# A DATA-DRIVEN APPROACH TO POWER SYSTEM DYNAMIC STATE ESTIMATION

A Thesis

by

DEEPIKA KUMARI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

| | |
|---|---|
| Chair of Committee, | Shankar P. Bhattacharyya |
| Co-Chair of Committee, | Raktim Bhattacharya |
| Committee Members, | Krishna Narayanan |
| | Ulisses Braga Neto |
| Head of Department, | Miroslav M. Begovic |

August 2017

Major Subject: Electrical Engineering

# ABSTRACT

State estimation is a key function in the supervisory control and planning of an electric power grid. Typically, the independent system operator (ISO) runs least-squares based static state estimation once every few minutes. Inherently, however, a power system is mostly in a transient state owing to load fluctuations, outages and network switching. In such a scenario, dynamic state estimation facilitates real-time monitoring and control of the system. Dynamic state estimation is implemented using Kalman filtering techniques. Popular estimators for nonlinear systems include the extended Kalman filter (EKF) and unscented Kalman filter (UKF). Practical implementation, however, is inhibited by the lack of an accurate system model and the high computational complexity of Kalman filtering methods.

I address the former issue of model unavailability and rely instead on measurement data from phasor measurement units for dynamic state estimation (DSE). I build an estimator for DSE which uses only measurement and input information, and operates without knowledge of the underlying system model. The algorithm considered uses a Gaussian process (GP) approximation of the state transition and observation functions in the implementation of a UKF-based state estimation.

I analyze the performance of the estimator for different scenarios using root mean squared (RMS) error as the metric. The estimator, when evaluated on the IEEE 14-bus test case, gives a minimum accuracy rate of over 94% over all considered scenarios.

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Shankar P. Bhattacharyya, whose support and insight greatly accelerated the completion of this work. It has been a godsend relief to be able to count on his patience and guidance throughout the course of my study.

I am grateful to Professors Narayanan, Braga-Neto and Bhattacharya for agreeing to be on my thesis committee and for their wonderful feedback.

I would also like to thank my parents and my sister, Meenakshi for supporting me emotionally and financially in my endeavour.

I am grateful to Prof. Le Xie, who gave me the inspiration and material to do this research. His course taught me the concepts I needed to push forward with this work.

Many thanks to the Texas A&M University staff, professors and particularly graduate student advisors for a wonderful, and comfortable learning experience at A&M.

Thank you,

Deepika Kumari

# CONTRIBUTORS AND FUNDING SOURCES

## Contributors

This work was supported by a thesis committee consisting of Professors Bhattacharyya, Narayanan and Braga-Neto of the Department of Electrical and Computer Engineering and Professor Bhattacharya of the Department of Aerospace Engineering.

The PSAT toolbox used to simulate the IEEE 14 bus system was provided by Dr. Federico Milano of University College Dublin School of Electrical and Electronic Engineering. The GPML toolbox used to fit the Gaussian Process to data was developed by Professor Carl Edward Rasmussen of University of Cambridge and Dr. Hannes Nickisch from Philips Research, Hamburg, Germany.

All other work conducted for the thesis was completed by the student independently.

## Funding Sources

There are no outside funding contributions to acknowledge related to the research and compilation of this document.

# NOMENCLATURE

| | |
|---|---|
| ISO | Independent System Operator |
| SCADA | Supervisory Control and Data Acquisition |
| DSE | Dynamic State Estimation |
| UKF | Unscented Kalman Filter |
| PMU | Phasor Measurement Unit |
| GP | Gaussian Process |
| RMS | Root Mean Squared |
| PSAT | Power System Analysis Toolbox |
| GPML | Gaussian Processes for Machine Learning |
| SE | State Estimation |
| EKF | Extended Kalman Filter |
| AVR | Automatic Voltage Regulator (Exciter) |

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

# 1. INTRODUCTION

State Estimation is a key functionality in power system operation as it provides the critical information for other fundamental processes like contingency analysis, load forecasting, optimal power flow and economic dispatch. Traditional state estimators compute the weighted least squares (WLS) estimate [3] of the network state using the data provided by the SCADA system. This static estimate is based on a steady-state model of the system, and does not consider the dynamic behaviour of the system. This scheme has been effective in estimating the state of transmission network in steady-state. Previously, it was the only state estimation deployed since SCADA measurements were too infrequent to consider dynamic state estimation (DSE).

With the rapidly rising deployment of phasor measurement units (PMUs), measurement acquisition can occur at rates of 10 to 60 samples per second. A power network equipped with PMU technology allows the estimator to capture transient behavior of the system. DSE has an edge over static SE primarily due to the predictive capability of Kalman filtering. The step-ahead prediction possible with DSE could open up new avenues in preventive control and, improved bad data detection and identification ([4], [5]). The main obstacles to the practical implementation of DSE are the difficulty in modeling large power networks with nonlinear dynamics and the high computational complexity of Kalman filtering methods. I address the former issue of model unavailibilty. I build an estimator for DSE which uses only measurement and input information, and operates without knowledge of the underlying system model.

## 1.1 Background

Consider a power transmission network with $n$ buses. Static SE aims to estimate the network state which is given by:

$$X_{net} := [\theta_1 \; V_1 \; \theta_2 \; V_2 \; \cdots \; \theta_n \; V_n]^T \tag{1.1}$$

where $\theta_i$ and $V_i$ are respectively the voltage phasor magnitude and phase angle at bus $i$, for $i = 1, 2, ....n$.

The dynamic state of the system is a different set of variables comprising rotor angles, speeds, transient and subtransient voltages. The vector formed by stacking the state variables of all the generators connected to the network, is the dynamic state vector of the system. Suppose the network has $k$ generators, with the $i^{th}$ generator having a state $X^i_{dyn}$. Then, the dynamic state of the network is as follows.

$$X_{dyn} := [X^1_{dyn} \; X^2_{dyn} \; \cdots \; X^k_{dyn}]^T \tag{1.2}$$

## 1.2 Outline

The relevant variables of my model are as follows.

$x_k \rightarrow$ state at time k

$u_k \rightarrow$ control input at time k

$z_k \rightarrow$ observation at time k

The general form of the process function is the following.

$$x_k = g(u_{k-1}, x_{k-1}) + \epsilon_k \tag{1.3}$$

$$\epsilon_k \sim \mathcal{N}(0, Q_k) \tag{1.4}$$

The observation function is represented in a similar fashion.

$$z_k = h(x_k) + \delta_k \tag{1.5}$$

$$\delta_k \sim \mathcal{N}(0, R_k) \tag{1.6}$$

Here, $\epsilon_k$ and $\delta_k$ represent the process noise and measurement noise respectively. The objective is to find an estimate $\hat{x}_k$, given $u_{k-1}$, $\hat{x}_{k-1}$ and $z_k$, in the absence of knowledge of mapping functions, $g$, and $h$.

Because $g$ and $h$ are nonlinear, even if $x_{k-1}$ is Gaussian, $x_k$ is not necessarily Gaussian. To proceed, $g$ and $h$ must be linearized. The extended Kalman filter (EKF) performs linearization using Taylor series expansion around the most recent estimate. UKF applies a more accurate, stochastic approximation using unscented transform. Unlike EKF, UKF does not require complex Jacobian computations.

The unknown functions $g$ and $h$ are approximated using GPs. GP is an easily adaptable non-parametric Bayesian modeling tool for supervised learning problems. Often, non-Gaussian processes can be approximated by a mixture of GPs. The use of a Gaussian process (GP) to model nonlinear dynamics is inspired by the fact that averaging over nonlinear regression models tends to a GP model [6].

The hyperparameters of the GP are learned from past measurements and corresponding state estimates. The learned GP, in conjunction with the unscented Kalman filter, facilitates sequential state estimation. The resulting GP-UKF algorithm was first introduced by Ko, Klein, Fox & Hahnel in 2007 [7]. The trained estimator is tested by repeatedly simulating cases from different scenarios - Network switching, parametric uncertainty, load perturbations and noisy measurements on IEEE 14-bus test case.

# 2. POWER SYSTEM DYNAMICS

## 2.1 Transmission Line Dynamics

The power transmission grid is predominantly a network of three-phase transmission lines carrying alternating current. A balanced phase transmission line is often represented using an equivalent pi circuit. The equivalent circuit for such a transmission line between bus $k$ and bus $m$ is shown in Fig. 2.1. Line resistance is the main cause of power loss in the transmission line and is modeled by resistor $R$. The line inductance $jX$ depends on the arrangement of conductors, spacing between them and their material. The shunt capacitance $B/2$ models the leakage of ac currents. It is the result of potential differences between the conductors as well as that between the conductors and ground. The capacitance element can be neglected for a short transmission line, one whose length is less than 80 km.

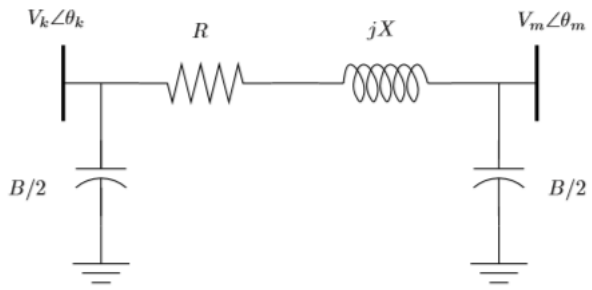



Figure 2.1: The $\pi$ model for transmission line [1]

The real and reactive powers entering bus $k$ are given by the equations below.

$$P_k = V_k^2(g_{km} + g_{k0}) - V_k V_m(g_{km}cos(\theta_k - \theta_m) + b_{km}sin(\theta_k - \theta_m)) \tag{2.1}$$

$$Q_k = -V_k^2(b_{km} + b_{k0}) - V_k V_m(g_{km}sin(\theta_k - \theta_m) - b_{km}cos(\theta_k - \theta_m)) \tag{2.2}$$

Similarly, the real and reactive powers leaving bus $m$ are given by the following equations.

$$P_m = V_m^2(g_{km} + g_{m0}) - V_k V_m(g_{km}cos(\theta_k - \theta_m) - b_{km}sin(\theta_k - \theta_m)) \tag{2.3}$$

$$Q_m = -V_m^2(b_{km} + b_{m0}) + V_k V_m(g_{km}sin(\theta_k - \theta_m) + b_{km}cos(\theta_k - \theta_m)) \tag{2.4}$$

The apparent power flow is $S = \sqrt{P^2 + Q^2}$.

## 2.2 Line Breakers

Circuit breakers are automatically operated switches meant to protect devices from excess current flow. They divert current flow from the line in case of a fault in the line or overloading. The breaker can be placed anywhere along the transmission line. When it is opened, current flow in that line drops to zero. This causes steep transients in the state of the system.

## 2.3 Generators

### 2.3.1 Slack generator

The slack generators are responsible for balancing the power system in terms of power generation, load demand and losses. Additionally, the voltage at the corresponding buses is kept constant at $1\angle 0^o$ pu. Hence, the slack generator is modeled as a $V\theta$ bus, having a constant bus voltage magnitude and phase angle. The network must possess at least one slack generator, since it provides the reference voltage for the rest of the system. If there are multiple slack generators, only one of them is used as the reference. Only the single

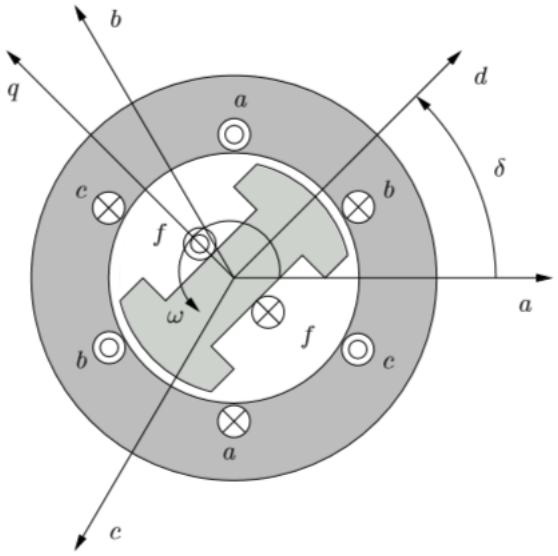



Figure 2.2: Park-Concordia model for synchronous machine [1]

slack bus model has been considered for simulation.

### 2.3.2 PV generator

A PV generator fixes the voltage magnitude and power injected at the bus where it is connected as follows.

$$P = P_g,\ V = V_o$$

### 2.3.3 Synchronous generator

The rotor of synchronous generator rotates in synchronization with the waveform of the generated voltage. It is the primary dynamic component of my simulated network. The generator dynamics are simulated using the Park-Concordia model for synchronous machines, shown in Fig.2.2. Two axes are defined with respect to the rotor. The direct axis, or $d$-axis is along the direction of magnetic flux generated by the field current. The quadrature axis, or $q$-axis leads the $d$-axis by $90^o$. The machine voltage expressed as

components along the $d$ and $q$ axes is related to the network phasor $V\angle\theta^o$ as follows.

$$v_d = V\sin(\delta - \theta) \tag{2.5}$$

$$v_q = V\cos(\delta - \theta) \tag{2.6}$$

A simple oscillation stabilizer is used to stabilize the field voltage. It uses a feedback of the rotor speed and active power produced by the machine as given in the following equation.

$$v_f^* = v_f + K_\omega(\omega - 1) - K_P(P(x, V, \theta) - P_o) \tag{2.7}$$

Here, $K_\omega$ is the speed feedback gain, $K_P$ is the active power feedback gain, and $P_o$ is the initial electric power generated by the machine.

In Table 2.1, we define the parameters of the machine.

$$\mathbf{x} = \begin{pmatrix} \delta \\ \omega \\ e'_q \\ e'_d \\ e''_q \\ e''_d \end{pmatrix},\ \mathbf{u} = \begin{pmatrix} P_m \\ P_e \\ v_f^* \end{pmatrix},\ \mathbf{z} = \begin{pmatrix} \theta \\ V \\ P \\ Q \end{pmatrix} \tag{2.8}$$

In Eq. (2.8), $\mathbf{x}$ represents the state of the generator dynamics, $\mathbf{u}$ is the control input and $\mathbf{z}$ comprises the observed variables. The variables involved in the these vectors are described briefly in Table 2.2. The sixth order dynamics of the Park-Concordia model for

| Symbol | Description |
|---|---|
| $x_l$ | Leakage Resistance |
| $r_a$ | Armature resistance |
| $x_d$ | d-axis synchronous reactance |
| $x'_d$ | d-axis transient reactance |
| $x''_d$ | d-axis subtransient reactance |
| $x'_q$ | q-axis transient reactance |
| $x''_q$ | q-axis subtransient reactance |
| $T'_{d0}$ | d-axis open-circuit transient time constant |
| $T''_{do}$ | d-axis open-circuit subtransient time constant |
| $T'_{q0}$ | q-axis open-circuit transient time constant |
| $T''_{qo}$ | q-axis open-circuit subtransient time constant |
| $H$ | Inertia constant |
| $M = 2H$ | Mechanical starting time |
| $D$ | Damping coefficient |
| $K_\omega$ | Speed feedback gain |
| $K_P$ | Active power feedback gain |
| $T_{AA}$ | d-axis additional time constant |

Table 2.1: Synchronous generator parameters

| Symbol | Description |
|---|---|
| $\delta$ | Rotor angle |
| $\omega$ | Rotor speed |
| $e'_q$ | q-axis transient voltage |
| $e'_d$ | d-axis transient voltage |
| $e''_q$ | q-axis subtransient voltage |
| $e''_d$ | d-axis subtransient voltage |
| $P_e$ | Electrical demand(load) power |
| $P_m$ | Mechanical power input |
| $v_f^*$ | Field voltage |
| $\theta$ | Bus voltage phase angle |
| $V$ | Bus voltage magnitude |
| $P$ | Net active power injection at bus |
| $Q$ | Net reactive power injection at bus |

Table 2.2: Synchronous generator variables

synchronous generators is described by the following set of differential equations.

$$
\begin{aligned}
\dot{\delta} &= \Omega_b(\omega - 1) \\
\dot{\omega} &= (P_m - P_e - D(\omega - 1))/M \\
\dot{e}'_q &= (-f_s(e'_q) - (x_d - x'_d - \frac{T''_{do}}{T'_{do}}\frac{x''_d}{x'_d}(x_d - x'_d))i_d + (1 - \frac{T_{AA}}{T'_{do}}v^*_f)/T'_{do} \\
\dot{e}'_d &= (-e'_d + (x_q - x'_q - \frac{T''_{qo}}{T'_{qo}}\frac{x''_q}{x'_q}(x_q - x'_q))i_q)/T'_{qo} \\
\dot{e}''_q &= (-e''_q + e'_q - (x'_d - x''_d + \frac{T''_{do}}{T'_{do}}\frac{x''_d}{x'_d}(x_d - x'_d))i_d + \frac{T_{AA}}{T'_{do}}v^*_F)/T''_{do} \\
\dot{e}''_d &= (-e''_d + e'_d + (x'_q - x''_q + \frac{T''_{qo}}{T'_{qo}}\frac{x''_q}{x'_q}(x_q - x'_q))i_q)/T''_{qo}
\end{aligned}
\tag{2.9}
$$

The power balance equation is as follows.

$$
P_e = (v_q + r_a i_q)i_q + (v_d + r_a i_d)i_d \tag{2.10}
$$

Coupling equations between the generator and the bus are given in Eq. (2.11).

$$
\begin{aligned}
0 &= v_q + r_a i_q - e'_q + (x'_d - x_l)i_d \\
0 &= v_d + r_a i_d - (x_q - x_l)i_q
\end{aligned}
\tag{2.11}
$$

The slack generator is simulated using the fifth order model which assumes that $e'_d = 0$ and $x_q = x'_q$.

## 2.4 Controls

### 2.4.1 Turbine governor

Turbine governors provide the primary frequency regulation for synchronous generators. Turbine governors are simulated using the type I model in PSAT [2]. The parameters of the type I turbine governor of Fig. 2.3 are defined in Table 2.3.

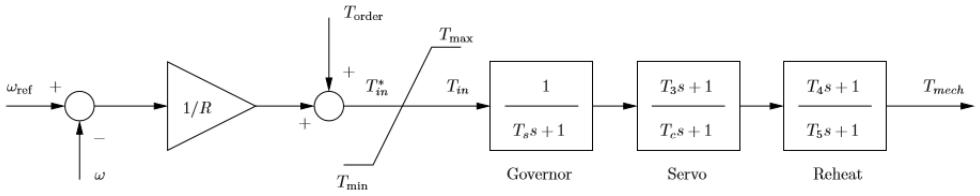



Figure 2.3: PSAT type I turbine governor [1]

| Symbol | Description |
|---|---|
| R | Droop |
| $T_{max}$ | Maximum turbine output |
| $T_{min}$ | Minimum turbine ouptut |
| $T_s$ | Governor time constant |
| $T_c$ | Servo time constant |
| $T_3$ | Transient gain time constant |
| $T_4$ | Power fraction time constant |
| $T_5$ | Reheat time constant |

Table 2.3: Synchronous generator variables

The set of equations corresponding to Fig. 2.3 are provided below.

$$
\begin{aligned}
T^*_{in} &= T_{order} + \frac{1}{R}(\omega_{ref} - \omega) \\
T_{in} &= \begin{cases} T^*_{in}, & T_{min} \le T^*_{in} \le T_{max} \\ T_{max}, & T^*_{in} > T_{max} \\ T_{min}, & T^*_{in} < T_{min} \end{cases} \\
\dot{t}_1 &= (T_{in} - t_1)/T_s \\
\dot{t}_2 &= ((1 - \frac{T_3}{T_c} t_1 - t_2)/T_c \\
\dot{t}_3 &= ((1 - \frac{T_4}{T_5})(t_2 + \frac{T_3}{T_c} t_1) - t_3)/T_5 \\
T_{mech} &= t_3 + \frac{T_4}{T_5}(t_2 - \frac{T_3}{T_c} t_1)
\end{aligned}
\tag{2.12}
$$

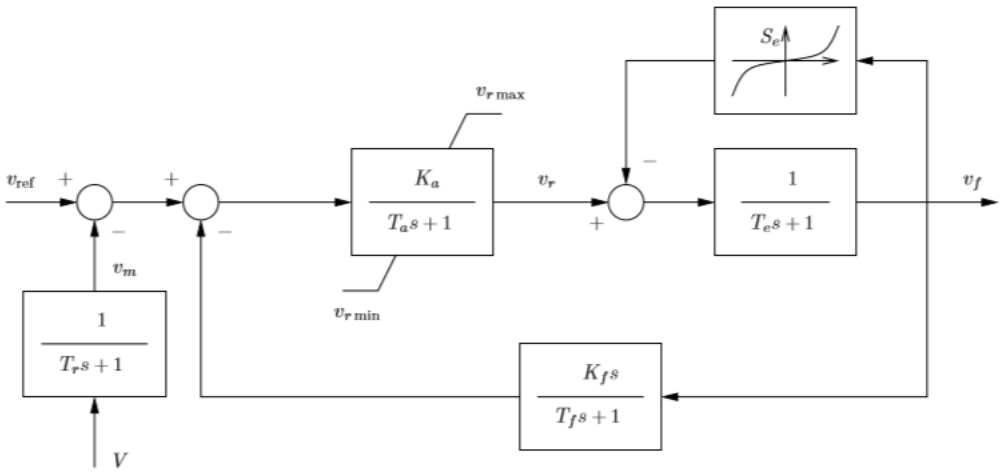



Figure 2.4: IEEE std model I AVR [1]

In Eq. (2.12), the variable $T_{mech}$ is equivalent to the mechanical power input to the generator, $P_m$ in Table 2.1.

### 2.4.2 Automatic voltage regulator

AVR provides the primary voltage regulation in power system. The IEEE standard model I AVR is illustrated in Fig. 2.4. The parameters used to model the AVR are specified in Table 2.4.

The ceiling function, $S_e(v_f)$ which characterizes the saturation of the exciter, is given by:

$$S_e(v_f) = A_e(e^{B_e|v_f|} - 1) \tag{2.13}$$

The block diagram in Fig. 2.4 corresponds to the following set of equations.

| Symbol | Description |
|---|---|
| $K_a$ | Amplifier gain |
| $T_a$ | Amplifier time constant |
| $K_f$ | Stabilizer gain |
| $T_f$ | Stabilizer time constant |
| $T_e$ | Field circuit time constant |
| $T_r$ | Measurement time constant |
| $A_e$ | First ceiling coefficient |
| $B_e$ | Second ceiling coefficient |

Table 2.4: AVR parameters

$$
\begin{aligned}
\dot{v}_m &= (v - v_m)/T_r \\
\dot{v}_{r1} &= (K_a(v_{ref} - v_m - v_{r2} - \frac{K_f}{T_f}v_f) - v_{r1})/T_a \\
v_r &= \begin{cases} v_{r1}, & v_{rmin} \leq v_{r1} \leq v_{rmax} \\ v_{rmax}, & v_{r1} > v_{rmax} \\ v_{r_min}, & v_{r1} < v_{rmin} \end{cases} \\
\dot{v}_{r2} &= -(\frac{K_f}{T_f}v_f + v_{r2})/T_f \\
\dot{v}_f &= -(v_f(1 + S_e(v_f)) - v_r)/T_e
\end{aligned}
\tag{2.14}
$$

## 2.5 Load

The load considered is a PQ load, which is characterized by constant active and reactive powers. After finding the initial power flow solution, i.e. before the time-domain simulation, these are converted to constant impedances.

## 2.6 Power Flow

Before running the time domain simulation, I first arrive at the power flow solution for the system. This forms the initial condition for the time-domain simulation. The objective

is to find the solution to a set of nonlinear equations of the form:

$$\begin{aligned} \dot{x} = 0 &= f(x, y) \\ 0 &= g(x, y) \end{aligned} \tag{2.15}$$

where the algebraic expressions $g$ ($g \in \mathcal{R}^m$) represent the power flow/balance equations, and $f$ ($f \in \mathcal{R}^n$) represents the differential equations. Correspondingly, $y$ ($y \in \mathcal{R}^m$) comprises algebraic variables such as bus voltages and power flows while $x$ ($x \in \mathcal{R}^n$) is the vector of state variables. The solution to the power flow problem is obtained using the Newton-Raphson method [8]. In brief, the algorithm involves iteratively updating the Jacobian of Eq. 2.15 and solving the following linear problem:

$$\begin{aligned} \begin{bmatrix} \Delta x^i \\ \Delta y^i \end{bmatrix} &= - \begin{bmatrix} F_x^i & -F_y^i \\ G_x^i & G_y^i \end{bmatrix}^{-1} \begin{bmatrix} f^i \\ g^i \end{bmatrix} \\ \begin{bmatrix} x^{i+1} \\ y^{i+1} \end{bmatrix} &= \begin{bmatrix} x^i \\ y^i \end{bmatrix} + \begin{bmatrix} \Delta x^i \\ \Delta y^i \end{bmatrix} \end{aligned} \tag{2.16}$$

where $F_x = \nabla_x f_x$, $F_y = \nabla_y f_y$, $G_x = \nabla_x g_x$ and $G_y = \nabla_y g_y$. The algorithm is stopped once the increments $\Delta x$ and $\Delta y$ become lower than some pre-decided tolerance level $\epsilon$, or if the maximum number of iterations is reached.

# 3. DYNAMIC STATE ESTIMATION

## 3.1 Notation

The relevant variables defined at sample time $k$ are given in Table 3.1.

| | |
|---|---|
| $x_k$ | The state vector |
| $u_k$ | The control input vector |
| $z_k$ | The incoming measurement vector |
| $\hat{x}_{k\|k}$ | A filtered vector: state estimate updated using current measurement |
| $\hat{x}_{(k+1)\|k}$ | A predicted state estimate |

Table 3.1: DSE variables for time instant $k$

## 3.2 Why DSE?

It has been established that PMU sensing can pave the way for dynamic state estimation. However, I am yet to justify the necessity of DSE. The prime advantage of Kalman filtering based DSE is the ability to predict the state of the system. This predictive capability can further be used in preventive control strategies, and load forecasting. When it comes to bad data detection, specifically gross measurement errors, DSE offers promising options. For instance, the detection can be performed by comparing the measurement values with the corresponding prediction. A few such DSE 'pre-filtering' methods have been proposed (Nishiya et al. [4], Leite da Silva [5]). It might also make the system robust to short periods of unobservability, since, pseudo-measurements can be obtained from the prediction $\hat{x}_{k|(k-1)}$.

## 3.3 Limitations of DSE

There are two main obstacles to the physical implementation of DSE in large scale power systems. Firstly, it is challenging to find a model that is sufficiently accurate yet simple enough for practical application. Additionally, the computational complexity of Kalman filtering approach is proportional to $n^3$, $n$ being the size of state vector. This work addresses the former concern, i.e, the lack of an accurate model by relying instead on PMU data. The algorithm can be sped up by utilizing sparse matrix operations and parallelizing computations (Karimipour et al. [9]).

## 3.4 Kalman Filtering

Consider a dynamic system modeled by a set of nonlinear equations.

$$\begin{aligned} x_k &= g(x_{k-1}) + \epsilon_{k-1} \\ \epsilon_{k-1} &\sim \mathcal{N}(0, Q_{k-1}) \rightarrow \text{Process Noise} \end{aligned} \tag{3.1}$$

The measurements at time instant $k$ are represented as a vector of non-linear functions of the state variables $x$.

$$\begin{aligned} z_k &= h(x_k) + \delta_k \\ \delta_k &\sim \mathcal{N}(0, R_k) \rightarrow \text{Measurement Noise} \end{aligned} \tag{3.2}$$

Initially, non-linear estimation problems were solved by a variant of the Kalman filter, namely the Extended Kalman Filter (EKF). EKF uses an approximate linearization of the system based on the Taylor series expansion of $g(x)$. It also uses the Jacobian matrices of nonlinear functions $g$ and $h$, which may require complex computation and may not always exist. Subsequently, extensions and generalizations have been developed for the Kalman filtering of nonlinear systems. One such algorithm is the Unscented Kalman Filter. When

$g$ and $h$ are highly nonlinear, EKF can perform poorly [10]. The UKF is seen to more accurately determine the mean and covariance, for certain systems [11]. Moreover, UKF does not involve Jacobian calculations which are far from trivial for a complex system.

### 3.4.1 Unscented transform

Consider the $n$-dimensional estimate $\mathbf{x}$ as having a Gaussian distribution with mean $\mu$ and covariance $\Sigma$. Unscented transformation involves sampling the distribution over the estimate at specific 'sigma' points. The sampling is not random, but rather has a specific algorithm that results in a sample mean and covariance of $\mu$ and $\Sigma$ respectively. Specifically, the sampling is:

$$\begin{aligned} \mathcal{X}^0 &= \mu \\ \mathcal{X}^i &= \mu + (\sqrt{(n+\lambda)\Sigma})_i \text{ for } i = 1, 2, ..., n \\ \mathcal{X}^i &= \mu - (\sqrt{(n+\lambda)\Sigma})_{i-n} \text{ for } i = n+1, ..., 2n \end{aligned} \tag{3.3}$$

where, $(\sqrt{(n+\lambda)\Sigma})_i$ is the $i^{th}$ column of the matrix square root and $\lambda$ is a weight parameter that controls how far apart the sample points lie.

The sampled points are subsequently passed through $g$ (or $h$), thereby analyzing how the noise-free system changes the shape of the Gaussian. The parameters of the resulting Gaussian approximation are extracted from the mapped points $\mathcal{Y}^i = g(\mathcal{X}^i)$ as follows.

$$\mu' = \sum_{i=0}^{2n} w^i \mathcal{Y}^i \tag{3.4}$$

$$\Sigma' = \sum_{i=0}^{2n} w^i (\mathcal{Y}^i - \mu')(\mathcal{Y}^i - \mu')^T \tag{3.5}$$

where the weights $w^i$ are chosen according to the following equations.

$$w^0 = \frac{\lambda}{n+\lambda} \tag{3.6}$$

$$w^i = \frac{1}{2(n+\lambda)} \; for \; i = 1, 2, ...2n \tag{3.7}$$

### 3.4.2 Unscented Kalman filter

Let the initial state mean and covariance be $m_o$ and $\Sigma_o$ respectively. The UKF algorithm has two stages.

*Prediction*

- Sample the posterior distribution of the previous estimate at $2n+1$ sigma points:

$$\mathcal{X}_{k-1} = (\mu_{k-1} \quad \mu_{k-1} + \gamma\sqrt{\Sigma_{k-1}} \quad \mu_{k-1} - \gamma\sqrt{\Sigma_{k-1}}) \tag{3.8}$$

- Pass sigma points, $\mathcal{X}_{k-1}$ through process function $g$ : $\bar{\mathcal{X}}_k = g(\mathcal{X}_{k-1})$

- Compute the predicted state by extracting a Gaussian distribution from $\bar{\mathcal{X}}_k$ with mean and covariance:

$$\hat{\mu}_k = \sum_{i=0}^{2n} w_m^i \bar{\mathcal{X}}_k^i \tag{3.9}$$

$$\hat{\Sigma_k} = \sum_{i=0}^{2n} w_m^i (\bar{\mathcal{X}}_k^i - \hat{\mu}_k)(\bar{\mathcal{X}}_k^i - \hat{\mu}_k)^T + Q_{k-1} \tag{3.10}$$

*Update*

- Sample the posterior distribution of the prediction at $2n+1$ sigma points:

$$\hat{\mathcal{X}}_k = (\hat{\mu}_k \quad \hat{\mu}_k + \gamma\sqrt{\hat{\Sigma}_k} \quad \hat{\mu}_k - \gamma\sqrt{\hat{\Sigma}_k}) \tag{3.11}$$

- Pass the sigma points, $\hat{\mathcal{X}}_k$ through observation function $h$ : $\hat{Z}_k = h(\hat{\mathcal{X}}_k)$

- Compute the predicted observation by extracting a Gaussian distribution from $\hat{Z}_k$ with mean and covariance:

$$\hat{z}_k = \sum_{i=0}^{2n} w_m^i \hat{Z}_k^i \tag{3.12}$$

$$S_k = \sum_{i=0}^{2n} w_c^i (\hat{Z}_k^i - \hat{z}_k)(\hat{Z}_k^i - \hat{z}_k)^T + R_k \tag{3.13}$$

$$\tag{3.14}$$

- Compute the cross covariance between the sigma points, $\hat{\mathcal{X}}_k$ and $\hat{Z}_k$.

$$\hat{\Sigma}_k^{x,z} = \sum_{i=0}^{2n} w_c^i (\hat{\mathcal{X}}_k^i - \hat{\mu}_k)(\hat{Z}_k^i - \hat{z}_k)^T \tag{3.15}$$

- Kalman gain: $K_k = \hat{\Sigma}_k^{x,z} S_k^{-1}$

- State estimate:

$$\mu_k = \hat{\mu}_k + K_k(z_k - \hat{z}_k) \tag{3.16}$$

$$\Sigma_k = \hat{\Sigma}_k - K_k S_k K_k^T \tag{3.17}$$

# 4. GAUSSIAN PROCESSES AND GP-UKF

## 4.1 Gaussian Processes

Modeling time-series data is challenging, because it is difficult to determine a model which can capture the nonlinearities of the data without overfitting. The use of Gaussian process (GP) to model nonlinear dynamics is inspired by the fact that averaging over nonlinear regression models tends to a GP model. Gaussian processes have been used as a nonlinear regression tool in many applications. For instance, Girard et al. used GPs for multiple-step ahead prediction for time-series data [12]. In fact, GPs have proved to be successful tools in complex nonlinear problems like tracking 3D human figures (Sidenbladh et al. [13]).

### 4.1.1 Advantages of GP models

The application of a GP model to machine learning problems is inspired by the following features.

- They are an easily adaptable non-parametric Bayesian modelling tool for supervised learning problems.
- Given the hyperparameter values (spread of the Gaussian kernel and the peak location), they can be optimized exactly. This allows for fine-tuning of the trade-off between fitting of data and smoothing.
- They are especially reliable on small datasets because of this fine-tuned smoothing and reasonably low computational burden.
- Often, non-Gaussian processes can be approximated by a mixture of GPs.

### 4.1.2 Limitations of GP

They do not scale well. The computational burden is $O(n^3)$, so if the number of features exceeds a few dozens, they might not be an efficient strategy. However, there are tricks for dimensionality reduction which might improve their applicability to high-dimension data.

### 4.1.3 GP for machine learning

Consider a general noisy observation of the form:

$$y_i = f(\mathbf{x_i}) + \epsilon, \quad \mathbf{x_i} \in \Re^n \tag{4.1}$$

where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ is noise and $y_i$ is a scalar

Let

$$\begin{aligned} \mathbf{X} &= [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}] \\ \mathbf{y} &= [y_1, y_2, ..., y_n]. \end{aligned}$$

Then the distribution on $y$ is given by Eq. (4.2).

$$p(\mathbf{y}|\mathbf{X}, \Theta) = \mathcal{N}(0, K(X, X) + \sigma_n^2 I) \tag{4.2}$$

$K(\mathbf{X}, \mathbf{X})$ is a kernel matrix such that $K_{ij} = k(\mathbf{x_i}, \mathbf{x_j})$, $k$ being a kernel function based on the measure of closeness between input vectors. The kernel function used to describe the GPs in this thesis, is the squared exponential kernel given below.

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 exp(\frac{-(\mathbf{x} - \mathbf{x}')W(\mathbf{x} - \mathbf{x}')^T}{2}) \tag{4.3}$$

In Eq. (4.3), $W$ is a diagonal matrix of the length scaling factors for each input dimension.

*Learning GP Hyperparameters*

The GP hyperparameter $\Theta = [W, \sigma_f^2, \sigma_n^2]$ can be learned by maximizing the log likelihood of the training outputs given the inputs.

$$\Theta_{max} = arg \max_{\Theta} log(p(\mathbf{y}|\mathbf{X}, \Theta)) \tag{4.4}$$

The optimization problem in Eq. (4.4) can be solved in a number of ways, such as using conjugate gradient descent [14].

$$log\ p(\mathbf{y}|\mathbf{X}, \Theta) = -\frac{1}{2}\mathbf{y}^T K_y^{-1}\mathbf{y} - \frac{1}{2}log|K_y| - \frac{n}{2}log\ 2\pi \tag{4.5}$$

where $K_y = K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I$.

Each of the three terms has a recognizable interpretation. The first term is the only one involving the target $\mathbf{y}$, and is responsible for data fitting. The term $log|K_y|/2$ serves as a complexity penalty, and $n\ log\ 2\pi/2$ is a normalization constant.

The partial derivative of the objective function with respect to the hyperparameters is given below.

$$\begin{aligned} \frac{\partial}{\partial\theta_j} log\ p(\mathbf{y}|\mathbf{X}, \Theta) &= \frac{1}{2}\mathbf{y}^T K_y^{-1}\frac{\partial K_y}{\partial\theta_j}K_y^{-1}\mathbf{y} - \frac{1}{2}tr(K_y^{-1}\frac{\partial K_y}{\partial\theta_j}) \\ &= \frac{1}{2}tr((\alpha\alpha^T - K_y^{-1})\frac{\partial K_y}{\partial\theta_i}) \end{aligned} \tag{4.6}$$

where $\alpha = K_y^{-1}\mathbf{y}$

Once $K_y^{-1}$ is known, the computational complexity of Eq. (4.6) is $O(n^2)$ per hyperparameter. Due to the relatively small computational overhead of computing derivatives, gradient based optimizer is apt for training the GP.

*GP prediction*

At a test point $\mathbf{x}_*$, the predicted observation $y_*$ has a Gaussian distribution with mean,

$$GP_\mu(x_*, D) = \mathbf{k}_*{}^T[K + \sigma_n^2 I]^{-1}\mathbf{y} \tag{4.7}$$

and covariance matrix,

$$GP_\Sigma(\mathbf{x}_*, D) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*{}^T[K + \sigma_n^2 I]^{-1}\mathbf{k}_* \tag{4.8}$$

where, $\mathbf{k}_* = k(\mathbf{x}_*, \mathbf{X})$.

## 4.2 GP-UKF

The GP-UKF algorithm is fundamentally similar to UKF but with one major difference. In place of $g$, I use the GP approximation $GP^g$ and in place of the observation function, $GP^h$. To determine these approximations, I collect training data. These examples are used to learn the hyperparameters that define the GP.

### 4.2.1 Learning prediction and observation models

The training data comprises the following.

Prediction data set: $D_g = \langle (X, U), X' \rangle$

Observation data set: $D_h = \langle (X, Z) \rangle$

where $X$ is the matrix of ground truth states, $X' = [\Delta x_1, \Delta x_2, ....\Delta x_k]$ is the matrix of state transitions when applying the controls in $U$. $Z$ is the matrix of observations.

GP approximation of functions $g$ and $h$ are $GP^g$ and $GP^h$ respectively.

$$\begin{aligned} x_k &= GP^g_\mu([x_{k-1}, u_{k-1}], D_g) + \epsilon_k \\ z_k &= GP^h_\mu(x_k, D_h) + \delta_k \\ \epsilon_k &\sim \mathcal{N}(0, GP^g_\Sigma([x_{k-1}, u_{k-1}], D_g)) \\ \delta_k &\sim \mathcal{N}(0, GP^h_\Sigma(x_k, D_h)) \end{aligned} \tag{4.9}$$

Each GP has a single global noise parameter $\sigma_n$. A separate GP must be learned for each output dimension. The resulting separate variances are collected on the diagonal of a Variance matrix.

### 4.2.2 The algorithm

**Data:**

- $(\mu_{k-1}, \Sigma_{k-1}, u_{k-1}, z_k)$
- Prediction model training data: $D_g$
- Observation model training data: $D_h$

**Result:** $\mu_k, \Sigma_k$

Initialization: $\mu_0 =$ sample mean of $D_g$, $\Sigma_0 =$ sample covariance of $D_g$;

Sigma points $\mathcal{X}_{k-1} = (\mu_{k-1} \quad \mu_{k-1} + \gamma\sqrt{\Sigma_{k-1}} \quad \mu_{k-1} - \gamma\sqrt{\Sigma_{k-1}})$ ;

**for** $i = 0, \ldots, 2n$ **do**

$\quad \bar{\mathcal{X}}_k^i = GP_\mu(u_{k-1}, \mathcal{X}_{k-1}^i, D_g)$ ;

**end**

$Q_k = GP_\Sigma(u_{k-1}, \mu_{k-1}, D_g)$ ;

$\hat{\mu}_k = \sum_{i=0}^{2n} w_m^i \bar{\mathcal{X}}_k^i$ ;

$\hat{\mathcal{X}}_k = (\hat{\mu}_k \quad \hat{\mu}_k + \gamma\sqrt{\hat{\Sigma}_k} \quad \hat{\mu}_k - \gamma\sqrt{\hat{\Sigma}_k})$ ;

**for** $i = 0, \ldots, 2n$ **do**

$\quad \hat{Z}_k^i = GP_\mu(\hat{\mathcal{X}}_k^i, D_h)$ ;

**end**

$R_k = GP_\Sigma(\hat{\mu}_k, D_h)$ ;

$\hat{z}_k = \sum_{i=0}^{2n} w_m^i \hat{Z}_k^i$ ;

$S_k = \sum_{i=0}^{2n} w_c^i (\hat{Z}_k^i - \hat{z}_k)(\hat{Z}_k^i - \hat{z}_k)^T + R_k$ ;

$\hat{\Sigma}_k^{x,z} = \sum_{i=0}^{2n} w_c^i (\hat{\mathcal{X}}_k^i - \hat{\mu}_k)(\hat{Z}_k^i - \hat{z}_k)^T$ ;

$K_k = \hat{\Sigma}_k^{x,z} S_k^{-1}$ ;

$\mu_k = \hat{\mu}_k + K_k(z_k - \hat{z}_k)$ ;

$\Sigma_k = \hat{\Sigma}_k - K_k S_k K_k^T$ ;

**Algorithm 1:** GP-UKF Algorithm [7]

# 5. EVALUATION OF GP-UKF DYNAMIC STATE ESTIMATOR

## 5.1 Simulation Data

The estimator has been evaluated on the IEEE 14-bus test case shown in Fig. 5.1. The complete data for the IEEE 14 bus system is given in Appendix A.
Firstly, the system must be initialized. This is done by running power flow operation on PSAT for the system. This finds the steady state solution for the various currents, voltages and power flows within the system. The results of power flow are listed in Appendix B.

### 5.1.1 Collecting training data

It is desirable to collect training data which reflects the dynamic behavior of the system. The idea is to simulate a sudden change in network topology which will cause the system to go into transient state. Multiple such stimulants can be given over a short period to get small amounts of varied data. A simple scenario that was utilized for this purpose involved opening both circuit breakers on the lines between bus 2 and bus 3, and between bus 2 and bus 4 at time, t = 1 second.

## 5.2 Training the Gaussian Process Model

The training data is used to 'infer' the posterior distribution on the process and observation functions, i.e. $p(g|D_g)$ and $p(h|D_h)$ respectively. The GPML toolbox [15] is equipped with a few inference methods.

### 5.2.1 Exact inference

By imposing a Gaussian likelihood on the training data, it is possible to perform exact inference of the posterior. In fact, the inference problem reduces to simply computing the mean and covariance of a multivariate Gaussian distribution. This can be done using basic

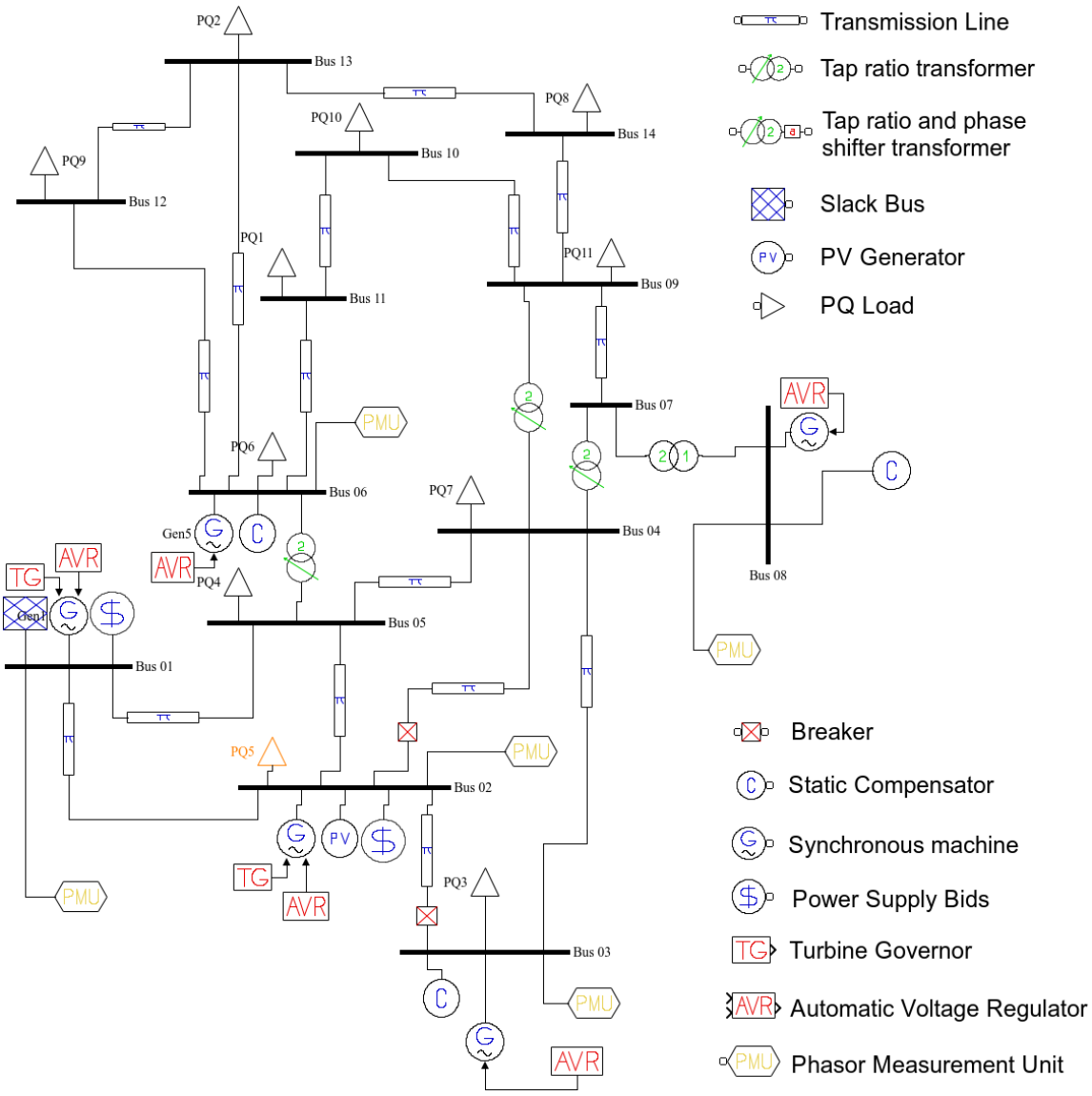



Figure 5.1: IEEE 14-bus system with legend [2]

matrix algebra.

### 5.2.2 The hyperparameters

*Covariance function*

A covariance function $k_\psi : \mathcal{X}$ x $\mathcal{X} \rightarrow \mathcal{R}$ with hyperparameters $\psi$ is defined for every GP $f$. It computes the covariance $k_\psi(\mathbf{x_i}, \mathbf{x_j}) = E[(f(\mathbf{x_i}) - m(\mathbf{x_i}))(f(\mathbf{x_j}) - m(\mathbf{x_j}))]$ between inputs $\mathbf{x_i}$ and $\mathbf{x_j}$. The covariance function used is the squared exponential function with automatic relevance detection, along with an additive measurement noise component. Mathematically, the covariance function is given as:

$$k(\mathbf{x}, \mathbf{y}) = \sigma_f^2 exp((\mathbf{x} - \mathbf{y})^T P^{-1}(\mathbf{x} - \mathbf{y})) + \sigma_n^2 \delta(\mathbf{x} - \mathbf{y}) \tag{5.1}$$

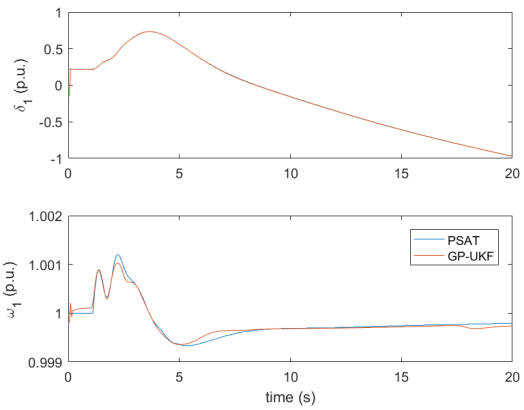
where the relevance matrix is $P = \Lambda^2, \Lambda = diag(\lambda), \lambda \in \mathcal{R}_+^D$. Thus, the hyperparameters to be trained are $\psi = \{\lambda, \sigma_f, \sigma_n\}$.

## 5.3 Training Phase

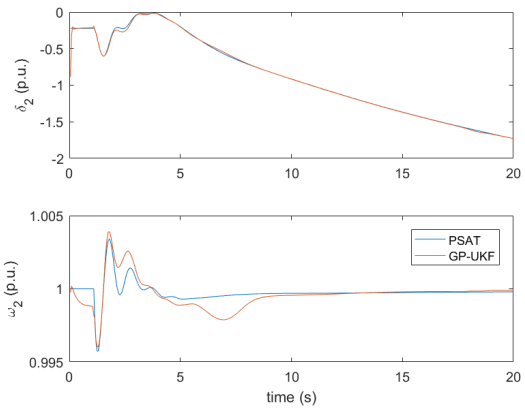
The trained estimator is first evaluated on the training data. The data for training is generated by opening circuit breakers on two lines - one between bus 2 and bus 4 and the other between bus 2 and bus 3, both at t = 1 second. The corresponding RMS error is plotted in Fig. 5.2f. The maximum training error reached is 1.53%. Thus, the GP model fits the training data with a 98.4% accuracy.

## 5.4 Validation Phase

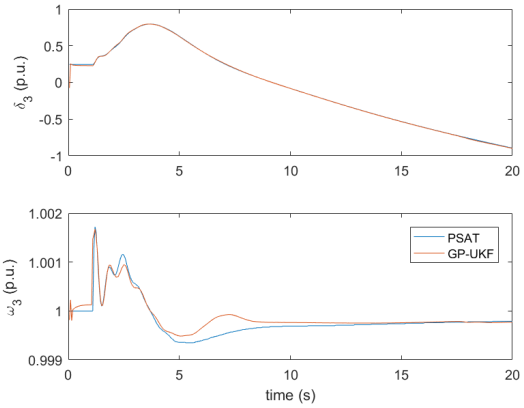
Validation is a technique used to decide between models or external fixed parameters. The data is split into training samples and validation sets. The trained estimators are evaluated based on the errors observed with the validation set. The estimator with the best performance is picked for testing. This strategy is used to pick between a handful of inference models, mean functions and covariance functions. It is also used to pick a



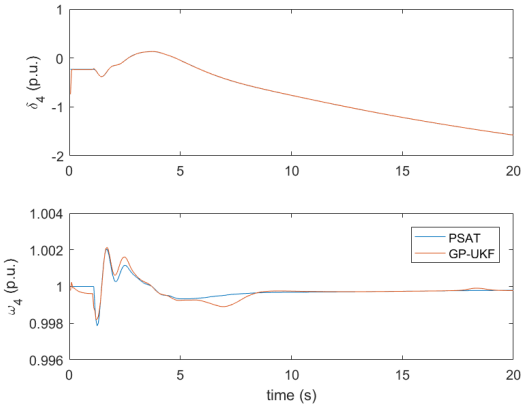

(a) Synchronous generator at bus 1



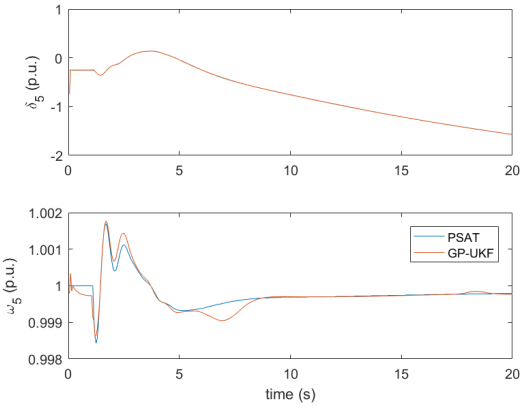

(b) Synchronous generator at bus 2



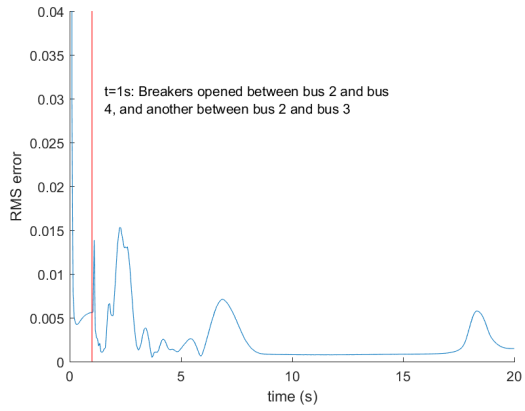

(c) Synchronous generator at bus 3




(d) Synchronous generator at bus 6




(e) Synchronous generator at bus 8




(f) RMS error

Figure 5.2: Training phase: Estimated (in red) and true (in blue) states - rotor angle and speed of synchronous machines at buses 1, 3, 2, 8 and 6 respectively.

good value for the weight parameter $\lambda$. Recall that, $\lambda$ determines the spacing between consecutive samples in the procedure for unscented transform. Larger $\lambda$ values would mean that samples are farther apart and capture more outlier data. However, these might perform poorly since outlier scenarios are rarer. Thus, there is a trade-off between fitting and generalization. This point is highlighted in the Fig. 5.3. Based on the validation error observed for different states, a value of $\lambda = 0.1$ is picked.

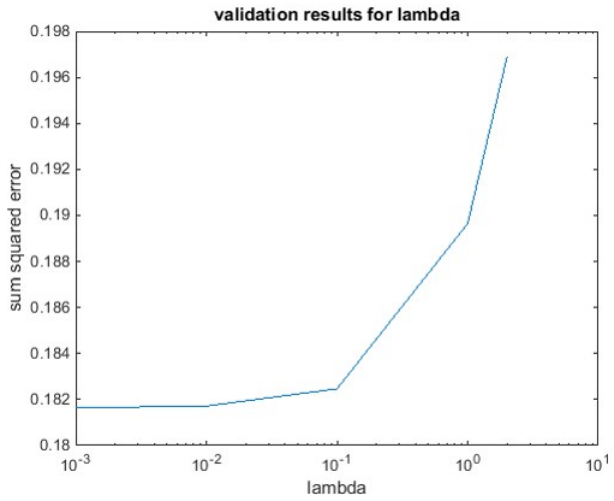



Figure 5.3: Sum squared error over all samples for the state variable $\delta$ (generator at bus 1) for different values of $\lambda$

## 5.5 Testing Conditions

### 5.5.1 Scenario 1: Topology change

Once I arrive at a set of learned hyperparameters, I start to evaluate the performance of the particular GP. To do this, I simulate scenarios which differ in general pattern from the training data. The first such scenario is a topology change using circuit breakers. Specifically, the opening of a breaker on the line between bus 2 and bus 4 at t = 1 second.

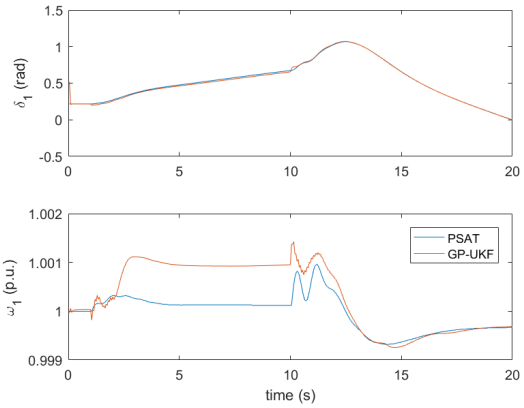
Additionally, the circuit breaker on the line between bus 2 and bus 3 is opened at t = 10 seconds. The measurement data is collected over a span of 20 seconds. These are then sequentially fed as input to the estimator. At each time step, a new estimate of the state is generated. The time sequence of state estimate is plotted and compared against the true state supplied by PSAT. The relative RMS error over all the states is plotted and the maximum error is determined. At each time step, the relative error between the true and estimated value is calculated for every state variable. The RMS value of these relative errors is plotted against time in Fig. 5.4f. The error is seen to reach a maximum of 5.4%.

### 5.5.2 Scenario 2: Parametric uncertainty

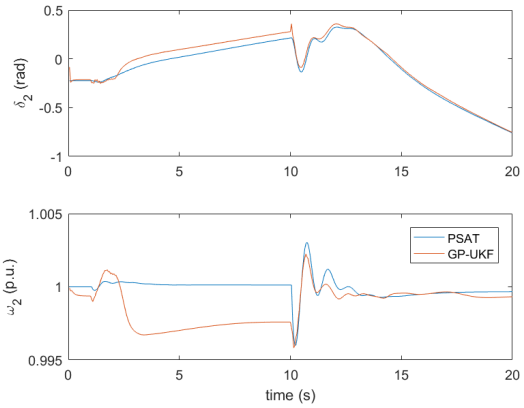
In order to evaluate the robustness of the trained estimator to changes in generator pattern, I introduce parametric uncertainty in generator parameters. One such experiment involves changing the parameters of the synchronous generator at bus 8. The estimator is trained on dynamic behaviour data obtained upon opening breakers between bus 2 and bus 3, and between bus 2 and bus 4 at t = 1 second. The testing scenario is the same as that used in scenario 1, but with perturbed generator parameters. For the synchronous machine at bus 8, the inertia constant is increased by 50% and damping ratio is halved. The performance of the estimator is illustrated in Fig. 5.5. The RMS error over the state variables reaches a maximum of 5.4%, giving an accuracy of 94.6% in RMS terms.

### 5.5.3 Scenario 3: Noisy measurements

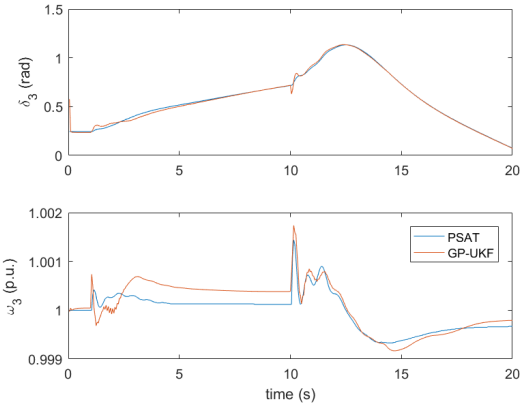
The IEEE standard for synchrophasor measurement does not specify noise characteristics or distribution. Typically, zero-mean additive Gaussian noise is assumed. A wide range of SNRs has been used to simulate noisy PMU measurements. Publications such as, [16], [17], [18], [19] use a standard deviation of 0.002, 0.02, 0.15, 0.03 pu respectively. An SNR of 92 dB has been used by Xie et al. [20]. A more recent paper by Brown et al. [21] found the PMU measurements on EPFL campus to be around 45 dB.



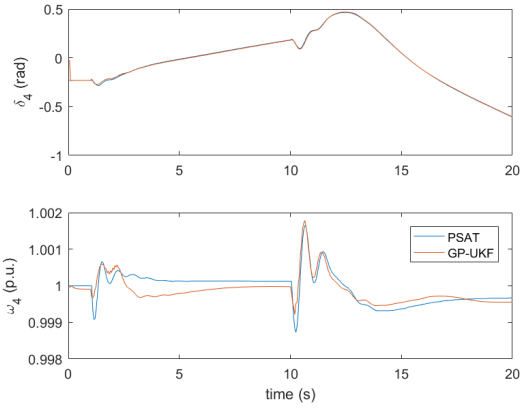

(a) Synchronous generator at bus 1



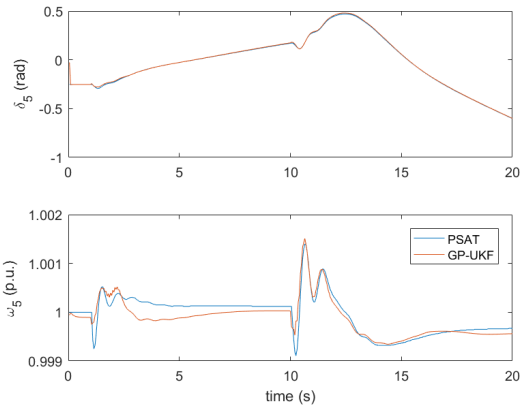

(b) Synchronous generator at bus 3



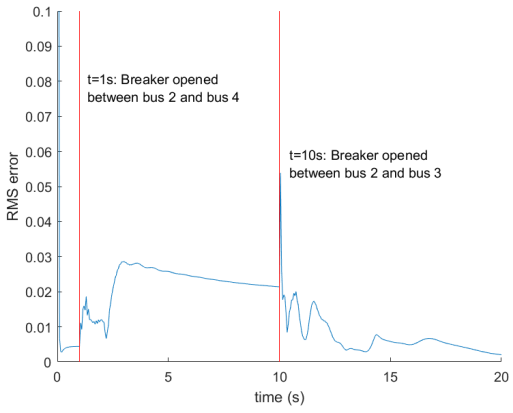

(c) Synchronous generator at bus 2




(d) Synchronous generator at bus 8

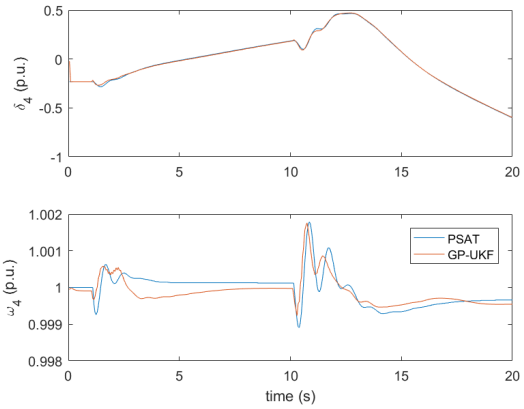



(e) Synchronous generator at bus 6

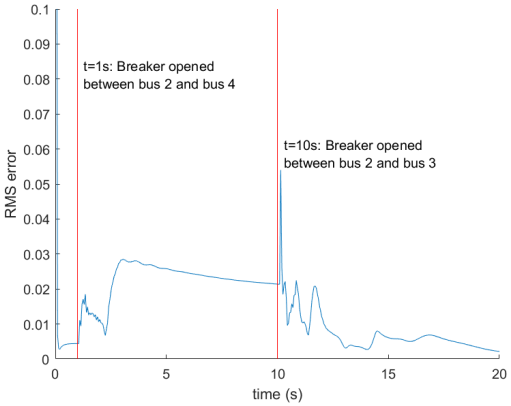



(f) RMS error

Figure 5.4: Scenario 1: Estimated (in red) and true (in blue) states - rotor angle and speed of synchronous machines at buses 1, 3, 2, 8 and 6 respectively



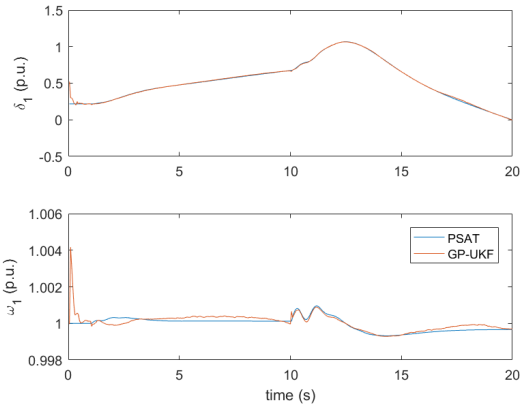

(a) Estimated (in red) and true (in blue) states - rotor angle and speed of generator at bus 8



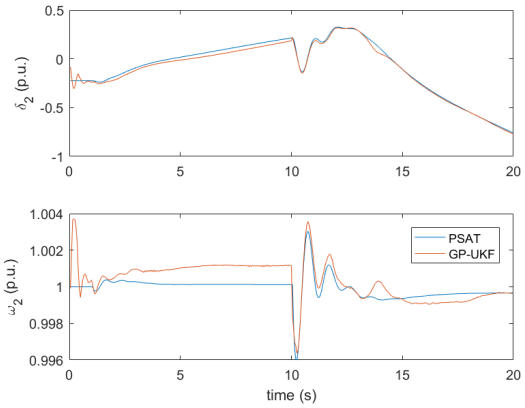

(b) RMS error

Figure 5.5: Scenario 2: Parametric uncertainty

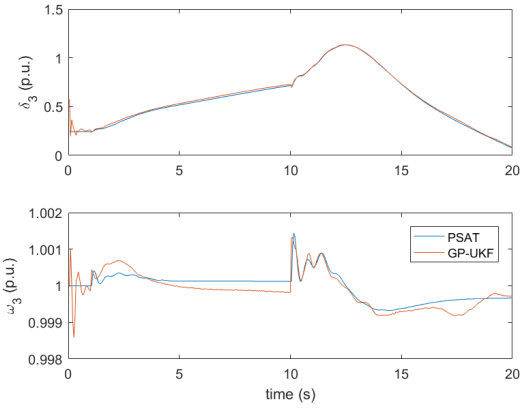
To see the impact of noisy measurements on the estimation scheme, noise is added to the voltage measurement at bus 8. An SNR of 65 dB is considered when adding zero-mean Gaussian noise to the measurement. This is done in addition to the opening of breakers on lines between bus 2 and bus 4 and that between bus 2 and bus 3 at 1 second and 10 seconds respectively. The RMS error over all the states for estimation with SNR 65 dB for bus 8 voltage measurement is given in Fig. 5.6f. The RMS error is below 2.2% after the initial estimation steps.



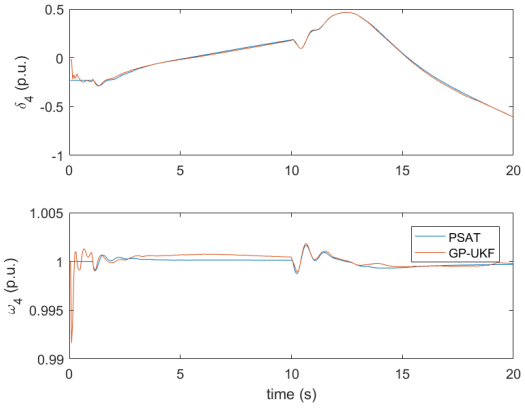

(a) Synchronous generator at bus 1



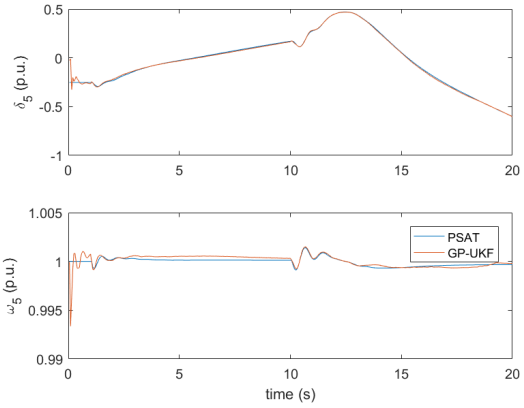

(b) Synchronous generator at bus 3



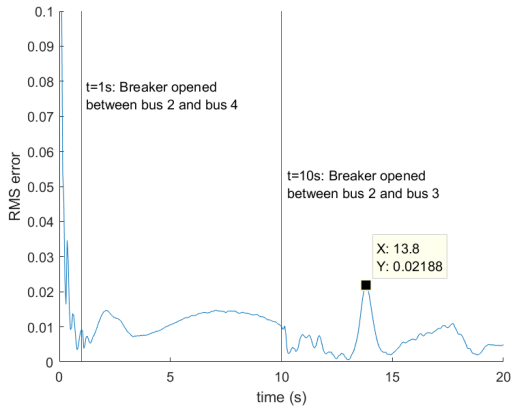

(c) Synchronous generator at bus 2




(d) Synchronous generator at bus 8




(e) Synchronous generator at bus 6

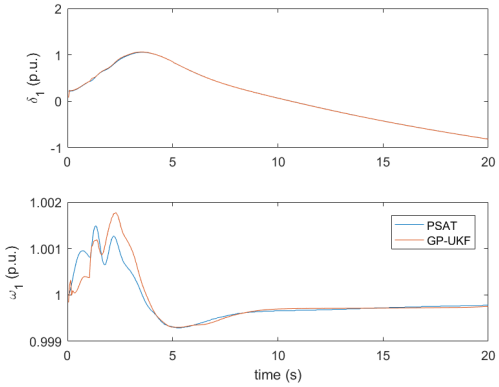



(f) RMS error

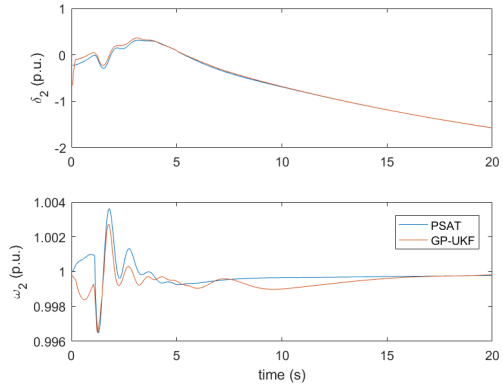
Figure 5.6: Scenario 3: Estimated (in red) and true (in blue) states - rotor angle and speed of synchronous machines at buses 1, 3, 2, 8 and 6 respectively.

### 5.5.4 Scenario 4: Load perturbations

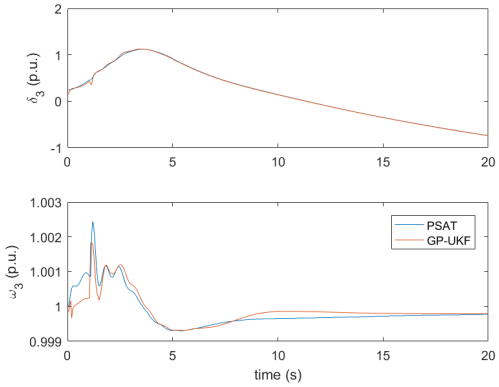
Changes in the loading conditions could be another cause for change in the system state. A good estimator would be able to predict the state even in the presence of load perturbations. To evaluate the performance of the GP-UKF estimation scheme, I employ the following methodology. First, I train the estimator as usual using the data collected over a time span of 20 seconds. During this period, I open the breakers on lines between bus 2 and bus 3, and between bus 2 and bus 4 at time, 1 second. Once the GP has been has been optimized to the training data, I test the estimator under altered conditions. In addition to the opening of breakers as described above, the load profile at bus 2 is changed in the following manner. Starting at t = 0, the active load at bus 2 is increased linearly at a rate of 0.217 p.u./s. The load is held constant once it reaches a value of 0.217 p.u. The estimated rotor angle and rotor speed is compared with the true values generated by PSAT as shown in Fig.5.7. The RMS value over the relative errors of all states is plotted in Fig. 5.7f. An overall accuracy of at least 96% is indicated by the RMS error.



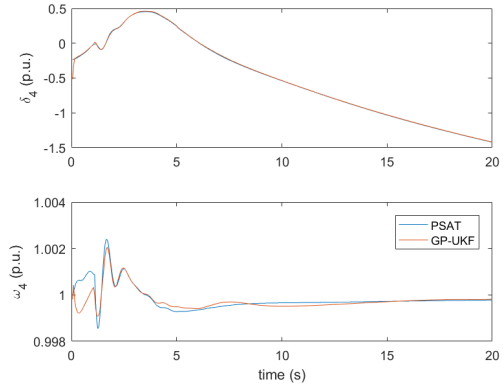

(a) Synchronous generator at bus 1



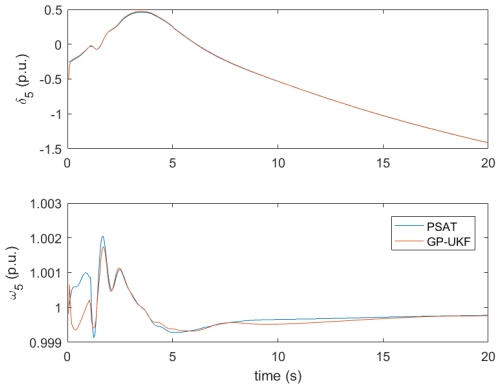

(b) Synchronous generator at bus 3



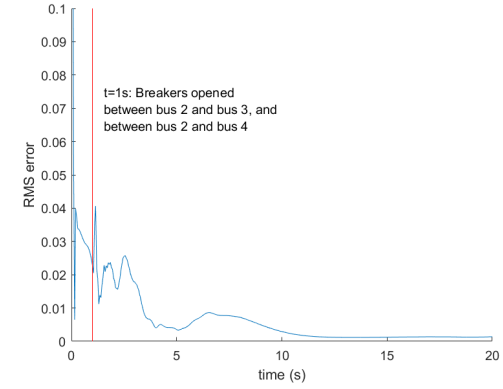

(c) Synchronous generator at bus 2




(d) Synchronous generator at bus 8




(e) Synchronous generator at bus 6




(f) RMS error

Figure 5.7: Scenario 4: Estimated (in red) and true (in blue) states - rotor angle and speed of synchronous machines at buses 1, 3, 2, 8 and 6 respectively.

# 6. CONCLUSIONS

The current standard for state estimation, based on a snapshot of the system, is performed every few minutes. No relevance is given to how the system evolves over time. Dynamic state estimation has been studied and envisioned as a realizable alternative. The predictive capabilities of Kalman filtering techniques provide information for preventive control. Additional benefits are improvements in observability analysis, bad data detection and detection of topology errors. Up until a decade ago, state estimation relied on SCADA measurements which were too far apart to perform dynamic state estimation. But with the widespread deployment of PMUs, system operators are privy to vast amounts of measurement data delivered every few milliseconds. However, dynamic state estimation remains an impractical functionality because it is difficult to find good models for large transmission networks.

This thesis investigates a data-driven approach to dynamic state estimation which could circumvent the need to model large, complex systems.

## 6.1 Contributions

This thesis studies and investigates the application of the GP-UKF algorithm for dynamic state estimation based solely on measurement and input data. By employing this algorithm, I avoid determining a model for the system, while retaining the benefits of Kalman filtering. The algorithm is evaluated on a simulated IEEE 14 bus test system.

**Network topology changes.** The estimator performs reasonably well when network switching occurs through the operation of breakers placed on specific lines. It is able to estimate the state of the network with an accuracy of 94.6%.

**Parametric uncertainty.** The estimator demonstrated some robustness to changes in system parameters. A combination of network switching and parametric perturbation was

simulated as a test case. The estimator performed just as well as it did with the network switching alone.

**Noisy measurement.** The measurements were corrupted with additive white Gaussian noise with SNR = 65 dB. The estimator performed at an accuracy rate of 97.8% when the noise was added to a single measurement.

**Load perturbation.** To determine how the estimator performs with the system under load perturbation, the active PQ load at a single bus is changed continuously over a short interval. The GP-UKF estimate matched the true state with a 96% accuracy.

Over all the scenarios and states, the largest error seen was in the tune of 5.5%. The estimation scheme is moderately accurate with the added advantage of not requiring a dynamic model of the system.

## 6.2 Limitations and Future Research

The estimation has shown some promising results, however it is not without some concerns. Each run of dynamic state estimation on an Intel i7-6500U CPU @2.50 GHz takes on average 1.5 seconds. PMU measurements are captured every few milliseconds. Parallel GPU implementation could cut this processing time by 90%, but that might still not be at par with the measurement rate. Some form of dimensionality reduction may become necessary to operate at an optimum rate.

The implementation itself is not without a few issues. Larger systems with renewable penetration might behave differently than the IEEE 14-bus test system. However, this thesis aims to initiate research into a more data-driven approach to state estimation in power networks, so that we may fully exploit the capabilities of PMU sensing technology.

REFERENCES


[1] F. Milano, "Power system analysis toolbox," *Documentation for PSAT version 2.0.0*, February 2008.

[2] F. Milano, "An open source power system analysis toolbox," *IEEE Transactions on Power Systems*, vol. 20, pp. 1199 – 1206, August 2005.

[3] F. C. Schweppe and J. Wildes, "Power system static state estimation - part i, ii iii," *IEEE Transactions on Power Apparatus and Systems*, vol. 89, pp. 120–135, January 1970.

[4] K. Nishiya, J. Hasegawa, and T. Koike, "Dynamic state estimation including anomaly detection and identification for power systems," *Proc. lEE*, vol. 129, no. 5, pp. 192–198, 1982.

[5] A. M. Leite da Silva, M. B. do Coutto Filho, and J. M. C. Cantera, "An efficient dynamic state estimation algorithm including bad data processing," *IEEE Transactions on Power Systems*, vol. 2, no. 4, pp. 1050–1058, 1987.

[6] J. Wang, A. Hertzmann, and D. M. Blei, "Gaussian process dynamical models," *Advances in Neural Information Processing Systems 18*, pp. 1441–1448, 2006.

[7] J. Ko, D. J. Kleint, D. Fox, and D. Haehnelt, "GP-UKF: Unscented Kalman filters with Gaussian process prediction and observation models," *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*.

[8] W. F. Tinney and C. E. Hart, "Power flow solution by Newton's method," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-86, pp. 1449–1460, November 1967.

[9] H. Karimipour and V. Dinavahi, "Extended Kalman filter-based parallel dynamic state estimation," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1539–1549, 2015.

[10] S. Julier and J. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," *Proc. of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 1997.

[11] F. Gustafsson and G. Hendeby, "Some relations between extended and unscented Kalman filters," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 545–555, 2012.

[12] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs - application to multiple-step ahead time series forecasting," *Proc. NIPS*, vol. 15, pp. 529–536, 2003.

[13] H. Sidenbladh, M. J. Black, and D. J. Fleet, "Stochastic tracking of 3D human figures using 2D motion," *European Conference on Computer Vision*, pp. 702–718, 2000.

[14] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. The MIT Press, 2005.

[15] C. E. Rasmussen and H. Nickisch, "The gpml toolbox version 4.0," October 2016.

[16] D. Shi, D. Tylavsky, and N. Logic, "An adaptive method for detection and correction of errors in PMU measurements," *IEEE Transactions on Smart Grid*, vol. 3, pp. 1575–1583, December 2012.

[17] E. Ghahremani and I. Kamwa, "Local and wide-area PMU-based decentralized dynamic state estimation in multi-machine power systems," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 547–562, 2016.

[18] Q. Huang, L. Shao, and N. Li, "Dynamic detection of transmission line outages using hidden Markov models," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 2026–2033, 2015.

[19] N. Zhou, "A cross-coherence method for detecting oscillations," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 623–631, 2016.

[20] L. Xie, Y. Chen, and P. R. Kumar, "Dimensionality reduction of synchrophasor data for early event detection: Linearized analysis," *IEEE Transactions on Power Systems*, vol. 29, p. 2784âĂŞ2794, November 2014.

[21] M. Brown, M. Biswal, S. Brahma, S. J. Ranade, and H. Cao, "Characterizing and quantifying noise in PMU data," *Power and Energy Society General Meeting (PESGM)*, 2016.

# APPENDIX A

## SIMULATION DETAILS OF NETWORK COMPONENTS

| Bus number | Power rating(MVA) | voltage rating(kV) | Active power(p.u.) | Reactive power(p.u.) | $V_{max}$ (p.u.) | $V_{min}$ (p.u.) |
|---|---|---|---|---|---|---|
| 11 | 100 | 13.8 | 0.035 | 0.018 | 1.2 | 0.6 |
| 13 | 100 | 13.8 | 0.135 | 0.058 | 1.2 | 0.6 |
| 3 | 100 | 69 | 0.942 | 0.19 | 1.5 | 0.8 |
| 5 | 100 | 69 | 0.076 | 0.016 | 1.2 | 0.6 |
| 2 | 100 | 69 | 0.217 | 0.127 | 1.2 | 0.8 |
| 6 | 100 | 13.8 | 0.112 | 0.075 | 1.5 | 0.6 |
| 4 | 100 | 69 | 0.478 | 0.04 | 1.2 | 0.6 |
| 14 | 100 | 13.8 | 0.149 | 0.05 | 1.2 | 0.5 |
| 12 | 100 | 13.8 | 0.061 | 0.016 | 1.2 | 0.6 |
| 10 | 100 | 13.8 | 0.09 | 0.058 | 1.2 | 0.6 |
| 9 | 100 | 13.8 | 0.295 | 0.166 | 1.2 | 0.6 |

Table A.1: Simulation details of PQ loads

| Bus number | Power rating (MVA) | Voltage rating (kV) | Voltage magnitude (p.u.) | $Q_{max}$ (p.u.) | $Q_{min}$ (p.u.) | $V_{max}$ (p.u.) | $V_{min}$ (p.u.) |
|---|---|---|---|---|---|---|---|
| 6 | 100 | 13.8 | 1.07 | 0.24 | -0.06 | 1.0701 | 0.6 |
| 3 | 100 | 69 | 1.01 | 0.4 | 0 | 1.0101 | 0.6 |
| 8 | 100 | 18 | 1.09 | 0.2517 | -0.06 | 1.0901 | 0.6 |

Table A.2: Simulation details of static synchronous compensators

| Bus number | Power rating (MVA) | Voltage rating (kV) | Active power (p.u.) | Voltage magnitude (p.u.) | $Q_{max}$ (p.u.) | $Q_{min}$ (p.u.) | $V_{max}$ (p.u.) | $V_{min}$ (p.u.) |
|---|---|---|---|---|---|---|---|---|
| 2 | 100 | 69 | 0.4 | 1.045 | 0.5 | -0.4 | 1.0451 | 0.8 |

Table A.3: Simulation details of PV generator

| Entity | Gen1 | Gen2 | Gen3 | Gen4 | Gen5 |
|---|---|---|---|---|---|
| Bus number | 1 | 3 | 2 | 8 | 6 |
| $x_l$ | 0.2396 | 0 | 0 | 0.134 | 0.134 |
| $r_a$ | 0 | 0.0031 | 0.0031 | 0.0014 | 0.0014 |
| $x_d$ | 0.8979 | 1.05 | 1.05 | 1.25 | 1.25 |
| $x'_d$ | 0.2998 | 0.185 | 0.185 | 0.232 | 0.232 |
| $x''_d$ | 7.4 | 6.1 | 6.1 | 4.75 | 4.75 |
| $x_q$ | 0.646 | 0.36 | 0.36 | 0.715 | 0.715 |
| $x'_q$ | 0.4 | 0.13 | 0.13 | 0.12 | 0.12 |
| $x''_q$ | 0 | 0.3 | 0.3 | 1.5 | 1.5 |
| $T'_{d0}$ | 0.03 | 0.04 | 0.04 | 0.06 | 0.06 |
| $T''_{do}$ | 0.646 | 0.98 | 0.98 | 1.22 | 1.22 |
| $T'_{q0}$ | 0.033 | 0.0990 | 0.099 | 0.21 | 0.21 |
| $T''_{qo}$ | 10.296 | 13.08 | 13.08 | 10.12 | 10.12 |
| $M$ | 2 | 2 | 2 | 2 | 2 |
| $D$ | 1 | 1 | 1 | 1 | 1 |

Table A.4: Simulation details of synchronous generators

| Bus number | $\omega_{ref}$ (p.u.) | R (p.u.) | $T_{max}$ (p.u.) | $T_{min}$ (p.u.) | $T_s$ (s) | $T_c$ (s) | $T_3$ (s) | $T_4$ (s) | $T_5$ (s) |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 0.02 | 1.2 | 0.3 | 0.1 | 0.45 | 0 | 12 | 50 |
| 1 | 1 | 0.02 | 1.2 | 0.3 | 0.1 | 0.45 | 0 | 12 | 50 |

Table A.5: Simulation details of turbine governors

| Entity | Gen1 | Gen2 | Gen3 | Gen4 | Gen5 |
|---|---|---|---|---|---|
| Bus number | 1 | 3 | 2 | 8 | 6 |
| $V_{r,max}$ | 7.32 | 4.38 | 4.38 | 6.81 | 6.81 |
| $V_{r,min}$ | 0 | 0 | 0 | 1.395 | 1.395 |
| $K_a$ | 200 | 20 | 20 | 20 | 20 |
| $T_a$ | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 |
| $K_f$ | 0.002 | 0.001 | 0.001 | 0.001 | 0.001 |
| $T_f$ | 1 | 1 | 1 | 1 | 1 |
| $T_c$ | 0.2 | 1.98 | 1.98 | 0.7 | 0.7 |
| $T_r$ | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| $A_e$ | 0.0006 | 0.0006 | 0.0006 | 0.0006 | 0.0006 |
| $B_e$ | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 |

Table A.6: Simulation details of AVRs

# APPENDIX B

## INITIAL POWER FLOW SOLUTION

This section gives the initialization of the network variables based on the steady-state solution of system equations.

| Bus number | Voltage magnitude (p.u.) | Voltage phase angle (rad) | Active power (p.u.) | Reactive power (p.u.) |
|---|---|---|---|---|
| 1 | 1.06 | 0 | 2.3258 | -0.14978 |
| 2 | 1.045 | -0.0871 | 0.183 | 0.36124 |
| 3 | 1.01 | -0.22267 | -0.942 | 0.08373 |
| 4 | 1.012 | -0.1785 | -0.478 | -0.04 |
| 5 | 1.016 | -0.15273 | -0.076 | -0.016 |
| 6 | 1.07 | -0.25161 | -0.112 | 0.1501 |
| 7 | 1.0493 | -0.23091 | 0 | 0 |
| 8 | 1.09 | -0.23091 | 0 | 0.25163 |
| 9 | 1.0328 | -0.25853 | -0.295 | -0.166 |
| 10 | 1.0318 | -0.26223 | -0.09 | -0.058 |
| 11 | 1.0471 | -0.25897 | -0.035 | -0.018 |
| 12 | 1.0534 | -0.26645 | -0.061 | -0.016 |
| 13 | 1.047 | -0.2671 | -0.135 | -0.058 |
| 14 | 1.0207 | -0.28018 | -0.149 | -0.05 |

Table B.1: Initialization of bus variables

| Symbolic variable | Gen1 | Gen2 | Gen3 | Gen4 | Gen5 |
|---|---|---|---|---|---|
| $\delta$ | 0.21704 | -0.22363 | 0.24427 | -0.2315 | -0.25217 |
| $\omega$ | 1 | 1 | 1 | 1 | 1 |
| $e'_q$ | 1.0515 | 1.0954 | 1.1676 | 1.3104 | 1.2708 |
| $e'_d$ | - | 0.00024 | 0.19109 | 0.00027 | 0.00023 |
| $e''_q$ | 1.0476 | 1.0687 | 1.1119 | 1.2008 | 1.171 |
| $e''_d$ | 0.08692 | 0.00037 | 0.29742 | 0.00059 | 0.00052 |

Table B.2: Initialization of state variables