIGNAL

APPROVED BY:

Prof. Dr. Ahmet Ademoğlu .....
(Thesis Advisor)

Prof. Dr. Ata Akın .....
(Thesis Co-advisor)

Assoc. Prof. Dr. Haluk Bingöl .....

Assoc. Prof. Dr. Ali Taylan Cemgil .....

Prof. Dr. Yekta Ülgen .....

Assist. Prof. Dr. Adil Deniz Duru .....

DATE OF APPROVAL: June 2015

Academic Ethics and Integrity Statement

(To be completed upon submission of Master and PhD Theses for examination)

I am familiar with the YÖK (Higher Council of Education) Academic Ethics and Integrity Policy and I understand the potential consequences should my thesis be found to contain plagiarized content or violate this policy in any other way.

Name: ..... Date: .....

Signature: .....

ACKNOWLEDGMENTS

Thanks to my God, Most Gracious, Most Merciful, to whom I have so deep love.

Thanks to my God, I had always the good opportunity to be educated in very good schools from my childhood. This was the same for Bosphorus University with its precious professors.

I loved many aspects of the science. I had the opportunity to study various branches of biology, including molecular biology and physiology, theoretical physics, chemistry and various branches of mathematics. From every aspect of the science, I had always great delight. Every such scientific branch, I saw as a beauty of God. The more my knowledge increased, the more my love increased to my God, to humanity, to every creature and every part of the space.

In my beautiful years, I have met many good colleagues and professors. For my thesis years, I thank to my professors and colleagues Ata Akın, Ali Taylan Cemgil, Karl Friston, Ahmet Ademoğlu, Haluk Bingöl, Umut Orguner, Murat Şamil Aslan, Behçet Uğur Töreyin and Mübeccel Demirekler. I also thank to Yekta Ülgen and Adil Deniz Duru for their presence in my jury.

I thank to my parents for their help in my life.

I thank to my dear precious friends for all their goodness.

I thank to my precious teacher whom I love so much.

I pray to my God that from my labors the humanity will benefit. I hope these efforts will benefit to various diseased people. I want to see the technology for as serving to spread wealth and goodness to the people whom I see as my brothers and

sisters. I pray to God that in a near future the war industry will transform mostly to the peace industry. The world can be a more beautiful place.

ABSTRACT

NONLINEAR STATE AND PARAMETER ESTIMATION OF THE HEMODYNAMIC MODEL USING fMRI BOLD SIGNAL

The joint estimation of the parameters and the states of the hemodynamic model from the blood oxygen level dependent (BOLD) signal is a challenging problem. In the functional magnetic resonance imaging (fMRI) literature, quite interestingly, many proposed algorithms work only as a filtering method. In the fMRI state estimation literature, extended Kalman filter (EKF) is asserted to be not robust and worse than standard particle filters (PF). We compared EKF with PF and observed that the contrary is true. We also implemented particle filter that approximates the proposal function by the extended Kalman filter. We compared Gaussian type approximated estimation techniques like extended Kalman filter (EKF), unscented Kalman filter (UKF), cubature Kalman filter (CKF) as well as stochastic inference techniques like standard particle filters (PF) and auxiliary particle filter (APF). Filtering makes the estimation of the hidden states and the parameters less reliable compared with the algorithms that use smoothing. We improved the EKF performance by adding smoother. The joint state and parameter estimation is improved substantially by performing the iterative EKS (IEKS) algorithm. We compared IEKS performance with the square-root cubature Kalman smoother (SCKS) algorithm. We show that its accuracy for the state and the parameter estimation is better and much faster than iterative SCKS. SCKS was found to be a better estimator than the dynamic expectation maximization (DEM), EKF, local linearization filter (LLF) and PF methods. We show in this thesis that IEKS is a better estimator than iterative SCKS under different process and measurement noise conditions.

Keywords: Hemodynamic model, particle filter, auxiliary particle filter, extended Kalman filter, smoother, cubature Kalman filter

ÖZET

HEMODİNAMİK MODELİN fMRI BOLD SİNYALİ KULLANARAK DOĞRUSAL OLMAYAN DURUM VE PARAMETRE TAHMİNİ

BOLD sinyali kullanarak hemodinamik modelin parametre ve durum tahmini başarması zor bir problemdir. fonksiyonel Manyetik Rezonans Görüntüleme (fMRG) literatüründe çoğu önerilen algoritmalar yalnız süzgeç tipindeki metodlardır. fMRG literatüründe, genişletilmiş Kalman süzgecinin (GKS) sağlam bir metod olmadığı ve PS'lerden kötü olduğu iddia edilmiştir. GKS ile PS'leri karşılaştırdık ve bunun doğru olmadığını gördük. Bu tezde ayrıca özel bir parçacık süzgeci uygulanmıştır. Bu süzgeç çeşidinde önerim fonksiyonu, genişletilmiş Kalman süzgeciyle (GKS) yaklaştırılmıştır. Bu tezde fMRG literatüründe kullanılan GKS, kokusuz Kalman süzgeci (KKS), küpleme Kalman süzgeci (KKS) ve standart paçacık süzgeci (PS) ve yardımcı parçacık süzgeci (YPS) gibi olasılıksal tahmin metodlarını karşılaştırdık. Süzgeçler, düzleştirici kullanan algoritmalara göre daha az güvenilirdir. Durum ve parametrelerin beraber tahmini genelleştirilmiş Kalman düzleştiricilerinin tekrarlı kullanımı ile önemli bir şekilde geliştirilmiştir. Tekrarlı kullanımla hem parametre hem durum tahmini önemli ölçüde iyileştirildi. Tekrarlı genelleştirilmiş Kalman düzleştiricisini (TGKS), karekök küpleme Kalman düzleştiricisi (KKKKD) algoritmasıyla karşılaştırdık. Hemodinamik durum ve parametre tahmininin bu yöntemle daha doğru ve daha hızlı olduğunu gördük. KKCKS metodu fMRI literatüründe dinamik tahmin maksimizasyonu (DTM), GKS, yerel düzleştirme süzgeci (YDS) ve PS algoritmalarından daha iyi bir tahmin metodu olduğu bulunmuştu. Bu çalışmada TGKS metodunun tekrarlı KKCKS'dan farklı durum ve gözlem gürültü şartlarında daha iyi bir tahmin metodu olduğunu gösterdik.

Anahtar Sözcükler: Hemodinamik model, Parçacık süzgeci, genişletilmiş Kalman süzgeci, düzleştirici, küpleme Kalman süzgeci

TABLE OF CONTENTS

ACKNOWLEDGMENTS iv
ABSTRACT vi
ÖZET vii
LIST OF FIGURES xi
LIST OF TABLES xvi
LIST OF SYMBOLS xviii
LIST OF ABBREVIATIONS xix
1. INTRODUCTION 1
1.1 Hemodynamic State Estimation 3
1.2 Contribution 4
1.3 Outline 7
2. ESTIMATION TECHNIQUES 8
2.1 State Estimation 8
2.2 Gaussian Approximated Inference 9
2.3 Stochastic Inference 10
2.4 Variational Approach 12
3. FRISTON-BUXTON HEMODYNAMIC MODEL 14
3.1 Functional Representation Overview 14
3.2 Hemodynamic Model System Representation 15
3.3 Neuronal Input and Hemodynamic Responses 20
4. NONLINEAR STATE ESTIMATION 24
4.1 Gaussian Approximated Inference Methods 24
4.1.1 Kalman Filter 24
4.1.2 Extended Kalman Filter 27
4.1.3 Square-Root Cubature Kalman Filter 30
4.2 Stochastic Type Approximation Methods 35
4.2.1 Particle Filter 35
4.2.2 Particle Filter with Extended Kalman Filter Proposal Function 36
5. DUAL ESTIMATION - JOINT PARAMETER AND STATE ESTIMATION 38

5.1 Iterative Extended Kalman Smoother 39
5.1.1 Extended Kalman Smoother 39
5.1.2 Parameter Estimation and Iterative EKS 40
5.2 Iterative Square-Root Cubature Kalman Filter 41
5.2.1 Square-Root Cubature Kalman Smoother 42
5.2.2 Parameter Estimation with SCKS 44
5.3 Toy Example: Linear Case 44
5.3.1 Visualization of Smoother Estimation 45
5.3.2 Toy Example Simulation Results for State Estimate for fixed Parameter 47
5.3.3 Toy Example Simulation Results for Parameter Estimation . . . 49
6. SIMULATION RESULTS FOR THE HEMODYNAMIC MODEL WITH KNOWN PARAMETERS 51
6.1 Findings and Contribution 51
6.2 Discussion 60
7. RESULTS FOR THE HEMODYNAMIC MODEL WITH UNKNOWN PARAMETERS - JOINT STATE AND PARAMETER ESTIMATION 66
7.1 Contribution: IEKS as an Efficient Hemodynamic Joint State and Parameter Estimator 66
7.1.1 Comparison of the Filtering Algorithms 67
7.1.2 Performance Improvement by EKS over EKF 68
7.1.3 Joint Parameter and State Estimation with Iterative EKS compared EKS 69
7.1.4 Joint Parameter and State Estimation with Iterative 70
7.1.5 Discussion 85
7.1.5.1 Parameter Sweep and the BOLD Signal 88
7.2 Real Data and the IEKS method 91
7.2.1 Background Information: Test Setup and BOLD Data 91
7.2.2 Extended Hemodynamic Model for Multi-Input System 93
7.2.3 Results: IEKS Model Inversion 94
7.2.4 Discussion 98
7.2.4.1 Parameter Variance and Convergence 99

7.2.4.2 IEKS Switched Parameter Variance 102
7.2.4.3 Estimated Hemodynamic Variables 104
8. CONCLUSION 106
8.1 Future Work 107
APPENDIX A. 109
A.1 Proof of Kalman Filter 109
A.2 Proof of RTS Kalman Smoother 110
A.3 QR Decomposition 112
REFERENCES 114

LIST OF FIGURES

Figure 1.1	Real BOLD signal with the permission of Karl Friston.	1
Figure 3.1	Boxcar neuronal input with short time activation.	21
Figure 3.2	Hemodynamic variables for boxcar neuronal input with short time activation.	21
Figure 3.3	BOLD signal for boxcar neuronal input with short time activation.	22
Figure 3.4	Boxcar neuronal input with long time activation.	22
Figure 3.5	Hemodynamic variables for boxcar neuronal input with long time activation.	23
Figure 3.6	BOLD signal for boxcar neuronal input with long time activation.	23
Figure 5.1	Kalman Smoother and Filter Confidence Region	47
Figure 5.2	The convergence of the parameter estimate.	50
Figure 6.1	Neuronal Inputs as Gaussian bumps.	53
Figure 6.2	Hemodynamic Variables for Gaussian bumped neuronal inputs.	53
Figure 6.3	BOLD Signal for Gaussian bumped neuronal inputs.	54
Figure 6.4	Filtering algorithms are tested for input as boxcar functions.	54
Figure 6.5	Errors of the algorithms at each time for the Parameter Set-3 with Gaussian Bump input	59
Figure 6.6	Large Scale Vision of Errors of the algorithms at each time for the Parameter Set-3 with Gaussian Bump input	59
Figure 6.7	Posterior $p(x_{150}\|y_{1:150})$ formed from the particles of the 1. hemodynamic state.	61
Figure 6.8	Posterior $p(x_{500}\|y_{1:050})$ formed from the particles of the 1. hemodynamic state.	61
Figure 6.9	Posterior $p(x_{750}\|y_{1:750})$ formed from the particles of the 1. hemodynamic state.	62
Figure 7.1	IEKS and Importance of Iteration.	70
Figure 7.2	The convergence of the parameter κ estimate: Scenario 1.	73
Figure 7.3	The convergence of the parameter τ estimate: Scenario 1.	73
Figure 7.4	The convergence of the parameter χ estimate: Scenario 1.	74

Figure 7.5	Scenario 1: Parameter Estimate Bounds.	74
Figure 7.6	Scenario 1: State Estimation Bounds.	75
Figure 7.7	The convergence of the parameter κ estimate: Scenario 2.	75
Figure 7.8	The convergence of the parameter τ estimate: Scenario 2.	76
Figure 7.9	The convergence of the parameter χ estimate: Scenario 2.	76
Figure 7.10	Scenario 2: Parameter Estimate Bounds.	77
Figure 7.11	Scenario 2: State Estimation Bounds.	77
Figure 7.12	The convergence of the parameter κ estimate: Scenario 3.	78
Figure 7.13	The convergence of the parameter τ estimate: Scenario 3.	78
Figure 7.14	The convergence of the parameter χ estimate: Scenario 3.	79
Figure 7.15	Scenario 3: Parameter Estimate Bounds.	79
Figure 7.16	Scenario 3: State Estimation Bounds.	80
Figure 7.17	The convergence of the parameter κ estimate: Scenario 4.	80
Figure 7.18	The convergence of the parameter τ estimate: Scenario 4.	81
Figure 7.19	The convergence of the parameter χ estimate: Scenario 4.	81
Figure 7.20	Scenario 4: Parameter Estimate Bounds.	82
Figure 7.21	Scenario 4: State Estimation Bounds.	82
Figure 7.22	The convergence of the parameter κ estimate: Scenario 5.	83
Figure 7.23	The convergence of the parameter τ estimate: Scenario 5.	83
Figure 7.24	The convergence of the parameter χ estimates: Scenario 5.	84
Figure 7.25	Scenario 5: Parameter Estimate Bounds.	84
Figure 7.26	Scenario 5: State Estimation Bounds.	85
Figure 7.27	BOLD signal change with respect to κ change. κ is changed over the interval [0.35;1.05].	89
Figure 7.28	BOLD signal change with respect to τ change. τ is changed over the interval [0.32;2.32].	90
Figure 7.29	BOLD signal change with respect to χ change. χ is changed over the interval [0.21;1.01].	90
Figure 7.30	BOLD signal change with respect to ϵ change. ϵ is changed over the interval [0.01;0.51].	91

Figure 7.31 Real BOLD data is collected from the motion sensitive area V5. The sampling time TR is 3.22 sec. The first 256 samples were used for model inversion. 92

Figure 7.32 In this setup, the person was exposed to 3 different neuronal inputs. They are Vision, Motion and Attention. Those neuronal inputs are modeled as boxcar functions. During the test, the person was subject to 5 different conditions. Those are Saturation, Fixation, Attention, No Attention and Motion View. Depending on the condition, for each neuronal input, we have the value of either 1 or 0. Following Friston et al. [13, 14, 15], DC values of the inputs are removed. 93

Figure 7.33 The simulations which converge to global optima θ_g. Specifically, we estimate the parameter set $\theta = [\epsilon_1, \epsilon_2, \epsilon_3, \kappa, \tau, \chi]$ which converged to the global optima θ_g. 96

Figure 7.34 The simulations which converge to global optima. For every iteration, the RMS error of the prediction is shown. At every step, the RMS errors are decreasing. After around 25 iterations, the algorithm converges. 96

Figure 7.35 The simulations which converge to local optima. Specifically, we estimate the parameter set $\theta = [\epsilon_1, \epsilon_2, \epsilon_3, \kappa, \tau, \chi]$ which converged local optima is θ_l. 97

Figure 7.36 The simulations which converge to local optima. For every iteration, the RMS error of the prediction is shown. At every step, the RMS errors are increasing. After around 200 iterations, the algorithm converges. 97

Figure 7.37 The estimated neuronal efficiency factors $\epsilon_1, \epsilon_2, \epsilon_3$ are shown. These factors are associated with neuronal inputs "Vision" u_1, "Motion" u_2 and "Attention" u_3. ϵ_2 gives information about how much the selected brain region is responsive to the motion input. The selected region was the motion sensitive area V5. The results are very similar to the work of Friston et al. [13]. 98

Figure 7.38 The importance of the parameter variance. Plots are taken from the converged global optimum θ_g. We chose the variance small enough ($\sigma_w^2 = \Delta t e^{-8}$) to ensure fixed value for parameter estimate throughout the time index. As can be seen from the graph the parameter is a rather fixed value as desired. 100

Figure 7.39 The simulations which converge to global optima θ_g. This time, the parameter variance is increased to $\sigma_w^2 = \Delta t e^{-6}$. Specifically, we estimate the parameter set $\theta = [\epsilon_1, \epsilon_2, \epsilon_3, \kappa, \tau, \chi]$ 101

Figure 7.40 The simulations which converge to global optima. This time, the parameter variance is increased to $\sigma_w^2 = \Delta t e^{-6}$. For every iteration, the RMS error of the prediction is shown. At every step, the RMS errors are decreasing. After around 25 iterations, the algorithm converges. 101

Figure 7.41 Importance of the parameter variance. This time, parameter variance increased to the value $\Delta t 10^{-6}$. We run the IEKS algorithm until convergence. For the converged θ, we plot the parameters with respect to the time index. There is only one optima, but we have lost the fixedness in time. The parameters fluctuate around the global optima point. 102

Figure 7.42 Convergence of the parameters when applying the switched parameter variance. First, IEKS begins with the high parameter noise ($\sigma_w^2 = \Delta t e^{-6}$). After the iteration number 10, we have the parameter noise variance as ($\sigma_w^2 = \Delta t e^{-8}$). This way, IEKS converged to the same global optima as in the low parameter variance case. There is only a single optimum. Furthermore, IEKS converged faster. 103

Figure 7.43 The estimated hemodynamic variables which converged at the global optima θ_g. The graph is in accordance with the theory and the previous results in the fMRI literature. During stimulus instances, an increased blood flow is observed. Subsequent the change in the increased blood flow, blood venus volume is increased and the initial dip of deoxyhemoglobin content is observed. 104

Figure 7.44 The prediction error in the BOLD signal. Here, sampling interval is $\Delta t = 3.22$ sec. The real BOLD signal is plotted by the blue color, whereas the prediction error is plotted by the red color. 105

LIST OF TABLES

Table 3.1	Parameter Values of the Hemodynamic State Transition Equations	17
Table 3.2	Parameter Values of the Hemodynamic Observation Equation	17
Table 5.1	Initial Values and Noise Parameters for Toy Example	45
Table 5.2	Monte Carlo State RMS Error for Kalman Filter and Smoother.	48
Table 5.3	Monte Carlo Estimate for θ.	50
Table 6.1	Noise Statistics and Initial Values.	52
Table 6.2	Parameter Values of the Hemodynamic Model.	55
Table 6.3	Parameter Set 1, Mean and Standard Deviation of the RMS State Errors: PF-EKF, PF, APF, EKF and SCKF for Gaussian Bump Type Neuronal Input.	56
Table 6.4	Parameter Set 1, Mean and Standard Deviation of the RMS State Errors: PF-EKF, PF, APF, EKF and SCKF for Boxcar Type Neuronal Input.	56
Table 6.5	Parameter Set 2, Mean and Standard Deviation of the RMS State Errors: PF-EKF, PF, APF, EKF and SCKF for Gaussian Bump Type Neuronal Input.	56
Table 6.6	Parameter Set 2, Mean and Standard Deviation of the RMS State Errors: PF-EKF, PF, APF, EKF and SCKFF for Boxcar Type Neuronal Input.	57
Table 6.7	Parameter Set 3, Mean and Standard Deviation of the RMS State Errors: PF-EKF, PF, APF, EKF and SCKF for Gaussian Bump Type Neuronal Input.	57
Table 6.8	Parameter Set 3, Mean and Standard Deviation of the RMS State Errors: PF-EKF, PF, APF, EKF and SCKF for Boxcar Type Neuronal Input.	57
Table 7.1	Noise Statistics and Initial Values.	67
Table 7.2	RMS State Errors: Sample Mean and Standard Deviation.	68
Table 7.3	Simulation Results for the Parameter κ Estimates.	71
Table 7.4	Simulation Results for the Parameter τ Estimates.	71

Table 7.5 Simulation Results for the Parameter χ Estimates. 72
Table 7.6 RMS State Errors: Sample Mean and Standard Deviation. 72
Table 7.7 Noise Statistics and Initial Values. 94

LIST OF SYMBOLS

$\mathbb{E}[.]$	Expectation of the inner argument
I_n	n dimensional identity matrix
$\mathcal{N}(\mu; \Sigma)$	Gaussian density with mean μ and covariance Σ
R^N	N dimensional Euclidean space
x	n dimensional vector
$y_{1:N}$	the sequence of $y_1, y_2, ..., y_N$
$p(x)$	probability density function of x
$p(x, y)$	the joint probability density function of x and y
$p(x\|y)$	Conditional probability density function of x given y
$x \sim p(x)$	x is sampled from the probability density function of $p(x)$
$\hat{x}_{k\|k}$	Filtered State Estimate
$\hat{x}_{k\|N}$	Smoothed State Estimate
$\hat{P}_{k\|k}$	Filtered Covariance Estimate
$\hat{P}_{k\|N}$	Smoothed Covariance Estimate
$\|.\|$	Euclidean norm

LIST OF ABBREVIATIONS

APF	Auxiliary Particle Filter
BOLD	Blood Oxygen Level Dependent Signal
CKF	Cubature Kalman Filter
EKF	Extended Kalman Filter
EKS	Extended Kalman Smoother
IEKS	Iterative Extended Kalman Smoother
KF	Kalman Filter
KS	Kalman Smoother
PF-EKF	Particle Filter with EKF approximated proposal
PDF	Probability Density Function
RMS	Root Mean Square
SCKF	Square-root Cubature Kalman Filter
SCKS	Square-root Cubature Kalman Smoother
UKF	Unscented Kalman Filter

1. INTRODUCTION

The hemodynamic model describes a nonlinear relationship between the neuronal activity and observed Blood Oxygen Level Dependent (BOLD) signal. This nonlinear relationship is described by stochastic differential equations [1, 2, 3]. With most of the imaging techniques, it is impossible to obtain a direct measurement of the neuronal activity. After the fast neuronal activity, we observe a response in the hemodynamic variables like cerebral blood flow (CBF), cerebral blood volume (CBV) and blood deoxyhemoglobin content [4]. But even then we do not directly observe the hemodynamic variables. In the fMRI technique, we observe the BOLD signal, which is a nonlinear combination of the blood venus volume and blood deoxyhemoglobin content [5]. A typical BOLD signal can be seen in the Figure 1.1 Hence it is important to understand the nature of the hidden hemodynamic states from the observed BOLD signal.

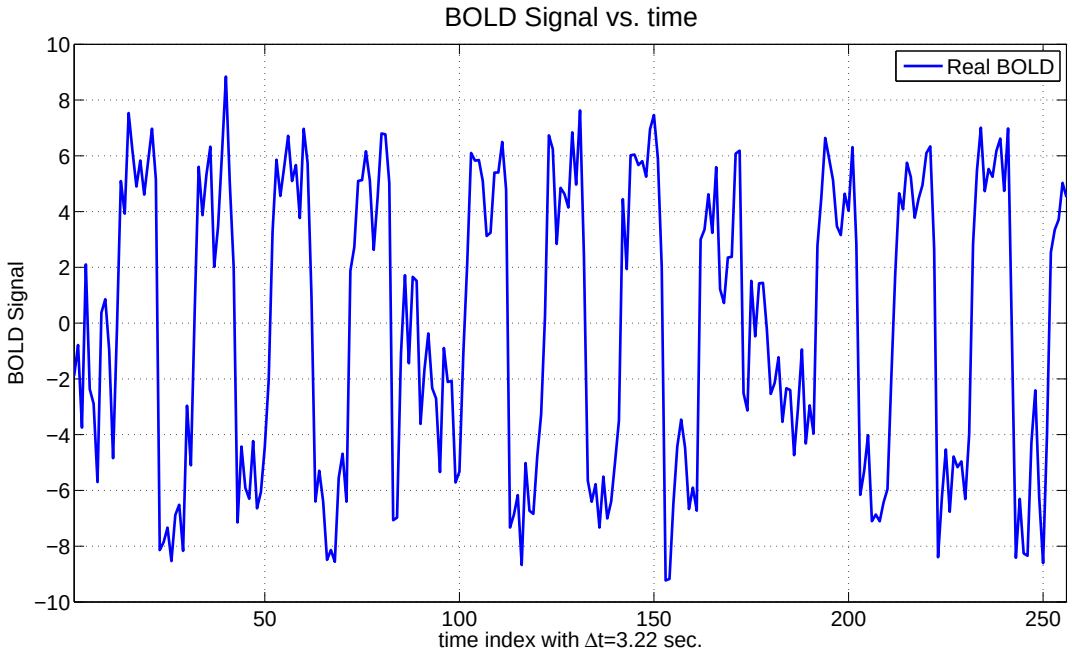

Figure 1.1 Real BOLD signal with the permission of Karl Friston.

In this thesis, we will be dealing with the problem of estimating the hemodynamic states: blood flow, blood volume, blood deoxyhemoglobin content and hemodynamic model parameters. Our approach will be based on the hemodynamic nonlinear dynamical system representation. Given the model, we will estimate the hemodynamic states and the parameters conditioned on the observed BOLD signal. More precisely, we will formulate the problem as a discrete-time state-space system:

$$x_{k+1} = f(\theta, x_k) + w_k \tag{1.1}$$

$$y_k = h(\theta, x_k) + v_k \tag{1.2}$$

where, f and h are nonlinear functions of the latent hemodynamic state x_k at time k. The state is a vector $x_k \in \mathbb{R}^4$. The BOLD signal at time k is described as $y_k \in \mathbb{R}$. The state transition noise w_k is Gaussian with $\mathcal{N}(0, Q_k)$ and the measurement noise v_k is Gaussian with $\mathcal{N}(0, R_k)$. By θ, we describe the set of parameters of the hemodynamic model. In Chapter 3, we give the details how this discrete time representation is obtained from the hemodynamic model. Having N observations $y_1, y_2, \cdots, y_N$ or $y_{1:N}$ and given the parameters θ, we will find the posterior distribution $p(x_{1:N}|y_{1:N}, \theta)$ from which we estimate the posterior mean for the hemodynamic states $E(x_{1:N}|y_{1:N}, \theta)$. As a harder problem we also work for unknown hemodynamic parameters. In this case, we want to estimate the parameters θ^* by treating them as slightly varying state variables. In Chapters 2, 4 and 5, more details are given regarding the mathematical theory of the state and parameter estimation.

The solution of this problem is also called the hemodynamic model inversion. The model inversion is a hard mathematical problem known as probabilistic inference. Most of the time, in practice to be applicable, it requires at least joint estimation of both latent hemodynamic state variables and parameters also known as system identification. The reason is that the hemodynamic response is not identical across the subjects, and even among different regions of the same subject [4, 6, 7]. In this thesis, we first work the case for the known parameter condition. We compared widely used methods. We provide a methodological improvement in the probabilistic inference of

the hemodynamic variables for a given neuronal input and system parameters. Subsequently, we work on the joint estimation of the hemodynamic model parameters and state estimation. For that purpose, we provide a simple, fast and an accurate model inversion technique.

The importance of the hemodynamic model inversion is manyfold for functional neuroimaging. Assuming that the hemodynamic model correctly describes the true behavior of the hemodynamic variables, it is surprising that just using the BOLD signal, we learn many things about the hidden temporal behavior of the cerebral blood flow, cerebral blood volume and blood de-oxyhemoglobin content. It is also possible to overcome the limits of temporal resolution of the hemodynamic response by identifying its inverted components on a finer time scale. From the clinical perspective, using the BOLD signal and model inversion techniques, we can estimate the system parameters. We can compare these parameters for both healthy and various diseased people. One other use is that we can monitor the change of the hemodynamic response and hemodynamic system parameters with respect to the medicament usage. We may identify parameter regions for healthy people. Depending on the parameter change with respect to the medicament usage, we may infer whether the medicament is useful or not.

1.1 Hemodynamic State Estimation

For the hemodynamic probabilistic inference, several attempts were presented in the fMRI model inversion literature. In the first attempts at applying the fMRI model inversion techniques, there was zero process noise in the state transition equations [1]. Friston et al. [1] modeled first the relation between the input and output by Volterra Kernels. Subsequently, they performed a Bayesian estimation technique to estimate the parameters [8]. Still, the assumption was zero state noise in the hemodynamic state equations. Riera et al. [9] utilized a type of extended Kalman filter (EKF). They introduced process noise in their method for the hemodynamic state equations. They performed EKF via the discretization method of Jimenez et al. [10]. They

did not, however, use the widely used Euler-Maruyama discretization method. UKF is performed by Hu et al. [11] for the system identification and state estimation of hemodynamic variables. Riera et al. [9] and Hu et al. [11] performed these techniques in a filtering style. Recently, Havlicek et al. [6] performed the square-root cubature Kalman filter (SCKF) for the hemodynamic state estimation. In this thesis, we also implemented SCKF, which is the numerical stable form of cubature Kalman filter.

In the fMRI literature, Johnston et al. [5] made direct usage of the particle filters. Murray and Storkey [12] implemented particle filters by using suboptimal proposal functions with a variety of discretization methods. They performed also particle smoothers.

For completeness, we also note the work of Friston et al. [13, 14, 15]. The algorithms they suggested are called dynamic expectation maximization (DEM), variational filtering (VF) and generalised filtering (GF). DEM, VF and GF are based on the variational approach. They perform dual estimation of the hemodynamic model. In these works, Friston et al. modified the variational approach for the dynamical systems [13, 14, 15]. In the variational approach, the joint probability density of the parameter set and the state conditioned on the observation sequence are assumed to be factorized in the parameter and state argument [16]. This assumption introduces simplicity in the calculation. At the same time, we note that this assumption is the source of error in the parameter and state estimation. Havlicek et al. [6] used square-root cubature Kalman smoother (SCKS) as the joint hemodynamic model parameter and state estimation. Under very low process noise conditions they also estimated the inputs.

1.2 Contribution

The hemodynamic model inversion is a hard nonlinear estimation problem. In this thesis, we have two contributions. The first contribution focuses on the model inversion under the known parameter condition. From the BOLD signal, by using the

functional representation of the hemodynamic model, we perform state estimation. We provide detailed comparison of the hemodynamic state estimation for Gaussian approximated and particle filter type algorithms under a wide range of process noise, different input, parameter sets and unkown initial condition. When we study the methods they make certain approximations. For example EKF, UKF, SCKF and LLF approximate the conditional probability density function of the state by Gaussian probability density function (pdf). For nonlinear problems this assumption is not proper and it introduces error in the state estimation. Particle filter methods are known to work with better numerical accuracy than Gaussian approximated approaches. However, for weakly-nonlinear systems particle filter do not benefit much. Particle filter performance also depends on the proposal functions from which samples are drawn. These samples aim to approximate the conditional probability density function of the state. Optimal proposal functions are difficult to calculate. In the hemodynamic state estimation literature, non-optimal proposal functions are used. In this thesis, we will examine the implementation and performance of an efficient state estimation technique for the fMRI signals. The method we apply approximates the optimal proposal function by extended Kalman filter method. The contribution and findings are:

- Contrary to the former claim, Extended Kalman filters estimate the hidden states better than particle filters under a wide range of noise conditions, different inputs, parameter sets and unknown initial condition.
- The hemodynamic model is weakly nonlinear system.
- PF-EKF can offer more accurate hemodynamic state estimation than the particle filter and auxiliary particle filters in most cases by approximating the optimal proposal by EKF. However, due to weak-nonlinearity of the hemodynamic model, Gaussian approximated methods are better in performance.

The second contribution we worked was a harder problem. We performed the joint estimation of the hemodynamic state and parameter estimation from the BOLD signal using the functional representation of the hemodynamic system. This kind

of estimation is also known as dual estimation. We proposed and implemented the usage of iterative extended Kalman smoother as the dual estimation. Regarding IEKS, contribution and findings are:

- The claim that EKF is not a robust model inversion [5] seems not to be valid for different process and measurement noise levels . We checked the EKF in the widely used parameter regime [6, 12, 13, 14, 15] and obtained results contrary to the literature.
- EKF's hemodynamic state estimation performance is even better than standard particle filters under a wide range of process and measurement noise conditions.
- EKS is robust and improves EKF especially in the higher hemoydnamic process/measurement noise conditions.
- IEKS substantially improves hemodynamic state estimation performance by utilizing the EKS algorithm iteratively.
- IEKS has better hemodynamic state estimation accuracy compared with iterative SCKS for different process and measurement noise conditions.
- IEKS has better parameter estimation accuracy compared with iterative SCKS for different hemodynamic model process and measurement noise conditions. The bias of the parameters is lower.
- IEKS is more than twice faster than iterative SCKS for the joint hemodynamic model inversion.
- Since SCKS alleged to be a better estimator among the other hemodynamic model inversion techniques [6], IEKS seems to be the best method among the hemodynamic model inversion methods.

1.3 Outline

The thesis is structured in the following order. In Chapter 2, we provide the conceptual overview of the estimation problem from the observed data using the functional representation of the system. In Chapter 3, we give the details of the functional representation of the hemodynamic model. In Chapter 4, we give the implementation details of the state estimation using the concepts explained in Chapter 2. This Chapter focuses on the filtering style of nonlinear state estimation. We compare for the hemodynamic model Gaussian approximated inferences and particle filter methods and provide the first methodological improvement called PF-EKF. In Chapter 5, we work on the joint estimation of the hemodynamic states and parameters. The nonlinear estimation in smoothing style is detailed. Parameter identification method is also discussed in this Chapter. We give the full details of the second contribution called IEKS. After presenting the methodological background in Chapter 6, we work out the PF-EKF method for the hemodynamic model for various process noise conditions. In Chapter 7, we implement the IEKS method for the hemodynamic model. We first perform model inversion using the simulated data. Having ground-truth validated the IEKS method, we also show its application in a rather complicated multi-input real BOLD signal. Conclusions and further recommendations are given in Chapter 8. The Appendix A provides the proof of Kalman approach of filtering and smoothing and the details of the QR decomposition.

2. ESTIMATION TECHNIQUES

In this section, we provide a review of state estimation techniques. We classify the algorithms under the Gaussian approximated and stochastic inference techniques. We give at the end of the section the scientific works done in the fMRI model inversion literature.

2.1 State Estimation

Due to nonlinearity of the system it is difficult to make probabilistic inferences from the observed data. In this thesis, our aim is to find the state estimation of type $p(x_k|y_{1:k})$, which is called filtering and $p(x_k|y_{1:N})$, which is called smoothing. In the filtering problem, for the time step k we take into account only the observations up to the time k. From these observations we calculate $p(x_k|y_{1:k})$. When the new observation arrives, we find the new conditional density $p(x_{k+1}|y_{1:k+1})$. Filtering disregards the information contained in the future values of y for the state x at time step k. However, in the smoothing we take into account all observed sequences. This provides us more accurate results for the state estimation, but at the cost of additional computation. Even the smoothing techniques require the calculation of the filtering. For that reason, we first concentrate on implementing a particular filtering technique to obtain the most accurate state estimation.

Once we have the expressions for $p(x_k|y_{1:k})$ and $p(x_k|y_{1:N})$ we can estimate the state sequence by taking the expectation of the variable x_k with respect to this probability distribution:

$$\mathbb{E}[x_k|y_{1:k}] = \int x_k p(x_k|y_{1:k}) dx_k \tag{2.1}$$

$$\mathbb{E}[x_k|y_{1:N}] = \int x_k p(x_k|y_{1:N}) dx_k \tag{2.2}$$

This result is the expected state conditioned on the observation sequence $y_{1:k}$ and $y_{1:N}$. Since it is impossible to calculate analytic expression for the $p(x_k|y_{1:k})$ and $p(x_k|y_{1:N})$, various approximation techniques are proposed.

2.2 Gaussian Approximated Inference

We have closed form expression for basic state-space systems that are linear and have Gaussian shaped noise. For such cases, Kalman filter [17] and smoother algorithms [18] work optimally. For linear systems, the form of the filtering and smoothing densities are expressed exactly as Gaussian densities.

$$p(x_k|y_{1:k}) = \mathcal{N}(\hat{x}_{k|k}; P_{k|k}) \tag{2.3}$$

$$p(x_k|y_{1:N}) = \mathcal{N}(\hat{x}_{k|N}; P_{k|N}) \tag{2.4}$$

Here $\hat{x}_{k|k}$ and $\hat{x}_{k|N}$ are the recursive filtered and smoothed state estimates at time k respectively. Similarly, $P_{k|k}$ and $P_{k|N}$ are the filtered and smoothed state covariances at time k.

Since we do not have closed form expressions for nonlinear systems, we have various approximation techniques. For Gaussian approximated methods, we can list extended Kalman filter (EKF) [19, 20], local linearization filter (LLF) [9], unscented Kalman filter (UKF) [21, 22, 23] and cubature Kalman filter (CKF) [24], which is a specific case of UKF [25]. These methods approximate conditional estimates of the states as Gaussian density.

$$p(x_k|y_{1:k}) \approx \mathcal{N}(\hat{x}_{k|k}; P_{k|k}) \tag{2.5}$$

For the extended Kalman filter algorithm, the state transition and measurement functions are linearized with a first-order Taylor series around the state estimates [6]. Approximating the nonlinear system with a linear state space form, the standard Kalman filter is applied. The LLF filter is the same as EKF except, in the prediction

update of the state, LLF uses the discretization proposed by Jimenez et al. [10]. In UKF and CKF, Gaussian densities are represented by deterministically chosen points called sigma points and cubature points for UKF and CKF, respectively [26]. For prediction and measurement updates of the states, the deterministic points are transferred from the nonlinear state transition and measurement functions, respectively. The positive aspect of UKF/CKF is that there is no need for the calculation of the Jacobian matrix of the state and measurement functions.

Similarly in the smoothing context various smoothing extensions are suggested for the above filtering methods. For the smoothing methods, they approximate the conditional estimates of the states again as Gaussian densities.

$$p(x_k|y_{1:N}) \approx \mathcal{N}(\hat{x}_{k|N}; P_{k|N}) \tag{2.6}$$

Extended Kalman smoother [18], uncented Kalman smoother [25] and cubature Kalman smoother [25] are the smoother extensions of the above filtered versions of EKF, UKF and CKF. Quite interestingly in the fMRI literature many of the Gaussian approximated filtering methods are not used in smoothing sense except for CKF Havlicek et al. [6].

2.3 Stochastic Inference

For stochastic inference, there are two broad categories of sampling algorithms: Markov chain Monte Carlo methods (MCMC) and sequential Monte Carlo (SMC) methods. Since we will be dealing with dynamical systems, in this thesis, we prefer to concentrate on SMC methods. We also give a literature survey about SMC methods used in fMRI at the end of this subsection.

Particle filters are sequential Monte Carlo methods that approximate the conditional density functions by a finite amount of particles and associated weights [27].

Monte Carlo methods aim to draw samples from complex target density functions $\pi(x)$. As a result, we represent the target function $\pi(x)$ with a discrete set of samples [28].

$$\pi(x) = \frac{1}{N}\sum_{i=1}^{N}\delta(x - x_i) \tag{2.7}$$

But most of the time, it is impossible to draw samples directly from complex density functions. The main algorithm to achieve this is called importance sampling, which lies at the hearth of the particle-type algorithms [28]. Importance sampling is performed by sampling from a known and easy-to-sample importance function $q(x)$. This necessitates compensating the weight due to discrepancy between the true density and the importance function. As a result, we have the following practically feasible form for generating samples:

$$\pi(x) = \sum_{i=1}^{N} w_i\delta(x - x_i) \tag{2.8}$$

where $w_i = \pi(x_i)/q(x_i)$ and x_i generated from $q(x)$ [28].

The good point of importance sampling is, if we have a test function ϕ, then we can make the approximation for large-enough N:

$$\int \phi(x)\pi(x)dx \approx \sum_{i=1}^{N} w_i\phi(x_i) \tag{2.9}$$

This is because bias and variance decreases asymptotically with increasing N [28].

The importance sampling algorithm can be easily modified for gradually increasing space dimensions. This is important, since in dynamical systems with each coming sequence, our space dimension increases. To handle such situations, the importance function is chosen so that it can be factorized as

$$q_{k+1}(x_{1:k+1}) = q_k(x_{1:k})q_{k+1}(x_{k+1}|x_{1:k}) \tag{2.10}$$

This expression tells us how to sample for the new dimension x_{k+1}. Having the previous sample points $x_{1:k}$, we just use the expression $q(x_{k+1}|x_{1:k})$ to sample for the new points. The weights are updated according to [28]:

$$w_{k+1} = w_k \frac{\pi_{k+1}(x_{1:k+1})}{\pi_k(x_{1:k}) q_{k+1}(x_{1:k+1}|x_{1:k})} \tag{2.11}$$

This algorithm, called Sequential Importance Sampling, has in practice a serious problem. After some time, only one of the weights of the particles becomes significant. As a result, practically there is only one sample that represents the conditional pdf [29]. This is a serious problem. To remedy this, Gordon proposed a resampling step and used the first working type of sequential Monte Carlo methods [27, 28]. In the resampling stage, from the unevenly distributed weighted samples, a new set of samples are obtained. By using this technique, we obtain a more stable approximation for successful estimation.

The above framework we explained is SMC, which works on a general setting. We did not specify the form of $\pi_k(x_{1:k})$. Those were the target densities we tried to approximate. Particle filter is just a special realization of the above SMC formulation for $\pi_k(x_{1:k}) = p(x_{1:k}|y_{1:k})$ [28].

2.4 Variational Approach

Yet another technique employed in fMRI literature is the variational approach. This is an approximation technique that was first suggested by Feynmann [30]. It was previously applied in statistical physics under the name of mean field theory [31]. For joint parameter and state estimation by the variational technique, the parameters and hidden states are assumed to be factorized. Subsequently, the difficult problem to find the posterior is approximated as:

$$p(\theta, x|y) \approx q(\theta)q(x) \tag{2.12}$$

Assuming such a factorization, then it is found that the individual factors are the expectations of the joint density $p(\theta, x, y)$ over other factors [16, 32, 33]. This is done iteratively until convergence. This principle of variational approach is worked in very detail for the dynamical systems by Friston et al. in a variety of works [13, 14, 15]. Through these algorithms, Friston et al. were able to determine the inputs, parameters and hidden states under certain noise conditions.

3. FRISTON-BUXTON HEMODYNAMIC MODEL

3.1 Functional Representation Overview

We work with the continuous time state space and observation formulation. Subsequently we will convert the model to the discrete time equivalent form. Hence, we will describe the system in the most general setting as:

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

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

Here, g is the nonlinear state transition function, h is the nonlinear measurement function. Both the functions have arguments as the hemodynamic state x_t at time t, parameter set θ and neuronal input $u(t)$. Hemodynamic state x_t at time t is a four-dimensional vector $x_t \in \mathbb{R}^4$. The BOLD signal y_t at time t is described as $y_t \in \mathbb{R}$.

Similar to Johnston et al. [5] we will perturb the system with Wiener noise and discretize the system by the Euler-Maruyama Method and arrive the discrete time form of the above nonlinear differential equation. We define the discrete time instants as $t = t_k \triangleq k\Delta t$, $k = 1, 2, \ldots$. The state variables, input and measured BOLD signals are discretized by defining $x_{k+1} = x(t_k + \Delta t)$, $x_k = x(t_k)$, $u_k = u(t_k)$, $y_k = y(t_k)$. By using these definitions we arrive the following nonlinear discrete time state-space model.

$$x_{k+1} = f(x_k, \theta, u_k) + w_k \tag{3.3}$$

$$y_k = h(x_k, \theta, u_k) + v_k \tag{3.4}$$

where f is:

$$f(x_k, \theta, u_k) = x_k + \Delta t g(x_k, \theta, u_k) \tag{3.5}$$

and

$$w_k = \mathcal{N}(0, Q_k) \tag{3.6}$$

Here, f is the nonlinear state transition function and h is the nonlinear measurement function. This time, we have the discretized version of the state x_k where the time index is k. The state is a vector $x_k \in \mathbb{R}^4$. The measured BOLD signal y_k is from the set $y_k \in \mathbb{R}$. The state noise w_k is of Gaussian type with $\mathcal{N}(0, Q_k)$. The measurement noise v_k is also Gaussian type with $\mathcal{N}(0, R_k)$. We denote by $y_1, y_2, \cdots, y_N$ or $y_{1:N}$ the discrete time observation sequence of length N. Given the neuronal input $u_{1:N}$ and plausible assumptions about the noise and N observations $y_1, y_2, \cdots, y_N$ or $y_{1:N}$, we want to estimate $x_{1:N}$.

3.2 Hemodynamic Model System Representation

In this section, we analyze the hemodynamic model in detail. We describe the dynamics first without any noise component as in [1]. Subsequently, we log-transform those state variables that describe actual physical entities like blood flow, blood vessel volume and deoxyhemoglobin content [34]. We perturb the system with the Wiener process. The last step is the Euler-Maruyama discretization of the SDE. The selected size of the discretization step is small enough to maintain the dynamics of the hemo-

dynamic model as in [5, 9]. The noise statistics are compatible with [6].

We will follow the Buxton-Friston hemodynamic model, which relates the neuronal activity to hemodynamic variables and eventually to the observed Blood Oxygen Level Dependent (BOLD) signal. The model consists of the neuronal activity $u(t)$, the blood flow h_2, blood vessel volume h_3 and the deoxyhemoglobin content h_4. After the neuronal activity $u(t)$, which is usually a rapid signal, a change in the blood flow h_2 is observed. After the flow change of the blood, changes in the blood venus volume h_3 and deoxyhemoglobin content h_4 are observed [1, 3, 8, 35]. The relation between the blood flow h_2 and neuronal activity $u(t)$ is a second order differential equation [11].

$$\ddot{h_2}(t) = \epsilon u(t) - \kappa \dot{h_2}(t) - \chi(h_2(t) - 1) \tag{3.7}$$

$$\dot{h_3}(t) = \tau(h_2(t) - F(h_3(t))) \tag{3.8}$$

$$\dot{h_4}(t) = \tau(h_2(t)E(h_2(t)) - F(h_3(t))\frac{h_4(t)}{h_3(t)}) \tag{3.9}$$

where

$$F(h_3(t)) = h_3(t)^{1/\alpha} \tag{3.10}$$

$$E(h_2(t)) = \frac{1}{\varphi}(1 - (1 - \varphi)^{1/h_2(t)}) \tag{3.11}$$

Here, ϵ is the neuronal efficacy factor, κ is a measure for signal decay h_1, τ is denoted as the transit time, χ is a measure for negative feedback of the blood flow h_2, α is Grubb's exponent and φ is the resting oxygen extraction fraction [6].

By introducing an abstract variable vasodilatory signal h_1, we work on the four-dimensional differential equation system.

Table 3.1
Parameter Values of the Hemodynamic State Transition Equations

Parameter	**Description**	**Value**
κ	Rate of Signal Decay	0.65
τ	Hemodynamic Transit Time	1.0204
χ	Rate of Flow Dependent Elimination	0.41
α	Grubb's exponent	0.32
φ	Resting Oxygen Extraction Fraction	0.34
ϵ	Neural Efficiency	0.5

Table 3.2
Parameter Values of the Hemodynamic Observation Equation

Parameter	**Description**	**Value**
V_0	Resting Blood Volume Fraction	0.04
k_1	Intravascular Coefficient	7φ
k_2	Concentration Coefficient	2
k_3	Extravascular Coefficient	$2\varphi - 2$

$$\dot{h_1}(t) = \epsilon u(t) - \kappa h_1(t) - \chi(h_2(t) - 1) \tag{3.12}$$

$$\dot{h_2}(t) = h_1(t) \tag{3.13}$$

$$\dot{h_3}(t) = \tau(h_2(t) - F(h_3(t))) \tag{3.14}$$

$$\dot{h_4}(t) = \tau(h_2(t)E(h_2(t)) - F(h_3(t))\frac{h_4(t)}{h_3(t)}) \tag{3.15}$$

where

$$F(h_3(t)) = h_3(t)^{1/\alpha} \tag{3.16}$$

$$E(h_2(t)) = \frac{1}{\varphi}(1 - (1 - \varphi)^{1/h_2(t)}) \tag{3.17}$$

We note that all the hemodynamic variables and the parameters are not directly observable. What we observe is just the BOLD signal, which is a nonlinear combination of blood vessel volume h_3 and deoxyhemoglobin content h_4. The observed BOLD signal equation is as follows:

$$y(t) = V_0(k_1(1 - h_4(t)) + k_2(1 - \frac{h_4(t)}{h_3(t)}) + k_3(1 - h_3(t))) \tag{3.18}$$

Here, k_1 is the intravascular coefficient, k_2 is the concentration coefficient and k_3 is the extravascular coefficient respectively and V_0 is resting blood volume fraction [3, 36, 37].

In this thesis, we perform model inversion under both known and unknown parameter condition. For the known parameter case, the parameters for the hemodynamic model and BOLD signal equation are taken as in Table 6.2 and Table 3.2, which is in accordance with [6]. Our present focus is on the hidden hemodynamic states induced

by (known) experimental or neuronal input.

We perform the nonlinear transform $x_i(t) = \log(h_i(t))$ to the x_2,x_3 and x_4, which ensures positive values for the hemodynamic states x_2,x_3 and x_4 [6, 14]. Subsequently, we utilize Euler-Maruyama method to obtain the following discrete form of the nonlinear state-space model.

$$x_1(t+\Delta t) \approx x_1(t) + \Delta t(\epsilon u(t) - \kappa x_1(t) - \chi(e^{x_2(t)} - 1)) + \sqrt{\Delta t}\beta_1(t) \tag{3.19}$$

$$x_2(t+\Delta t) \approx x_2(t) + \Delta t(\frac{x_1(t)}{e^{x_2(t)}}) + \sqrt{\Delta t}\beta_2(t) \tag{3.20}$$

$$x_3(t+\Delta t) \approx x_3(t) + \Delta t(\frac{\tau(e^{x_2(t)} - F(e^{x_3(t)}))}{e^{x_4(t)}}) + \sqrt{\Delta t}\beta_3(t) \tag{3.21}$$

$$x_4(t+\Delta t) \approx x_4(t) + \Delta t(\frac{\tau(e^{x_2(t)}E(e^{x_2(t)}) - F(e^{x_3(t)})\frac{e^{x_4(t)}}{e^{x_3(t)}})}{e^{x_4(t)}}) + \sqrt{\Delta t}\beta_4(t) \tag{3.22}$$

Hence we arrived at the final form for the discrete time state-space model:

$$x_{k+1} = f(\theta, x_k) + w_k \tag{3.23}$$

$$y_k = h(\theta, x_k) + v_k \tag{3.24}$$

where the state transition function $f(\theta, x_k)$ and the measurement function $h(\theta, x_k)$ are given as follows:

$$f(\theta, x_k) = \begin{bmatrix} x_{k,1} + \Delta t(\epsilon u_k - \kappa x_{k,1} - \chi(e^{x_{k,2}} - 1)) \\ x_{k,2} + \Delta t(\frac{x_{k,1}}{e^{x_{k,2}}}) \\ x_{k,3} + \Delta t(\frac{\tau(e^{x_{k,2}} - F(e^{x_{k,3}}))}{e^{x_{k,4}}}) \\ x_{k,4} + \Delta t(\frac{\tau(e^{x_{k,2}}E(e^{x_{k,2}}) - F(e^{x_{k,3}})\frac{e^{x_{k,4}}}{e^{x_{k,3}}})}{e^{x_{k,4}}}) \end{bmatrix} \tag{3.25}$$

$$h(\theta, x_k) = V_0(k_1(1 - e^{x_{k,4}}) + k_2(1 - \frac{e^{x_{k,4}}}{e^{x_{k,3}}}) + k_3(1 - e^{x_{k,3}})) \tag{3.26}$$

Here, $x_{k,i}$ for $i = 1, 2, 3, 4$ are the individual components of the discrete state vector x_k. The noise components w_k and v_k are Gaussian densities with $\mathcal{N}(0, Q_k)$ and $\mathcal{N}(0, R_k)$, respectively.

3.3 Neuronal Input and Hemodynamic Responses

In this section, we examine the hemodynamic responses for the basic type neuronal inputs. Neuronal input is modeled as the standard boxcar function. During the activation time, the boxcar function takes the value 1, whereas in the non-activated neuronal input the boxcar function takes the value 0. We examine both short time and long time activated neuronal input. Typical graph is shown in Figure 3.1. We note that, in our plots, following the style of Friston et al. and Havlicek et al. we plot the log-transformed hemodynamic variables as in [6, 13, 14, 15].

We simulated the hemodynamic response using the nonlinear state transition function given in Eq. 3.25 and the measurement function given in Eq. 3.26. The associated responses of the hemodynamic variables and the BOLD signal are plotted in Figures 3.2 and 3.3.

For longer activated neurons we observe saturation affect as in [5]. For that purpose, we increased the activation time as can be seen from Figure 3.4. The saturation affect of the hemodynamic variable and the BOLD signal is shown in Figures 3.5 and 3.6.

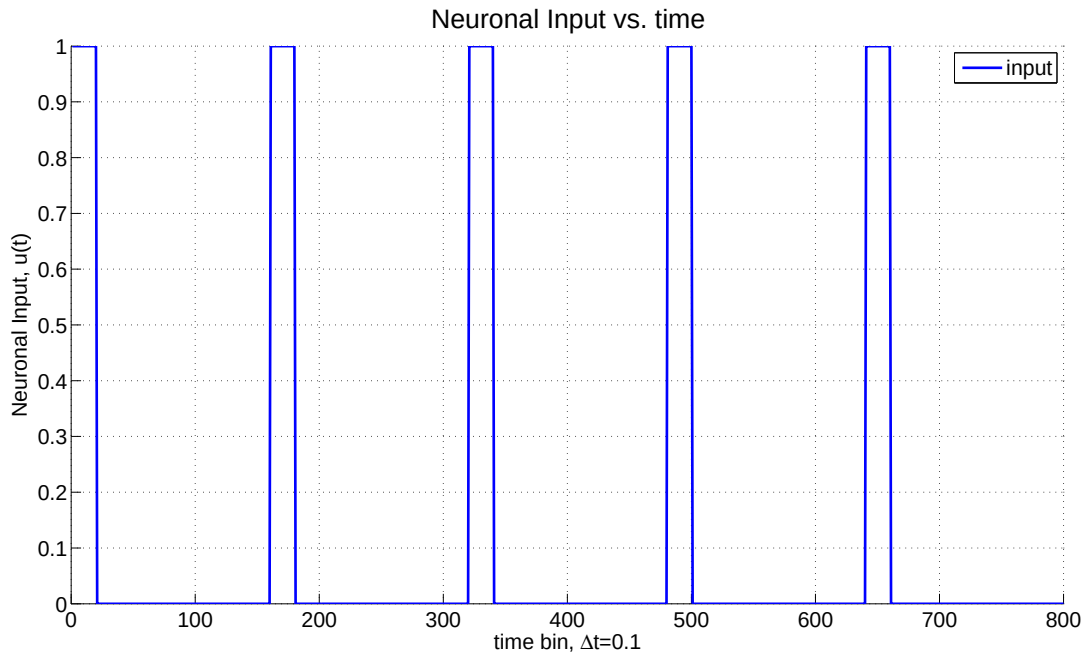

Figure 3.1 Boxcar neuronal input with short time activation.

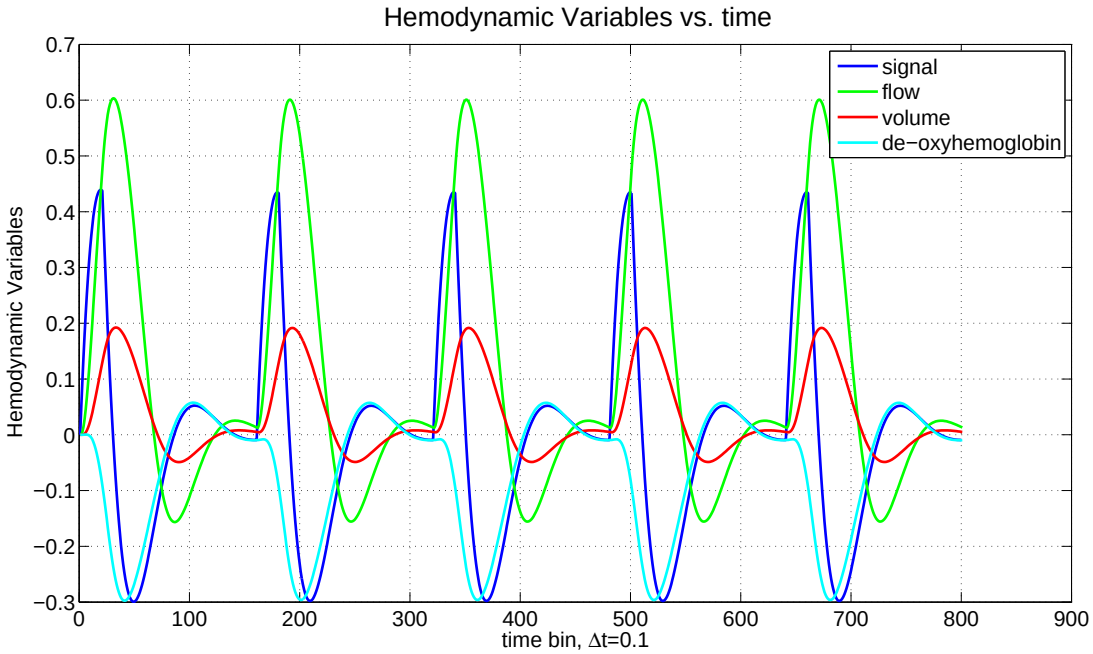

Figure 3.2 Hemodynamic variables for boxcar neuronal input with short time activation.

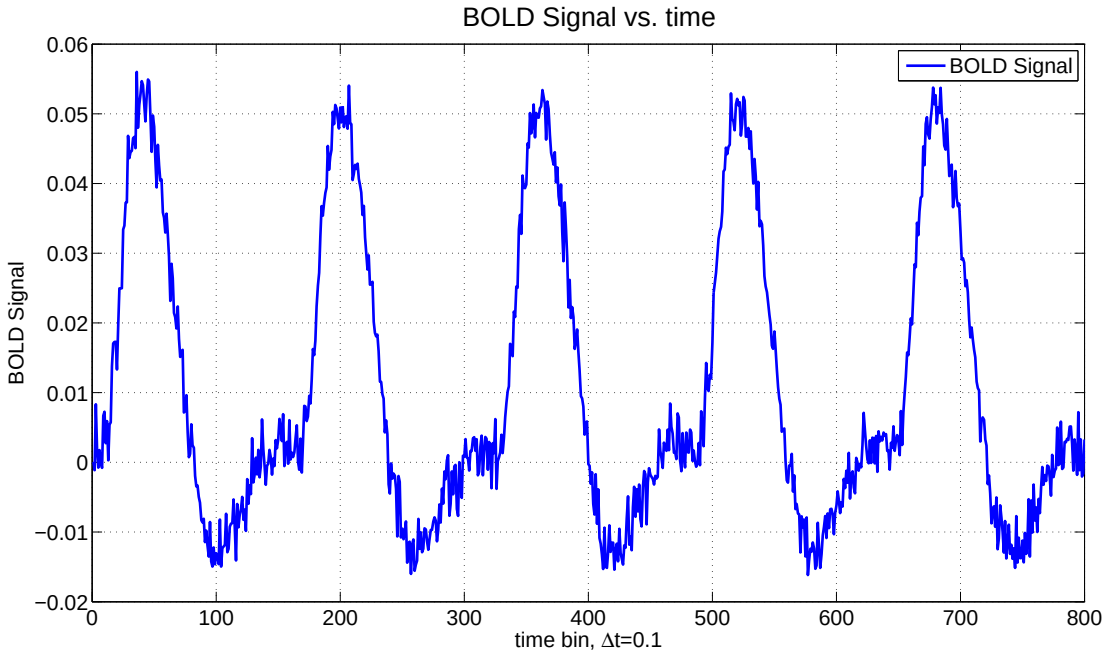

Figure 3.3 BOLD signal for boxcar neuronal input with short time activation.

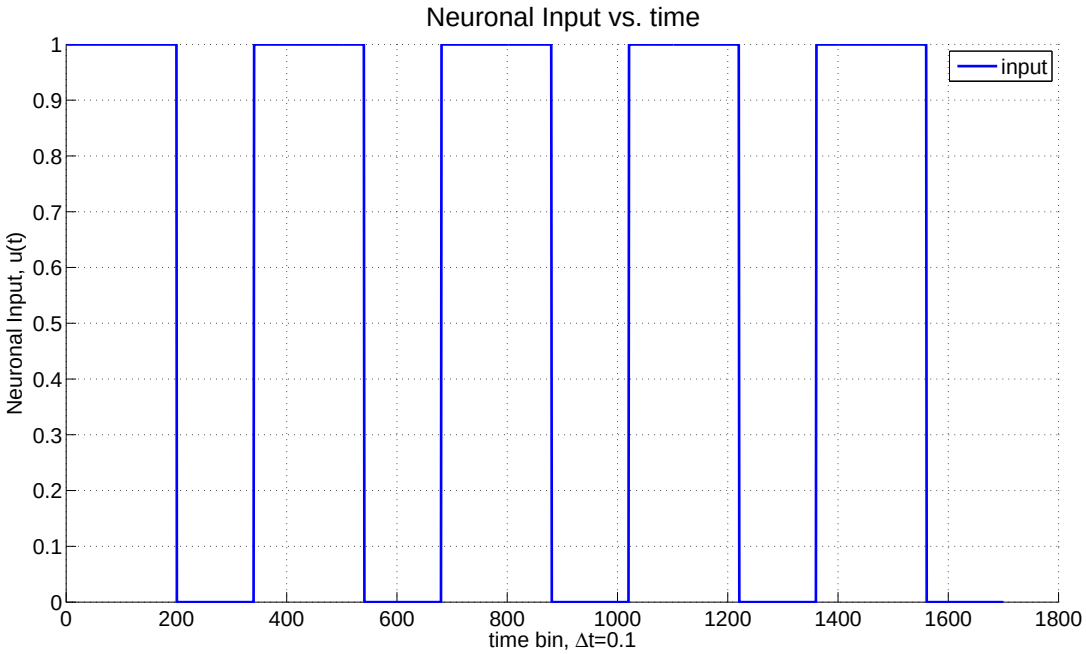

Figure 3.4 Boxcar neuronal input with long time activation.

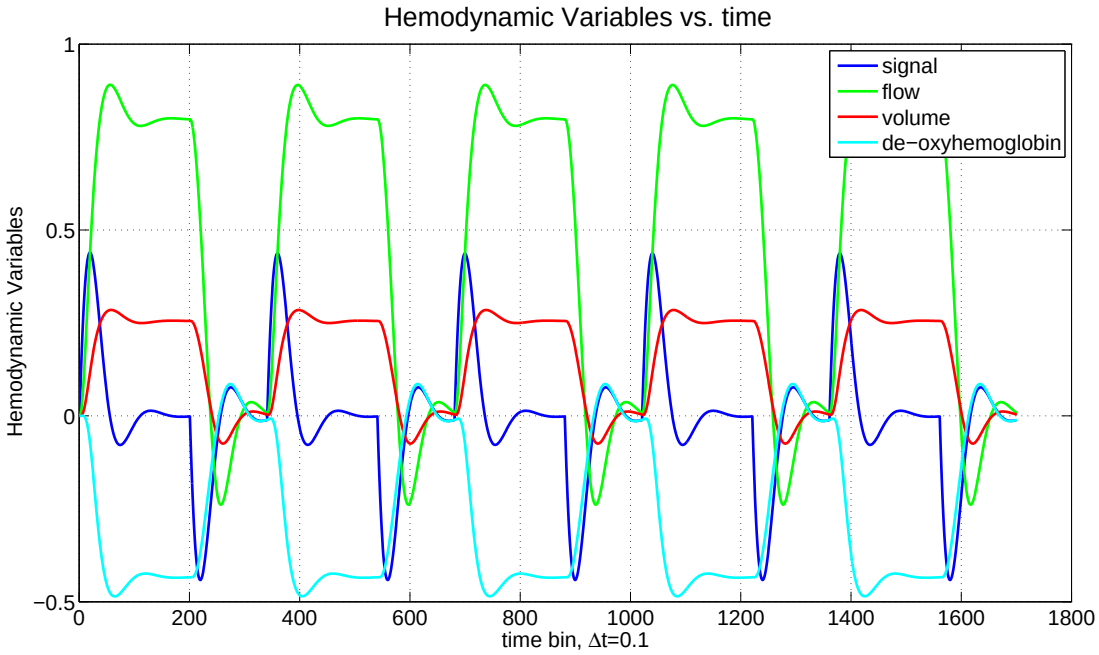

Figure 3.5 Hemodynamic variables for boxcar neuronal input with long time activation.

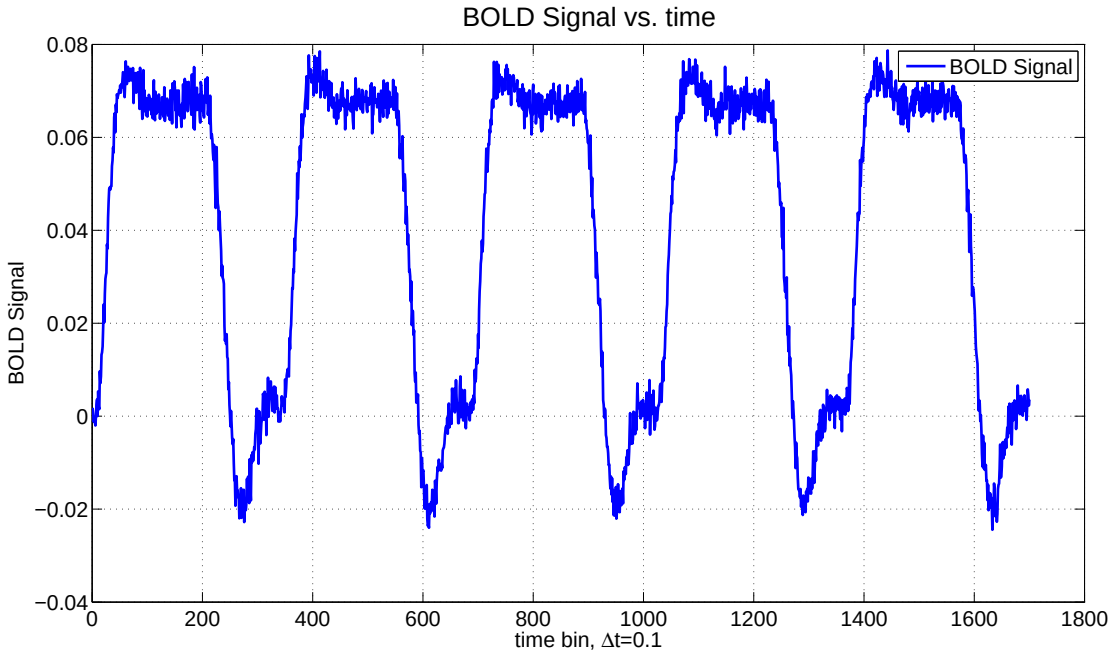

Figure 3.6 BOLD signal for boxcar neuronal input with long time activation.

4. NONLINEAR STATE ESTIMATION

In this Chapter, we will see the implementation details of the theory presented in Chapter 2. As pointed earlier, for nonlinear problems - in most of the cases -, it is impossible to find an analytical expression for the posteriors $p(x_k|y_{1:k})$ and $p(x_k|y_{1:N})$. For that purposes, two types of approximation methods are used.

1. Gaussian approximated methods
2. Stochastic type approximation methods

4.1 Gaussian Approximated Inference Methods

In this subsection, first the details of the Kalman filter, extended Kalman filter and square-root cubature Kalman filter are given. The extended Kalman filter and cubature Kalman filter algorithms are classified under the Gaussian approximated inference algorithms. They approximate the filtered state estimate $p(x_k|y_{1:k})$ by Gaussian probability density functions. Since densities are Gaussian, both of the algorithms iteratively update the mean and covariance of the state estimates.

4.1.1 Kalman Filter

Let us consider that we have the following linear and Gaussian additive state space representation.

$$x_{k+1} = Ax_k + w_k \tag{4.1}$$

$$y_k = Cx_k + v_k \tag{4.2}$$

Here x_k at time k is a n-dimensional vector $x_k \in \mathbb{R}^n$. The measurement signal y_k at time k is l-dimensional vector described as $y_k \in \mathbb{R}^l$. A is a matrix with $A \in \mathbb{R}^{n\mathbf{x}n}$. C is a matrix with $C \in \mathbb{R}^{l\mathbf{x}n}$. The state noise w_k is of Gaussian type with $\mathcal{N}(0, Q_k)$. The measurement noise v_k is also Gaussian type with $\mathcal{N}(0, R_k)$. Our aim is to find $p(x_k|y_{1:k})$. The solution to this problem can be found analytically. This density is a Gaussian pdf which is expressed as:

$$p(x_k|y_{1:k}) = \mathcal{N}(\hat{x}_{k|k}; P_{k|k}) \tag{4.3}$$

The conditional density of x_k given the measurements up-to time k is a Gaussian with mean $\hat{x}_{k|k}$ and covariance $P_{k|k}$. The optimum solution for the filter is given by Kalman filter algorithm. With each new coming observation y_{k+1}, Kalman filter updates the mean and covariance sequentially.

Kalman Filter Algorithm is composed of the following steps.

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} \tag{4.4}$$

Here $\hat{x}_{k-1|k-1}$ is the predicted state mean at time $k-1$ conditioned on the observation sequence $y_{1:k-1}$ and $\hat{x}_{k|k-1}$ is the predicted state mean at time k. For the predicted state mean $\hat{x}_{k|k-1}$, we just make a predicton before the observation y_k arrives. For linear systems, we just propagate the predicted state mean $x_{k-1|k-1}$ by multiplying with the state transition matrix A.

Similar to mean update, for the prediction update of the state covariance $P_{k|k-1}$ we perform the following step:

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q_{k-1} \tag{4.5}$$

Here $P_{k-1|k-1}$ is the state covariance prediction at time $k-1$ conditioned on the observation sequence $y_1, y_2, ..., y_{k-1}$. With these two steps, we calculate the mean and

covariance of the predicted state probability density function $p(x_k|y_{1:k-1})$. This density is also a Gaussian pdf with

$$p(x_k|y_{1:k-1}) = \mathcal{N}(\hat{x}_{k|k-1}; P_{k|k-1}) \tag{4.6}$$

This concludes the prediction step update. Next at time k, we have the new observation y_k. Our aim is to update the condtional density $p(x_k|y_{1:k})$. For that purpose, first we calculate the so-called innovation v_k.

$$v_k = y_k - C\hat{x}_{k|k-1} \tag{4.7}$$

The logic for the innovation is simple. From the predicted state $\hat{x}_{k|k-1}$, we calculate the predicted response. The difference between the real response and the predicted response is innovation v_k. Next we calculate the measurement prediction covariance S_k called and the Kalman gain K_k.

$$S_k = CP_{k|k-1}C^T + R_k \tag{4.8}$$

$$K_k = P_{k|k-1}C^T S_k^{-1} \tag{4.9}$$

From these three quantities: innovation v_k, measurement prediction covariance S_k and Kalman gain K_k, we estimate the mean of the state $\hat{x}_{k|k}$ according to:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k v_k \tag{4.10}$$

The next step is to estimate the state covariance $P_{k|k}$.

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T \tag{4.11}$$

For the proof of this nontrivial equation, the reader is referred to the Appendix A. We summarize the Kalman Filter in Algorithm 1.

Algorithm 1 Kalman Filter

Initialize:

Initialize the mean estimate $\hat{x}_{0|0}$ and covariance estimate $P_{0|0}$.

for k=1 to N **do**

Prediction Step:

$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1}$

$P_{k|k-1} = AP_{k-1|k-1}A^T + Q_{k-1}$

Measurement Step:

$v_k = y_k - C\hat{x}_{k|k-1}$

$S_k = CP_{k|k-1}C^T + R_k$

$K_k = P_{k|k-1}C^T S_k^{-1}$

$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k v_k$

$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T$

end for

4.1.2 Extended Kalman Filter

For the EKF algorithm, the system is linearized in the estimated state values. After the linearization step, the standard Kalman filter algorithm is applied, which is known to operate optimally in linear systems [17]. So the extended Kalman filter is a kind of approximation technique. It has the further assumption that the filtered and the predicted states can be approximated by Gaussian densities:

$$p(x_k|y_{1:k}) \approx \mathcal{N}(\hat{x}_{k|k}, P_{k|k}) \tag{4.12}$$

$$p(x_{k+1}|y_{1:k}) \approx \mathcal{N}(\hat{x}_{k+1|k}, P_{k+1|k}) \tag{4.13}$$

Here, $\hat{x}_{k|k}$ and $\hat{x}_{k+1|k}$ are the filtered and the predicted mean values of the densities, whereas $P_{k|k}$ and $P_{k+1|k}$ are the filtered and the predicted covariances of the states.

Subsequently, the EKF algorithm recursively updates these statistics as follows:

Prediction update:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}) \tag{4.14}$$

$$P_{k|k-1} = F(\hat{x}_{k-1|k-1})P_{k-1|k-1}F(\hat{x}_{k-1|k-1})^T + Q_{k-1} \tag{4.15}$$

Hence, the state update $\hat{x}_{k|k-1}$ is done by propagating the estimate $\hat{x}_{k-1|k-1}$ through the nonlinear state equation. The analytic calculation of the the predicted covariance $P_{k|k-1}$ is not possible most of the time. In order to calculate the predicted covariance $P_{k|k-1}$, the state equation is linearized at the estimated state value $\hat{x}_{k-1|k-1}$. The Jacobian matrix of f at the state value x is denoted by F with the component values evaluated as:

$$F_{ij}(x) = \frac{\partial f(x_i)}{\partial x_j} \tag{4.16}$$

We note that the partial differentiations are implemented by simple first order Euler numerical approximation. This concludes the prediction step. The next step is the measurement update. The first step is to calculate the so-called innovation v_k.

Measurement update:

If there is a measurement at the time k, we perform the updates similar to those for the Kalman filter. In order to calculate the Kalman gain, K_k, the measurement function h is linearized at the predicted state $\hat{x}_{k|k-1}$ similar to the prediction step. The Jacobian matrix of h at the state value x is denoted by H with the component values evaluated as:

$$H_{ij}(x) = \frac{\partial h(x_i)}{\partial x_j} \tag{4.17}$$

We evaluate the partial differentiation in the Jacobian matrix with first order Euler

approximation. Innovation v_k, measurement prediction covariance S_k and Kalman gain K_k is calculated as:

$$v_k = y_k - h(\hat{x}_{k|k-1}) \tag{4.18}$$

$$S_k = H(\hat{x}_{k|k-1})P_{k|k-1}H(\hat{x}_{k|k-1})^T + R_k \tag{4.19}$$

$$K_k = P_{k|k-1}H(\hat{x}_{k|k-1})^T S_k^{-1} \tag{4.20}$$

$$\tag{4.21}$$

Having the innovation v_k, measurement prediction covariance S_k and Kalman gain K_k the same update formula as in Kalman filter is applied:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k v_k \tag{4.22}$$

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T \tag{4.23}$$

We summarize the extended Kalman filter in Algorithm 2.

Algorithm 2 Extended Kalman Filter

Initialize:

Initialize the mean estimate $\hat{x}_{0|0}$ and covariance estimate $P_{0|0}$.

for k=1 to N **do**

Prediction Step:

$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1})$

$P_{k|k-1} = F(\hat{x}_{k-1|k-1})P_{k-1|k-1}F(\hat{x}_{k-1|k-1})^T + Q_{k-1}$

Measurement Step:

$v_k = y_k - h(\hat{x}_{k|k-1})$

$S_k = H(\hat{x}_{k|k-1})P_{k|k-1}H(\hat{x}_{k|k-1})^T + R_k$

$K_k = P_{k|k-1}H(\hat{x}_{k|k-1})^T S_k^{-1}$

$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k v_k$

$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T$

end for

We note that in case there is no measurement available at time k, we leave the

estimated state $\hat{x}_{k|k}$ and estimated covariance $P_{k|k}$ to the same values in the prediction update step.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} \tag{4.24}$$

$$P_{k|k} = P_{k|k-1} \tag{4.25}$$

4.1.3 Square-Root Cubature Kalman Filter

In the literature, it is reported that in some highly nonlinear systems EKF can diverge [24]. In order to circumvent such problems recently a method called cubature Kalman filter was suggested [24]. In this technique, the filtered state density $p(x_k|y_{1:k})$ and predicted state density $p(x_{k+1}|y_{1:k})$ are again approximated by Gaussian densities. However these densities are further approximated with deterministically chosen the so-called cubature points. Using these points, state and measurement updates are performed very similar to the Kalman Filter algorithm. In the Kalman algorithm, the covariance matrix is updated in every step. However, the two properties of the covariance matrix, symmetry and positive definiteness, must be preserved [24]. In nonlinear problems, there are several reasons which cause these two properties to break down [29]. This problem results in the divergence of the algorithm. There are various ad-hoc methods to circumvent this issue. One stable solution is the square-root formulation [24]. In the square-root formulation, the square root of the error covariance matrix is updated in every step.

Recently, Havlicek et al. [6] used square-root cubature Kalman filter (SCKF) with the smoother to make inference of the state, parameter and even input. He reported quite satisfactory results and even better performance than the DEM algorithm under certain noise conditions. DEM is a variational based approach suggested by Friston et al. [14] which will not be discussed in this thesis. In the work [6], square-root cubature Kalman smoother is applied by augmenting parameters with the state. In this thesis, for comparison purposes, we used SCKF without augmenting the noise with the state, contrary to the method in [6]. The reason is that we already assume

correct noise statistics. Moreover, the noise augmentation is not suggested for additive noises [25].

As we mentioned, cubature Kalman filter assumes the filtered state estimate $p(x_k|y_{1:k})$ and the predicted state estimate $p(x_{k+1}|y_{1:k})$ have the Gaussian type probability density functions.

$$p(x_k|y_{1:k}) = \mathcal{N}(\hat{x}_{k|k}, P_{k|k}) \tag{4.26}$$

$$p(x_{k+1}|y_{1:k}) = \mathcal{N}(\hat{x}_{k+1|k}, P_{k+1|k}) \tag{4.27}$$

These densities are approximated by utilizing a set of deterministic points called cubature points [24]. This technique resembles the UKF algorithm. In UKF, these preselected points are called sigma points [22]. Actually, the cubature points are kind of sigma points by setting the parameters of UKF to specific values [25]. The square root of $P_{k-1|k-1}$ is denoted as $S_{k-1|k-1}$. We assume at time $t = k-1$ state estimate $\hat{x}_{k-1|k-1}$ and square-root covariance $S_{k-1|k-1}$ is given. Subsequently, we update the state $\hat{x}_{k|k}$ and square-root covariance estimate $S_{k|k}$ using SCKF Algorithm as follows:

Prediction Step:

1) By using $S_{k-1|k-1}$, the cubature points $X_{i,k-1|k-1}$ are found:

$$X_{i,k-1|k-1} = S_{k-1|k-1}\xi_i + \hat{x}_{k-1|k-1} \tag{4.28}$$

where $\xi = \sqrt{n}[I_n, -I_n]$, ξ_i is the i-th column of ξ for $i = 1, 2, \cdots, 2n$ and I_n is the n-dimensional identity matrix.

2) Next the cubature points $X_{i,k-1|k-1}$ are propagated through the state update function, f:

$$X^*_{i,k|k-1} = f(X_{i,k-1|k-1}, u_{k-1}) \tag{4.29}$$

Here u_k stands for the input sequence at time k.

3) From these predicted cubature points, mean of the predicted state estimate $\hat{x}_{k|k-1}$ is calculated by the formula:

$$\hat{x}_{k|k-1} = \frac{1}{m}\sum_{i=1}^{m} X^*_{i,k|k-1} \tag{4.30}$$

where $m = 2n$.

4) For the Square-Root Error covariance $S_{k|k-1}$:

$$X_{k|k-1} = \frac{1}{\sqrt{m}}\left[X^*_{1,k|k-1} - \hat{x}_{k|k-1} \quad X^*_{2,k|k-1} - \hat{x}_{k|k-1} \quad \cdots \quad X^*_{m,k|k-1} - \hat{x}_{k|k-1}\right] \tag{4.31}$$

$$S_{k|k-1} = qr([X_{k|k-1} \;\; S_{Q,k-1}]) \tag{4.32}$$

Here $S_{Q,k-1}$ is the square root of the covariance Q_{k-1} and qr is the QR decomposition [6]. $S = qr(X)$ results in the lower triangular matrix as used by [6, 24]. For QR decomposition, please check the Appendix A.3. This concludes the prediction step. Subsequently, we present the measurement step.

Measurement Update:

When there is a measurement at time k, than the updates are:

1) From the predicted mean $\hat{x}_{k|k-1}$ and predicted square-root covariance $S_{k|k-1}$, we build the new cubature points $X_{i,k|k-1}$:

$$X_{i,k|k-1} = S_{k|k-1}\xi_i + \hat{x}_{k|k-1} \tag{4.33}$$

for $i = 1, 2, \cdots, 2n$

2) In order to find the innovation, the cubature points $X_{i,k|k-1}$ are propagated through the observation equation:

$$Z_{i,k|k-1} = h(X_{i,k|k-1}, u_k) \tag{4.34}$$

3) From the propagated cubature points $Z_{i,k|k-1}$, mean predicted measurement estimate

$\hat{z}_{k|k-1}$ is calculated:

$$\hat{z}_{k|k-1} = \frac{1}{m}\sum_{i=1}^{m} Z_{i,k|k-1} \tag{4.35}$$

4) Similarly the square root of the innovation covariance matrix $S_{zz,k|k-1}$ is calculated as follows:

$$Z_{k|k-1} = \frac{1}{\sqrt{m}}\left[Z_{1,k|k-1} - \hat{z}_{k|k-1} \quad Z_{2,k|k-1} - \hat{z}_{k|k-1} \quad \cdots \quad Z_{m,k|k-1} - \hat{z}_{k|k-1}\right] \tag{4.36}$$

$$S_{zz,k|k-1} = qr([Z \;\; S_{R,k}]) \tag{4.37}$$

Here $S_{R,k}$ is the square root of the covariance R_k.

6) In order to evaluate the Kalman gain, the cross covariance $P_{xz,k|k-1}$ is calculated as follows:

$$P_{xz,k|k-1} = X_{k|k-1}Z_{k|k-1}^{T} \tag{4.38}$$

where

$$X_{k|k-1} = \frac{1}{\sqrt{m}}\left[X_{1,k|k-1} - \hat{x}_{k|k-1} \quad X_{2,k|k-1} - \hat{x}_{k|k-1} \quad \cdots \quad X_{m,k|k-1} - \hat{x}_{k|k-1}\right] \tag{4.39}$$

7) The Kalman gain:

$$K_k = P_{xz,k|k-1}S_{zz,k|k-1}^{-T}S_{zz,k|k-1}^{-1} \tag{4.40}$$

8) The correction in the predicted estimate $\hat{x}_{k|k-1}$ is done by using the Kalman gain.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - \hat{z}_{k|k-1}) \tag{4.41}$$

9) For the measurement filtered estimate of the square-root covariance matrix $S_{k|k}$ is:

$$S_{k|k} = qr([X_{k|k-1} - K_kZ_{zz,k|k-1} \;\; K_kS_{R,k}]) \tag{4.42}$$

For the proof of this last step the reader is referred to [24]. We summarize the square-root cubature Kalman filter in Algorithm 3.

Algorithm 3 Square-Root Cubature Kalman Filter

Initialize:

Initialize the mean estimate $\hat{x}_{0|0}$ and square-root covariance estimate $S_{0|0}$.

for k=1 to N **do**

Prediction Step:

$X_{i,k-1|k-1} = S_{k-1|k-1}\xi_i + \hat{x}_{k-1|k-1}$

where $\xi = \sqrt{n}[I_n, -I_n]$, ξ_i is the i-th column of ξ for $i = 1, 2, \cdots, 2n$ and I_n is the n-dimensional identity matrix.

$X^*_{i,k|k-1} = f(X_{i,k-1|k-1}, u_{k-1})$

$\hat{x}_{k|k-1} = \frac{1}{m}\sum_{i=1}^{m} X^*_{i,k|k-1}$ with $m = 2n$

$X_{k|k-1} = \frac{1}{\sqrt{m}}\left[X^*_{1,k|k-1} - \hat{x}_{k|k-1} \quad X^*_{2,k|k-1} - \hat{x}_{k|k-1} \quad \cdots \quad X^*_{m,k|k-1} - \hat{x}_{k|k-1}\right]$

$S_{k|k-1} = qr([X_{k|k-1}\, S_{Q,k-1}])$ where $S_{Q,k-1}$ is the square root of the covariance Q_{k-1}.

Measurement Step:

$X_{i,k|k-1} = S_{k|k-1}\xi_i + \hat{x}_{k|k-1}$

$Z_{i,k|k-1} = h(X_{i,k|k-1}, u_k)$

$\hat{z}_{k|k-1} = \frac{1}{m}\sum_{i=1}^{m} Z_{i,k|k-1}$

$Z_{k|k-1} = \frac{1}{\sqrt{m}}\left[Z_{1,k|k-1} - \hat{z}_{k|k-1} \quad Z_{2,k|k-1} - \hat{z}_{k|k-1} \quad \cdots \quad Z_{m,k|k-1} - \hat{z}_{k|k-1}\right]$

$S_{zz,k|k-1} = qr([Z \;\; S_{R,k}])$ where $S_{R,k}$ is the square root of the covariance R_k.

$X_{k|k-1} = \frac{1}{\sqrt{m}}\left[X^*_{1,k|k-1} - \hat{x}_{k|k-1} \quad X^*_{2,k|k-1} - \hat{x}_{k|k-1} \quad \cdots \quad X^*_{m,k|k-1} - \hat{x}_{k|k-1}\right]$

$P_{xz,k|k-1} = X_{k|k-1}Z^T_{k|k-1}$

$K_k = P_{xz,k|k-1}S^{-T}_{zz,k|k-1}S^{-1}_{zz,k|k-1}$

$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - \hat{z}_{k|k-1})$

$S_{k|k} = qr([X_{k|k-1} - K_kZ_{zz,k|k-1} \;\; K_kS_{R,k}])$

end for

If there is no measurement available, than

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} \tag{4.43}$$

$$S_{k|k} = S_{k|k-1} \tag{4.44}$$

4.2 Stochastic Type Approximation Methods

Since we are dealing with nonlinear dynamical systems we focus here on the sequential type algorithms. In this section, we see the details of the particle filters and the method implemented in this thesis called particle filter with EKF approximated proposal function.

4.2.1 Particle Filter

In the particle filter approach, the conditional density function $p(x_{k-1}|y_{1:k-1})$ is approximated by M number of samples $\tilde{x}^i_{k-1}$ and their weights w^i_{k-1} as follows:

$$p(x_{k-1}|y_{1:k-1}) \approx \sum_{i=1}^{M} w^i_{k-1}\delta(x_{k-1} - \tilde{x}^i_{k-1}) \tag{4.45}$$

The predicted samples x^i_k are drawn using the proposal function $q(x_k|x_{k-1}, y_k)$, and the weights w^i_k are found by the following formula [28]:

$$w^i_k \propto w^i_{k-1}\frac{p(y_k|x_k)p(x_k|x_{k-1})}{q(x_k|x_{k-1}, y_k)} \tag{4.46}$$

The ideal choice for the proposal function is $q(x_k|x_{k-1}, y_k) = p(x_k|x_{k-1}, y_k)$, which is difficult to calculate. If we choose $q(x_k|x_{k-1}, y_k) = p(x_k|x_{k-1})$, we arrive at the standard particle filter weight update rule:

$$w^i_k \propto w^i_{k-1}p(y_k|x_k) \tag{4.47}$$

Furthermore, as a last step, we apply resampling to prevent particle collapse.

Resampling is done by drawing samples from the discrete probability set with the particles $\{x_k^i\}_{i=1}^M$ and the probabilities (weights) $\{w_k^i\}_{i=1}^M$ [38]. After the resampling, we set the weights $w_k^i = 1/M$. We summarize the steps in Algorithm 4.

Algorithm 4 Particle Filter

Initialize:

Draw the particles $\{x_0^i\}_{i=1}^M \sim p(x_0)$

for k=1 to N **do**

Prediction Step:

$$\{x_k^i\}_{i=1}^M \sim p(x_k|x_{k-1}^i) \tag{4.48}$$

Measurement Step:

$$w_k^i \propto p(y_k|x_k^i) \tag{4.49}$$

Resample Step Draw samples from the discrete probability set with the particles $\{x_k^i\}_{i=1}^M$ and the probabilities (weights) $\{w_k^i\}_{i=1}^M$ [38]. Set the weights $w_k^i = 1/M$

end for

4.2.2 Particle Filter with Extended Kalman Filter Proposal Function

The choice of the proposal function q is important. It is difficult to sample from the optimal choice of q. The closed form calculation is again not possible most of the time. If we linearize the measurement function, then the proposal function can be approximated with [38]:

$$q(x_k|x_{k-1}^i, y_k) \approx \mathcal{N}(f(x_{k-1}^i) + K_k^i(y_k - \hat{y}_k^i), (H_k^{i,T} R_k^{-1} H_k^i + Q_{k-1})^{-1}) \tag{4.50}$$

where

$$H_{ij}(x) = \frac{\partial h(x_i)}{\partial x_j} \tag{4.51}$$

$$\hat{y}_k^i = h(f(\hat{x}_{k-1}^i)) \tag{4.52}$$

$$K_k = Q_{k-1} H_k^{i,T} (H_k^i Q_{k-1} H_k^{i,T} + R_k)^{-1} \tag{4.53}$$

By choosing $q(x_k|x_{k-1}, y_k)$ as above, the weight update rule becomes [38]

$$w_k^i \propto w_{k-1}^i \mathcal{N}(f(x_{k-1}^i) + K_k^i(y_k - \hat{y}_k^i), (H_k^i Q_{k-1} H_k^{i,T} + R_k)^{-1}) \tag{4.54}$$

We summarize the PF-EKF in Algorithm 5

Algorithm 5 Particle Filter with EKF Proposal

Initialize:

Draw the particles by using $\{x_0^i\}_{i=1}^M \sim p(x_0)$

for k=1 to N **do**

Prediction Step:

Draw the particles by using the Eq. 4.50

Measurement Step:

Update the weight by using the Eq. 4.54

Resample Step Draw samples from the discrete probability set with the particles $\{x_k^i\}_{i=1}^M$ and the probabilities (weights) $\{w_k^i\}_{i=1}^M$ [38]. Set the weights $w_k^i = 1/M$

end for

We noted that The ideal choice for the proposal function for particle filters is $q(x_k|x_{k-1}, y_k) = p(x_k|x_{k-1}, y_k)$. In the fMRI literature, non-optimal proposal functions are chosen as in [5, 12]. By using EKF approximation to the optimal proposal function, we improved the state estimation accuracy in most of the cases.

5. DUAL ESTIMATION - JOINT PARAMETER AND STATE ESTIMATION

In estimation theory, there are two main categories for the state estimation. These are the filtering and smoothing. In the filtering of the state estimate, the observation sequence $y_1, y_2, \cdots, y_k$ or $y_{1:k}$ is used to find the statistical information about the states $x_1, x_2, \cdots, x_k$. We are interested to finding the conditional probability density function (pdf) $p(x_k|y_{1:k})$ for the filtering. Since by the new observation y_{k+1} we can find the conditional density $p(x_{k+1}|y_{1:k+1})$, filtering is also known as online estimation, whereas, in the smoothing approach, all the observation sequence including the future data ($y_{1:N}$, $N > k$) are used to find $p(x_k|y_{1:N})$ [18]. Since for every time step we need the complete observation sequence, this kind of processing is also known as offline estimation. The use of future data contains information about the past data and, as a result, improves the state and parameter estimates of the system. The choice of the state estimation technique also has an important effect on the parameter estimation. Having a more accurate state estimation algorithm also improves the parameter estimation of the system. In the fMRI model inversion literature, most of the state estimation techniques work only in the filtering sense. However, in most hemoydynamic data analysis cases, the computation is not performed in real time. By using smoothing techniques, the estimation of both the state and the parameters of the hemodynamic systems is improved. Trade-off is that incorporating smoothing into existing algorithms leads to an increased computation time.

We estimate the parameters and the states by using the BOLD signal and the functional representation of the system. In the joint parameter and state estimation part, we use the same functional model:

$$x_{k+1} = f(\theta, x_k) + w_k \tag{5.1}$$

$$y_k = h(\theta, x_k) + v_k \tag{5.2}$$

where, f and h are nonlinear functions of the state x_k, where k is the time index. The state in our case is a vector $x_k \in \mathbb{R}^4$, because there are four hemodynamic state variables. The observed BOLD signal at time k is denoted by $y_k \in \mathbb{R}$, because the BOLD signal is one dimensional. The state transition noise w_k is Gaussian with $\mathcal{N}(0, Q_k)$, and the observation noise v_k is Gaussian with $\mathcal{N}(0, R_k)$. The set of parameters of the model is denoted by θ. Given N observations $y_1, y_2, \cdots, y_N$ or $y_{1:N}$, our aim is to find the parameter set θ and the states $x_1, x_2, \cdots, x_N$ or $x_{1:N}$.

5.1 Iterative Extended Kalman Smoother

Iterative extended Kalman smoother is performed by the repeated use of the extended Kalman filter and smoother algorithm. At each step we first perform extended Kalman filter as explained in Section 4.1.2. Having the EKF step performed the smoother is performed as explained in the next section.

5.1.1 Extended Kalman Smoother

By using the EKF algorithm, the state x_k is predicted from the observed sequence $y_{1:k}$. From the future data, once we have them, a more accurate state estimation can be accomplished. This is achieved through the EKS algorithm [18]. The EKS algorithm is also easy to implement. The recursion is done by going backward in time. The recursion step begins from $k = N - 1$.

First, the state is predicted as:

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}) \tag{5.3}$$

and the covariance is predicted by the formula

$$P_{k+1|k} = F(\hat{x}_{k|k}) P_{k|k} F(\hat{x}_{k|k})^T + Q_k \tag{5.4}$$

The Kalman gain J_k for the smoother is:

$$J_k = P_{k|k} F(\hat{x}_{k|k})^T P_{k+1|k}^{-1} \tag{5.5}$$

Smoother estimates for the state mean $\hat{x}_{k|N}$ and state covariance $P_{k|N}$ are found by using the Kalman gain K_k and the filter estimates of the state mean $\hat{x}_{k|k}$ and the state covariance $P_{k|k}$:

$$\hat{x}_{k|N} = \hat{x}_{k|k} + J_k(\hat{x}_{k+1|N} - \hat{x}_{k+1|k}) \tag{5.6}$$

$$P_{k|N} = P_{k|k} + J_k(P_{k+1|N} - P_{k+1|k})J_k^T \tag{5.7}$$

For a nice proof of the algorithm please check the appendix. We summarize the steps of EKS in Algorithm 6.

Algorithm 6 Extended Kalman Smoother

Initialize:

Perform EKF summarized in Algorithm 2.

for k=N-1 to 1 **do**

$\hat{x}_{k+1|k} = f(\hat{x}_{k|k})$

$P_{k+1|k} = F(\hat{x}_{k|k}) P_{k|k} F(\hat{x}_{k|k})^T + Q_k$

$J_k = P_{k|k} F(\hat{x}_{k|k})^T P_{k+1|k}^{-1}$

$\hat{x}_{k|N} = \hat{x}_{k|k} + J_k(\hat{x}_{k+1|N} - \hat{x}_{k+1|k})$

$P_{k|N} = P_{k|k} + J_k(P_{k+1|N} - P_{k+1|k})J_k^T$

end for

5.1.2 Parameter Estimation and Iterative EKS

EKF and EKS can also be used to estimate the parameters of the system by a very simple method. In this approach, the parameters are treated as time-varying variables with small noise perturbations as in [6, 11]. As a result, the parameter updates

become:

$$\theta_{k+1,1} = \theta_{k,1} + w_{k,1} \tag{5.8}$$

$$\theta_{k+1,2} = \theta_{k,2} + w_{k,2} \tag{5.9}$$

$$\cdots \tag{5.10}$$

$$\theta_{k+1,p} = \theta_{k,p} + w_{k,p} \tag{5.11}$$

where $\theta_{k,i}$ stands for the i-th parameter at time k, $i = 1, 2, \cdots, p$. The new state $x_{a,k}$ becomes an extended version of the original state x_k.

$$x_{a,k} = \begin{bmatrix} x_k \\ \theta_{k,1} \\ \theta_{k,2} \\ \vdots \\ \theta_{k,p} \end{bmatrix} \tag{5.12}$$

For SCKS, Havlicek et al. used the similar technique for parameter estimation [6]. Since we augmented the parameters to the state, EKS also estimates for the parameters besides the states. At the end of the EKS algorithm, we obtain an estimate of the parameter set θ. Using the estimates of the parameters θ as the new initial estimate for the parameters, we iterate the EKF and EKS algorithm until convergence similar to [6].

5.2 Iterative Square-Root Cubature Kalman Filter

Similar to iterative EKS, iterative SCKS performed in this thesis begins with SCKF algorithm detailed in Chapter 4. Next step is smoother part of SCKS.

5.2.1 Square-Root Cubature Kalman Smoother

By applying the smoother to the output of the cubature Kalman filter, the state estimation errors decrease more. This time the recursion goes backward in time. Having the mean $\hat{x}_{N|N}$ and the square root of the covariance estimate $\hat{S}_{N|N}$, the recursion step begins from $k = N - 1$. The steps are as follows:

1) Evaluate the cubature points:

$$X_{i,k|k} = S_{k|k}\xi_i + \hat{x}_{k|k} \tag{5.13}$$

for $i = 1, 2, \cdots, 2n$.

2) Propagate the cubature points $X_{i,k|k}$ through the state update equation:

$$X_{i,k+1|k} = f(X_{i,k|k}, u_k) \tag{5.14}$$

3) From these predicted points $X_{i,k+1|k}$, smoothed mean estimate of the state $\hat{x}_{k+1|k}$ is calculated.

$$\hat{x}_{k+1|k} = \frac{1}{m}\sum_{i=1}^{m} X_{i,k+1|k} \tag{5.15}$$

4) Square-root error covariance $S_{k+1|k}$ is:

$$S_{k+1|k} = qr([X_{k+1|k} \;\; S_Q]) \tag{5.16}$$

where

$$X_{k+1|k} = \frac{1}{\sqrt{m}}\left[X_{1,k+1|k} - \hat{x}_{k+1|k} \quad X_{2,k+1|k} - \hat{x}_{k+1|k} \quad \cdots \quad X_{m,k+1|k} - \hat{x}_{k+1|k}\right] \tag{5.17}$$

6) Cross-covariance $D_{k+1|k}$ is calculated according to the formula:

$$D_{k+1|k} = X_{k|k}X_{k+1|k}^T \tag{5.18}$$

where

$$X_{k|k} = \frac{1}{\sqrt{m}}\left[X_{1,k|k} - \hat{x}_{k|k} \quad X_{2,k|k} - \hat{x}_{k|k} \quad \cdots \quad X_{m,k|k} - \hat{x}_{k|k}\right] \tag{5.19}$$

7) The Kalman gain C_k:

$$C_k = D_{k+1|k} S_{k+1|k}^{-T} S_{k+1|k}^{-1} \tag{5.20}$$

8) The smoothed mean estimate $\hat{x}_{k|N}$:

$$\hat{x}_{k|N} = \hat{x}_{k|k} + C_k(\hat{x}_{k+1|N} - \hat{x}_{k+1|k}) \tag{5.21}$$

9) The smoothed square-root covariance $S_{k|N}$:

$$S_{k|N} = qr([X_{k|k} - C_k X_{k+1|N} \;\; C_k S_{k+1|N}]) \tag{5.22}$$

The recursion is performed backward in time up-to $k = 1$. This concludes the smoothing part.

We summarize the steps of SCKS in Algorithm 7.

Algorithm 7 Square-root Cubature Kalman Smoother

Initialize:

Perform SCKF summarized in Algorithm 3.

for k=N-1 to 1 **do**

$X_{i,k|k} = S_{k|k}\xi_i + \hat{x}_{k|k}$ for $i = 1, 2, \cdots, 2n$.

$X_{i,k+1|k} = f(X_{i,k|k}, u_k)$

$\hat{x}_{k+1|k} = \frac{1}{m}\sum_{i=1}^{m} X_{i,k+1|k}$

$S_{k+1|k} = qr([X_{k+1|k} \;\; S_Q])$

$X_{k+1|k} = \frac{1}{\sqrt{m}}\left[X_{1,k+1|k} - \hat{x}_{k+1|k} \quad X_{2,k+1|k} - \hat{x}_{k+1|k} \quad \cdots \quad X_{m,k+1|k} - \hat{x}_{k+1|k}\right]$

$D_{k+1|k} = X_{k|k} X_{k+1|k}^T$

$X_{k|k} = \frac{1}{\sqrt{m}}\left[X_{1,k|k} - \hat{x}_{k|k} \quad X_{2,k|k} - \hat{x}_{k|k} \quad \cdots \quad X_{m,k|k} - \hat{x}_{k|k}\right]$

$C_k = D_{k+1|k} S_{k+1|k}^{-T} S_{k+1|k}^{-1}$

$\hat{x}_{k|N} = \hat{x}_{k|k} + C_k(\hat{x}_{k+1|N} - \hat{x}_{k+1|k})$

$S_{k|N} = qr([X_{k|k} - C_k X_{k+1|N} \;\; C_k S_{k+1|N}])$

end for

5.2.2 Parameter Estimation with SCKS

For the estimation of the parameter set θ, the same technique presented in Section 5.1.2 is utilized similar to [6]. In the following section, a toy example is considered.

5.3 Toy Example: Linear Case

In this section, we want to show the importance of the smoothing and the usage of EKS. In the first two subsections we make state estimation by assuming the parameter is known. In the first subsection, we visualize the state estimation improvement by the smoother over the filter. In the second subsection, we perform Monte Carlo analysis to show the state estimation improvement. In the third subsection, we assume that the parameter is also unknown. With this condition, we estimate the parameter by the methods we described in the Sections 5.1.2 and 5.2.2. We perform again Monte Carlo simulation. We compare the mean and variance of the estimates. We show the gradual improvement in every iteration. In this section, we work on a linear state space system. The system representation is as follows:

$$x_{k+1} = Ax_k + w_k \tag{5.23}$$

$$y_k = Cx_k + v_k \tag{5.24}$$

The state in our case is a vector $x_k \in \mathbb{R}^2$. The measurement at time k is denoted by $y_k \in \mathbb{R}$. The state transition noise w_k is Gaussian with $\mathcal{N}(0, Q_k)$ and the observation noise v_k is Gaussian with $\mathcal{N}(0, R_k)$. Here the state transition matrix A is also dependent

Table 5.1
Initial Values and Noise Parameters for Toy Example

Variable	Value
Initial State Value X_0	[1 1]
Initial State pdf $p(X_0)$	$\mathcal{N}(1, 0.01)$ for each component
Initial Parameter pdf θ_0	$\mathcal{N}(0.8, 1/12)$
Parameter Noise w_θ	$\mathcal{N}(0, 10^{-5})$
State Noise w_k	$\mathcal{N}(0, e^{-3})$ for each component
Measurement Noise v_k	$\mathcal{N}(0, e^{-3})$

on the θ parameter.

$$A = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} \tag{5.25}$$

C matrix is chosen as:

$$C = \begin{bmatrix} 1 & 1 \end{bmatrix} \tag{5.26}$$

Initial state conditions, parameter, state and measurement noise figures of the system are presented in Table 5.1.

5.3.1 Visualization of Smoother Estimation

In this toy example, the true value for the parameter θ is arbitrarily chosen as 0.8. We generate one simulation from the model. From the observed data we estimate the state by Kalman Filter (KF) and Kalman Smoother (KS). Note that since the system is linear, EKF boils down to KF. This is true for EKS and KS, as well.

In order to better understand the performance of the smoother algorithms, we plot the confidence region of the state estimates of Kalman Filter and Smoother. The

confidence region for Kalman estimate is an ellipsoid. As a result, for the two dimensional case it is possible to visualize the filter and smoother estimates. We plot the ellipsoids for both the Kalman Smoother and Filter estimates. The contour graph of the ellipsoid is expressed as:

$$(x - \hat{x})P^{-1}(x - \hat{x})^T = \gamma \tag{5.27}$$

where $\hat{x}$ is the state estimate, P is the covariance of the estimate, γ determines the confidence region contour.

The cholesky decomposition gives us a simple transformation such that the above equation is transformed into:

$$\left\|U^{-T}(x - \hat{x})\right\|^2 = \gamma \tag{5.28}$$

where $P = U^T U$. The solution for x can be given as:

$$x = \hat{x} + \sqrt{\gamma}U^T[cos(\alpha)\ \ sin(\alpha)] \tag{5.29}$$

where $0 \leq \alpha \leq 2\pi$.

Note that $U^{-1}(x - \hat{x})$ is χ^2 distributed with two degrees of freedom. For 99% confidence region, we choose γ according to the formula:

$$\gamma = \mathsf{chi2inv}(0.99, 2) \tag{5.30}$$

Here chi2inv(Pr,n) is the MATLAB command for the inverse of the χ^2 cumulative distribution function (cdf) where the first argument (Pr) is the probability value of the cdf and the second argument (n) is the number of degrees of freedom (in this case 2).

As can be seen in Figure 5.3.1, the ellipsoids for the smoothers are tighter. This reflects their covariance is narrower for smoothers, which means that the estimation

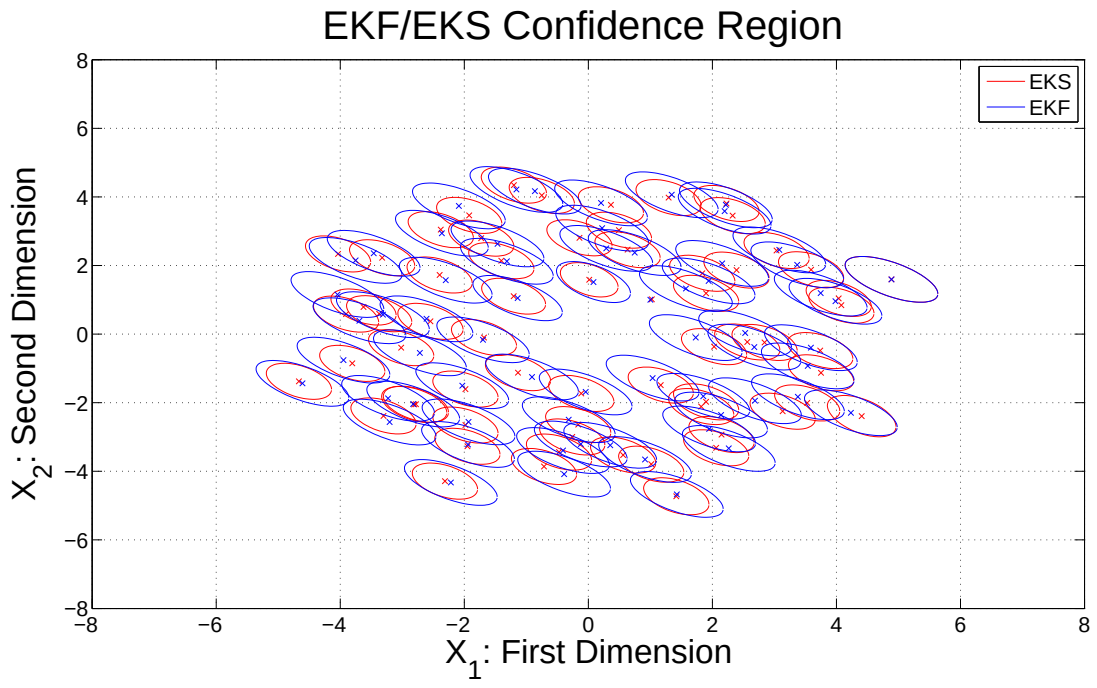

Figure 5.1 Kalman Smoother and Filter Confidence Region

of smoothers is more accurate. To further emphasize the accuracy of smoothers, in Section 5.3.2, we provide Monte Carlo simulation results of the smoother and filter algorithms, as well.

We note in Figure 5.3.1 most of the ellipsoids have similar shape. This is due to the fast convergence of the covariance matrix of the state. After a few steps, the covariance matrix converges. As a result, we have the same ellipsoid shifted just by the means of the states.

5.3.2 Toy Example Simulation Results for State Estimate for fixed Parameter

In this section, we assess the performance of EKF with respect to EKS for the linear system by running a Monte Carlo analysis. Note that for this fully linear case, running EKF (or EKS) indeed is the same as running Kalman filter (or Kalman

Table 5.2
Monte Carlo State RMS Error for Kalman Filter and Smoother.

Method	**RMS Error**
Kalman Filter	$0.3264 \pm 8.3620\text{x}10^{-4}$
Kalman Smoother	$0.2444 \pm 6.4533\text{x}10^{-4}$

Smoother). We perform 100 Monte Carlo runs. For each run, the Root Mean Square (RMS) errors for each method is calculated for comparison. The RMS error in a specific run is defined as

$$e_{rms} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} \|\hat{x}_k - x_{k,true}\|^2} \tag{5.31}$$

Here $\hat{x}_k$ is the result of the state estimation algorithm at time k which corresponds to either the filtered value, $\hat{x}_{k|k}$ for the EKF case or the smoothed value $\hat{x}_{k|N}$ for the EKS case, $x_{k,true}$ represents the ground-truth and $\|.\|$ is the Euclidean Norm.

After collecting the RMS state estimation errors for each run as described in Eq. 7.7, we calculate the sample mean and the sample standard deviation for the RMS state estimation error across all the runs as shown in Table 5.2.

As we can see from Table 5.2, smoothing clearly improves the state estimation performance for this known parameter case. In the next subsection, we investigate the unknown parameter case.

5.3.3 Toy Example Simulation Results for Parameter Estimation

In this case, we treat the parameter θ as unknown. When we augment the parameter to the state vector, then the resulting system with the augmented state vector becomes nonlinear. This is because the parameter enters to the system in a nonlinear fashion. In the previous case study, we observed that smoothing results in more accurate state estimation than the filtering. Therefore, in the current case study, we consider only the smoothing algorithms, EKS and SCKS and compare their performances in terms of estimating the unknown parameter θ. We perform 140 Monte Carlo runs for the comparison. In each run, the initial estimate, which is selected as the same value for both of the algorithms, is assumed to be Gaussian distributed around the true value θ with a variance of 1/12. Starting from the same initial estimate, both algorithms are run forward (i.e., *filtering*) and then backward (i.e., *smoothing*) which accounts for one iteration. Then, in the second iteration, we continue in the same manner, i.e., forward and then backward passes again. We repeate this procedure for several iterations in each Monte Carlo run as in [6]. The Monte Carlo averaged estimation results of EKS and SCKS algorithms together with the corresponding performance variation (1-σ upper and lower bounds) are shown in Fig. 5.2 with respect to the number of iterations performed.

As shown in Figure 5.2, on the average, both of the algorithms converged to the true value of the parameter θ, which is 0.8, very quickly as the number of iterations increased. The converged values for each of the algorithms together with their corresponding 1-σ bounds are listed in Table 5.3. Note that performance of both algorithms are the same in terms of the averaged converged value, but SCKS is slightly better than EKF in terms of the 1-σ bound.

After examining performances of the algorithms on this toy example, we apply them to our realistic problem, namely, system identification and state estimation of the hemodynamic model. In the next section, we will see the details of the hemodynamic model.

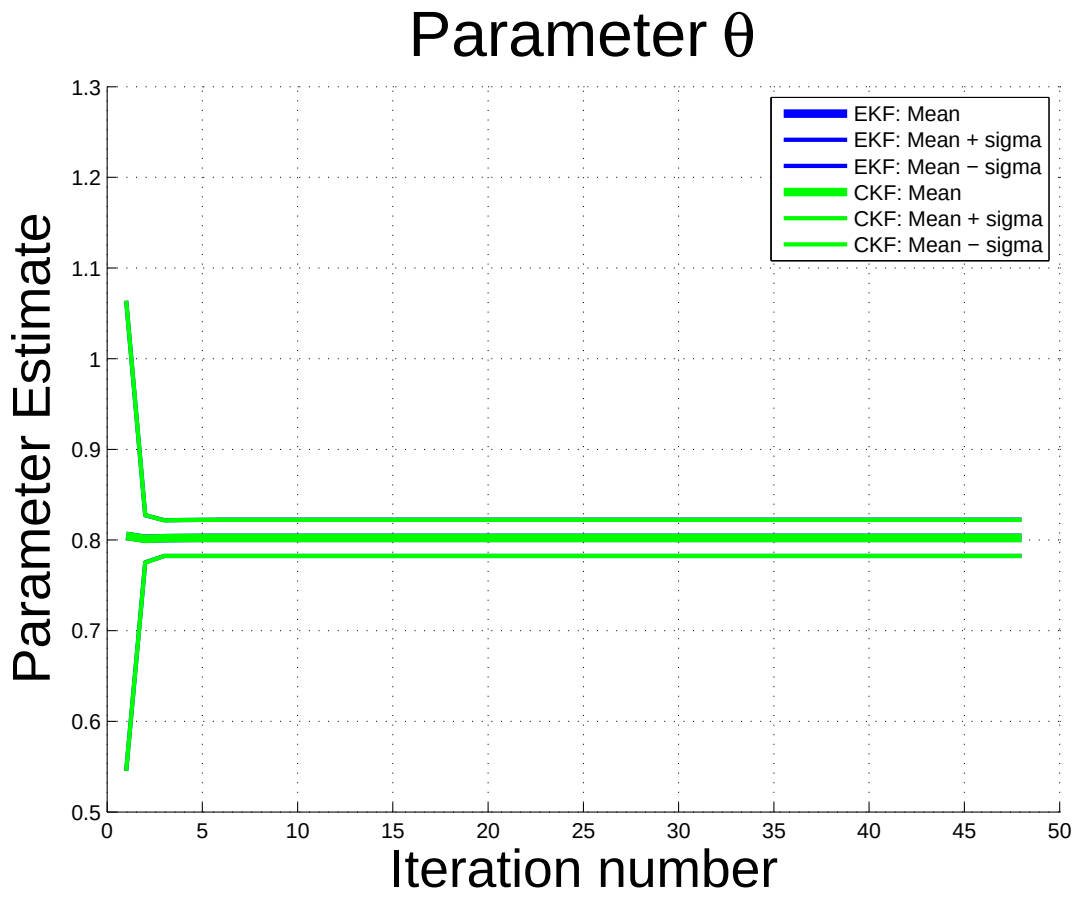

Figure 5.2 The convergence of the parameter estimate.

Table 5.3
Monte Carlo Estimate for θ.

Method	True Value	Estimated
EKS	0.8	$0.8023 \pm 4.00\text{x}10^{-4}$
SCKS	0.8	$0.8023 \pm 3.97\text{x}10^{-4}$

6. SIMULATION RESULTS FOR THE HEMODYNAMIC MODEL WITH KNOWN PARAMETERS

6.1 Findings and Contribution

We compare different state estimation techniques with the method we implement. We generate the data from the model and test the state estimation performance of the methods. We test the validity of our approach on a synthetic data set as generated from the hemodynamic model. We compare the algorithms under a range of different noise levels, different inputs, parameter sets and unknown initial condition. We compare PF-EKF with the standard particle filter (PF), auxiliary particle filter (APF), extended Kalman filter (EKF) and cubature Kalman filter (SCKS). For the details of the APF algorithm, the reader is referred to [28].

In the Discussion section, we interpret the results. Findings and contributions are as follows:

- Contrary to the former notion, Extended Kalman filters estimate the hidden states better than particle filters under a wide range of noise conditions, different inputs, parameter sets and unknown initial condition.
- The hemodynamic model is weakly nonlinear system.
- PF-EKF can offer more accurate hemodynamic state estimation than the particle filter and auxiliary particle filters in most cases by approximating the optimal proposal by EKF. However, due to weak-nonlinearity of the hemodynamic model, Gaussian approximated methods are better in performance.

The initial condition, measurement and process noise statistics are listed in Table 6.1. We did not initiate the state from a known state. We took the initial state

Table 6.1
Noise Statistics and Initial Values.

Variable	Value
$p(X_0)$	$\mathcal{N}(0, 0.01)$ for each component
Measurement Noise v_k	$\mathcal{N}(0, \sigma_v^2 = e^{-12})$
Scenario 1: w_k	$\mathcal{N}(0, \sigma_w^2 = \Delta t e^{-16})$ for each state component
Scenario 2: w_k	$\mathcal{N}(0, \sigma_w^2 = \Delta t e^{-12})$ for each state component
Scenario 3: w_k	$\mathcal{N}(0, \sigma_w^2 = \Delta t e^{-8})$ for each state component

pdf $p(X_0)$ as $\mathcal{N}(0, 0.01)$. The measurement noise is in accordance with [6, 14]. The process noise varies in a wide range, which includes the values used in the literature [6, 14]. In order to compare the performance of the algorithms, we generated simulated data using Eq. 3.5 to produce 100 runs for each level of noise -(Scenarios 1 to 3 in Table 6.1). For the simulation, we assumed that for each state update, we have a corresponding measurement. The input is first taken as in [6] as Gaussian bumps with different amplitudes centered in the time points (10, 15, 39, 48) as shown in Figure 6.1. The length of this input signal is 64 seconds. Δt is taken as 0.1 second as in [5]. A typical hemodynamic response and simulated BOLD signal obtained from the Gaussian bump signal is also plotted in Figures 6.2 and 6.3. To test the algorithm performances with respect to the different inputs we also check the input for the widely used box-car functions as can be seen in Figure 6.4.

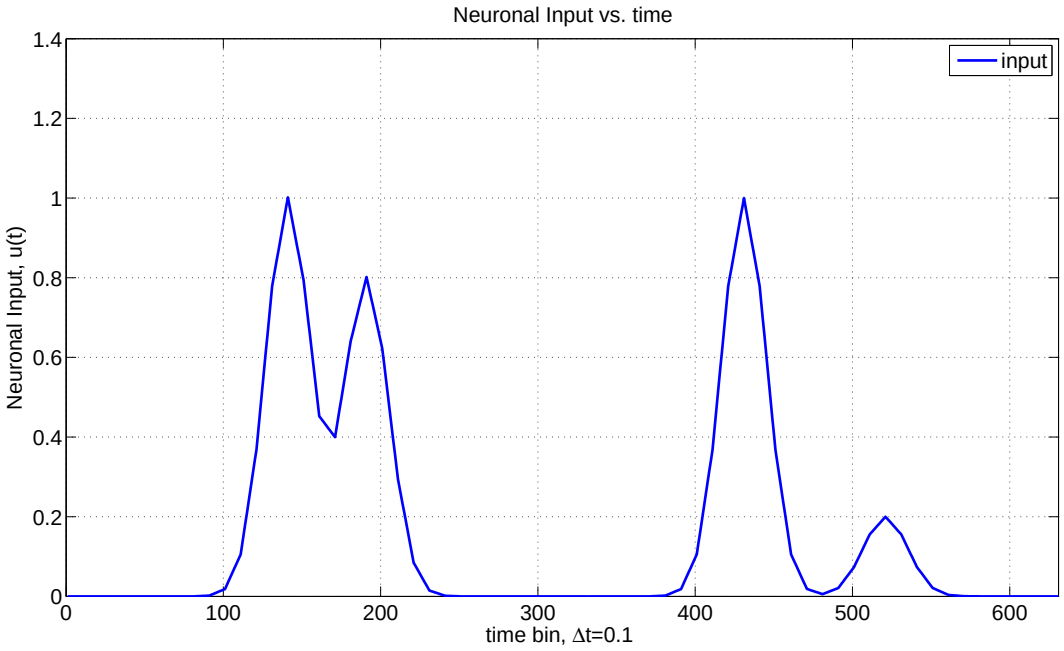

Figure 6.1 Neuronal Inputs as Gaussian bumps.

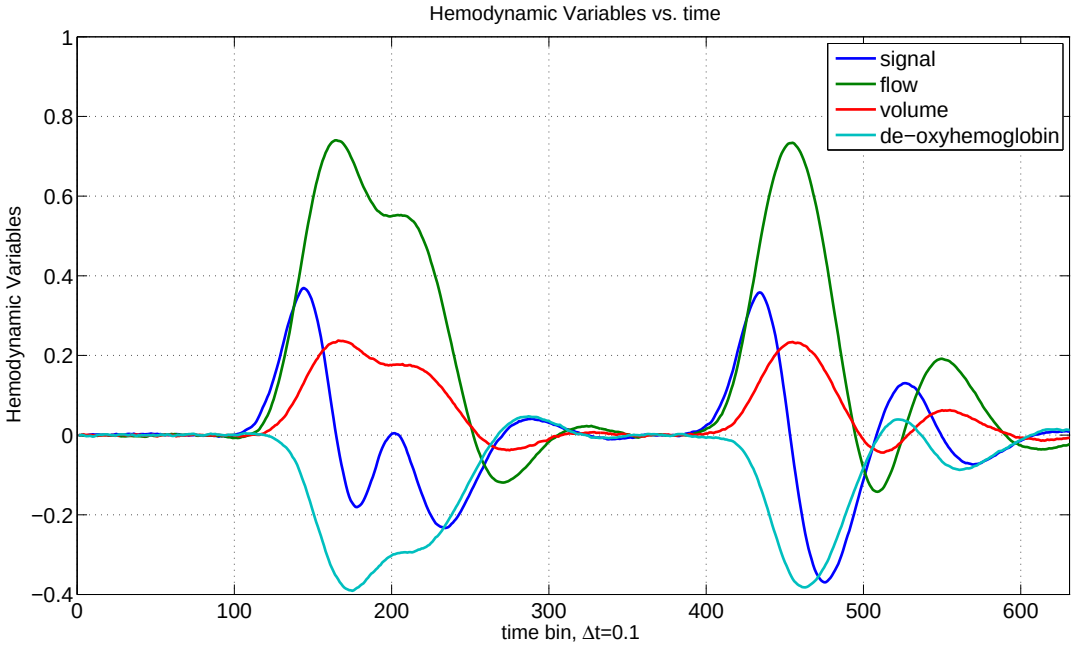

Figure 6.2 Hemodynamic Variables for Gaussian bumped neuronal inputs.

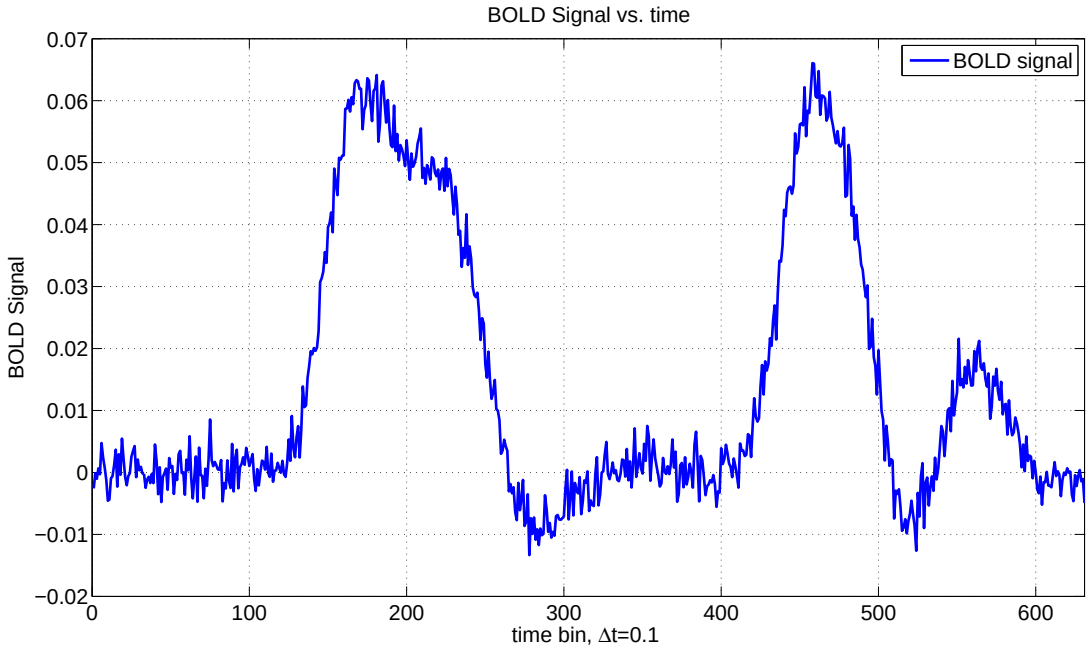

Figure 6.3 BOLD Signal for Gaussian bumped neuronal inputs.

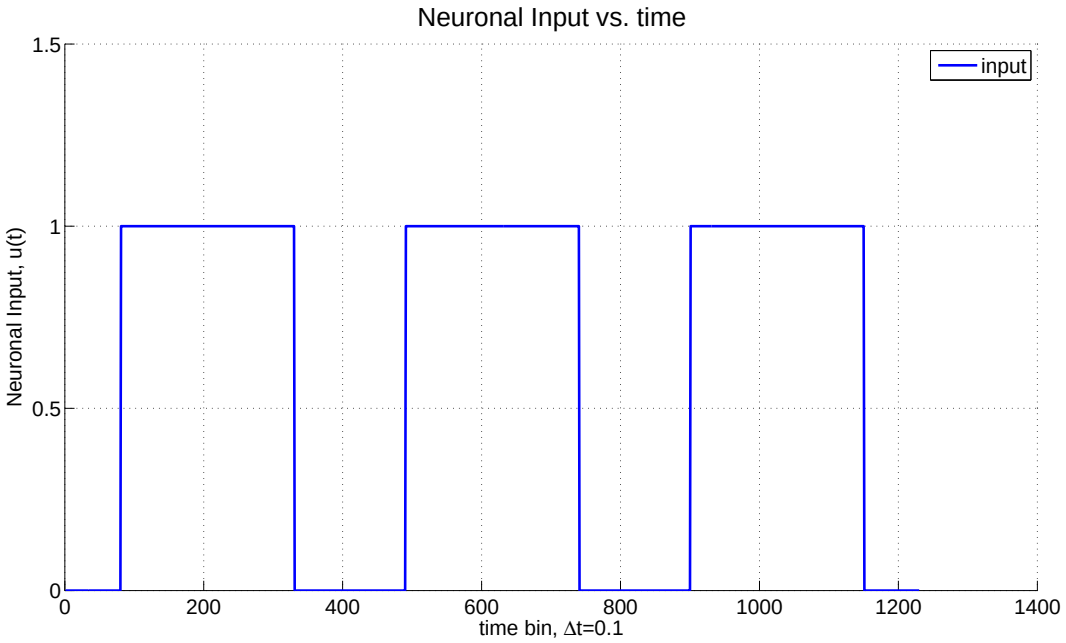

Figure 6.4 Filtering algorithms are tested for input as boxcar functions.

We tested the model inversion techniques under 3 different parameter sets as

Table 6.2
Parameter Values of the Hemodynamic Model.

Parameter	**Description**	**Set-1**	**Set-2**	**Set-3**
κ	Rate of Signal Decay	0.8	0.8	0.65
τ	Hemodynamic Transit Time	1	1	1.02
χ	Rate of Flow Dependent Elimination	0.4	0.4	0.41
α	Grubb's exponent	0.3	0.3	0.32
φ	Resting Oxygen Extraction Fraction	0.3	0.3	0.34
ϵ	Neural Efficiency	0.5	0.5	0.5
V_0	Resting Blood Volume Fraction	0.02	0.02	0.04
σ_v^2	Measurement Variance	0.001	e^{-12}	e^{-12}

shown in Table 6.2. The first one uses the parameter set used by Johnston et al. [5]. They made the assertion that EKF is not robust. In this scenario, they tested with a rather high measurement noise condition. The second parameter set uses again their parameter set with moderate level of measurement noise condition which is in agreement with [6, 14]. For further test the algorithms, we switch to the widely used parameter set used in [9, 11, 12, 13, 14, 15]. These 3 sets compromise nearly all the parameter sets used in the hemodynamic model inversion literature.

For each run, Root Mean Square (RMS) errors of the states are found. The RMS error in a specific run is defined as follows:

$$e_{rms} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} \|\hat{x}_k - x_{k,true}\|^2} \tag{6.1}$$

where $\hat{x}_k$ is the state estimation done by the filtering algorithm. $\|.\|$ is the Euclidean Norm, finally $x_{k,true}$ is the true value for the state. We summarize the results for the first,second and third parameter sets with Gaussian bumps and box-car

Table 6.3
Parameter Set 1, Mean and Standard Deviation of the RMS State Errors: PF-EKF, PF, APF, EKF and SCKF for Gaussian Bump Type Neuronal Input.

Scenario	PF-EKF	PF	APF	EKF	SCKF
Scenario 1	0.0279 ±0.0116	0.0283 ±0.0114	0.0282 ±0.0115	0.0271 ±0.0112	0.0271 ±0.0112
Scenario 2	0.0284 ±0.0108	0.0284 ±0.0110	0.0288 ±0.0111	0.0276 ±0.0108	0.0276 ±0.0108
Scenario 3	0.0530 ±0.0080	0.0531 ±0.0079	0.0533 ±0.0080	0.0513 ±0.0076	0.0513 ±0.0076

Table 6.4
Parameter Set 1, Mean and Standard Deviation of the RMS State Errors: PF-EKF, PF, APF, EKF and SCKF for Boxcar Type Neuronal Input.

Scenario	PF-EKF	PF	APF	EKF	SCKF
Scenario 1	0.0196 ±0.0080	0.0199 ±0.0081	0.0200 ±0.0081	0.0189 ±0.0076	0.0189 ±0.0076
Scenario 2	0.0215 ±0.0081	0.0217 ±0.0081	0.0212 ±0.0081	0.0207 ±0.0079	0.0207 ±0.0079
Scenario 3	0.0495 ±0.0051	0.0495 ±0.0051	0.0495 ±0.0050	0.0478 ±0.0049	0.0478 ±0.0049

functions in Tables 6.3, 6.4, 6.5, 6.6, 6.7 and 6.8 (Entries of these tables contain the sample mean +/− sample standard deviation of the RMS state errors for the Monte Carlo simulations.)

In all process noise conditions, Gaussian approximated methods performed better than the other state estimation techniques (PF-EKF, PF and APF). In the fMRI literature even 100 particles were alleged to outperform [5]. We set 1000 particles, which is 10 times more particles than the ones used in Johnston et al. [5]. We begin with Scenario 1, which has low process noise conditions. This case was used by Havlicek

Table 6.5
Parameter Set 2, Mean and Standard Deviation of the RMS State Errors: PF-EKF, PF, APF, EKF and SCKF for Gaussian Bump Type Neuronal Input.

Scenario	PF-EKF	PF	APF	EKF	SCKF
Scenario 1	0.0206 ±0.0086	0.0201 ±0.0079	0.0201 ±0.0077	0.0175 ±0.0058	0.0175 ±0.0058
Scenario 2	0.0212 ±0.0078	0.0212 ±0.0083	0.0213 ±0.0081	0.0184 ±0.0053	0.0185 ±0.0053
Scenario 3	0.0419 ±0.0050	0.0421 ±0.0050	0.0420 ±0.0051	0.0398 ±0.0038	0.0398 ±0.0038

Table 6.6
Parameter Set 2, Mean and Standard Deviation of the RMS State Errors: PF-EKF, PF, APF, EKF and SCKFF for Boxcar Type Neuronal Input.

Scenario	PF-EKF	PF	APF	EKF	SCKF
Scenario 1	0.0145 ±0.0071	0.0142 ±0.0061	0.0145 ±0.0061	0.0122 ±0.0040	0.0122 ±0.0040
Scenario 2	0.0159 ±0.0057	0.0158 ±0.0053	0.0159 ±0.0053	0.0139 ±0.0035	0.0139 ±0.0034
Scenario 3	0.0397 ±0.0031	0.0398 ±0.0031	0.0396 ±0.0030	0.0379 ±0.0025	0.0379 ±0.0025

Table 6.7
Parameter Set 3, Mean and Standard Deviation of the RMS State Errors: PF-EKF, PF, APF, EKF and SCKF for Gaussian Bump Type Neuronal Input.

Scenario	PF-EKF	PF	APF	EKF	SCKF
Scenario 1	0.0244 ±0.0123	0.0237 ±0.0110	0.0242 ±0.0098	0.0163 ±0.0055	0.0163 ±0.0055
Scenario 2	0.0243 ±0.0125	0.0251 ±0.0115	0.0248 ±0.0111	0.0178 ±0.0052	0.0178 ±0.0052
Scenario 3	0.0426 ±0.0076	0.0431 ±0.0060	0.0426 ±0.0069	0.0393 ±0.0038	0.0393 ±0.0037

Table 6.8
Parameter Set 3, Mean and Standard Deviation of the RMS State Errors: PF-EKF, PF, APF, EKF and SCKF for Boxcar Type Neuronal Input.

Scenario	PF-EKF	PF	APF	EKF	SCKF
Scenario 1	0.0189 ±0.0077	0.0196 ±0.0086	0.0190 ±0.0076	0.0137 ±0.0036	0.0137 ±0.0036
Scenario 2	0.0194 ±0.0098	0.0186 ±0.0080	0.0182 ±0.0078	0.0134 ±0.0037	0.0134 ±0.0036
Scenario 3	0.0391 ±0.0037	0.0393 ±0.0037	0.0391 ±0.0035	0.0366 ±0.0026	0.0366 ±0.0026

et al. [6]. We increased step by step the process noise. The last scenario corresponds to the high process noise condition used by Friston et al. [34]. The computation time is about 420, 2.25, 2.1, 0.6 and 1.1 seconds for PF-EKF, APF, PF, EKF and SCKF, respectively.

We want also see the performance of the algorithms by showing the errors at each time. Errors are calculated according to the Eq. 6.2.

$$e_k = \sum_{k=1}^{4} (\hat{x}_{k,i} - x_{k,i})^2 \tag{6.2}$$

where $\hat{x}_{k,i}$ and $x_{k,i}$ are the individual components of the estimated and true state values, respectively.

We plot the log-error results of the algorithms with respect to time in Figure 6.5. In Figure 6.6 an enlarged version of the individual performances of the algorithms can be seen. As shown in these figures, Gaussian approximated inference methods are better than sampling based methods. We also note that all the algorithms recover the initial high uncertainty of the state by time. Gaussian approximated methods are also faster in this recovery stage.

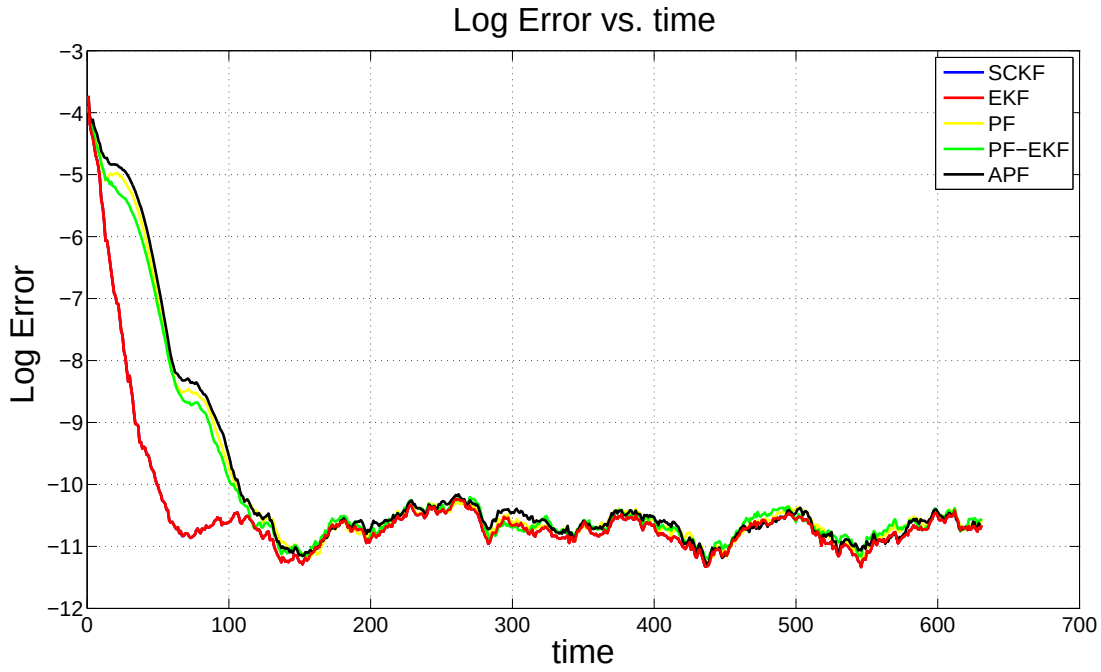

Figure 6.5 Errors of the algorithms at each time for the Parameter Set-3 with Gaussian Bump input

Figure 6.6 Large Scale Vision of Errors of the algorithms at each time for the Parameter Set-3 with Gaussian Bump input

By attacking the hemodynamic model with different noise conditions, inputs, parameter sets, and unknown initial condition, we showed that the hemodynamic system is weakly-nonlinear. We give more details, further insights and important outcomes of this finding in the Discussion section.

6.2 Discussion

State filtering algorithms are important in order to understand hemodynamic responses since they are not directly observable. Also, filtering methods form the bases for more advanced type of estimation problems, including state smoothing, parameter estimation (system identification) and even input estimation. In this thesis, we applied different filtering techniques used in the fMRI literature. The main finding in this Chapter is that the hemodynamic model is a weakly nonlinear system. For nonlinear hemodynamic models, it is known that sampling based methods can give better results than Gaussian approximated methods. However, for the case of the hemodynamic model, we found that the contrary is true. The important point here is in the fMRI literature EKF was alleged to be poor in performance compared to particle filters. We show that in this technical note, the contrary is true. This finding is important because based on that we will suggest an advanced Kalman type algorithm for performing not only state estimation but also hemodynamic parameter estimation. Particle filter type algorithms are suitable where there is strong nonlinearity. For weakly nonlinear system the posterior densities can be well approximated by Gaussian densities. This is indeed the case for the hemodynamic model. To see this fact, we used the histograms formed by the particles of the PF method for the 3. parameter set with box-car type input. We chose 3 randomly points in time and plotted the histograms of the first state for the times $t = 150, 500, 750$ in Figures 6.7, 6.8 and 6.9. Those are the typical plots for the other states. As can be seen from the graphs, they are well Gaussian shaped figures. Particle Filters work especially good for multi-modal posteriors. However, we observe that it is not the case for the hemodyamic model.

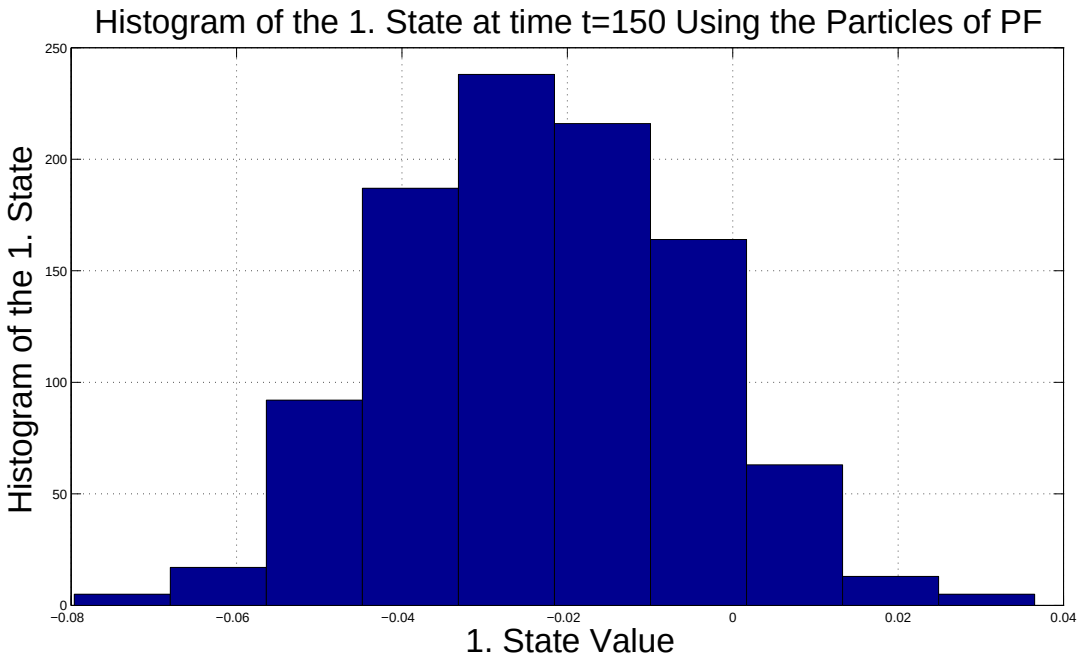

Figure 6.7 Posterior $p(x_{150}|y_{1:150})$ formed from the particles of the 1. hemodynamic state.

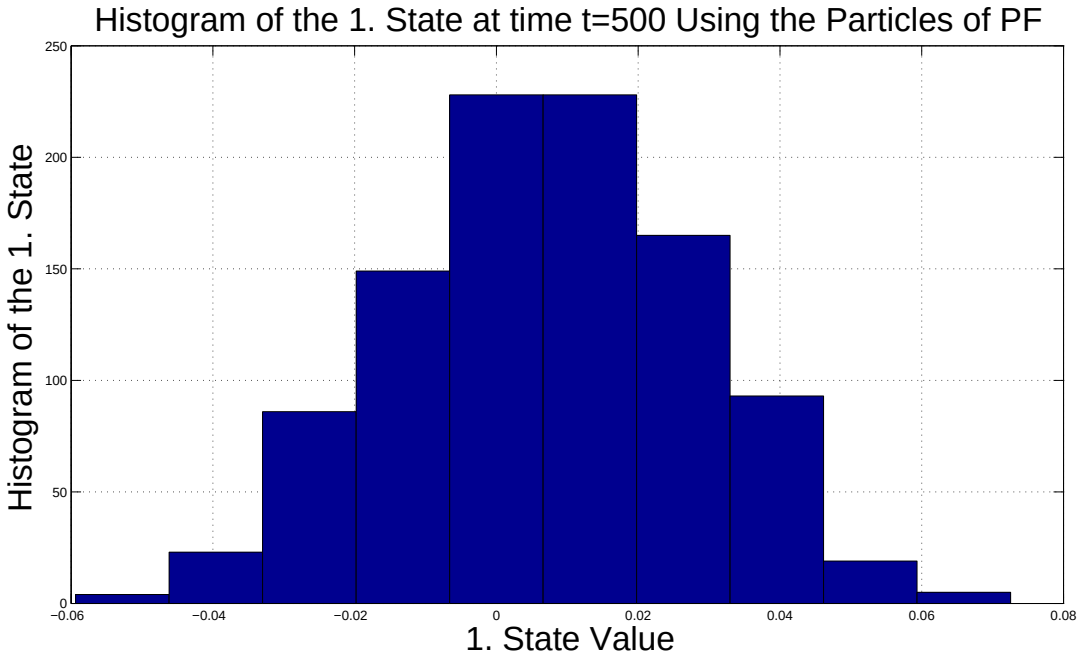

Figure 6.8 Posterior $p(x_{500}|y_{1:050})$ formed from the particles of the 1. hemodynamic state.

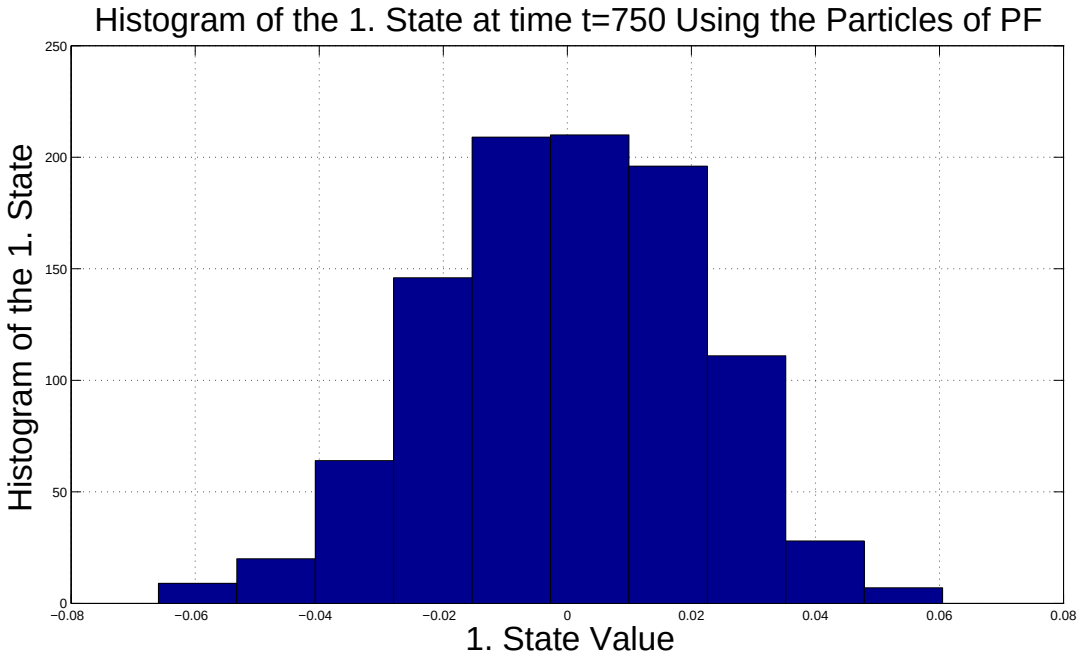

Figure 6.9 Posterior $p(x_{750}|y_{1:750})$ formed from the particles of the 1. hemodynamic state.

SCKF and EKF performed closely. They both approximate the conditional pdf $p(x_k|y_{1:k})$ as a Gaussian pdf. In the first 4 scenarios we used quite low particle numbers. For Particle Filter type algorithms, for all scenarios we chose 1000 particles, which are 10 times more than the ones used by Johnston et al. [5]. In regard to SMC methods, the choice of the proposal function is crucial, because standard particle filters can work poor in performance since they do not use optimal proposal functions. In many applications, standard particle filter algorithm proposed by Gordon et al. [27] is used. The non-optimal choice of the proposal function decreases the performance of the state estimation. The optimal importance function is $q = p(x_k|x_{k-1}^i, y_k)$, which is difficult to calculate. By using the EKF approximation of the optimal importance function, improved sample diversity is achieved for most process noise conditions, however the improvement is not huge. The reason is again weak-nonlinearity. One drawback of PF-EKF is computational complexity. In the PF-EKF method, for every particle, a specific Kalman Filter algorithm is run. As a result, PF-EKF is also the most time consuming algorithm. By utilizing PF-EKF, APF and PF we show the set of particle

filters do not outperform Gaussian approximated methods due to weakly nonlinearity of the hemodyanmic model. Clearly, EKF and SCKF were faster than PF-EKF.

In the literature, the UKF method is also used by Hu et al. [11]. However, the cubature Kalman filter is actually a special case of the Unscented Kalman Filter [25]. Furthermore, the difference between the PF-EKF and SCKS was clear from the simulation results. Also in the literature, the local linearization filter (LLF) is used. LLF is just a special type of EKF that differs only in the discretization step used in the prediction update [5, 9]. In LL the discretization type is the one proposed by [39]. For the discretization type, we chose the standard Euler-Maruyama method similar to [5]. The time step was small enough to reveal the dynamics of the system. The time step $\Delta t = 0.1$ was also smaller in size than the ones used by Havlicek et al. [25].

In this thesis, we did not compare PF-EKF with variational based approaches like DEM, GF and VF proposed by Friston et al. [13, 14, 15]. Because DEM, GF and VF perform dual estimation. They perform both hemodynamic parameter estimation -system identification- and state estimation. In this thesis, we are restricted on the state estimation. However, we compared our algorithm with the filter part of the SCKS algorithm implemented by Havlicek et al. [6]. Havlicek et al. compared SCKS with DEM and concluded that SCKS was a better implementation [6]. PF-EKF was better in state accuracy compared to SKCF. Since SCKF was Gaussian based approximation method, we suspected that EKF could also work satisfactorily. Actually first Riera performed a kind of extended Kalman filter which differs in the prediction update due to the discretization type they used [9]. However, Johnston [5] later compared EKF, PF and LLF and concluded that EKF is not stable. We checked this assumption in the widely used physiologically parameter regime [6, 12, 13, 14, 15] and concluded that EKF can be a successful candidate for state estimation either. The RMS errors in state estimation were in close proximity with SCKF suggested by Havlicek et al. [6]. Even the EKF performance always slightly better than SCKF. This was in accordance with our expectations, since SCKF and EKF have the same basic assumption. They approximate the conditional pdf $p(x_k|y_{1:k})$ with Gaussian density.

In the hemodynamic state estimation literature, Murray and Storkey [12] worked several SMC methods. They checked the particle filters (PF), auxiliary particle filters (APF) and regularised particle filters (RPF) all of which can be categorized under non-optimal partical filter methods. They performed smoothing to improve the accuracy. In this thesis, we did not implement RPF, however we implemented both PF and APF. In their report, in the smoother context the performance of APF and PF was similar for Euler-Maruyama discretization. Even APF was slightly worse. In the APF method, samples are chosen according to the one step ahead measurement data. In some non-linear estimation problems, this method can improve the RMS error performance [26]. We had comparable results for APF and PF in our simulations. The implementation of APF is slightly more complex compared to PF.

The proposal type of the PF method used by Johnston et al. [5] was not optimal. In PF, the optimal proposal function $p(x_k|x_{k-1}^i, y_k)$ is approximated by $p(x_k|x_{k-1}^i)$. This makes the steps of the PF very easy to implement. However as we noted before, using the rather complicated proposal function of PF-EKF, the performance is improved. Johnston et al. used very few particles [5]. With 100 particles, they performed the state estimation. This is quite unexpected for SMC methods for a nonlinear dynamical system with 4-dimensions . However we also note that Johnston et al. added the random perturbation to the system only in the first dimension [5]. In our implementation, we added noise perturbation in all components which is in accordance with [6, 13, 14, 15].

As a further note, for multi-input neuronal activations, the hemodynamic model is only changed in its first equation. Having k number of inputs $u_i(t)$ where $i = 1, 2, \cdots, k$, we modify the equation as [15]:

$$\dot{h_1}(t) = \sum_{i=1}^{k} \epsilon_i u_i(t) - \kappa h_1(t) - \chi(h_2(t) - 1) \tag{6.3}$$

Here ϵ_i for $i = 1, 2, \cdots, k$ are the neuronal efficiency factors corresponding to each

neuronal input $u_i(t)$. Having an improved version of the filtering algorithm, we plan to work for more sophisticated smoothing and hemodynamic parameter estimation problems in future work.

7. RESULTS FOR THE HEMODYNAMIC MODEL WITH UNKNOWN PARAMETERS - JOINT STATE AND PARAMETER ESTIMATION

In this Chapter, we discuss the various aspects of the IEKS method in detail. In the first section, we work the filter, smoother and iterative smoother parts for state and parameter estimation using the simulated data. Extensive comparison is made with respect to the currently best-known hemodynamic model inversion technique called SCKS [6]. In the subsequent section, more insight is given to the parameters. We observe the changes in the BOLD signal by changing the parameters in an interval. In the last section, having ground-truth validated the IEKS method, we turn our focus to the application of the IEKS method for a rather complicated multi-input driven BOLD data.

7.1 Contribution: IEKS as an Efficient Hemodynamic Joint State and Parameter Estimator

In this section, we perform Monte Carlo simulations under five different scenarios with different process and measurement noise conditions. The ground-truth values for all the parameters of the hemodynamic model defined in Equations (3.23), (3.25) are taken as shown in Table 6.2 and 3.2 which is in accordance with [34]. For each scenario, 100 Monte Carlo runs are performed. In the first set of scenarios 1, 2, and 3, we fix the measurement noise standard deviation to $\sigma_v = e^{-6}$ as in [6] and change process noise levels. In scenario 1, the process noise standard deviation is as in [6] (i.e., $\sigma_w = \sqrt{\Delta t}e^{-8}$). In scenarios 2 and 3, we try more challenging process noise conditions with $\sigma_w = \sqrt{\Delta t}e^{-6}$ and $\sigma_w = \sqrt{\Delta t}e^{-4}$. In the second set of scenarios, 3, 4, and 5, we fix the process noise standard deviation to $\sigma_w = \sqrt{\Delta t}e^{-4}$, which is the value in scenario 3, and change the measurement noise levels. We increase the measurement noise variance to $\sigma_v^2 = e^{-11}$ and $\sigma_v^2 = e^{-10}$ for Scenarios 4 and 5, respectively.

Table 7.1
Noise Statistics and Initial Values.

Variable	Value
Initial State Value X_0	[0 0 0 0]
Initial State pdf	$\mathcal{N}(0, 0.01)$ for each component
Initial Parameter pdf	$\mathcal{N}(\theta_{true}, 1/12)$ for each component
Parameter Noise w_θ	$\mathcal{N}(0, 10^{-5})$ for each component
Scenario 1: w_k, v_k	$\mathcal{N}(0, \sigma_w^2 = \Delta t e^{-16})$ for each state component, $\mathcal{N}(0, \sigma_v^2 = e^{-12})$
Scenario 2: w_k, v_k	$\mathcal{N}(0, \sigma_w^2 = \Delta t e^{-12})$ for each state component, $\mathcal{N}(0, \sigma_v^2 = e^{-12})$
Scenario 3: w_k, v_k	$\mathcal{N}(0, \sigma_w^2 = \Delta t e^{-8})$ for each state component, $\mathcal{N}(0, \sigma_v^2 = e^{-12})$
Scenario 4: w_k, v_k	$\mathcal{N}(0, \sigma_w^2 = \Delta t e^{-8})$ for each state component, $\mathcal{N}(0, \sigma_v^2 = e^{-11})$
Scenario 5: w_k, v_k	$\mathcal{N}(0, \sigma_w^2 = \Delta t e^{-8})$ for each state component, $\mathcal{N}(0, \sigma_v^2 = e^{-10})$

A priori information about the states are taken as shown in Table 7.1 similar to [6]. The state components are all initialized with 0.

In the simulation, for some rare cases, we observed that the second state x_2 can go to $-\infty$. This corresponds to 0 for the original untransformed variable h_2. Since the transformation is of the exponential type, we set a lower limit $x_2 = -4$ ($h_2 = 0.0183$) for each time step at the filtering step of EKS. The same limit is also put for the other variables. The same thing is also done for PF. PF is even more prone to diverge these limits. The reason is the following: EKF tracks the mean of the state, which is expected to be in the stable region. But the PF tries to represent the conditional pdf $p(x_k|y_{1:k})$, including the extreme conditions also. For that reason, they are more prone to diverge for those particles. The same limits are also put for SCKS.

7.1.1 Comparison of the Filtering Algorithms

In this section, we want to show that the commonly held view that EKF is not an appropriate filtering algorithm is not right. Johnston et al. [5] compared EKF state estimation with particle filter and concluded that PF outperforms EKF, and it is stated that EKF diverges most of the time, and as a result, it is not robust and poor in

Table 7.2
RMS State Errors: Sample Mean and Standard Deviation.

Scenario Number	EKF	PF	EKS
Scenario 1	0.0070 ± 0.0031	0.0075 ± 0.0030	0.0066 ± 0.0029
Scenario 2	0.0095 ± 0.0026	0.0098 ± 0.0026	0.0092 ± 0.0023
Scenario 3	0.0408 ± 0.0034	0.0411 ± 0.0035	0.0344 ± 0.0028
Scenario 4	0.0433 ± 0.0041	0.0434 ± 0.0042	0.0381 ± 0.0036
Scenario 5	0.0454 ± 0.0051	0.0455 ± 0.0050	0.0423 ± 0.0048

performance. We see in this section that, on the contrary, not only state estimation but also joint estimation of parameter and state is robust and performs better than PF. In this subsection, we first compare the EKF and PF state estimation and show that EKF performs better. In Table 7.2, the Monte Carlo results of EKF with PF are compared. In all scenarios, EKF performs better than PF, and it is robust. The particle number is chosen as 500, which is fivefold more than the case where the original comparison was made [5].

7.1.2 Performance Improvement by EKS over EKF

In the fMRI state estimation literature, extended Kalman-type estimation algorithms are only used in the filtering sense [5, 9]. Even the parameter estimation algorithms which rested on extended Kalman filter algorithms, rely on the filtering algorithm [9]. In this subsection, assuming that we have known and fixed parameters, our aim is to estimate the hidden hemodynamic states by using extended Kalman filter and smoother algorithms. We show the performance improvement in Table 7.2. We summarize the EKF and EKS state estimation errors in RMS. For high process and measurement noise conditions, the improvement is more apparent.

7.1.3 Joint Parameter and State Estimation with Iterative EKS compared EKS

A priori information about the parameters are taken as shown in Table 7.1 similar to [6]. The state components are all initialized with 0. Note that the initial values of all the parameters are assumed to be Gaussian distributed around their true values with a variance of 1/12 as in [6]. In the literature, Riera [9] used extended Kalman filter with the discretization proposed by Jimenez et al. [10] without any iteration. This is the standart usage of EKF and EKS. For that reason, in this section we also compare the EKS state RMS error with IEKS utilized in this thesis. Figure 7.1.3 also shows the importance of the iteration. This figure is a typical representative for showing the decrease of RMS state error with respect to the iteration number. The example plot is a particular MC mean estimate for Scenario 2. By increasing the iteration number, the RMS state errors decrease substantially. Obviously, this performance shows that iterative EKS outperforms the direct usage of EKS without iteration.

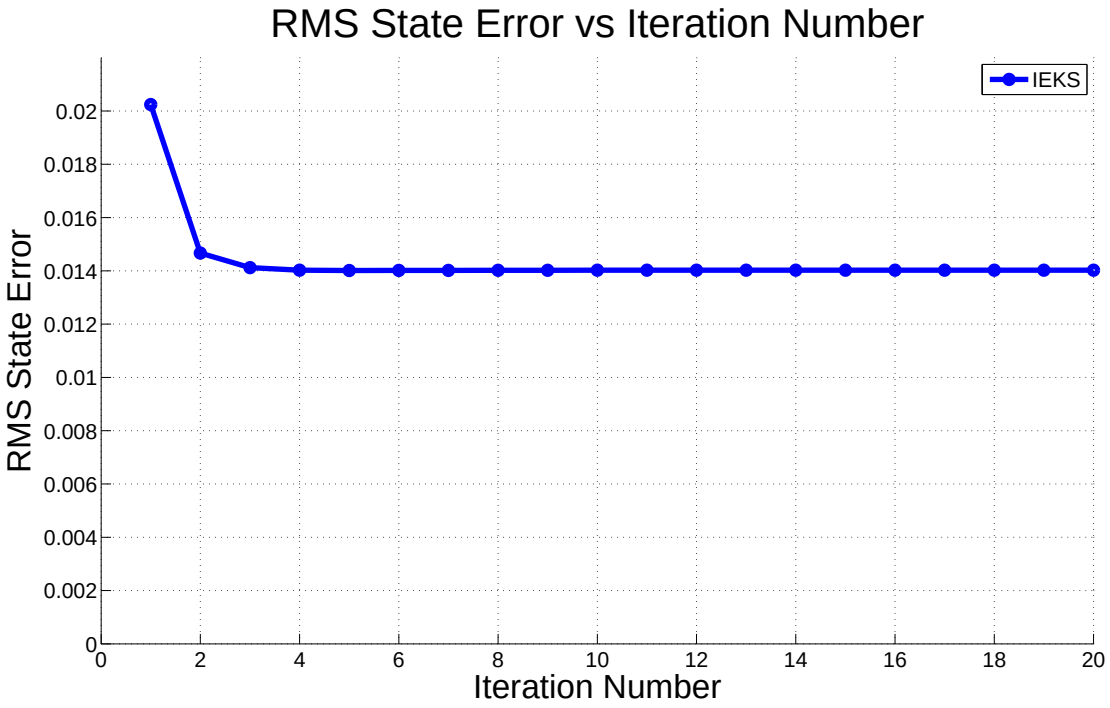

Figure 7.1 IEKS and Importance of Iteration.

7.1.4 Joint Parameter and State Estimation with Iterative

We also summarized the Monte Carlo parameter estimation results of the algorithms IEKS and iterative SCKS in Tables 7.3, 7.4, and 7.5. The true value for the parameters were $\kappa = 0.65$, $\tau = 1.0204$ and $\chi = 0.41$. In all five different process/measurement noise levels, for all parameter values, the accuracy of the estimation of EKS was better than that of SCKS. Only in one case were they equal. We note that the bias of the EKS Monte Carlo estimate is less than the bias of the SCKS case. Similarly, for the state estimation, the error in RMS is summarized in Table 7.6. Although the difference was not big, for all cases, IEKS was better than the SCKS method.

For all scenarios, Figures 7.2, 7.3, 7.4, 7.7, 7.8, 7.9, 7.12, 7.13, 7.14, 7.17, 7.18, 7.19, 7.22, 7.23 and 7.24 show the gradual parameter convergence of the algorithms with respect to the iteration number. Figures 7.5, 7.6, 7.10, 7.11, 7.15, 7.16, 7.20, 7.21,

Table 7.3
Simulation Results for the Parameter κ Estimates.

Scenario	**EKS κ Estimate**	**SCKS κ Estimate**	**EKS κ Bias**	**SCKS κ Bias**
Scenario 1	0.6489 $\pm$ 0.0282	0.6511 $\pm$ 0.0280	0.0011	0.0011
Scenario 2	0.6494 $\pm$ 0.0289	0.6517 $\pm$ 0.0288	0.0006	0.0017
Scenario 3	0.6545 $\pm$ 0.0556	0.6580 $\pm$ 0.0556	0.0045	0.0080
Scenario 4	0.6561 $\pm$ 0.0627	0.6588 $\pm$ 0.0630	0.0061	0.0088
Scenario 5	0.6560 $\pm$ 0.0748	0.6571 $\pm$ 0.0752	0.0060	0.0071

Table 7.4
Simulation Results for the Parameter τ Estimates.

Scenario	**EKS τ Estimate**	**SCKS τ Estimate**	**EKS τ Bias**	**SCKS τ Bias**
Scenario 1	1.0219 $\pm$ 0.0739	1.0282 $\pm$ 0.0740	0.0015	0.0078
Scenario 2	1.0224 $\pm$ 0.0739	1.0288 $\pm$ 0.0740	0.0020	0.0084
Scenario 3	1.0372 $\pm$ 0.1327	1.0460 $\pm$ 0.1335	0.0168	0.0256
Scenario 4	1.0492 $\pm$ 0.1665	1.0578 $\pm$ 0.1679	0.0288	0.0374
Scenario 5	1.0721 $\pm$ 0.2266	1.0791 $\pm$ 0.2277	0.0517	0.0587

7.25 and 7.26 visualize the estimation bounds for parameter and state RMS errors.

Furthermore, IEKS was much faster than SCKS. Both algorithms are robust under different measurement noise conditions. As expected, by increasing the measurement noise, the gradual decrease of the performance of the estimates is observed. The bias and sample variance estimates are increased by increasing the process and measurement noise variance.

Table 7.5
Simulation Results for the Parameter χ Estimates.

Scenario	**EKS χ Estimate**	**SCKS χ Estimate**	**EKS χ Bias**	**SCKS χ Bias**
Scenario 1	0.4116 $\pm$ 0.0092	0.4131 $\pm$ 0.0093	0.0016	0.0031
Scenario 2	0.4111 $\pm$ 0.0092	0.4127 $\pm$ 0.0092	0.0011	0.0027
Scenario 3	0.4100 $\pm$ 0.0164	0.4116 $\pm$ 0.0165	0.0000	0.0016
Scenario 4	0.4112 $\pm$ 0.0182	0.4136 $\pm$ 0.0184	0.0012	0.0036
Scenario 5	0.4124 $\pm$ 0.0219	0.4158 $\pm$ 0.0221	0.0024	0.0058

Table 7.6
RMS State Errors: Sample Mean and Standard Deviation.

Scenario Number	**EKS**	**SCKS**
Scenario 1	0.0128 $\pm$ 0.0038	0.0130$\pm$ 0.0039
Scenario 2	0.0140 $\pm$ 0.0035	0.0143$\pm$ 0.0036
Scenario 3	0.0374 $\pm$ 0.0046	0.0376$\pm$ 0.0047
Scenario 4	0.0418 $\pm$ 0.0053	0.0420$\pm$ 0.0054
Scenario 5	0.0483 $\pm$ 0.0071	0.0486$\pm$ 0.0074

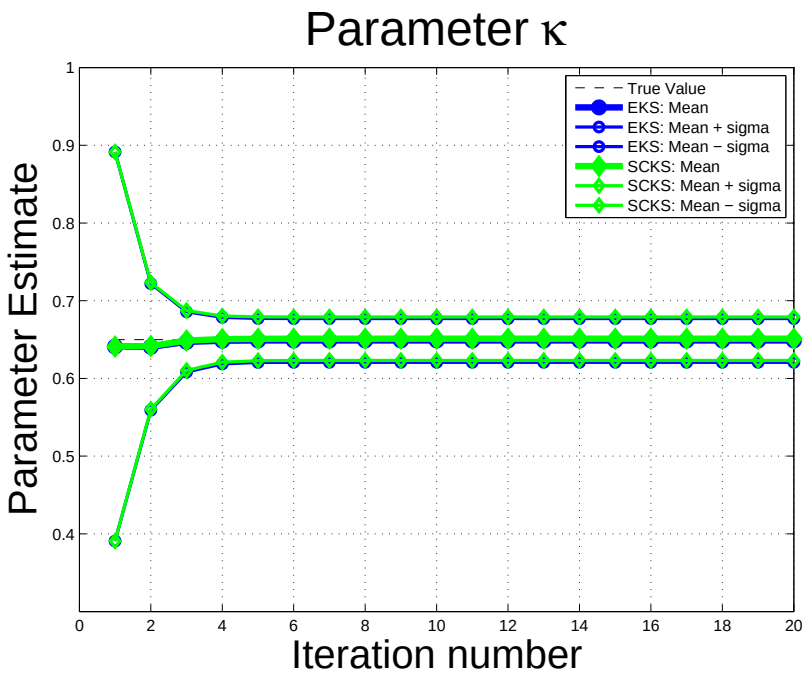

Figure 7.2 The convergence of the parameter κ estimate: Scenario 1.

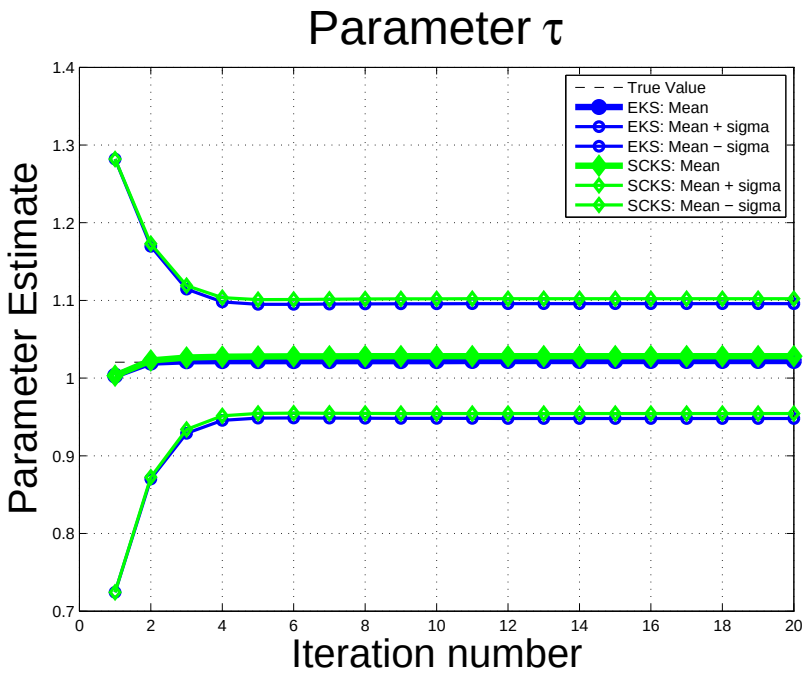

Figure 7.3 The convergence of the parameter τ estimate: Scenario 1.

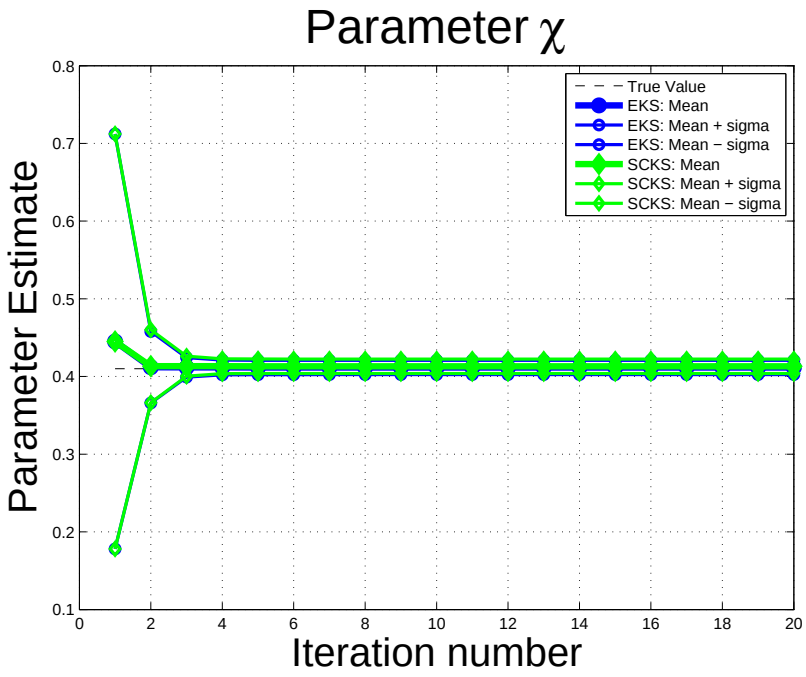

Figure 7.4 The convergence of the parameter χ estimate: Scenario 1.

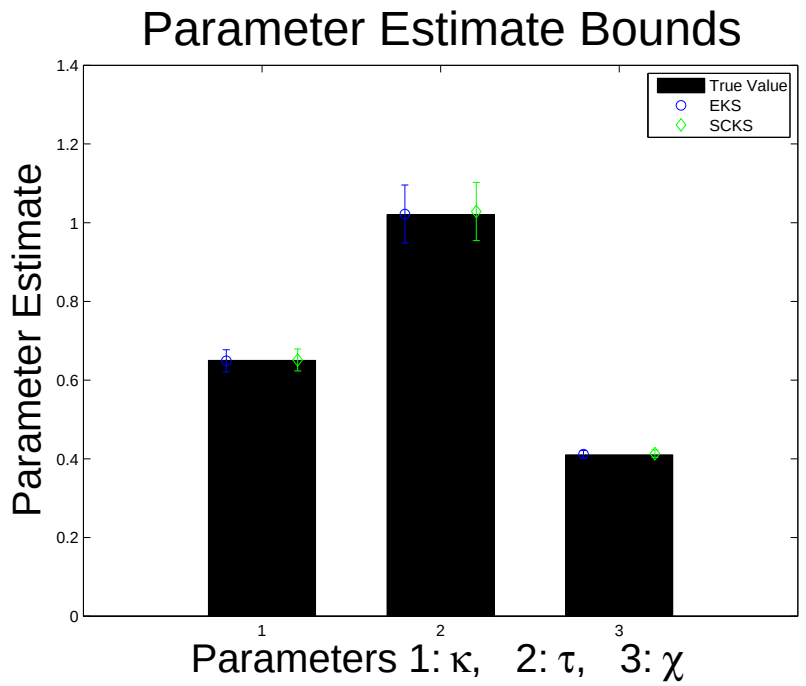

Figure 7.5 Scenario 1: Parameter Estimate Bounds.

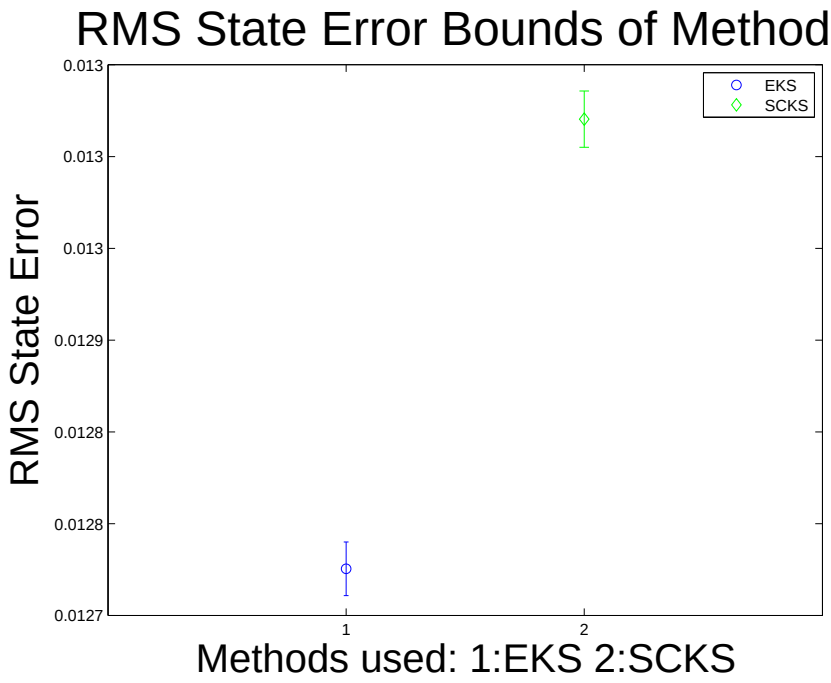

Figure 7.6 Scenario 1: State Estimation Bounds.

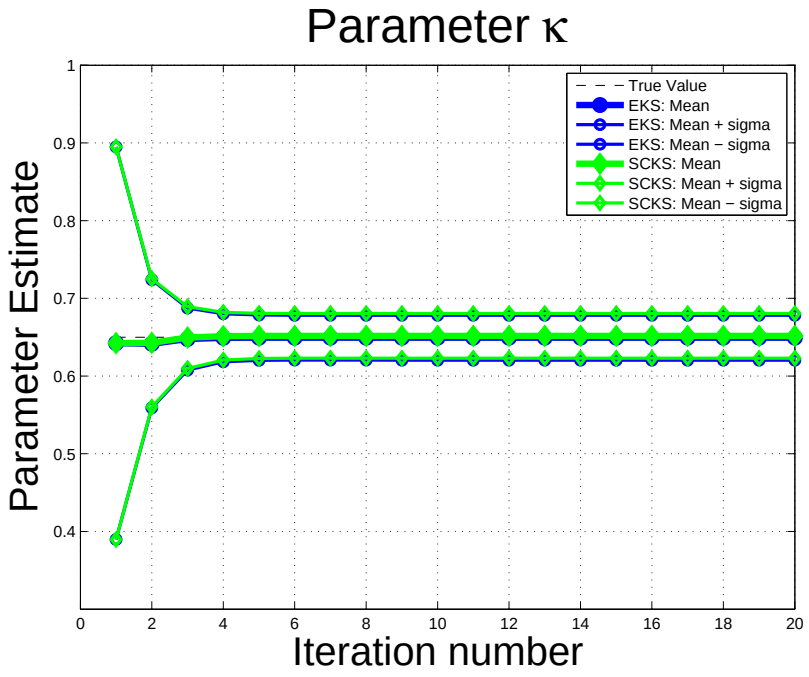

Figure 7.7 The convergence of the parameter κ estimate: Scenario 2.

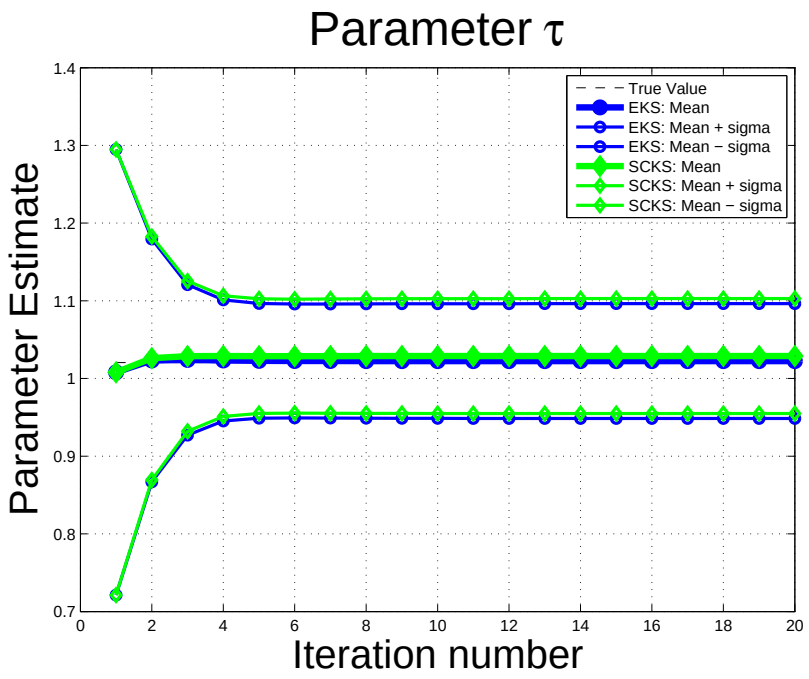

Figure 7.8 The convergence of the parameter τ estimate: Scenario 2.

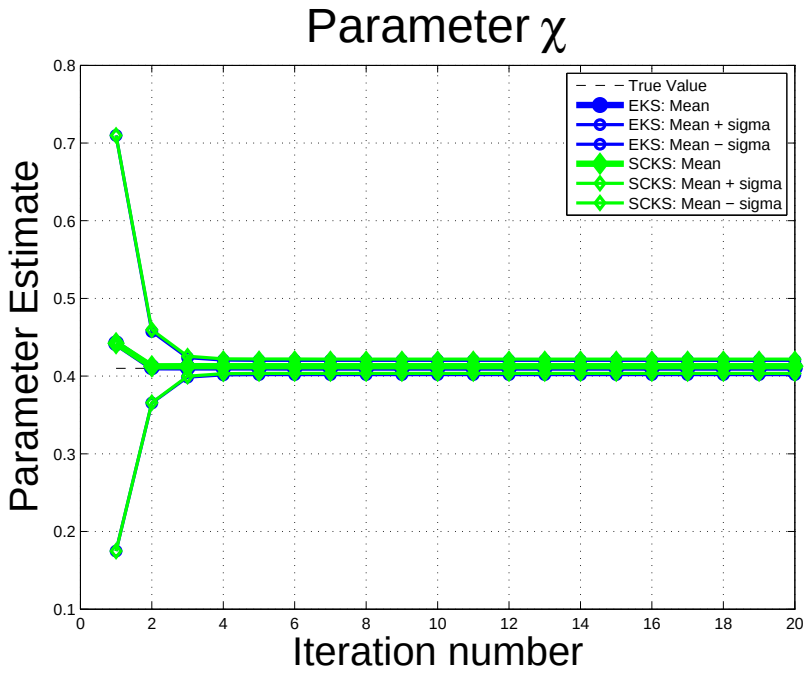

Figure 7.9 The convergence of the parameter χ estimate: Scenario 2.

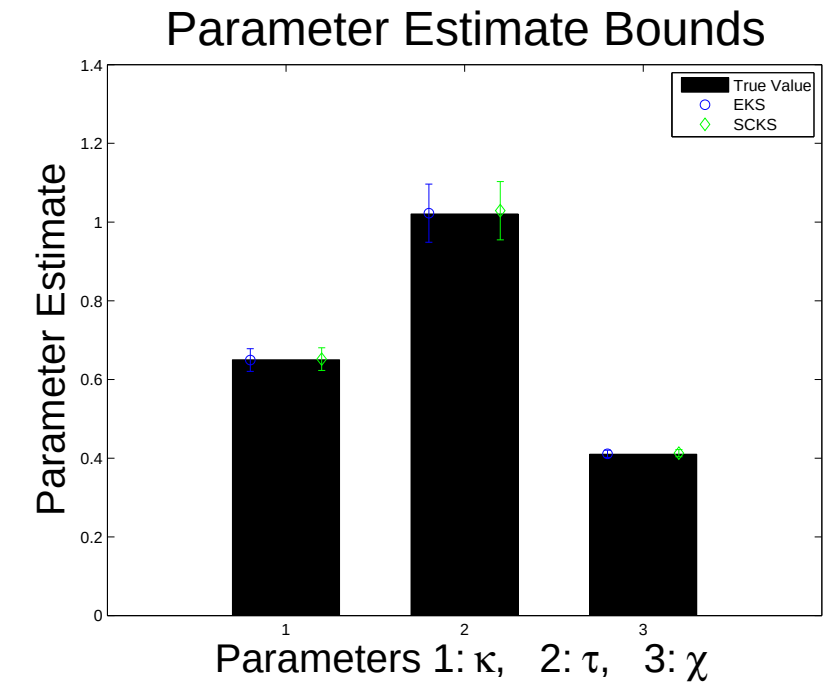

Figure 7.10 Scenario 2: Parameter Estimate Bounds.

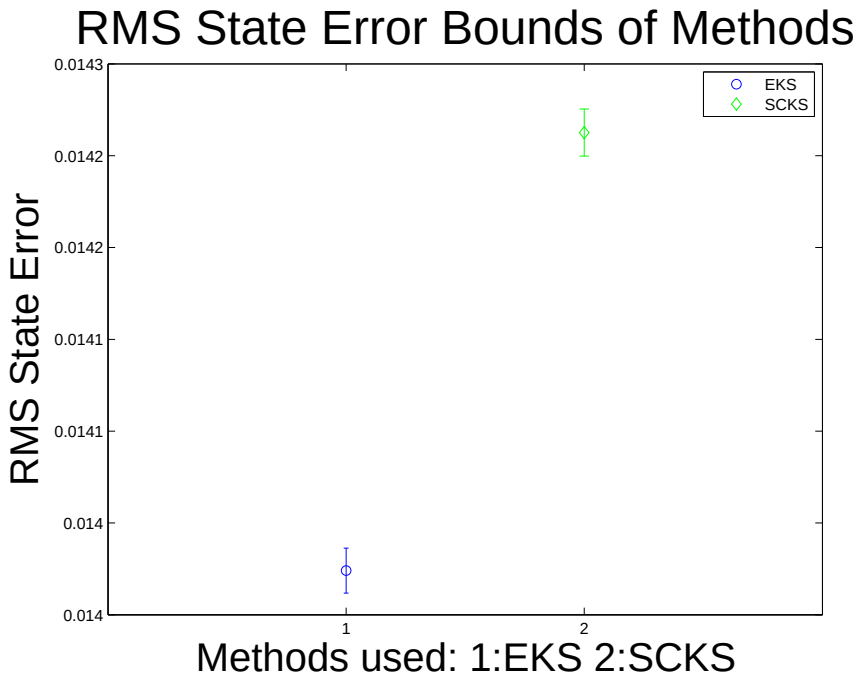

Figure 7.11 Scenario 2: State Estimation Bounds.

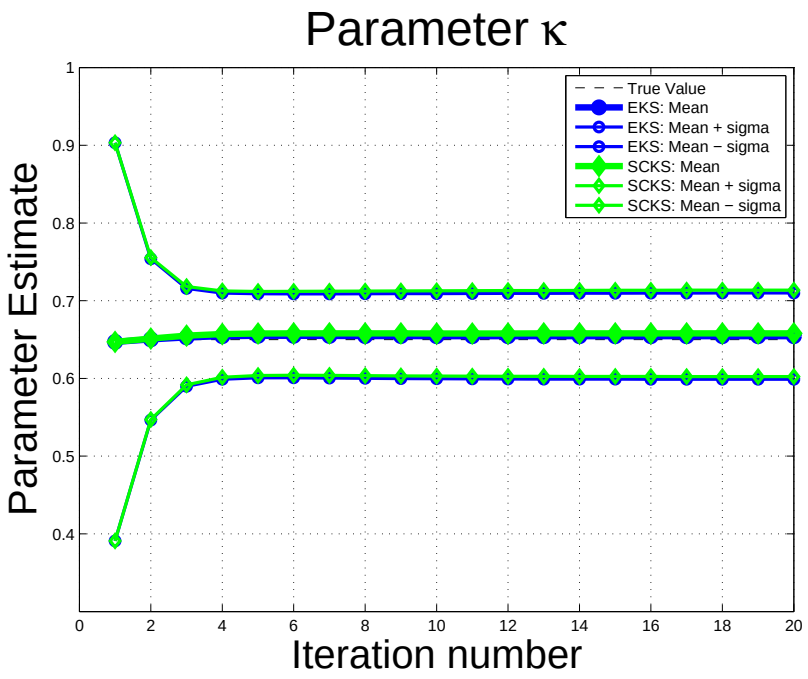

Figure 7.12 The convergence of the parameter κ estimate: Scenario 3.

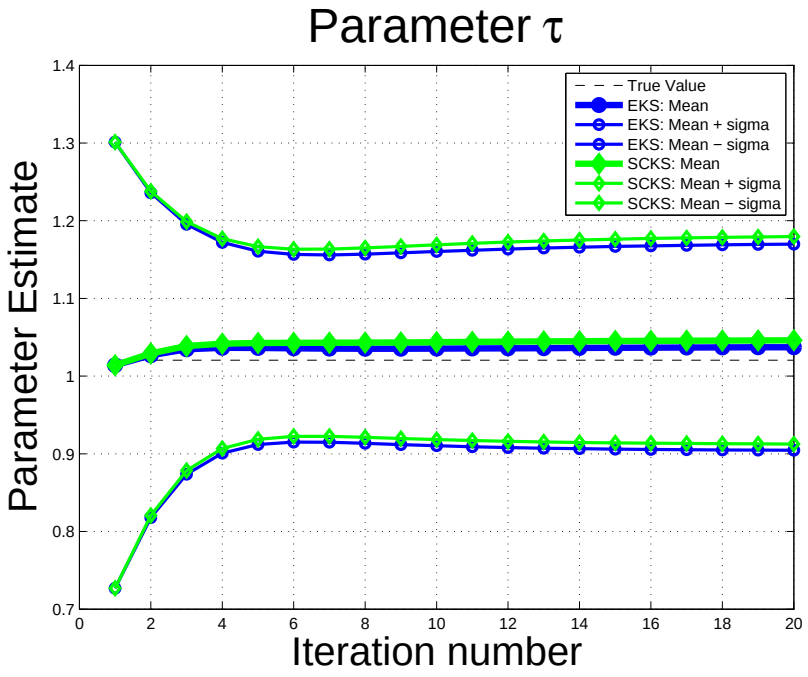

Figure 7.13 The convergence of the parameter τ estimate: Scenario 3.

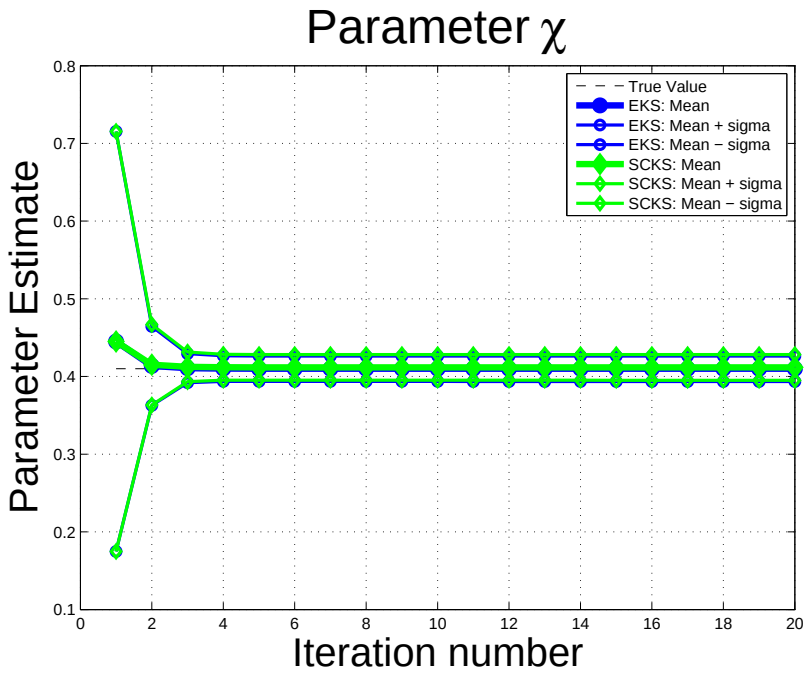

Figure 7.14 The convergence of the parameter χ estimate: Scenario 3.

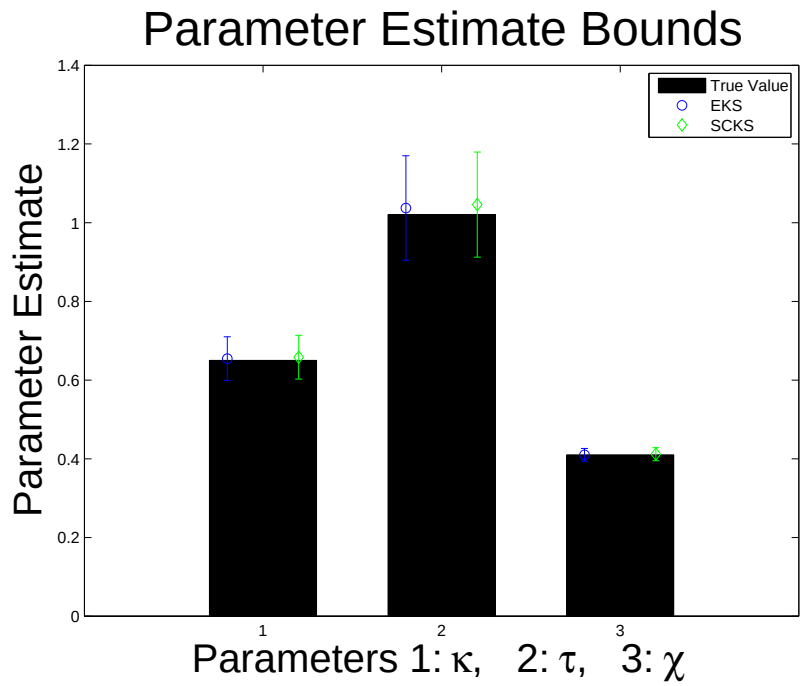

Figure 7.15 Scenario 3: Parameter Estimate Bounds.

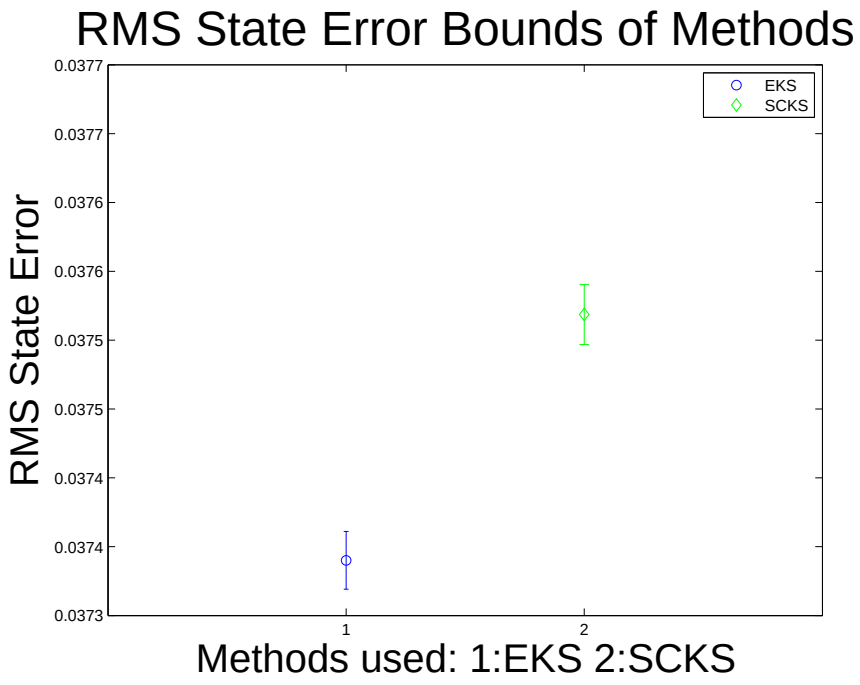

Figure 7.16 Scenario 3: State Estimation Bounds.

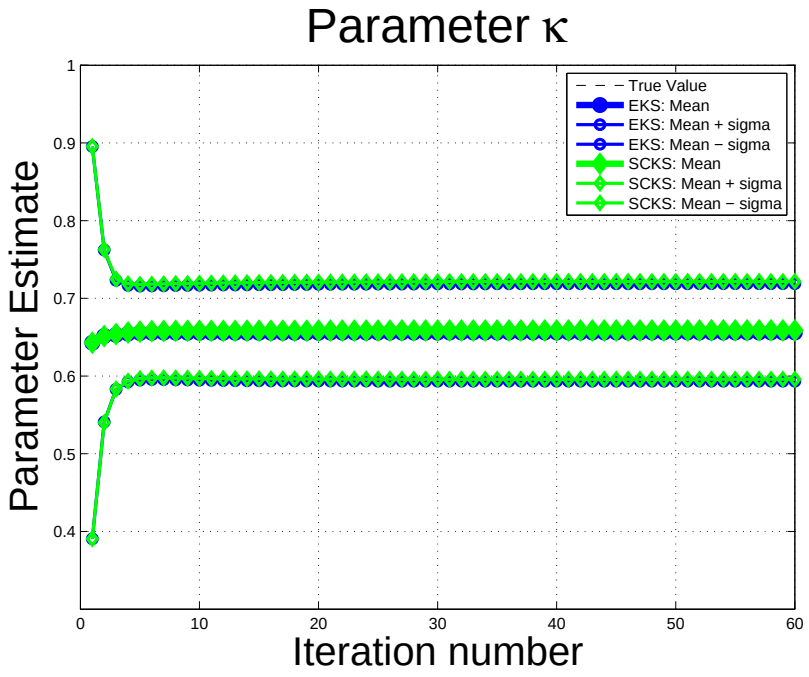

Figure 7.17 The convergence of the parameter κ estimate: Scenario 4.

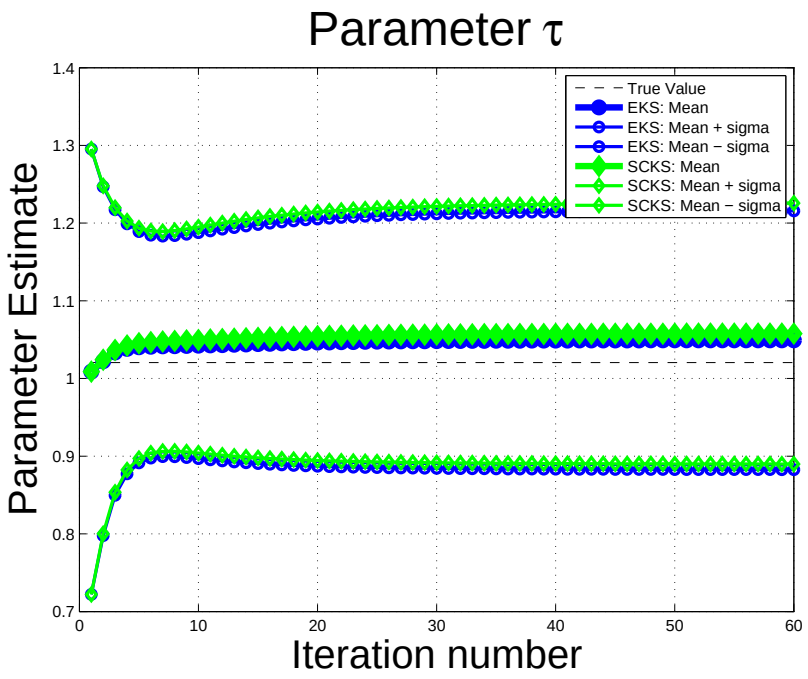

Figure 7.18 The convergence of the parameter τ estimate: Scenario 4.

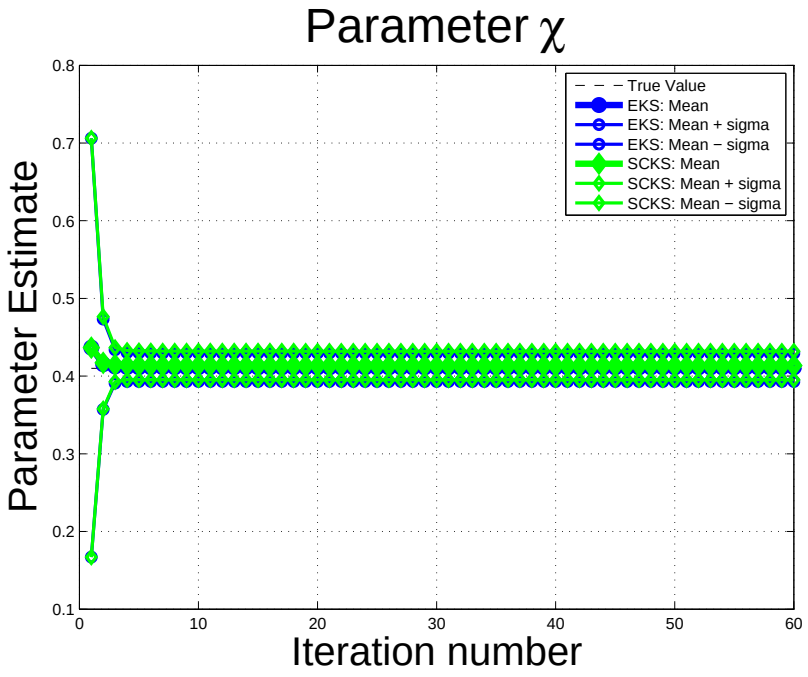

Figure 7.19 The convergence of the parameter χ estimate: Scenario 4.

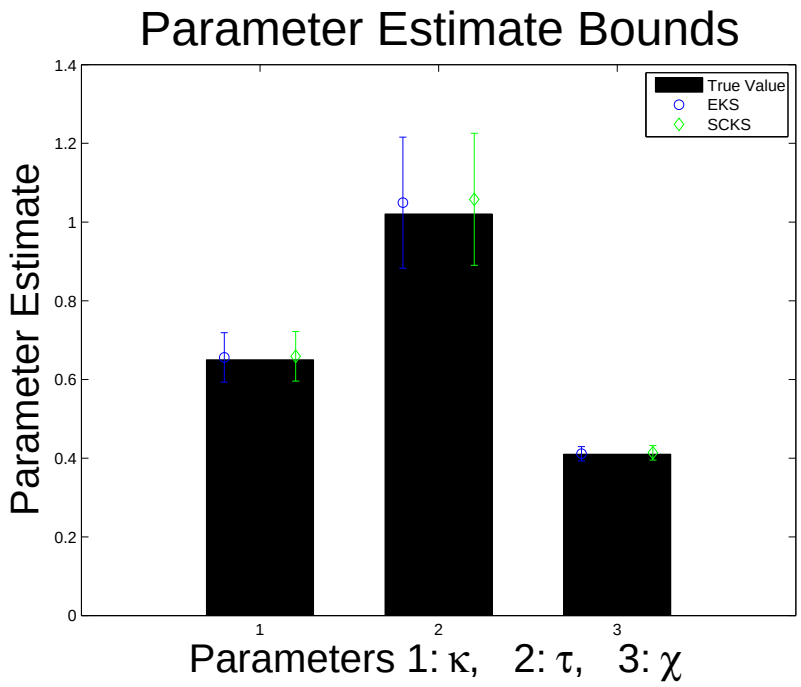

Figure 7.20 Scenario 4: Parameter Estimate Bounds.

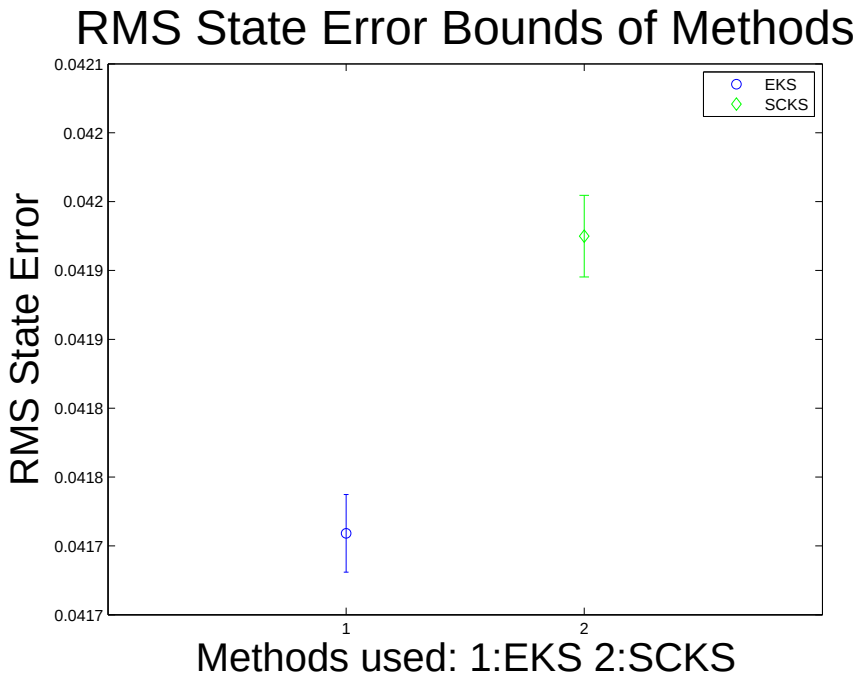

Figure 7.21 Scenario 4: State Estimation Bounds.

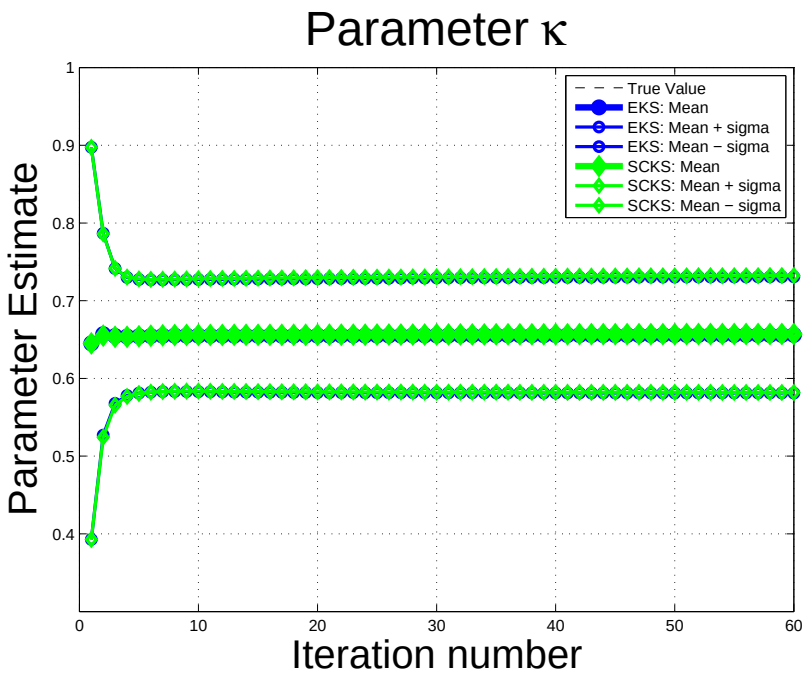

Figure 7.22 The convergence of the parameter κ estimate: Scenario 5.

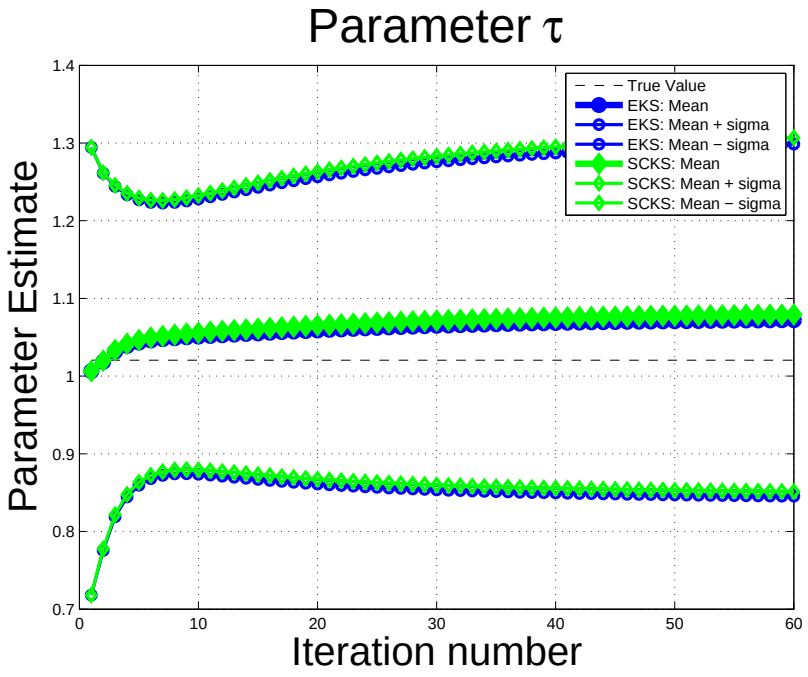

Figure 7.23 The convergence of the parameter τ estimate: Scenario 5.

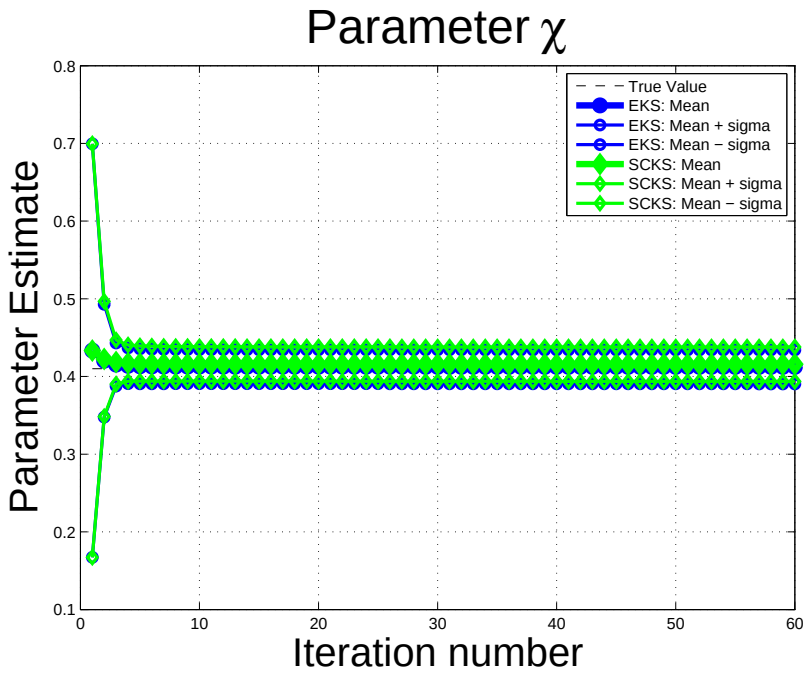

Figure 7.24 The convergence of the parameter χ estimates: Scenario 5.

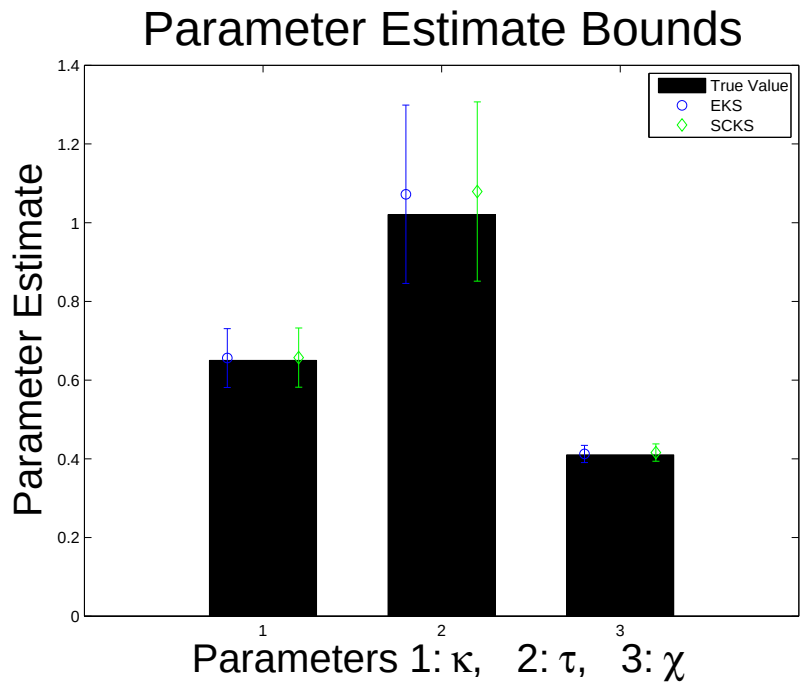

Figure 7.25 Scenario 5: Parameter Estimate Bounds.

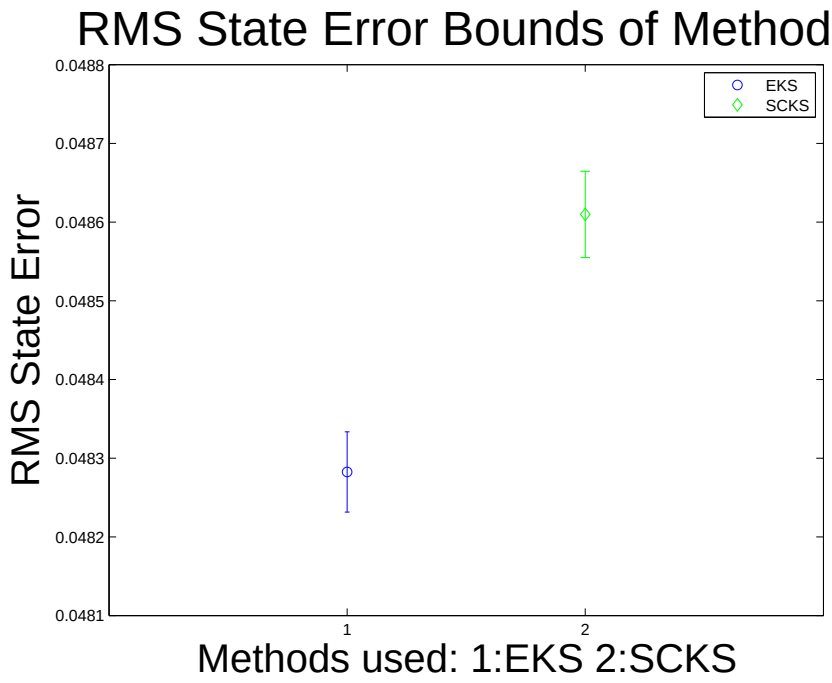

Figure 7.26 Scenario 5: State Estimation Bounds.

7.1.5 Discussion

In this thesis, as the second main contribution, we worked on the joint parameter and state estimation of the hemodynamic nonlinear dynamic system representation. As pointed earlier, the blood oxygen level dependent (BOLD) signal is a measure for the localized hemodynamic response, which is observed after neuronal activation [5]. This response is a nonlinear function of the blood flow and the blood oxygen content [6]. The general shape of the BOLD response can be represented as a mixture of gamma functions [40]. Alternatively, the total system representing the hemodynamic response can be modeled by nonlinear differential equations [1, 2, 3]. In the first approach, where the BOLD signal is represented as the sum of gamma functions, the method called generalized linear model (GLM) is utilized for analysis of the BOLD signal. However, we followed in this thesis the second main approach which based on the stochastic differential modeling of the hemodynamic system. In this approach, people worked on advanced model inversion techniques instead of GLM [5, 6, 8, 9, 11, 13, 14, 15]. Our

algorithms are classified under the same category.

The waveforms of the hemodynamic variables change from subject to subject, even in different times during the day and different regions of the brain [4], [7]. Model inversion techniques which perform joint parameter and state estimation are capable to detect these differences by having different estimated parameter sets. The IEKS method is one of the few model inversion techniques which can be used for that purpose.

In the fMRI model inversion literature, in one of the keystone papers [5], it was asserted that EKF is not a robust method for state estimation, and its performance was poor compared with PF. The assertion that EKF is poorer in performance was repeated in the fMRI literature in several landmark papers without examining specifically for the hemodynamic modeling case [6, 14]. In this thesis, we examined EKF for a variety of noise conditions and concluded that this is not the case. It even performed better than particle filter. We take fivefold more particles than the study in [5], which makes that assertion. Standard particle filters were alleged to perform better than the other hemodynamic state estimation algorithms. In this thesis, we concluded that, on the contrary, EKF performed better than the other filtering algorithms. The reason is that standard particle filters use a nonoptimal proposal function, which degrades the performance. There is still room for particle filters, which may use more sophisticated proposal functions. The first contribution PF-EKF used this fact and resulted in more accurate state estimation than PF, APF, EKF and SCKF under known parameter condition.

In this thesis, the importance of the smoothing for both the hemodynamic state and the parameter estimation is emphasized. To our knowledge extended Kalman-type algorithms in the fMRI literature have not been used with smoothing so far [5, 9]. The extended Kalman filter is modified by incorporating the smoother. For fixed hemodynamic parameters, we checked the state estimation improvement for various noise conditions. Especially for higher conditions, the improvement was more apparent compared to SCKS.

Furthermore, in the standard application of the filtering and smoothing, EKF and EKS are performed only one time. By iteratively calling EKF and EKS after each new parameter estimate, we noticed the huge improvement over non-iterative usage. This improvement is shown in Figure 7.1.3. As a result, EKS is just the special case of IEKS with just one iteration. Like Havlicek et al. [6] and Hu et al. [11], we treated the parameters as time-varying variables by adding artificial dynamics. We iterated the algorithm until the IEKS algorithm converged. The improvement of the parameter estimation of the hemodynamic model with respect to the iteration number is shown under a variety of noise conditions in Figures 7.2, 7.3, 7.4, 7.7, 7.8, 7.9, 7.12, 7.13, 7.14, 7.17, 7.18, 7.19, 7.22, 7.23 and 7.24 . Havlicek et al. compared the iterative usage of SCKS with DEM and concluded that SCKS is more accurate in their paper [6]. In this report, IEKS is compared with iterative SCKS under different noise conditions. The first scenario we used has the same noise conditions as in [6]. This model has a very low process noise covariance. Even in this condition, our method was more accurate for both the state and the parameter estimation. Overall, both methods were good. For low process and measurement noise, it is quite reasonable to approximate the densities by Gaussian density, because the system is almost deterministic. As a result, we expected that EKS and SCKS perform well. However, in nonlinear systems, even the additive noise is Gaussian; the conditional densities $p(x_k|y_{1:k})$ and $p(x_{k+1}|y_{1:k})$ actually deviate from the Gaussian assumption. The higher the covariances of the noises, the more severe these conditional densities deviate from the Gaussian assumption. Although this is the case, for the hemodynamic model, Gaussian approximated approaches worked quite well. We tested the algorithms under five different process/measurement noise conditions. IEKS was more accurate than iterative SCKS for both parameter and state estimation for all cases.

Besides the ease of implementation of EKS, it was also much faster than SCKS. EKS is around 2.3 times faster than SCKS.

In the fMRI model inversion literature, Riera et al. [9] used a kind of extended Kalman filter where the discretization was not based on the Euler-Maruyama method. They used the method proposed by Jimenez et al. [10]. Hu et al., on the other hand,

used the unscented Kalman filter (UKF) [22] for the estimation of the hemodynamic model states and parameters [11]. Riera et al. [9] and Hu et al. [11] used these techniques in a forward pass manner without smoothing. The early attempts in the fMRI model inversion literature disregarded the state noise. For example, in [1] Friston et al. models the input and the output relation via Volterra kernels. Later, Friston used a Bayesian estimation technique [8] to find the posterior of the parameters again under zero state noise assumption. Later, Friston et al. suggested advanced techniques based on variational filtering in which he also included the nonzero state noises in the model [13, 14, 15]. Variational filtering techniques assume factorization of the parameters and states. Dynamic expectation maximization (DEM) is the most popular of them. As Havlicek et al. pointed out, DEM also works in the forward pass manner [6]. Johnston et al. [5] used particle filters, without using any smoothing. However, Murray and Storkey [12] used particle smoothers. Most recently, Havlicek et al. [6] used the square-root cubature Kalman smoother (SCKS) and obtained quite a remarkable success compared with DEM under certain noise conditions. Most of the signal analysis techniques in the fMRI literature can be regarded as a kind of Gaussian approximated model inversion technique [6, 9, 11]. IEKS is included in the same category.

So overall, EKF is not an unreliable model inversion method. It is better than the PF algortihm claimed as the most successful one in [5]. The iterative EKS is substantially more accurate than the formerly used EKF-type algorithms in the fMRI model inversion literature. EKS is robust under a wide range of noise conditions. IEKS is faster and has lower parameter bias and more accurate state estimation than the SCKS method, which seems to work best among the current fMRI model inversion methods [6] for all the noise levels we worked. We discuss the features of IEKS more in the last section when utilizing IEKS for real BOLD data.

7.1.5.1 Parameter Sweep and the BOLD Signal. In this section we want to observe the changes in the hemodynamic state variables and the BOLD signal with respect to the parameter changes. We let the parameter values as in Tables 6.2 and 3.2. We sweep the parameters for κ in the interval [0.35 1.05], τ in the interval [0.32

2.32], χ in the interval [0.21 1.01], ϵ in the interval [0.01 0.51]. Figures 7.27, 7.28, 7.29 and 7.30 the BOLD signal change with respect to the parameters $\kappa, \tau, \chi, \epsilon$, respectively.

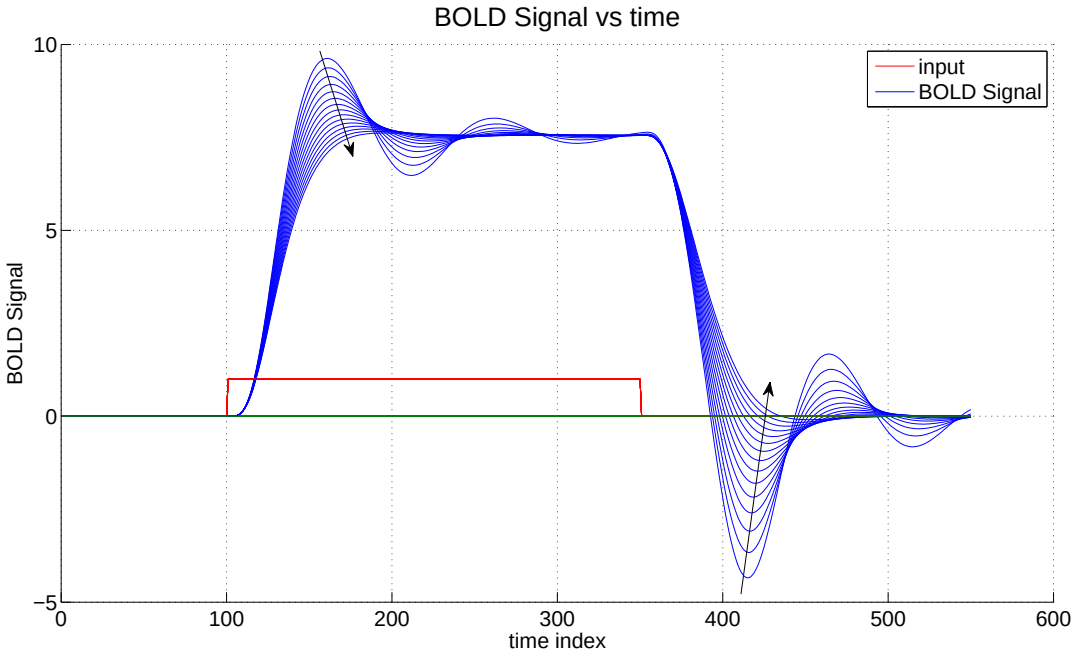

Figure 7.27 BOLD signal change with respect to κ change. κ is changed over the interval [0.35;1.05].

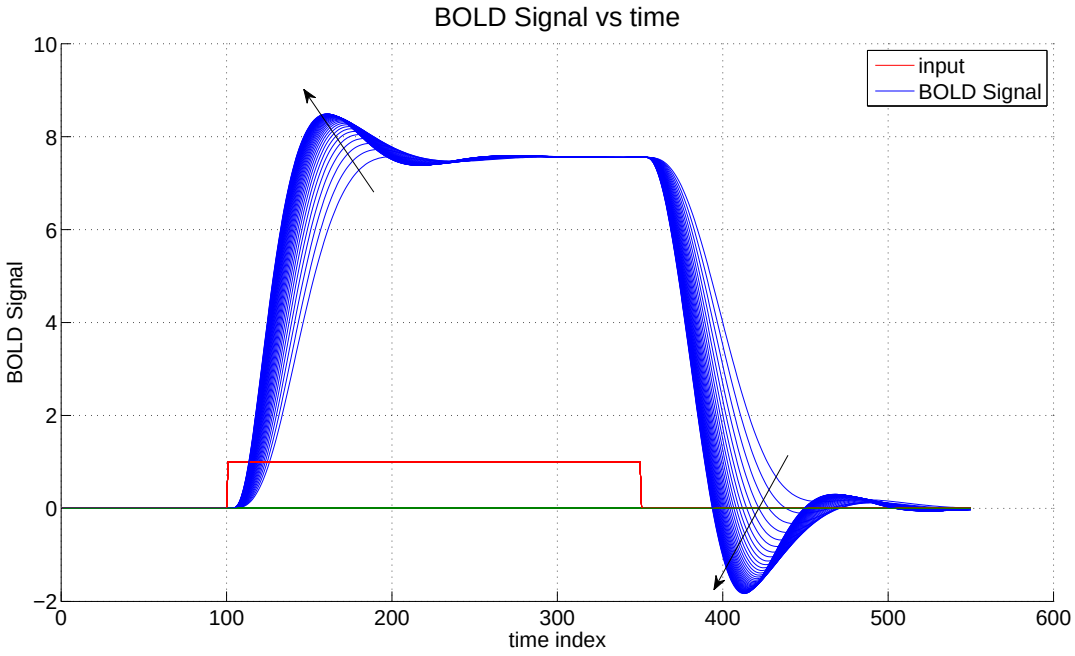

Figure 7.28 BOLD signal change with respect to τ change. τ is changed over the interval [0.32;2.32].

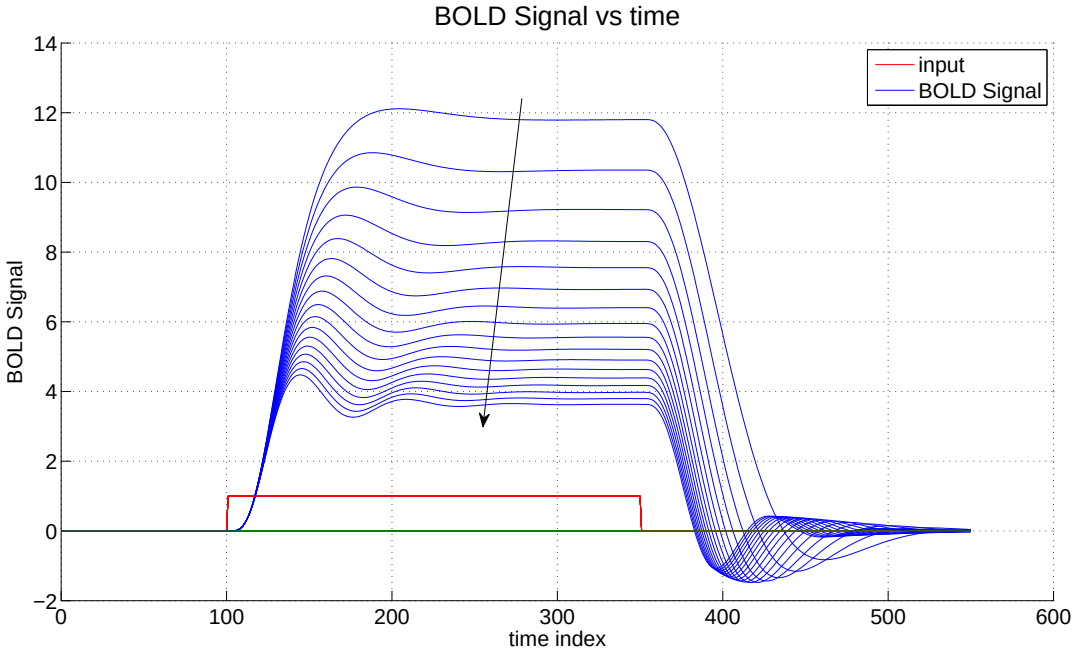

Figure 7.29 BOLD signal change with respect to χ change. χ is changed over the interval [0.21;1.01].

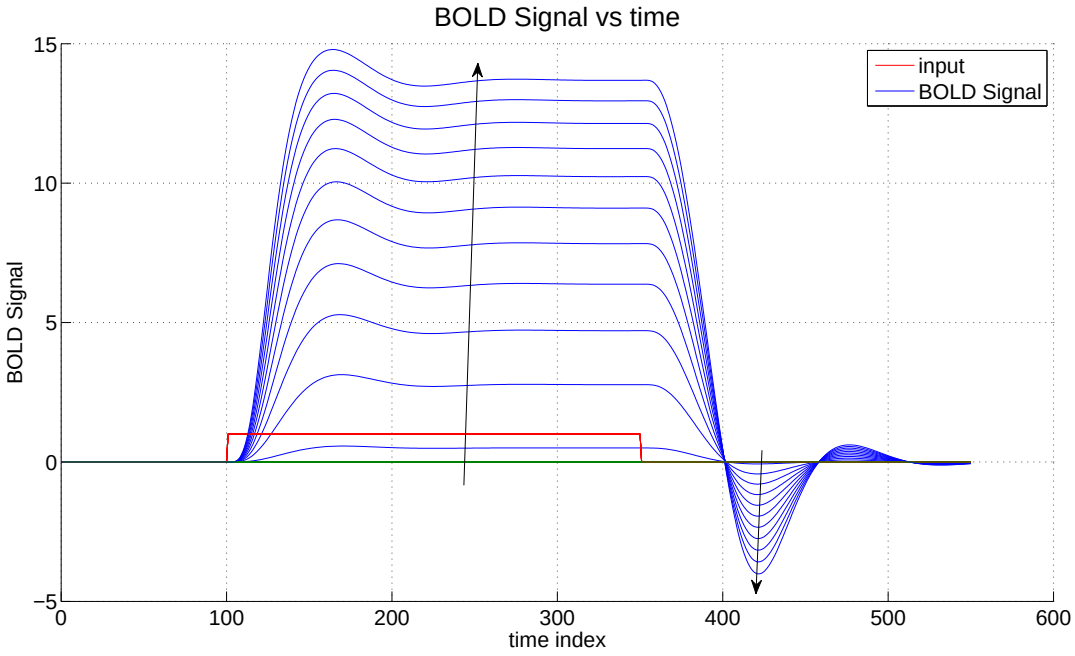

Figure 7.30 BOLD signal change with respect to ϵ change. ϵ is changed over the interval [0.01;0.51].

7.2 Real Data and the IEKS method

7.2.1 Background Information: Test Setup and BOLD Data

In this section, we want to show the validity of the IEKS method on a real BOLD data. This data was also used by Friston et al. to show the practical applicability of their model inversion methods DEM, VF and GF [13, 14, 15] [1]

The BOLD data was collected from the motion sensitive area V5. During the test, the subject was exposed to 5 different conditions. The first one was to allow for magnetic saturation effects [13, 14, 15]. In the second one called "Fixation", subject was viewing a fixation point in the screen. In the third one called "Attention", subject was viewing 250 dots. Those dots were moving radially by 4.7° from the center. At the

[1] We thank Professor Karl Friston to give consent us to use BOLD data in our researches.

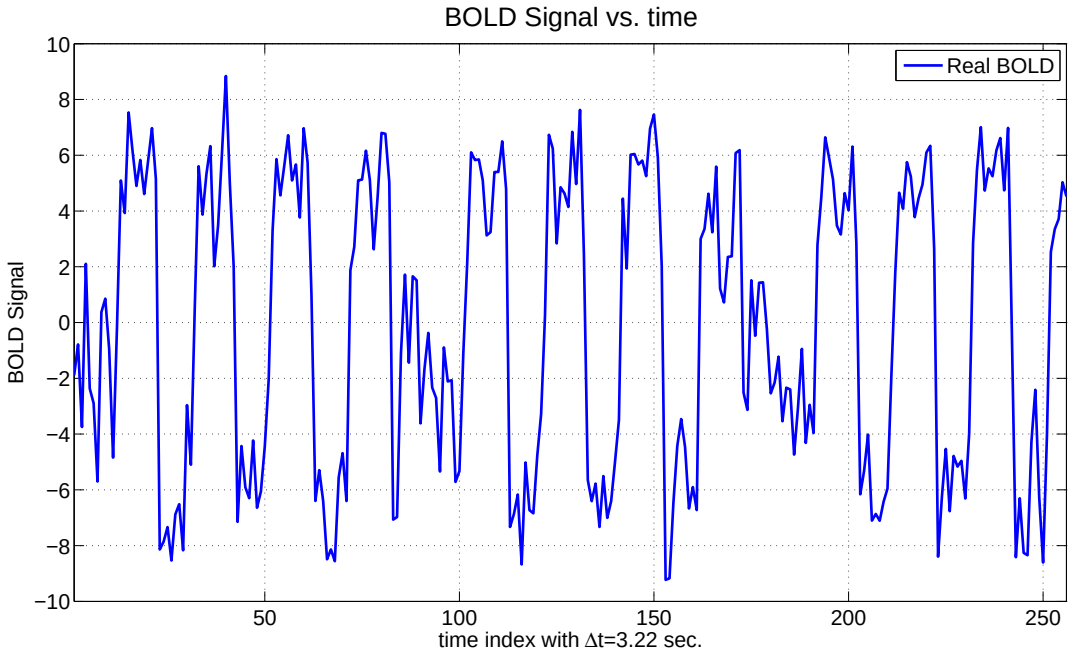

Figure 7.31 Real BOLD data is collected from the motion sensitive area V5. The sampling time TR is 3.22 sec. The first 256 samples were used for model inversion.

same time, the subject was supposed to detect the changes in radial velocity [13, 14, 15]. In the fourth one called "No Attention", the person was just viewing the moving dots. In the fifth one, the person was supposed to view the stationary points. During the entire test, the subject was exposed to the fixation and visual stimulus conditions alternatively.

Friston et al.[13, 14, 15] modeled these five different conditions as a combination of 3 different neuronal inputs . The first one is visual stimulus, the second one is motion stimulus and the third one is the attention stimulus represented by u_1, u_2 and u_3, respectively. Each neuronal input is modeled by boxcar functions. For example for the third condition "Attention", in those time instants u_1, u_2 and u_3 have all the value 1. The reason is that in the "Attention" condition the person is supposed to view, track the motion and shall attend the changes in radial velocities [13, 14, 15]. In the "No attention" condition u_1, u_2 and u_3 are 1, 1 and 0, respectively. The reason is that in this condition the person views, tracks the motion but there is nothing for attention. In the fifth condition u_1, u_2 and u_3 are 1, 0 and 0. The reason is that there

is visual stimulus, but the dots are not moving and there is nothing for the attention. Similarly, in the "Fixation" condition u_1, u_2 and u_3 are all 0. One further note is that Friston et al. [13, 14, 15] have the real BOLD signal with DC value 0 and remove the DC values from u_1, u_2 and u_3 . As a result, they used the neuronal input sequences depicted in Figure 7.32 [13, 14, 15].

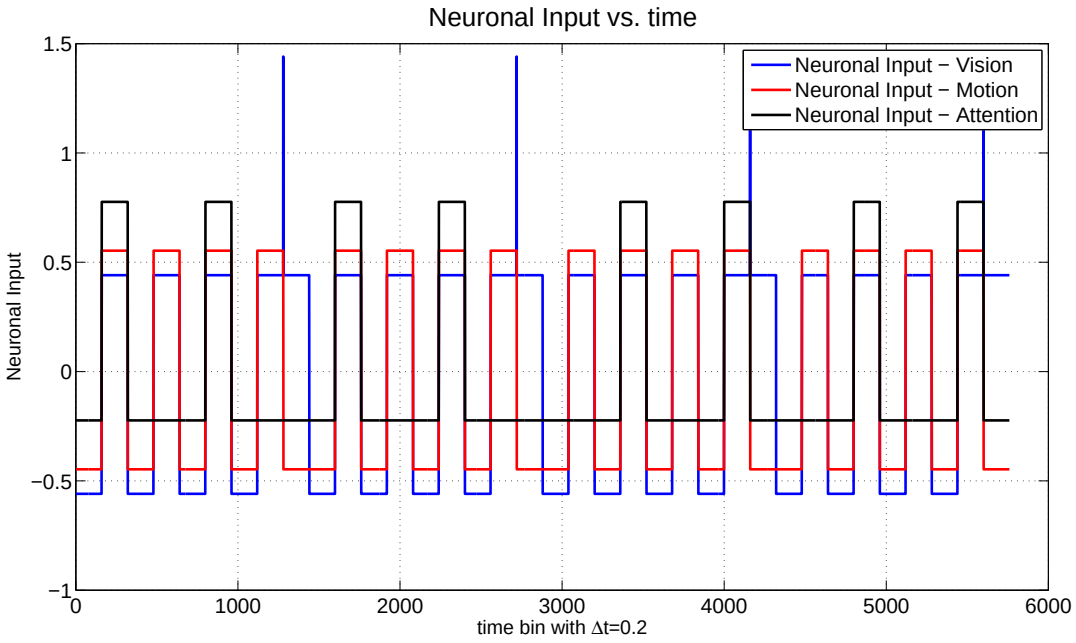

Figure 7.32 In this setup, the person was exposed to 3 different neuronal inputs. They are Vision, Motion and Attention. Those neuronal inputs are modeled as boxcar functions. During the test, the person was subject to 5 different conditions. Those are Saturation, Fixation, Attention, No Attention and Motion View. Depending on the condition, for each neuronal input, we have the value of either 1 or 0. Following Friston et al. [13, 14, 15], DC values of the inputs are removed.

7.2.2 Extended Hemodynamic Model for Multi-Input System

Since we have in this setup a rather complicated multi-input system, we extend the nonlinear hemodynamic system to the following nonlinear differential equation system.

Table 7.7
Noise Statistics and Initial Values.

Variable	Value
Initial State Value X_0	[0 0 0 0]
Initial State pdf	$\mathcal{N}(0, 0.01)$ for each component
Initial Parameter pdf	$\mathcal{N}(0, 1/12)$ for $\epsilon_1, \epsilon_2, \epsilon_3$
Initial Parameter pdf	$\mathcal{N}(0.65, 1/12), \mathcal{N}(1.02, 1/12), \mathcal{N}(0.41, 1/12)$ for κ, τ, χ respectively
Parameter Noise w_θ	$\mathcal{N}(0, \Delta t 10^{-8})$ for each component
Process Noise: w_k	$\mathcal{N}(0, \sigma_w^2 = \Delta t e^{-8})$ for each state component
Measurement Noise: v_k	$\mathcal{N}(0, \sigma_v^2 = e^{-12})$

$$\dot{h_1}(t) = \epsilon_1 u_1(t) + \epsilon_2 u_2(t) + \epsilon_3 u_3(t) - \kappa h_1(t) - \chi(h_2(t) - 1) \tag{7.1}$$

$$\dot{h_2}(t) = h_1(t) \tag{7.2}$$

$$\dot{h_3}(t) = \tau(h_2(t) - F(h_3(t))) \tag{7.3}$$

$$\dot{h_4}(t) = \tau(h_2(t)E(h_2(t)) - F(h_3(t))\frac{h_4(t)}{h_3(t)}) \tag{7.4}$$

Here ϵ_1, ϵ_2 and ϵ_3 are the neuronal efficiency factors for the neuronal inputs u_1, u_2 and u_3 respectively. In this setup ϵ_1, ϵ_2 and ϵ_3 give information how much the selected brain region is sensitive to the vision input u_1, to motion input u_2 and to attention input u_3, respectively.

In order to have a best estimate for the parameter set $\theta = [\epsilon_1, \epsilon_2, \epsilon_3, \kappa, \tau, \chi]$. We performed a Monte Carlo simulation and began with different initial parameter sets. The initial parameter set and noise conditions are chosen according to Table 7.7.

7.2.3 Results: IEKS Model Inversion

In this section, we apply the IEKS method to the BOLD data. We perform Monte Carlo simulation with 100 runs to estimate the parameters. The initial param-

eter set $\theta = [\epsilon_1, \epsilon_2, \epsilon_3, \kappa, \tau, \chi]$ is chosen according to the prior statistics given in Table 7.7. We iterate until the algorithm converges. The IEKS algorithm converged to 2 values which are global and local optima. We denote the two parameter sets as θ_g and θ_l.

$$\theta_g = [0.1024, 0.2102, 0.0175, 0.7285, 0.4981, 0.6460] \tag{7.5}$$

$$\theta_l = [0.0329, 0.0776, 0.0091, 0.6349, 2.2990, 0.2381] \tag{7.6}$$

When we note the neuronal efficiency factors ϵ_1, ϵ_2 and ϵ_3, we observe close resemblance of the converged sequences to the ones reported by Friston et al. [13, 14, 15]. Both θ_g and θ_l shows the neuronal activation most for ϵ_2 which corresponds for the "Motion" condition. Somewhat for ϵ_1 which corresponds to the "Vision" condition and almost nothing for "Attention" condition. The difference is in the strength.

For each simulation and at every iteration, RMS error in the BOLD signal prediction is evaluated according to the formula.

$$e_{rms} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} \|\hat{y}_k - y_k^p\|^2} \tag{7.7}$$

where y_k and y_k^p is the real and predicted BOLD signal, respectively.

RMS errors for global optimum value are lower than the ones for local optimum. Furthermore, at every iteration, the global optimum RMS values are improved, whereas the local optimum RMS errors are getting worse. The parameter estimates which converge to global optima and local optima are shown in Figures 7.33 and 7.35. Similarly, the RMS errors in prediction can be seen in Figures 7.34 and 7.36.

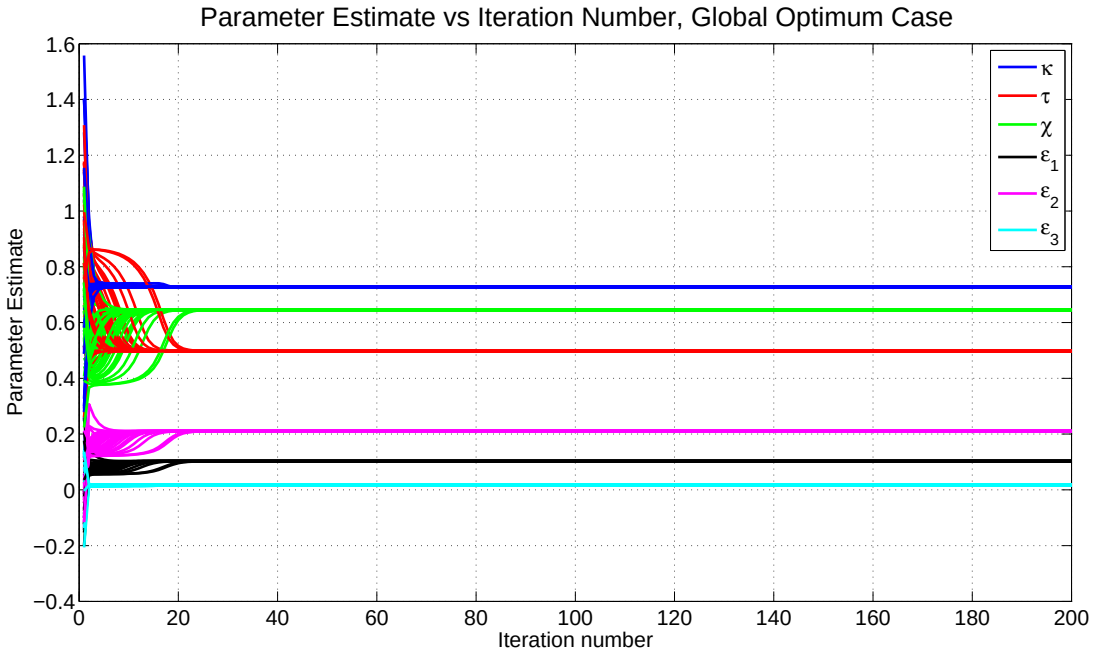

Figure 7.33 The simulations which converge to global optima θ_g. Specifically, we estimate the parameter set $\theta = [\epsilon_1, \epsilon_2, \epsilon_3, \kappa, \tau, \chi]$ which converged to the global optima θ_g.

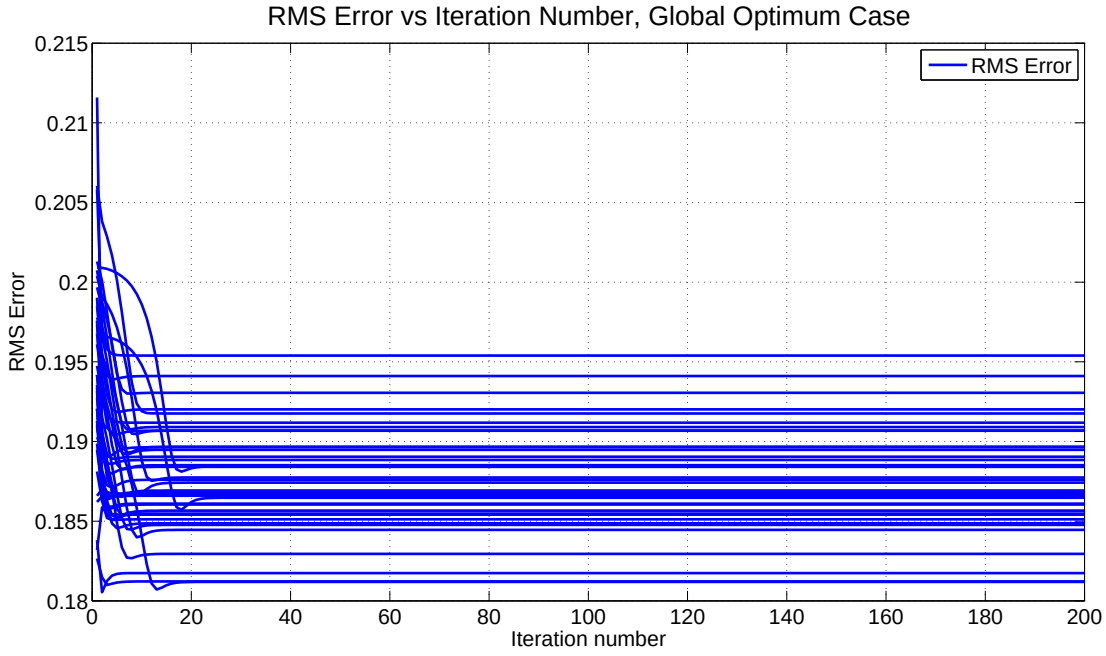

Figure 7.34 The simulations which converge to global optima. For every iteration, the RMS error of the prediction is shown. At every step, the RMS errors are decreasing. After around 25 iterations, the algorithm converges.

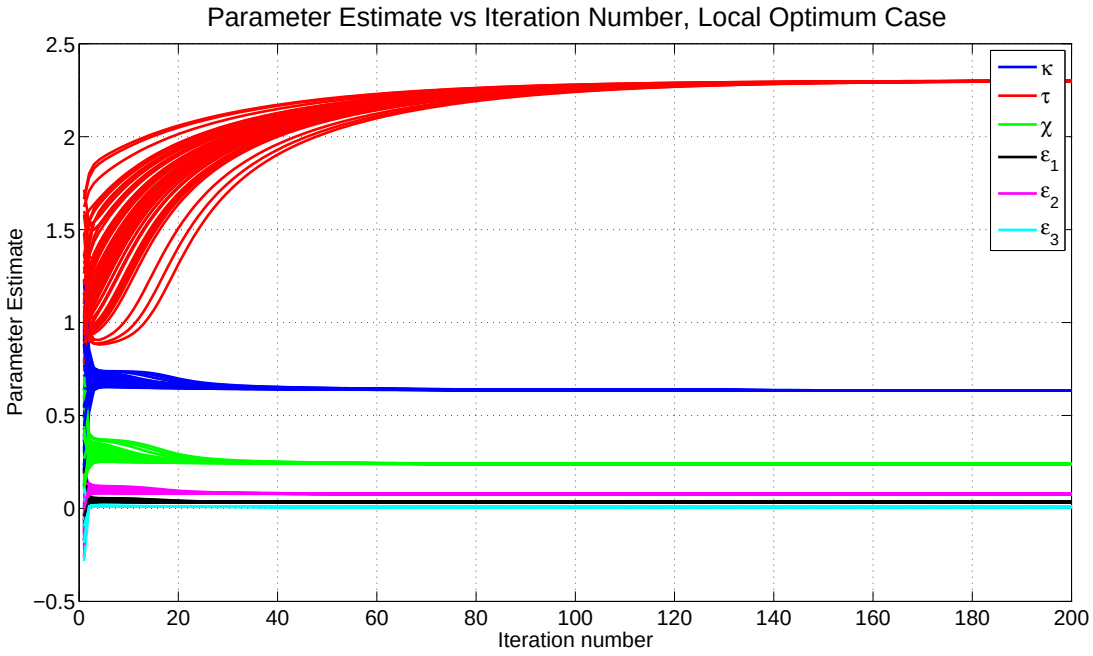

Figure 7.35 The simulations which converge to local optima. Specifically, we estimate the parameter set $\theta = [\epsilon_1, \epsilon_2, \epsilon_3, \kappa, \tau, \chi]$ which converged local optima is θ_l.

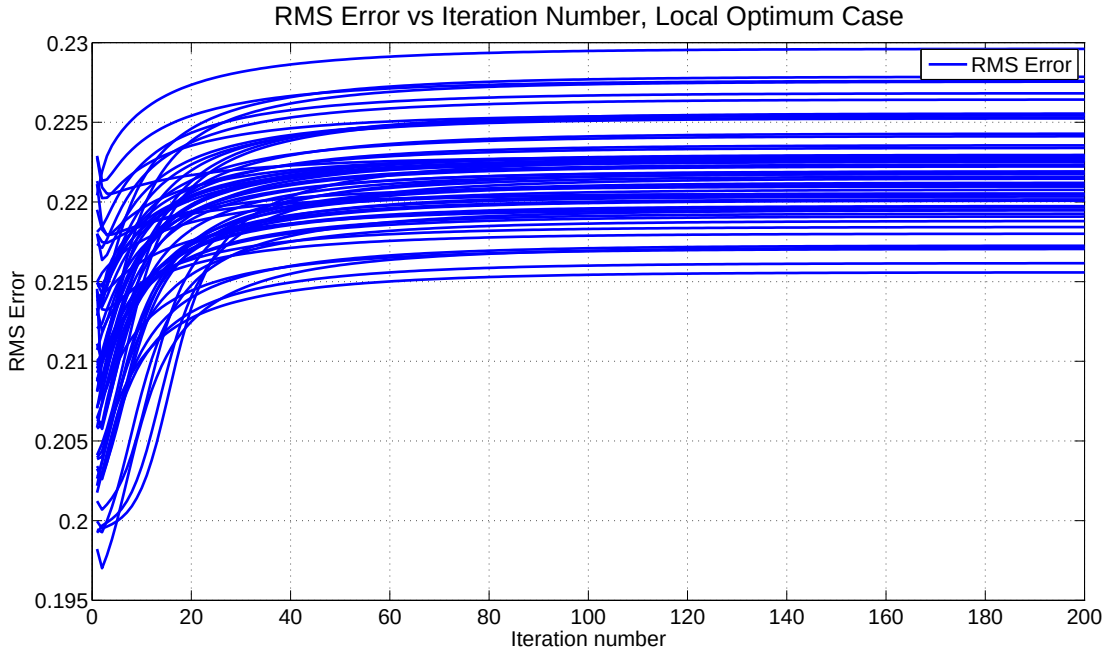

Figure 7.36 The simulations which converge to local optima. For every iteration, the RMS error of the prediction is shown. At every step, the RMS errors are increasing. After around 200 iterations, the algorithm converges.

7.2.4 Discussion

The IEKS method converged to the parameter set θ_g. When we check to which input the selected region is most sensitive, then we observe that ϵ_2 is, as expected, significantly bigger than ϵ_1 and ϵ_3. This result is in close agreement with Friston et al. [13]. They used DEM for model inversion. As here, they have relatively high ϵ_2 estimated region. We also visualize the results in Figure 7.37. Almost nothing for ϵ_3 for "Attention" case. We have the similar moderate result for ϵ_2 for "Vision" condition.

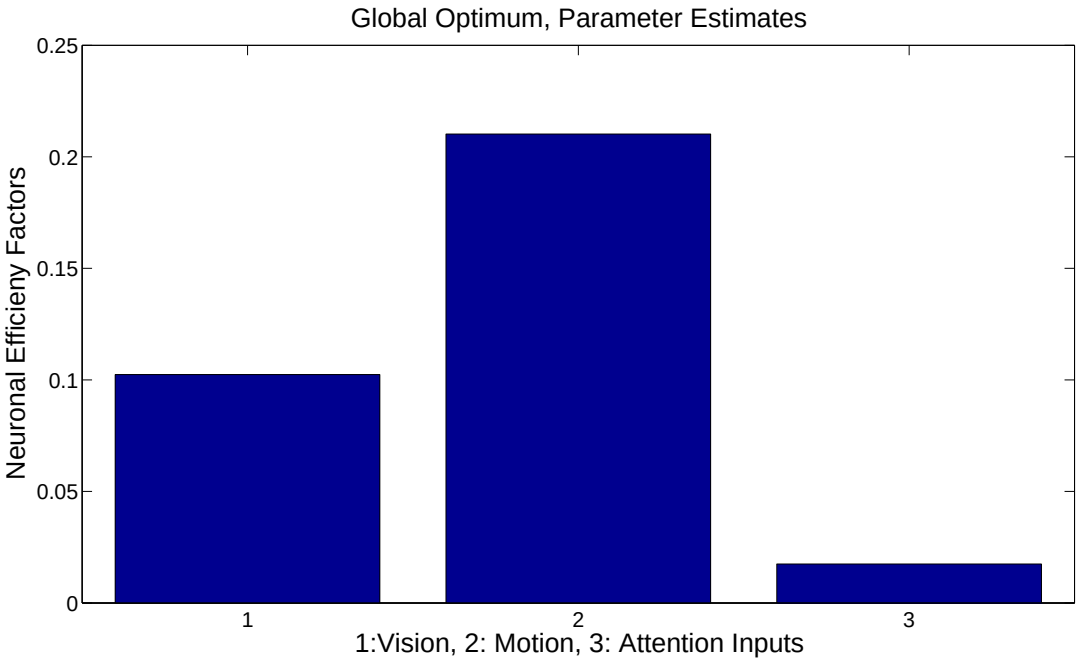

Figure 7.37 The estimated neuronal efficiency factors $\epsilon_1, \epsilon_2, \epsilon_3$ are shown. These factors are associated with neuronal inputs "Vision" u_1, "Motion" u_2 and "Attention" u_3. ϵ_2 gives information about how much the selected brain region is responsive to the motion input. The selected region was the motion sensitive area V5. The results are very similar to the work of Friston et al. [13].

Interestingly, for the local converged sequence, we have similar estimates to Friston et al. [15] where they used GF algorithm. Their estimates for the neuronal efficiency coefficients $\epsilon_1, \epsilon_2, \epsilon_3$ are in close agreement with the ones for θ_l.

One note to our observations is we have multi-modal convergence. In Section 7.1.5.1, we plotted the BOLD response changes with respect to the parameter changes.

When one begins with an arbitrary set of θ value then the first predicted BOLD response is quite different from the real one. The amplitude of the response is changed by either changing ϵ or χ value. Assuming the first predicted BOLD response is low in amplitude, in order to decrease the error, the most reasonable change is either having higher ϵ or lower χ value. This is exactly seen in the difference of the θ_g and θ_l values. For the local optima θ_l, the low strength in the ϵ parameters are compensated by the low χ value. According to our observation, the methods of Friston et al. [13, 15] converged either to the local or global optima depending on their initial condition.

DEM, VF and GF of Friston et al. [13, 14, 15] rest on a prior pdf for the parameters. They impose some predefined pdf with some prior mean and variance statistics. For that reason, we think that the algorithm may be more dependent on the initial estimate. For testing this, we also initialized the θ value to the prior means of Friston et al. [15] with $\theta = [0, 0, 0, 1.2, 2.14, 0.31]$. With this initial parameter set θ, we ended as expected to the local optima which is in accordance with GF result.

Now, we turn our focus to the importance of the parameter variance to see more insight of the IEKS algorithm.

7.2.4.1 Parameter Variance and Convergence. When applying IEKS, we chose the parameter variance as $\mathcal{N}(0, \Delta t 10^{-8})$ for each parameter. We should note that we treat the parameters as artificially varying variables with small enough steps.

$$\theta_{k+1,1} = \theta_{k,1} + w_{k,1} \tag{7.8}$$

$$\theta_{k+1,2} = \theta_{k,2} + w_{k,2} \tag{7.9}$$

$$\cdots \tag{7.10}$$

$$\theta_{k+1,p} = \theta_{k,p} + w_{k,p} \tag{7.11}$$

Having small variance is important. When the algorithm converges, we prefer a rather fixed value throughout the time index. For the converged parameter sequence, the end result of the IEKS algorithm looks like in Figure 7.38.

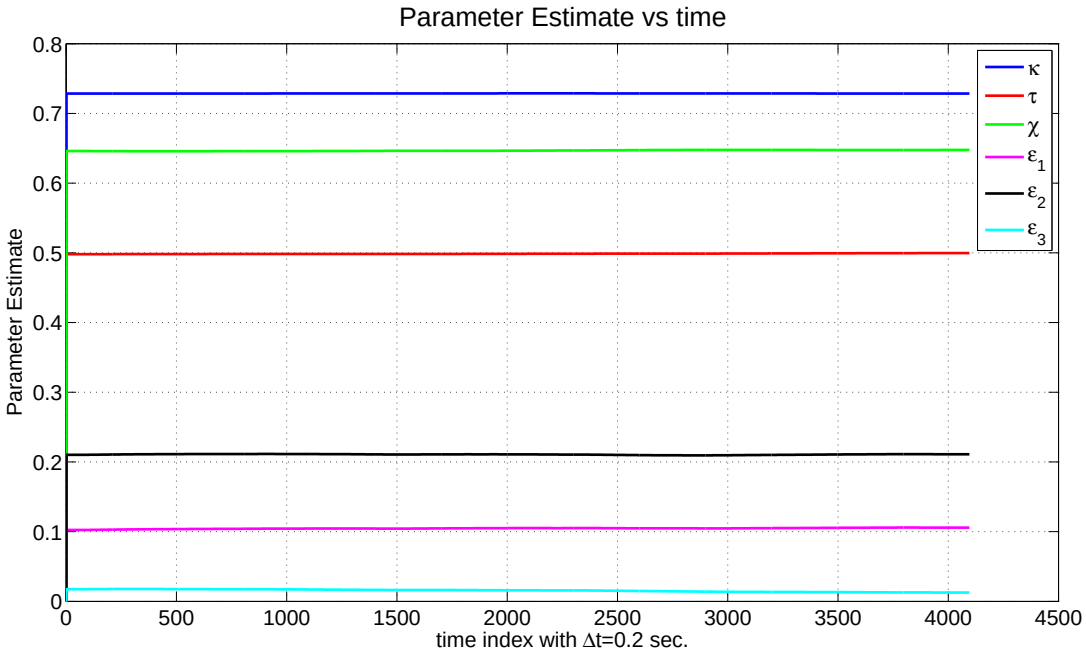

Figure 7.38 The importance of the parameter variance. Plots are taken from the converged global optimum θ_g. We chose the variance small enough ($\sigma_w^2 = \Delta t e^{-8}$) to ensure fixed value for parameter estimate throughout the time index. As can be seen from the graph the parameter is a rather fixed value as desired.

Let us see what happens if we increase the parameter variance. We performed again Monte Carlo simulation. We chose, in this case, perturbation as $\mathcal{N}(0, \Delta t 10^{-6})$. Now, there is an interesting fact here. We have only a single optimum. All the parameters converge to the same $\theta = [0.0055, 0.2452, 0.0459, 0.7643, 0.4735, 0.7320]$. Figures 7.39 and 7.40 show the convergence of the runs with different initial conditions. One thing we note is that the convergence speed is increased. In less than 15 iteration number, the algorithm converged.

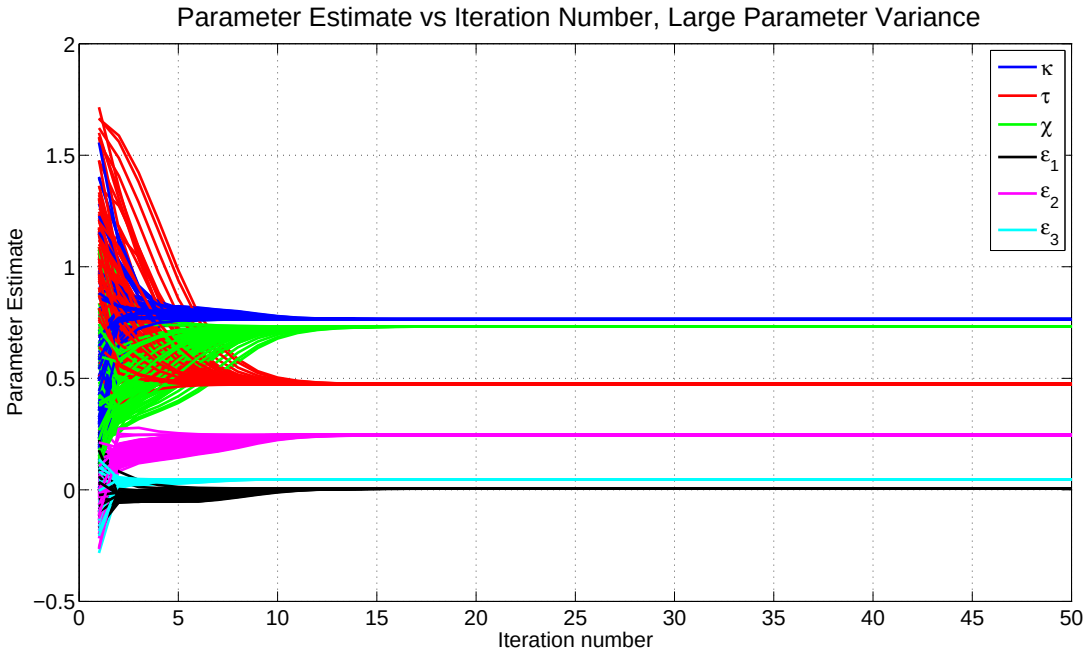

Figure 7.39 The simulations which converge to global optima θ_g. This time, the parameter variance is increased to $\sigma_w^2 = \Delta t e^{-6}$. Specifically, we estimate the parameter set $\theta = [\epsilon_1, \epsilon_2, \epsilon_3, \kappa, \tau, \chi]$

.

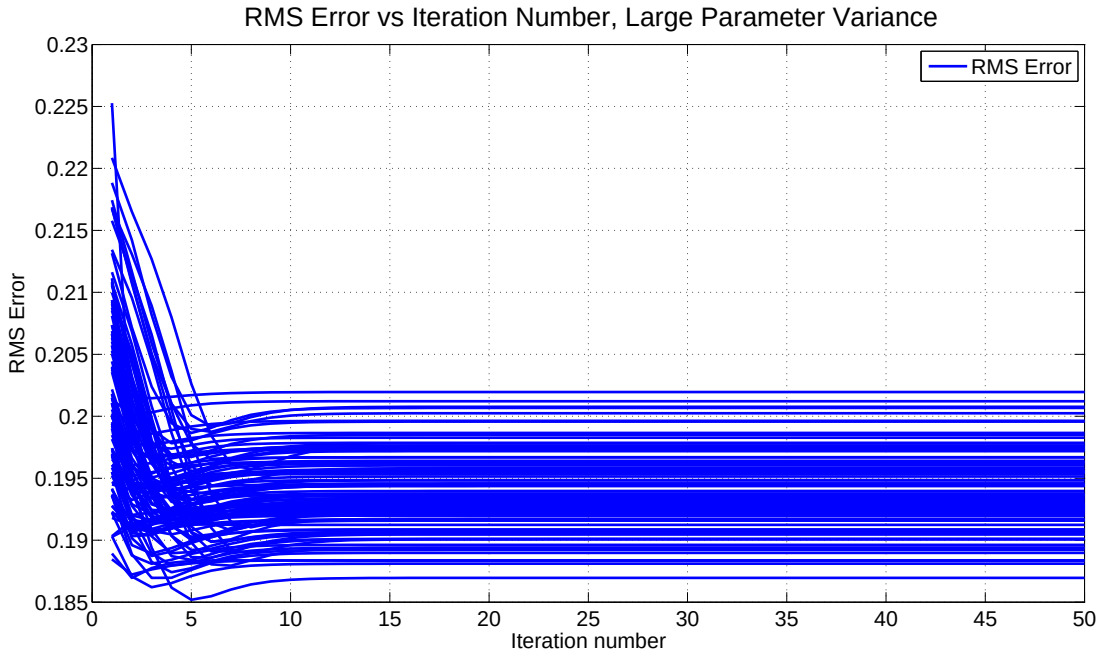

Figure 7.40 The simulations which converge to global optima. This time, the parameter variance is increased to $\sigma_w^2 = \Delta t e^{-6}$. For every iteration, the RMS error of the prediction is shown. At every step, the RMS errors are decreasing. After around 25 iterations, the algorithm converges.

We have the following important observation. Having increased parameter variance seems to force the IEKS method to search in a wider parameter region, which eased IEKS to escape from the local optima. However, this is at the cost of loosing the precision. To see this fact, we also plot the parameter estimates with respect to the time index. Figure 7.41 shows the fluctuation of the parameter estimate around the global optima.

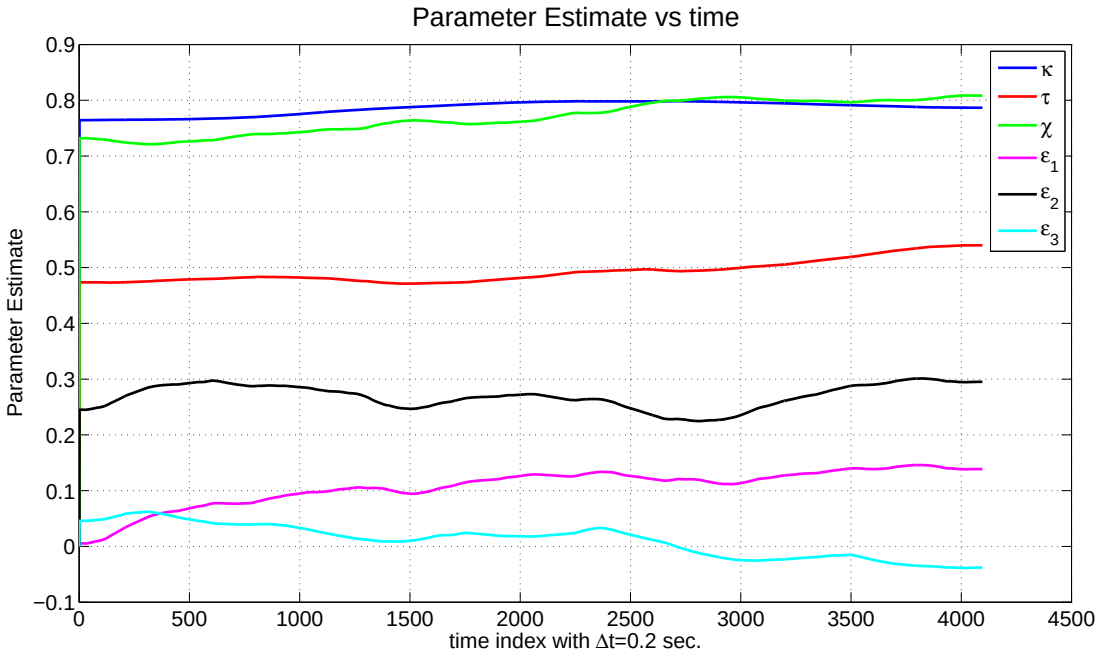

Figure 7.41 Importance of the parameter variance. This time, parameter variance increased to the value $\Delta t 10^{-6}$. We run the IEKS algorithm until convergence. For the converged θ, we plot the parameters with respect to the time index. There is only one optima, but we have lost the fixedness in time. The parameters fluctuate around the global optima point.

Although we solved the parameter identification problem in the former section completely, in the next subsection we will perform an interesting refinement to the algorithm.

7.2.4.2 IEKS Switched Parameter Variance. In nonlinear systems, it is quite often that there exists multi-modal optima. In our case, for low parameter noise, IEKS converged also to local optima. By performing multiple runs we could identify the global optimum. We verified that the converged one has also fixed value through the

time index. One other observation was having high parameter noise enables for the IEKS method to search in a wider parameter regime. As a result, IEKS could escape the local optima. However, when IEKS converges then the parameter is not fixed as desired. It fluctuates around the global optima. At this point, we suggest a very simple modification to the algorithm. First, begin with the high parameter noise, subsequently switch to the low parameter noise. We performed this switch at the iteration number 10. As a result, we have a very nice result which is shown in Figure 7.42

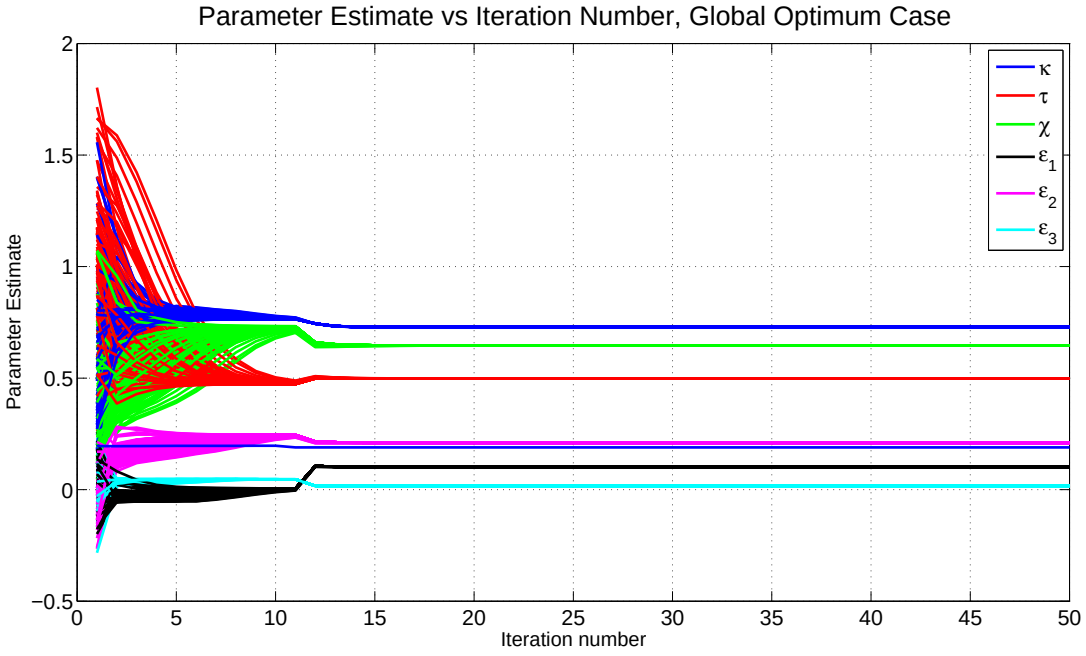

Figure 7.42 Convergence of the parameters when applying the switched parameter variance. First, IEKS begins with the high parameter noise ($\sigma_w^2 = \Delta t e^{-6}$). After the iteration number 10, we have the parameter noise variance as ($\sigma_w^2 = \Delta t e^{-8}$). This way, IEKS converged to the same global optima as in the low parameter variance case. There is only a single optimum. Furthermore, IEKS converged faster.

This way, our algorithm converged to the same global optima as in the low parameter noise condition for all initial conditions. There is no more local optima. At the switch time, we see the further improvement in the estimation. As a result, in less than 15 iteration numbers, IEKS method converged to the true global optima irrespective of the initial condition.

7.2.4.3 Estimated Hemodynamic Variables. At this point, we will check the estimated hemodynamic variables at the global optima of the estimated parameter set. Following Friston et al. and Havlicek et al. [6, 13, 14, 15], we plot the signal h_1 and the log-transformed hemodynamic variables blood flow h_2, blood venus volume h_3 and blood deoxyhemoglobin content h_4 . The result is shown in Figure 7.43. During stimulus instances, an increased blood flow is observed. Subsequently, blood venus volume is increased. At first, initial dip of deoxyhemoglobin is also observed. Those results are in agreement with the theory and results of Friston et al. [13, 14, 15].

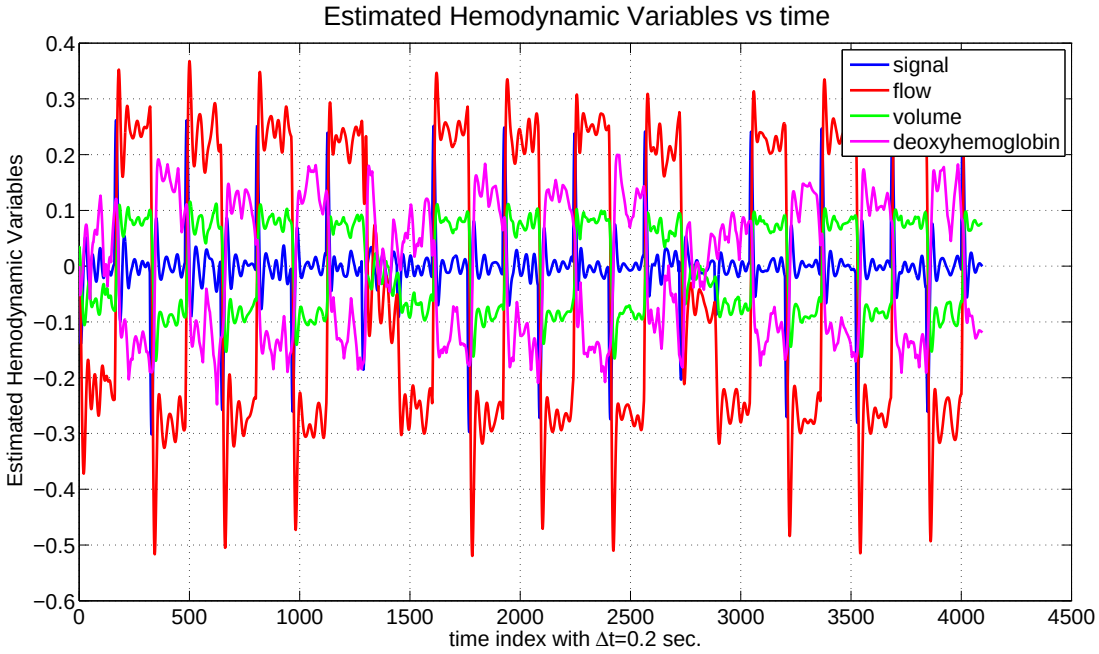

Figure 7.43 The estimated hemodynamic variables which converged at the global optima θ_g. The graph is in accordance with the theory and the previous results in the fMRI literature. During stimulus instances, an increased blood flow is observed. Subsequent the change in the increased blood flow, blood venus volume is increased and the initial dip of deoxyhemoglobin content is observed.

By using the predicted hemodynamic variables, we calculated the predicted response and plotted the prediction error in Figure 7.44.

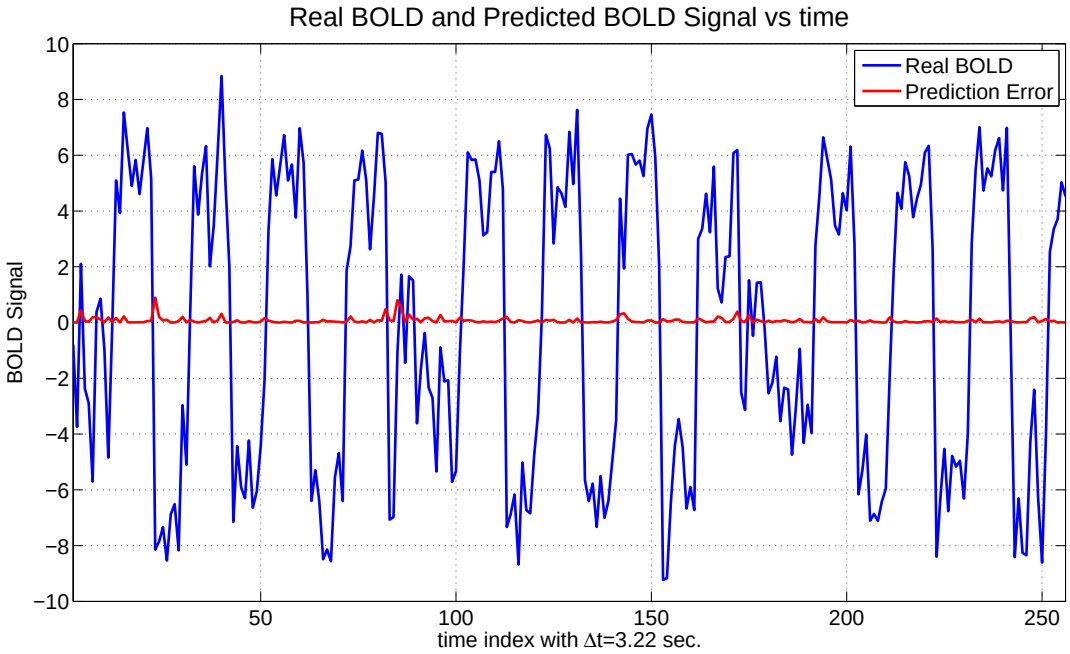

Figure 7.44 The prediction error in the BOLD signal. Here, sampling interval is $\Delta t = 3.22$ sec. The real BOLD signal is plotted by the blue color, whereas the prediction error is plotted by the red color.

8. CONCLUSION

In this thesis, we worked on the nonlinear state estimation of the hemodynamic model using the Friston-Buxton model. We worked for both known and unknown parameter case. For the known parameter case, we showed that hemodynamic models are suitable for Extended Kalman filter algorithms. We saw that Gaussian approximated inference methods were better than particle filter type algorithms for the hemodynamic state estimation. Standard particle filters use non-optimal proposal function which degrades the accuracy of the method. We also implemented particle filter with EKF approximated optimal proposal function (PF-EKF). By using the EKF approximation of the optimal importance function, improved sample diversity is achieved for most process noise conditions, however the improvement is not huge. Particle filter type algorithms like APF, PF, PF-EKF were not better than Gaussian approximated inference methods. This was the contrary to the former belief that EKF is poor in performance. We checked this assumption under a wide range of noise conditions, different inputs, nearly all parameter sets used in the literature and unknown initial condition. We also checked the histograms of the posteriors of the particle filters and concluded that the hemodynamic model is weakly nonlinear.

The second part of the thesis was the unknown parameter case. In this thesis, the iterative extended Kalman smoother method was implemented for the fMRI model inversion for both hemodynamic state and parameter estimation. Current extended Kalman-type fMRI model inversion algorithms work only in the filtering sense. By utilizing smoothers, we noticed improvement for the hemodynamic state estimation in a wide range of noise conditions. The joint state and parameter estimation of the hemodynamic model was performed by treating the parameters as time-varying variables. The joint estimation of the parameters and states was robust by using the iterative EKS algorithm. The IEKS algorithm was compared with EKS and the iterative square-root cubature Kalman smoother algorithm for different process and measurement noise conditions. In all conditions, IEKS outperformed EKS. Especially

in the higher noise conditions, the improvement was more profound. IEKS was also more accurate than the iterative SCKS method in terms of lower parameter bias and lower state RMS error. EKS was also more than twice faster than SCKS. Since SCKS is known to be the most accurate model inversion technique, the IEKS algorithm is a powerful and robust alternative to the former fMRI model inversion techniques.

We also tested IEKS for a rather complicated multi-input hemodynamic model. The dataset we work was used in a variety of works in the fMRI literature [13, 14, 15]. From the real BOLD signal, IEKS successfully estimated hemodynamic variables and parameters. Those results were in agreement with the literature. In nonlinear estimation problems, there exits quite often multi-modal optimum values. We observed that former model inversion algorithms stuck in local optima in some cases. We analyzed the IEKS method in depth by studying different parameter variances. By a small refinement of the algorithm we were able to escape from the local optima either. The refinement was using different parameter variances in different iteration numbers. First, by using high parameter variance IEKS method traces in a rather high parameter regime, in the successive steps the variance is decreased to converge to global optima and to have rather fixed parameter value throughout the time index. As a result, IEKS was both successful in theory and application.

8.1 Future Work

The first contribution of the thesis was PF-EKF, which was an improved filtering method compared to the techniques used in the fMRI literature. We plan to work on further improving the hemodynamic state estimation by utilizing smoothing techniques. Smoothing of the hemodynamic variables is not much evaluated in the fMRI literature compared to the filtering methods. We want to make a thorough analysis of the possible hemodynamic smoothing techniques. We want to suggest a new implementation like we did in the filtering problem. For the second contribution, having ground-truth validated the IEKS method and its application to the real data, we want also work on the functional near infrared spectroscopy (fNIRS) model inversion. fNIRS and

fMRI have the same hemodynamic state transition function but differ in measurement function. For that reason we hope that our method is also applicable for fNIRS systems. We want also look for the application opportunities of IEKS for Brain Machine Interface (BMI) and clinical applications.

APPENDIX A.

A.1 Proof of Kalman Filter

Here, a short sketch of the proof of the Kalman filter is given. We will use two properties [41].

Property 1:

For a jointly Gaussian pdf given by:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \mathcal{N}(\begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix}, \begin{bmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{bmatrix}) \tag{A.1}$$

Here μ_X and μ_Y are the means of the individual components of the 2-dimensional vector $\begin{bmatrix} X \\ Y \end{bmatrix}$. Σ_X, Σ_{XY}, Σ_{YX} and Σ_Y are the individual components of the covariance matrix of the 2-dimensional vector $\begin{bmatrix} X \\ Y \end{bmatrix}$. The conditional statistics of $p(X|Y)$ is given as follows:

$$p(X|Y) = \mathcal{N}(\mu_{X|Y}, \Sigma_{X|Y}) \tag{A.2}$$

where $\mu_{X|Y}$ and $\Sigma_{X|Y}$ is given by:

$$\mu_{X|Y} = \mu_X + \Sigma_{XY}\Sigma_Y^{-1}(Y - \mu_Y) \tag{A.3}$$

$$\Sigma_{X|Y} = \Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX} \tag{A.4}$$

Property 2:

$$p(x_k, y_k|y_{1:k-1}) = \mathcal{N}(\begin{bmatrix} \mu_{k|k-1} \\ C\mu_{k|k-1} \end{bmatrix}, \begin{bmatrix} \Sigma_{k|k-1} & \Sigma_{k|k-1}C^T \\ C\Sigma_{k|k-1} & C\Sigma_{k|k-1}C^T + R \end{bmatrix}) \tag{A.5}$$

When we apply property 1 for the pdf given in the property 2 we end up with the conditional Gaussian

$$p(x_k|y_k, y_{1:k-1}) = p(x_k|y_{1:k})\mathcal{N}(x_k; \mu_{k|k}; \Sigma_{k|k}) \tag{A.6}$$

where with the mean and covariance updates:

$$\mu_{k|k} = \mu_{k|k-1} + \Sigma_{k|k-1}C^T(C\Sigma_{k|k-1}C^T + R)^{-1}(y_k - C\mu_{k|k-1}) \tag{A.7}$$

$$\Sigma_{k|k} = \Sigma_{k|k-1} - \Sigma_{k|k-1}C^T(C\Sigma_{k|k-1}C^T + R)^{-1}C\Sigma_{k|k-1} \tag{A.8}$$

Note that Property 2 is straightforward. In the Kalman filter algorithm, the expression $\Sigma_{k|k-1}C^T(C\Sigma_{k|k-1}C^T + R)^{-1}$ is denoted as Kalman gain.

A.2 Proof of RTS Kalman Smoother

Following Cemgil [41], we again use two properties. Afterwards we will be able to obtain an expression of x_k in terms of x_{k+1}, from which we will easily derive the update equation of statistics of $p(x_k|y_{1:K})$ in terms of the statistics $p(x_{k+1}|y_{1:K})$. Again we remind the property for the joint Gaussian pdf's.

Property 1:

For jointly Gaussian pdf given by:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \mathcal{N}(\begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix}, \begin{bmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{bmatrix}) \tag{A.9}$$

$$p(x_k|x_{k+1}, y_{1:k}) = \mathcal{N}(x_k; \bar{\mu}; \bar{\Sigma}) \tag{A.10}$$

where with the mean and covariance updates:

$$\mu_{X|Y} = \mu_X + \Sigma_{XY}\Sigma_Y^{-1}(Y - \mu_Y) \tag{A.11}$$

$$\Sigma_{X|Y} = \Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX} \tag{A.12}$$

This time we have the property similarly:
Property 2:

$$p(x_k, x_{k+1}|y_{1:k}) = \mathcal{N}(\begin{bmatrix} \mu_{k|k1} \\ A\mu_{k|k} \end{bmatrix}, \begin{bmatrix} \Sigma_{k|k} & \Sigma_{k|k}A^T \\ A\Sigma_{k|k} & A\Sigma_{k|k}A^T + Q \end{bmatrix}) \tag{A.13}$$

It is very easy to prove the mean and individual covariance components of property 2.

Now again applying Property 1 for the joint pdf given in the Property 2. We end up

$$p(x_k|x_{k+1}, y_{1:k}) = \mathcal{N}(x_k, \tilde{\mu}, \tilde{\Sigma}) \tag{A.14}$$

$$\tilde{\mu} = \mu_{k|k} + \Sigma_{k|k}A^T(A\Sigma_{k|k}A^T + R)^{-1}(x_{k+1} - A\mu_{k|k}) \tag{A.15}$$

$$\tilde{\Sigma} = \Sigma_{k|k} - \Sigma_{k|k}A^T(A\Sigma_{k|k}A^T + Q)^{-1}A\Sigma_{k|k} \tag{A.16}$$

$$= \Sigma_{k|k} - \Sigma_{k|k}A^T(A\Sigma_{k|k}A^T + Q)^{-1}A\Sigma_{k|k} \tag{A.17}$$

$$= \Sigma_{k|k} - J_kA\Sigma_{k|k} \tag{A.18}$$

We worked on the conditional pdf $p(x_k|x_{k+1}, y_{1:k})$ which is a Gaussian density. For a Gaussian random variable, the mean and covariance are enough to describe its pdf. From these expressions, our aim is to obtain a relationship between x_k and x_{k+1}. This expression is:

$$x_k = (I - J_kA)\mu_{k|k} + J_kx_{k+1} + e_k \tag{A.19}$$

where

$$e_k = \mathcal{N}(0, \tilde{\Sigma}) \tag{A.20}$$

$$J_k = \Sigma_{k|k} A^T (A\Sigma_{k|k} A^T + Q)^{-1} \tag{A.21}$$

Note that it is routine to show that this expression contains the statistics information which we obtained in Eq. A.14. After obtaining this intermediate step which exhibits the relation between the random variables x_k and x_{k+1} we can derive the smoothed statistics of these random variables. We project these random variables to the space generated by the random variables $y_1, y_2, ..., y_K$. Otherwise stated, we take the conditional expectation of the above expression in both sides of equality. We arrive first the mean statistic.

$$\mu_{k|K} = \mu_{k|k} + J_k(\mu_{k+1|K} - \mu_{k+1|k}) \tag{A.22}$$

In order to calculate $\Sigma_{k|K}$

$$\Sigma_{k|K} = cov(x_k - \mu_{k|K}) \tag{A.23}$$

$$= cov(J_k(x_{k+1} - \mu_{k+1|K}) + e_k) \tag{A.24}$$

$$= J_k \Sigma_{k+1|K} J_k^T + \Sigma_{k|k} - J_k A \Sigma_{k|k} \tag{A.25}$$

$$= \Sigma_{k|k} + J_k \Sigma_{k+1|K} J_k^T - J_k \Sigma_{k+1|k} \Sigma_{k+1|k}^{-1} A \Sigma_{k|k} \tag{A.26}$$

$$= \Sigma_{k|k} + J_k \Sigma_{k+1|K} J_k^T - J_k \Sigma_{k+1|k} J_k^T \tag{A.27}$$

$$= \Sigma_{k|k} + J_k (\Sigma_{k+1|K} - \Sigma_{k+1|k}) J_k^T \tag{A.28}$$

which concluded the proof RTS Kalman Smoother.

A.3 QR Decomposition

Let X be any matrix. The QR decomposition of the matrix X^T results in an orthogonal matrix Q and upper triangular matrix R such that the following property

is true [6].

$$X^T = QR \tag{A.29}$$

For the QR decomposed matrices we have the following remarkable property.

$$XX^T = R^T R \tag{A.30}$$

REFERENCES

1. Friston, K. J., A. Mechelli, R. Turner, and C. Price, "Nonlinear responses in fMRI: the Balloon model, Volterra kernels, and other hemodynamics," *Neuroimage*, Vol. 12, no. 4, pp. 466–477, 2000.

2. Buxton, R. B., K. Uludağ, D. J. Dubowitz, and T. T. Liu, "Modeling the hemodynamic response to brain activation," *Neuroimage*, Vol. 23, pp. S220–S233, 2004.

3. Buxton, R. B., E. C. Wong, and L. R. Frank, "Dynamics of blood flow and oxygenation changes during brain activation: the balloon model," *Magnetic Resonance in Medicine*, Vol. 39, no. 6, pp. 855–864, 1998.

4. Aguirre, G., E. Zarahn, and M. D'esposito, "The variability of human, BOLD hemodynamic responses," *Neuroimage*, Vol. 8, no. 4, pp. 360–369, 1998.

5. Johnston, L. A., E. Duff, I. Mareels, and G. F. Egan, "Nonlinear estimation of the BOLD signal," *Neuroimage*, Vol. 40, no. 2, pp. 504–514, 2008.

6. Havlicek, M., K. J. Friston, J. Jan, M. Brazdil, and V. D. Calhoun, "Dynamic modeling of neuronal responses in fMRI using cubature Kalman filtering," *Neuroimage*, Vol. 56, no. 4, pp. 2109–2128, 2011.

7. Handwerker, D. A., J. M. Ollinger, and M. D'Esposito, "Variation of BOLD hemodynamic responses across subjects and brain regions and their effects on statistical analyses," *Neuroimage*, Vol. 21, no. 4, pp. 1639–1651, 2004.

8. Friston, K. J., "Bayesian estimation of dynamical systems: an application to fMRI," *Neuroimage*, Vol. 16, no. 2, pp. 513–530, 2002.

9. Riera, J. J., J. Watanabe, I. Kazuki, M. Naoki, E. Aubert, T. Ozaki, and R. Kawashima, "A state-space model of the hemodynamic approach: nonlinear filtering of BOLD signals," *Neuroimage*, Vol. 21, no. 2, pp. 547–567, 2004.

10. Jimenez, J., I. Shoji, and T. Ozaki, "Simulation of stochastic differential equations through the local linearization method. a comparative study," *Journal of Statistical Physics*, Vol. 94, no. 3-4, pp. 587–602, 1999.

11. Hu, Z., X. Zhao, H. Liu, and P. Shi, "Nonlinear analysis of the BOLD signal," *EURASIP Journal on Advances in Signal Processing*, Vol. 2009, p. 1, 2009.

12. Murray, L., and A. Storkey, "Particle smoothing in continuous time: A fast approach via density estimation," *Signal Processing, IEEE Transactions on*, Vol. 59, no. 3, pp. 1017–1026, 2011.

13. Friston, K., J. Mattout, N. Trujillo-Barreto, J. Ashburner, and W. Penny, "Variational free energy and the Laplace approximation," *Neuroimage*, Vol. 34, no. 1, pp. 220–234, 2007.

14. Friston, K. J., N. Trujillo-Barreto, and J. Daunizeau, "DEM: a variational treatment of dynamic systems," *Neuroimage*, Vol. 41, no. 3, pp. 849–885, 2008.

15. Friston, K., K. Stephan, B. Li, and J. Daunizeau, "Generalised filtering," *Mathematical Problems in Engineering*, Vol. 2010, 2010.

16. Bishop, C. M., *et al.*, *Pattern recognition and machine learning*, Vol. 1, springer New York, 2006.

17. Kalman, R. E., "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, Vol. 82, no. 1, pp. 35–45, 1960.

18. Rauch, H. E., C. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, Vol. 3, no. 8, pp. 1445–1450, 1965.

19. Jazwinski, A., "Filtering for nonlinear dynamical systems," *Automatic Control, IEEE Transactions on*, Vol. 11, no. 4, pp. 765–766, 1966.

20. Cox, H., "On the estimation of state variables and parameters for noisy dynamic systems," *Automatic Control, IEEE Transactions on*, Vol. 9, no. 1, pp. 5–12, 1964.

21. Julier, S. J., J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *American Control Conference, Proceedings of the 1995*, Vol. 3, pp. 1628–1632, IEEE, 1995.

22. Julier, S. J., and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, Vol. 92, no. 3, pp. 401–422, 2004.

23. Sarkka, S., "Unscented Rauch–Tung–Striebel Smoother," *Automatic Control, IEEE Transactions on*, Vol. 53, no. 3, pp. 845–849, 2008.

24. Arasaratnam, I., *Cubature Kalman filtering: theory & applications*. PhD thesis, PhD thesis, ECE Department, McMaster University, 2009.

25. Hartikainen, J., A. Solin, and S. Särkkä, "Optimal filtering with Kalman filters and smoothers," *Department of Biomedica Engineering and Computational Sciences, Aalto University School of Science, 16th August*, 2011.

26. Ristic, B., S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*, Vol. 685, Artech house Boston, 2004.

27. Gordon, N. J., D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proceedings F (Radar and Signal Processing)*, Vol. 140, pp. 107–113, IET, 1993.

28. Doucet, A., and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of Nonlinear Filtering*, Vol. 12, pp. 656–704, 2009.

29. Arulampalam, M. S., S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on*, Vol. 50, no. 2, pp. 174–188, 2002.

30. Feynman, R., "Statistical Mechanics, a set of lectures, california, institute of technology," 1972.

31. Parisi, G., *Statistical field theory*, Addison-Wesley, 1988.

32. Jaakkola, T. S., "Tutorial on Variational Approximation Methods," *Advanced mean field methods: theory and practice*, p. 129, 2001.

33. Borman, S., "The expectation maximization algorithm-a short tutorial," 2009.

34. Friston, K. J., N. Trujillo-Barreto, and J. Daunizeau, "DEM: a variational treatment of dynamic systems," *Neuroimage*, Vol. 41, no. 3, pp. 849–885, 2008.

35. Mandeville, J. B., J. J. Marota, C. Ayata, G. Zaharchuk, M. A. Moskowitz, B. R. Rosen, and R. M. Weisskoff, "Evidence of a cerebrovascular postarteriole windkessel with delayed compliance," *Journal of Cerebral Blood Flow & Metabolism*, Vol. 19, no. 6, pp. 679–689, 1999.

36. Stephan, K. E., N. Weiskopf, P. M. Drysdale, P. A. Robinson, and K. J. Friston, "Comparing hemodynamic models with DCM," *Neuroimage*, Vol. 38, no. 3, pp. 387–401, 2007.

37. Obata, T., T. T. Liu, K. L. Miller, W.-M. Luh, E. C. Wong, L. R. Frank, and R. B. Buxton, "Discrepancies between BOLD and flow dynamics in primary and supplementary motor areas: application of the balloon model to the interpretation of bold transients," *Neuroimage*, Vol. 21, no. 1, pp. 144–153, 2004.

38. Gustafsson, F., "Particle filter theory and practice with positioning applications," *Aerospace and Electronic Systems Magazine, IEEE*, Vol. 25, no. 7, pp. 53–82, 2010.

39. Ozaki, T., "A bridge between nonlinear time series models and nonlinear stochastic dynamical systems: a local linearization approach," *Statistica Sinica*, Vol. 2, no. 1, pp. 113–135, 1992.

40. Boynton, G. M., S. A. Engel, G. H. Glover, and D. J. Heeger, "Linear systems analysis of functional magnetic resonance imaging in human v1," *The Journal of Neuroscience*, Vol. 16, no. 13, pp. 4207–4221, 1996.

41. Cemgil, A. T., "Kalman proof." `https://dl.dropboxusercontent.com/u/9787379/cmpe58k/bsml_lecture09.pdf`, 2015. [Online; accessed 19-July-2015].