Georgia Institute of Technology

ECE 6323: Power System Protection

Term Project

Intelligent Merging Unit

Student
Cristian Gomez Peces

Instructor
A.P. Meliopoulos

April 24, 2020

Contents

List of Figures

List of Tables

1 Introduction

Merging units are proven to be very powerful tools to ensure the security of a power system. Since it is essential for a relay to work with valid measurements, merging units are appropriate to acquire, process and validate data obtained from the system [1]. In this way, the relay is provided with reliable data to compute its security functions.

1 .1 Case Study

The model of instrumentation channel is one of the multiple ways we have to perform a dynamic state estimation given the voltage measurements. One of the advantages of this estimation is that it ultimately has the capability to correct errors introduced by the instrumentation channel, since the measurement model is known, and we can apply the state estimation process for any instrumentation channel independently from how many channels exist. The physical high-level architecture of the system is shown in Figure 1.

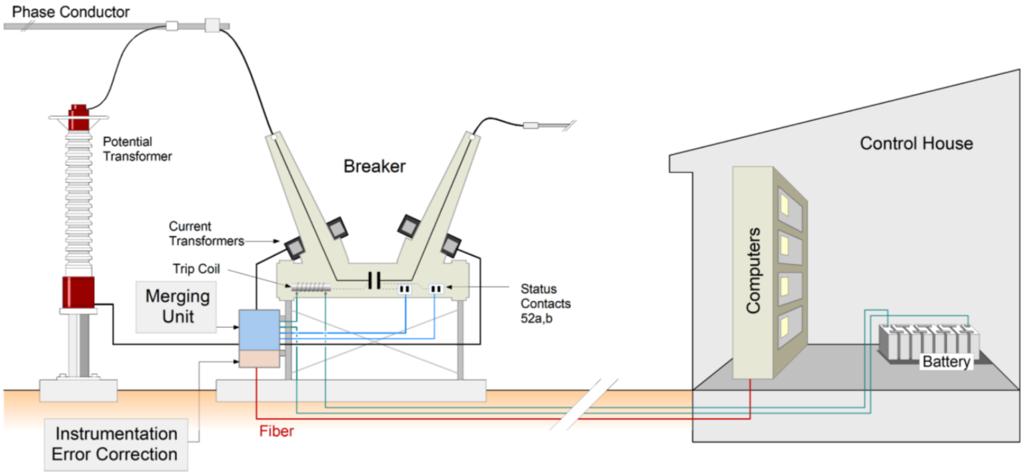


Figure 1: High-Level Architecture of the System

As can be observed, the system of study counts on a potential transformer to capture the measurements that will be processed by the merging unit. These data will tell whether any action of the relay is required. The instrumentation subsystem compacts voltage and current transformers, copper wires and the merging unit/relay input circuit. These elements are also sources that may introduce errors in the measurements that can distort waveforms of high-magnitude currents and voltages [2]. These errors may lead to malfunctions of the relay or even cause its mal-operation. In order to avoid this situation, it is essential to identify when errors become greater by establishing a certain confidence level. This project is focused on the current instrumentation channel and aims to detect faults by estimating the state and evaluating the confidence level of the measurements.

1 .2 Objectives

The project addresses 6 main tasks to perform correctly the dynamic state estimation:

1. Write MATLAB code to read files in COMTRADE format. This function will include a feedback tool to display the read data.
2. Develop the mathematical model of the instrumentation channel, that is, define the variables that completely describe the operating point of the circuit.
3. Deduce the equations of all measurements, actual, derived, virtual and pseudo as a function of the state of the system.
4. Develop the algorithm to solve the dynamic state estimation problem, given the state and confidence level as output.
5. Create a program that encapsules the read of data, state estimation, confidence level computation and display selected results. This will be implemented as a Graphical User Interphase (GUI).

Section 2 introduces the problem formulation. The model of the instrumentation channels is presented as well as the methodology to perform the dynamic state estimation. Section 3 discusses more in depth the methodology applied to get the measures and mathematical elements that are used to calculate the state estimation, in addition to some of the assumptions made for the model and algorithm. In the same section, multiple development perspectives of the project are presented. Here it is explained how the GUI is designed and the folders structure is also introduced. Section 4 shows the results gathered from the state estimation by studying two different fault events. Finally, section 5 brings together all the comments and conclusions from previous sections. The 2 appendices included at the end of the document include the code developed for the GUI and the dynamic state estimation itself, respectively.

2 Problem Formulation

The ultimate goal is to detect whether a fault has taken place in the system or not. In order to do that, we measure how the system is performing by computing a dynamic state estimation. Every single point in time will have an associated confidence level to that estimation. In normal operation conditions, this confidence should be equal to 100%. However, if the system experiences a fault, the confidence level will drop down, and the merging unit is required to communicate with the relay to take actions when it surpasses the threshold of the significance level.

Before we compute the estimation, it is required to have some knowledge of the model of the instrumentation channels (that is, the components that constitute the circuit of the channel), since its dynamic behavior will define the dynamic equations for the state estimation.

2 .1 Model

Figure 2 shows the equivalent circuit of the current instrumentation channel:

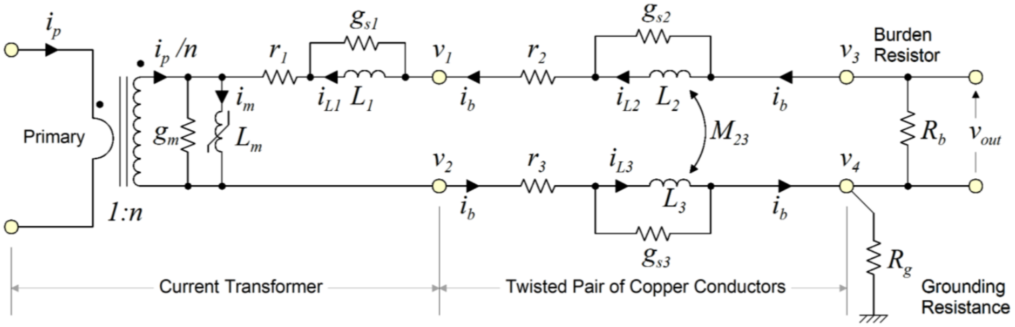


Figure 2: Equivalent circuit of the current instrumentation channel

Every measurement can be written in terms of the state variables. The vector that we define for the state variables is shown in Eq. 1 below.

$$x(t) = [v_1(t)\ v_2(t)\ v_3(t)\ v_4(t)\ i_P(t)\ i_{L_1}(t)\ i_{L_2}(t)\ i_{L_3}(t)\ i_m(t)\ \lambda(t)] \tag{1}$$

Where $\lambda(t)$ is the flux leakage of the current transformer. The value of the elements that compound our model are shown in Table 2 .1. These parameters will be essential when determining measurements from state variables.

Parameter	Value	Parameter	Value
n	400	$R_b = 1/g_b$	0.1 Ω
M_{23}	287.0 μH	λ_0	0.1876 Wb
g_{s1}	1.9635 S	i_0	6.09109 A
g_{s2}	0.1497 S	L_0	2.36 μH
g_{s3}	0.1497 S	L_1	26.526 μH
r_1	0.005 Ω	L_2	348.0 μH
r_2	0.4469 Ω	L_3	348.0 μH
r_3	0.4469 Ω	g_m	0.001 S

Table 1: Value of parameters

We can divide the measurements into 4 groups: Actual, Virtual, Derived and Pseudo measurements. Equations 2-23 that follow below show how to compute them in terms of the state variables.

Actual measurements (a total of 1 equation):

$$V_{out}(t) = v_3(t) - v_4(t) \tag{2}$$

Virtual measurements (a total of 10 equations):
KCL at node 0 yields:

$$0 = -g_m e(t) - i_m(t) + \frac{1}{n} i_p(t) + i_{L_1}(t) + g_{s1} L_1 \frac{\mathrm{d}i_{L_1}(t)}{\mathrm{d}t} \tag{3}$$

KCL at node 1 yields:

$$0 = g_m e(t) + i_m(t) - \frac{1}{n} i_p(t) - i_{L_2}(t) - g_{s2}\left(L_2 \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_3}(t)}{\mathrm{d}t}\right) \tag{4}$$

KCL at node 2 yields:

$$0 = -g_m e(t) - i_m(t) + \frac{1}{n} i_p(t) + i_{L_3}(t) + g_{s3}\left(L_3 \frac{\mathrm{d}i_{L_3}(t)}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}t}\right) \tag{5}$$

KVL loop: node 1 to transformer to node 2, yields:

$$0 = -v_1(t) + v_2(t) + e(t) + L_1 \frac{\mathrm{d}i_{L_1}(t)}{\mathrm{d}t} + r_1\left(g_m e(t) + i_m(t) - \frac{1}{n} i_p(t)\right) \tag{6}$$

KVL loop: node 3 to node 1, yields:

$$\begin{aligned} 0 = &-v_3(t) + v_1(t) + r_2(i_{L_2}(t) + g_{s_2}\left(L_2 \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}dt} - M_{23} \frac{\mathrm{d}i_{L_3}(t)}{\mathrm{d}t}\right) + \\ &+ L_2 \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_3}(t)}{\mathrm{d}t} \end{aligned} \tag{7}$$

KVL loop: node 2 to node 4, yields:

$$\begin{aligned} 0 = &-v_2(t) + v_4(t) + r_3\left(i_{L_3}(t) + g_{s_3}(L_3 \frac{\mathrm{d}i_{L_3}(t)}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}t}\right) + \\ &+ L_3 \frac{\mathrm{d}i_{L3}(t)}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}t} \end{aligned} \tag{8}$$

KCL at node 3 yields:

$$0 = i_{L_2}(t) + g_{s2}\left(L_2 \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_3}(t)}{\mathrm{d}t}\right) + g_b(v_3(t) - v_4(t)) \tag{9}$$

KCL at node 4 yields:

$$0 = -i_{L_3}(t) - g_{s3}\left(L_3 \frac{\mathrm{d}i_{L_3}(t)}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}t}\right) + g_b(v_4(t) - v_3(t)) \tag{10}$$

Transformer magnetizing leg yields:

$$0 = e(t) - \frac{\mathrm{d}\lambda(t)}{\mathrm{d}t} \tag{11}$$

And also, if we take into account that:

$$0 = i_m(t) - i_0 \left|\frac{\lambda(t)}{\lambda_0}\right|^n - \frac{1}{L_0}\lambda(t)$$

And use as an example n=11, the above (odd n) simplifies to:

$$0 = i_m(t) - i_0 \left(\frac{\lambda(t)}{\lambda_0}\right)^n - \frac{1}{L_0}\lambda(t) \tag{12}$$

Derived measurements (a total of 5 equations):

$$i_b^m(t) = -g_b(v_3(t) - v_4(t)) \tag{13}$$

$$i_b^m(t) = i_{L_1}(t) + g_{s1}L_1\frac{\mathrm{d}i_{L_1}}{\mathrm{d}t} \tag{14}$$

$$i_b^m(t) = i_{L_2}(t) + g_{s2}\left(L_2\frac{\mathrm{d}i_{L_2}}{\mathrm{d}t} - M_{23}\frac{\mathrm{d}i_{L_3}}{\mathrm{d}t}\right) \tag{15}$$

$$i_b^m(t) = i_{L_3}(t) + g_{s3}\left(L_3\frac{\mathrm{d}i_{L_3}}{\mathrm{d}t} - M_{23}\frac{\mathrm{d}i_{L_2}}{\mathrm{d}t}\right) \tag{16}$$

$$i_b^m(t) = g_m e(t) + i_m(t) - \frac{1}{n}i_p(t) \tag{17}$$

Pseudo measurement, node 4 is grounded (a total of 1 equation):

$$0^m = v_4(t) \tag{18}$$

So we have a total of 17 measurements and 11 states. However, we can reduce the order of Eq. 13 by quadratizing it and obtaining multiple second-order equations as follows:

$$0 = y_1(t) - \left(\frac{\lambda(t)}{\lambda_0}\right)^2 \tag{19}$$

$$0 = y_2(t) - (y_1(t))^2 \tag{20}$$

$$0 = y_3(t) - (y_2(t))^2 \tag{21}$$

$$0 = y_4(t) - y_1(t) \cdot y_3(t) \tag{22}$$

$$0 = i_m(t) - i_0 \left(\frac{\lambda(t)}{\lambda_0}\right) y_4(t) - \frac{1}{L_0}\lambda(t) \tag{23}$$

So finally, our measurement model yields to 21 equations and 15 states:

$$\begin{aligned} x(t) = [&v_1(t)\ v_2(t)\ v_3(t)\ v_4(t)\ \lambda(t) \\ &y_1(t)\ y_2(t)\ y_3(t)\ y_4(t) \\ &i_p(t)\ i_m(t)\ i_{L_1}(t)\ i_{L_2}(t)\ i_{L_3}(t)] \end{aligned} \tag{24}$$

The error standard deviation are determined by the following convention. The actual measurements have 0.05%, virtual have 0.005%, derived have 0.05% and pseudo have 10%. Taking equations 2-23 into account, we compact the measurement functions in terms of the state in Table 2 .1.

# Eq.	Measurement Model derived	σ
1	$v_{out}(t) = v_3(t) - v_4(t)$	0.005V
2	$0 = -g_m e(t) - i_m(t) + \frac{1}{n} i_p(t) + i_{L_1}(t) + g_{s1} L_1 \frac{\mathrm{d}i_{L_1}(t)}{\mathrm{d}t}$	0.005A
3	$0 = g_m e(t) + i_m(t) - \frac{1}{n} i_p(t) - i_{L_2}(t) - g_{s2} \left(L_2 \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_3}(t)}{\mathrm{d}t} \right)$	0.005A
4	$0 = -g_m e(t) - i_m(t) + \frac{1}{n} i_p(t) + i_{L_3}(t) + g_{s3} \left(L_3 \frac{\mathrm{d}i_{L_3}(t)}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}t} \right)$	0.005A
5	$0 = -v_1(t) + v_2(t) + e(t) + L_1 \frac{\mathrm{d}i_{L_1}(t)}{\mathrm{d}t} + r_1 \left(g_m e(t) + i_m(t) - \frac{1}{n} i_p(t) \right)$	0.0005V
6	$0 = -v_3(t) + v_1(t) + r_2(i_{L_2}(t) + g_{s_2} \left(L_2 \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}dt} - M_{23} \frac{\mathrm{d}i_{L_3}(t)}{\mathrm{d}t} \right) + L_2 \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_3}(t)}{\mathrm{d}t}$	0.0005V
7	$0 = -v_2(t) + v_4(t) + r_3 \left(i_{L_3}(t) + g_{s_3}(L_3 \frac{\mathrm{d}i_{L_3}(t)}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}t} \right) + L_3 \frac{\mathrm{d}i_{L3}(t)}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}t}$	0.0005V
8	$0 = i_{L_2}(t) + g_{s2} \left(L_2 \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_3}(t)}{\mathrm{d}t} \right) + g_b(v_3(t) - v_4(t))$	0.005A
9	$0 = -i_{L_3}(t) - g_{s3} \left(L_3 \frac{\mathrm{d}i_{L_3}(t)}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_2}(t)}{\mathrm{d}t} \right) + g_b(v_4(t) - v_3(t))$	0.005A
10	$0 = e(t) - \frac{\mathrm{d}\lambda(t)}{\mathrm{d}t}$	0.0005V
11	$0 = y_1(t) - \left(\frac{\lambda(t)}{\lambda_0} \right)^2$	0.00005
12	$0 = y_2(t) - (y_1(t))^2$	0.00005
13	$0 = y_4(t) - y_1(t) \cdot y_3(t)$	0.00005
14	$0 = i_m(t) - i_0 \left(\frac{\lambda(t)}{\lambda_0} \right) y_4(t) - \frac{1}{L_0} \lambda(t)$	0.00005
15	$i_b^m(t) = i_{L_2}(t) + g_{s2} \left(L_2 \frac{\mathrm{d}i_{L_2}}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_3}}{\mathrm{d}t} \right)$	0.0003A
16	$i_b^m(t) = -g_b(v_3(t) - v_4(t))$	0.05A
17	$i_b^m(t) = i_{L_1}(t) + g_{s1} L_1 \frac{\mathrm{d}i_{L_1}}{\mathrm{d}t}$	0.05A
18	$i_b^m(t) = i_{L_2}(t) + g_{s2} \left(L_2 \frac{\mathrm{d}i_{L_2}}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_3}}{\mathrm{d}t} \right)$	0.05A
19	$i_b^m(t) = i_{L_3}(t) + g_{s3} \left(L_3 \frac{\mathrm{d}i_{L_3}}{\mathrm{d}t} - M_{23} \frac{\mathrm{d}i_{L_2}}{\mathrm{d}t} \right)$	0.05A
20	$i_b^m(t) = g_m e(t) + i_m(t) - \frac{1}{n} i_p(t)$	0.05A
21	$0 = v_4(t)$	1V

Table 2: Measurement equations as function of the state

2 .2 Dynamic State Estimation

The state estimation is solved as a minimization problem. Let z_i be the measurement i that corresponds to the measurement model equation $h(x)$, which depends on the state. We use the Wighten Least Squares method to estimate the real measure by finding the state that minimizes the sum of residuals. That is:

$$\min J = \sum_{i=1}^{n} \left(\frac{z_i - h_i(x)}{\sigma_i} \right)^2 = \sum_{i=1}^{n} s_i^2 = \eta^T W \eta = \zeta \tag{25}$$

Where sigma is the standard deviation associated to that measurement, $\eta = z - h(x)$ and W=diag$\left(..., \frac{1}{\sigma_i}, ...\right)$. The Gauss-Newton iterative method provides a way to find the state that minimizes J by performing an iterative algorithm. Let H be jacobian matrix of the measurement equations $\left(H = \frac{\partial h(x)}{\partial x}\right)$ and $\hat{x}^v$ the iteration of the best estimate of state vector. We can compute the following calculation until the difference between $\hat{x}^v$ and $\hat{x}^{v+1}$ is very small (that is, when the optimal reached):

$$\hat{x}^{v+1} = \hat{x}^v + \left(H^T W H\right) H^T W (z - h(\hat{x}^v)) \tag{26}$$

Once optimal state is reached, the confidence level can be computed by summing up all the residues from the 21 measurements. The sum, noted as ζ, will follow a Chi-Square probability distribution function with $\nu = 21 - 15 = 6$ degrees of freedom (number of total measurements minus number of state variables). Hence, we can perform a Chi-Square test and get the confidence level β as shown in Eq. 27:

$$\beta = 1 - \Pr[\zeta, \nu] \tag{27}$$

Once we get β, we can set a level of significante to perform the test and reject the null hypothesis of all measurements correct when β drops to low percentages, i.e. when a fault occurs. In this project, it was selected a significance level of 10%, that is, we acknowledge a fault has occur when the confidence level is under 90%.

3 Methodology

From section 2.2 we have seen that we need 5 elements to compute the dynamic state estimation: $\hat{x}^v$, H, W, z, and $h(\hat{x}^v)$.

For the first iteration we assume a state vector of zeros. The Gauss-Newton method will lead to the correct estimation after several iterations. We also assume a null previous state as there is no way to calculate the first state without previous state. W is calculated with the standard deviations discussed in section 2.1 and the following measurements vector $z(t)$ is used for each iteration i (deduced also from the 21 equations in section 2.1):

$$z(t) = [v_{out}(t)\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ i_b^m(t)\ i_b^m(t)\ i_b^m(t)\ i_b^m(t)\ i_b^m(t)\ 0\ 0\ 0\ 0\ 0\ 0] \tag{28}$$

Note that v_{out} is the data to be read from the files in COMTRADE format, and $i_b^m(t)$ is derived from v_{out} as $i_b^m(t) = -v_{out}/R_b$, according the instrumentation channel equivalent circuit in Figure 2 and the Ohm's law. The order in which $z(t)$ elements are sorted follows the same criteria as stated in Table 2 .1 in the matrix H computation section.

The computation of the H matrix and measurement vector in terms of the state $h(\hat{x}^v)$ requires a longer process and therefore it is discussed in a separate section below.

3 .1 Computation of h(x) and Jacobian H

Some of the KVL equations applied at nodes that involved inductances voltages have derivative elements of the inductance current. This is an important issue since it is not possible to compute the jacobian for those equations, since we have no knowledge about the function of these currents. One method to address this problem is by computing the numerical integral for those equations that have derivative terms. We can use both the trapezoidal or the quadratic numerical integration. In this case, the trapezoidal integration was chosen for simplicity. The goal here is to get rid of the derivative terms and get a derivable expression for those equations. The trapezoidal integration can be computed using the following rules:

$$x(\tau) = x(t-h)\left(\frac{t-\tau}{h}\right) + x(t)\left(\frac{\tau-t+h}{h}\right) \Rightarrow \begin{cases} \int_{t-h}^{t}\left(\frac{dx(\tau)}{d\tau}\right)d\tau = (x(t)-x(t-h)) \\ \int_{t-h}^{t} x(\tau)d\tau = (x(t)+x(t-h)) \end{cases}$$

In which h is the time between samples. We only need to apply the integration for those equations that include a derivative term. That is, equations 3 to 11 and from 14 to 16 will change. In addition, since the left hand side will also be integrated, in those equation in which i_b^m appears a factor of $2/h$ should multiply the right hand side, followed by the subtraction of the term $i_b^m(t-h)$. These final measure model equations h(x) are shown on Table 2 .1 :

# Eq.	Final Measurement Model	σ
1	$v_{out}(t) = v_3(t) - v_4(t)$	0.005V
2	$0 = -g_m\frac{h}{2}(e(t-h)+e(t)) - \frac{h}{2}(i_m(t-h)+i_m(t)) + \frac{h}{2n}(i_p(t-h)+i_p(t)) +$ $+\frac{h}{2}(i_{L_1}(t-h)+i_{L_1}(t)) + g_{s1}L_1(i_{L_1}(t) - i_{L_1}(t-h)$	0.005A
3	$0 = g_m\frac{h}{2}(e(t-h)+e(t)) + \frac{h}{2}(i_m(t-h)+i_m(t)) - \frac{h}{2n}(i_p(t-h)+i_p(t))$ $-\frac{h}{2}(i_{L_2}(t-h)+i_{L_2}(t)) - g_{s2}L_2(i_{L_2}(t) - i_{L_2}(t-h) + g_{s2}M_{23}(i_{L_3}(t) - i_{L_3}(t-h)$	0.005A
4	$0 = -g_m\frac{h}{2}(e(t-h)+e(t)) - \frac{h}{2}(i_m(t-h)+i_m(t)) + g_{s3}L_3(i_{L_3}(t) - i_{L_3}(t-h)$ $+\frac{h}{2n}(i_p(t-h)+i_p(t)) + \frac{h}{2}(i_{L_3}(t-h)+i_{L_3}(t)) - g_{s3}M_{23}(i_{L_2}(t-h)+i_{L_2}(t))$	0.005A
5	$0 = -\frac{h}{2}(v_1(t-h)+v_1(t)) + \frac{h}{2}(v_2(t-h)+v_2(t)) + \frac{h}{2}(e(t-h)+e(t)) + L_1(i_{L_1}(t)$ $-i_{L_1}(t-h)) + r_1 g_m\frac{h}{2}(e(t-h)+e(t)) + r_1\frac{h}{2}(i_m(t)+i_m(t-h)) - \frac{r_1 h}{2n}(i_p(t-h)+i_p(t))$	0.0005V

# Eq.	Final Measurement Model	σ
6	$0=-\frac{h}{2}(v_3(t-h)+v_3(t))+\frac{h}{2}(v_1(t-h)+v_1(t))+\frac{r_1h}{2}(i_{L_2}(t)$ $+r_2g_{s2}L_2(i_{L_2}(t)-i_{L_2}(t-h))-r_2g_{s2}M_{23}(i_{L_3}(t)$ $+i_{L_2}(t-h))-i_{L_3}(t-h))+L_2(i_{L_2}(t)-i_{L_2}(t-h))-M_{23}(i_{L_3}(t)-i_{L_3}(t-h))$	0.0005V
7	$0=-\frac{h}{2}(v_2(t-h)+v_2(t))+\frac{h}{2}(v_4(t-h)+v_4(t))+\frac{r_3h}{2}(i_{L_3}(t)+i_{L_3}(t-h))$ $+r_3g_{s3}L_3(i_{L_3}(t)-i_{L_3}(t-h))-r_3g_{s3}M_{23}(i_{L_2}(t)-i_{L_2}(t-h))$ $+L_3(i_{L_3}(t)-i_{L_3}(t-h))-M_{23}(i_{L_2}(t)-i_{L_2}(t-h))$	0.0005V
8	$0=\frac{h}{2}(i_{L_2}(t)+i_{L_2}(t-h))+g_{s2}L_2(i_{L_2}(t)-i_{L_2}(t-h))$ $-g_{s2}M_{23}(i_{L_3}(t)-i_{L_3}(t-h))+\frac{g_bh}{2}(v_3(t-h)+v_3(t))-\frac{g_bh}{2}(v_4(t-h)+v_4(t))$	0.005A
9	$0=-\frac{h}{2}(i_{L_3}(t)+i_{L_3}(t-h))-g_{s3}L_3(i_{L_3}(t)-i_{L_3}(t-h))$ $+g_{s3}M_{23}(i_{L_2}(t)-i_{L_2}(t-h))+\frac{g_bh}{2}(v_4(t-h)+v_4(t))-\frac{g_bh}{2}(v_3(t-h)+v_3(t))$	0.005A
10	$0=\frac{h}{2}(e(t-h)+e(t))-(\lambda(t)-\lambda(t-h))$	0.0005V
11	$0=y_1(t)-\left(\frac{\lambda(t)}{\lambda_0}\right)^2$	0.00005
12	$0=y_2(t)-(y_1(t))^2$	0.00005
13	$0=y_4(t)-y_1(t)\cdot y_3(t)$	0.00005
14	$0=i_m(t)-i_0\left(\frac{\lambda(t)}{\lambda_0}\right)y_4(t)-\frac{1}{L_0}\lambda(t)$	0.00005
15	$i_b^m(t)=\frac{2}{h}(\frac{h}{2}(i_{L_1}(t)+i_{L_1}(t-h))+g_{s1}L_1(i_{L_1}(t)-i_{L_1}(t-h)))-i_b^m(t-h)$	0.0003A
16	$i_b^m(t)=-g_b(v_3(t)-v_4(t))$	0.05A
17	$i_b^m(t)=i_{L_1}(t)+g_{s1}L_1\frac{\mathrm{d}i_{L_1}}{\mathrm{d}t}$	0.05A
18	$i_b^m(t)=\frac{2}{h}(\frac{h}{2}(i_{L_2}(t)+i_{L_2}(t-h))+g_{s2}L_2(i_{L_2}(t)-i_{L_2}(t-h))$ $-g_{s2}M_{23}(i_{L_3}(t)-i_{L_3}(t-h)))-i_b^m(t-h)$	0.05A
19	$i_b^m(t)=\frac{2}{h}(-\frac{h}{2}(i_{L_3}(t)+i_{L_3}(t-h))-g_{s3}L_3(i_{L_3}(t)-i_{L_3}(t-h))$ $+g_{s3}M_{23}(i_{L_2}(t)-i_{L_2}(t-h)))-i_b^m(t-h)$	0.05A
20	$i_b^m(t)=g_me(t)+i_m(t)-\frac{1}{n}i_p(t)$	0.05A
21	$0=v_4(t)$	1V

Table 3: Final measurement equations as function of the state

For the computation of the jacobian H, we need to calculate de partial derivative with respect each of the equations shown above. Note that the states $(t-h)$ remain are considered constant when deriving. The matrix H will have the following shape:

$$H = \begin{bmatrix} \frac{dh_1(x)}{dx_1} & \cdots & \frac{dh_1(x)}{dx_{15}} \\ \vdots & \ddots & \vdots \\ \frac{dh_{21}(x)}{dx_1} & \cdots & \frac{dh_{21}(x)}{dx_{15}} \end{bmatrix} =$$

$$= \begin{bmatrix}
0 & 0 & h_{3,1} & h_{4,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & h_{5,2} & 0 & 0 & 0 & 0 & 0 & h_{11,2} & h_{12,2} & h_{13,2} & 0 & 0 \\
0 & 0 & 0 & 0 & h_{5,3} & 0 & 0 & 0 & 0 & 0 & h_{11,3} & h_{12,3} & 0 & h_{14,3} & h_{15,3} \\
0 & 0 & 0 & 0 & h_{5,4} & 0 & 0 & 0 & 0 & 0 & h_{11,4} & h_{12,4} & 0 & h_{14,4} & h_{15,4} \\
h_{1,5} & h_{2,5} & 0 & 0 & h_{5,5} & 0 & 0 & 0 & 0 & 0 & h_{11,5} & h_{12,5} & h_{13,5} & 0 & 0 \\
h_{1,6} & 0 & h_{3,6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{14,6} & h_{15,6} \\
0 & h_{2,7} & 0 & h_{4,7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{14,7} & h_{15,7} \\
0 & 0 & h_{3,8} & h_{4,8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{14,8} & h_{15,8} \\
0 & 0 & h_{3,9} & h_{4,9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{14,9} & h_{15,9} \\
0 & 0 & 0 & 0 & h_{5,10} & h_{6,10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & h_{6,11} & h_{7,11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & h_{7,12} & h_{8,12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{8,13} & h_{9,13} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & h_{7,14} & 0 & h_{9,14} & h_{10,14} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & h_{6,15} & 0 & 0 & 0 & h_{10,15} & 0 & h_{12,15} & 0 & 0 & 0 \\
0 & 0 & h_{3,16} & h_{4,16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{13,17} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{14,18} & h_{15,18} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{14,19} & h_{15,19} \\
0 & 0 & 0 & 0 & h_{5,20} & 0 & 0 & 0 & 0 & 0 & h_{11,20} & h_{12,20} & 0 & 0 & 0 \\
0 & 0 & 0 & h_{4,21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Each element h_{ij} computation follows below:

$$\begin{aligned}
h_{3,1} &= 1 \\
h_{4,1} &= -1 \\
h_{5,2} &= -g_m \cdot h/2 \\
h_{11,2} &= \frac{h}{2n} \\
h_{12,2} &= -h/2 \\
h_{13,2} &= g_{s1} \cdot L_1 + h/2 \\
h_{5,3} &= g_m \cdot h/2 \\
h_{11,3} &= -\frac{h}{2n} \\
h_{12,3} &= h/2 \\
h_{14,3} &= -g_{s2} \cdot L_2 - h/2 \\
h_{15,3} &= g_{s2} \cdot M_{23}
\end{aligned}$$

$$\begin{aligned}
h_{5,4} &= -g_m \cdot h/2 \\
h_{11,4} &= \frac{h}{2n} \\
h_{12,4} &= -h/2 \\
h_{14,4} &= -M_{23} \cdot g_{s3} \\
h_{15,4} &= g_{s3} \cdot L_3 + h/2 \\
h_{1,5} &= -h/2 \\
h_{2,5} &= h/2 \\
h_{5,5} &= h/2 + r_1 \cdot h/2 \cdot g_m \\
h_{11,5} &= -r_1 \cdot \frac{h}{2n} \\
h_{12,5} &= r_1 \cdot h/2 \\
h_{13,5} &= L_1
\end{aligned}$$

$$\begin{aligned}
h_{1,6} &= h/2 \\
h_{3,6} &= -h/2 \\
h_{14,6} &= r_2 \cdot (h/2 + g_{s2} \cdot L_2) + L_2 \\
h_{15,6} &= -r_2 \cdot g_{s2} \cdot M_{23} - M_{23} \\
h_{2,7} &= -h/2 \\
h_{4,7} &= h/2 \\
h_{14,7} &= -r_3 \cdot g_{s3} \cdot M_{23} - M_{23} \\
h_{15,7} &= r_3 \cdot (h/2 + g_{s3} \cdot L_3) + L_3 \\
h_{3,8} &= g_b \cdot h/2 \\
h_{4,8} &= -g_b \cdot h/2 \\
h_{14,8} &= h/2 + g_{s2} \cdot L_2 \\
h_{15,8} &= -g_{s2} \cdot M_{23} \\
h_{3,9} &= -h/2 \cdot g_b \\
h_{4,9} &= h/2 \cdot g_b \\
h_{14,9} &= g_{s3} \cdot M_{23} \\
h_{15,9} &= -h/2 - g_{s3} \cdot L_3 \\
h_{5,10} &= h/2 \\
h_{6,10} &= -1 \\
h_{6,11} &= -2/(\lambda_0^2) \cdot \lambda \\
h_{7,11} &= 1
\end{aligned}$$

$$\begin{aligned}
h_{7,12} &= -2 \cdot y1 \\
h_{8,12} &= 1 \\
h_{8,13} &= -2 \cdot y2 \\
h_{9,13} &= 1 \\
h_{7,14} &= -y3 \\
h_{9,14} &= -y1 \\
h_{10,14} &= 1 \\
h_{6,15} &= -i_0/\lambda_0 \cdot y4 - 1/L_0 \\
h_{10,15} &= -i_0/\lambda_0 \cdot \lambda \\
h_{12,15} &= 1 \\
h_{3,16} &= -g_b \\
h_{4,16} &= g_b \\
h_{13,17} &= (h/2 + g_{s1} \cdot L_1) \cdot 2/h \\
h_{14,18} &= h/2 + g_{s2} \cdot L_2 \cdot 2/h \\
h_{14,19} &= g_{s3} \cdot M_{23} \cdot 2/h \\
h_{15,19} &= (-h/2 - g_{s3} \cdot L_3) \cdot 2/h \\
h_{5,20} &= g_m \\
h_{11,20} &= -1/n \\
h_{12,20} &= 1 \\
h_{4,21} &= 1
\end{aligned}$$

3 .2 Description of Algorithm

Figure 3 shows the workflow design implemented in the algorithm. Once the input data is acquired, we enter into a loop for every sample of the measured output voltage. Until the last sample is reached, the Gauss-Newton method is applied over each sample point until two consecutive iterations provide very similar states, that is, the sum of absolute differences between x^v and x^{v+1} should be lower than a certain tolerance. This tolerance has been set to 0.01 in the code. When the final state is reached, we record it in addition to the confidence level associated to that operation point given the measures vector z and the measurement model $h(x)$ in terms of the estimated state. These 2 loops keep operating until all the states for each of the samples have been computed.

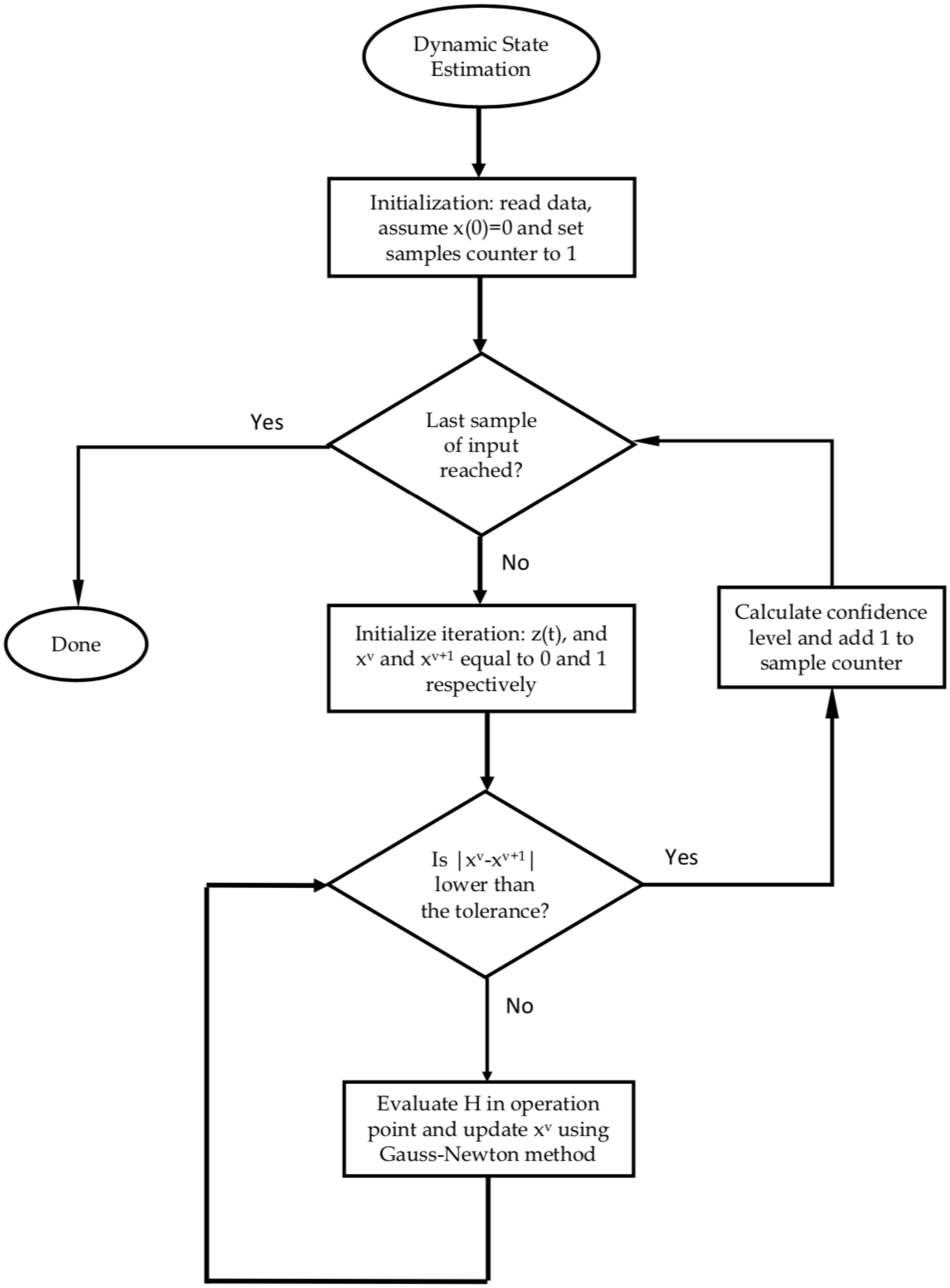

Dynamic State Estimation
Initialization: read data, assume x(0)=0 and set samples counter to 1
Last sample of input reached?
Yes
No
Done
Initialize iteration: z(t), and x^v and x^{v+1} equal to 0 and 1 respectively
Calculate confidence level and add 1 to sample counter
Is $|x^v-x^{v+1}|$ lower than the tolerance?
Yes
No
Evaluate H in operation point and update x^v using Gauss-Newton method

Figure 3: Algorithm Flowchart

3 .3 Development

This section discusses how the software was implemented to not only create the algorithm of the estimation but also how the GUI was designed, how the files are structured within the program and the time assigned to each task.

3 .3.1 Project plan

The project was completed in 5 weeks. Figure 4 shows the Gantt diagram followed to achieve the objective tasks.

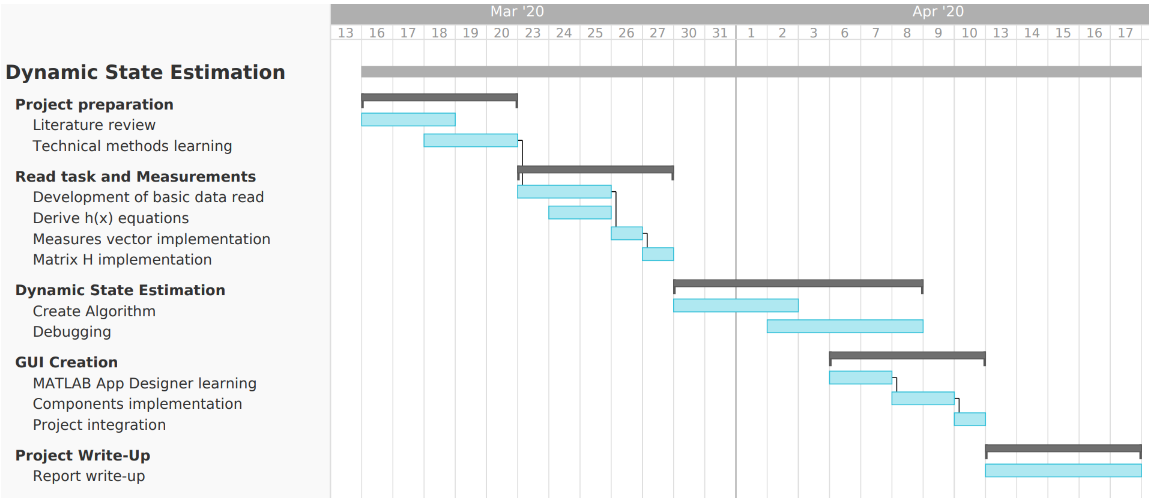


Figure 4: Project Plan

1. The first week was focused on the reading through the specifications, analyzing the methods and learning about the trapezoidal integration to use it in the measurement model equations.

2. During the second week, a very simple approach to read files in COMTRADE format was develop. In addition, all the equations that needed to be integrated were derived to compute H and the vector of measurements.

3. The concentration of the third was mainly on the dynamic state estimation algorithm and bugs debugging. Most of the bugs had been introduced during the derivation of the measurement equations and also within the loop of the algorithm.

4. The creation of GUI took place during the fourth week. The basic concepts of MATLAB App Designer were learnt and later on implemented to read and display data in COMTRADE format. The integration with the rest of the MATLAB files was successfully developed by the end of the week.

5. The fifth and last week was used to compact and write up all the steps done for this report.

3 .3.2 Directory Structure

The directory structure under which the program operates is shown on Figure 5. As can be observed, it follows a hierarchical structure in which the App Designer file is the main program and depends upon the output from master.m, which in turn also uses other tools. The arrows point towards the file that will use a feature from the file in question.

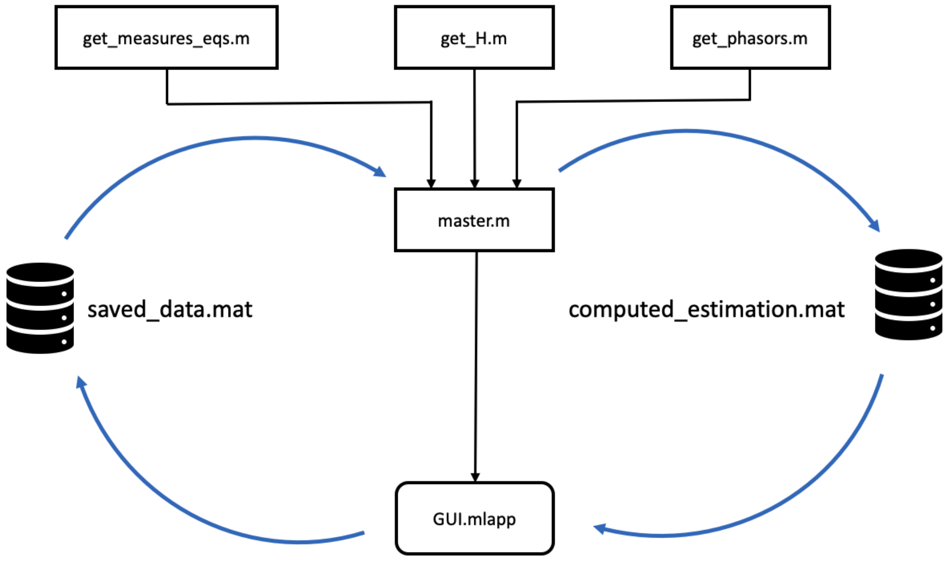


Figure 5: Directory Structure

Each of the files has a function:

- `get_measures_eqs.m` gets the vector of measures $h(x)$ that is used in the iterative process of the state estimation from the state vector that is passed as input.
- `get_H.m` gets the elements of the matrix H, given the current and previous state, which are passed as required arguments in addition to the instrumentation channel model parameters.
- `get_phasors.m` provides the phasor magnitude of a certain waveform. This is used under a flag in `master.m` in case we want our results in phasors.
- `master.m` is the file where the dynamic state estimation algorithm is coded. Hence it requires the output of the functions above mentioned and provides the estimation given an input previously read by the graphical user interphase. The output is stored in `computed_estimation.mat`.
- `GUI.mlapp` is the top-level file. It provides a visual to read COMTRADE files, store and display the data, communicate with `master.m` to perform the dynamic state estimation and get the final resutls from its output. That is, the confidence level computation and the decision making dialog to warn whether any relay action is required or not.
- `saved_data.mat` Stores the data read by the GUI and that is required by `master.m` for the state estimation. It contains all the input information: total samples, sampling rate, time scale, frequency and the input v_{out} array.

- `computed_estimation.mat` Stores the results obtained in the estimation. This file contains the resulting state variables, estimated measurements and confidence level along time.

3 .3.3 GUI Design

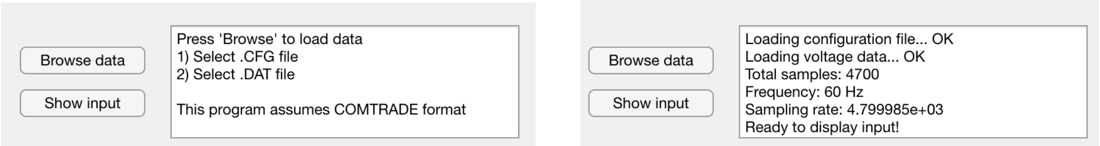
The GUI was designed to achieve several goals: allow the user to select the input data visually, show both the input and the computed state estimation, provide feedback in text frames, determine whether relay actions are required to the user, display a confidence level plot once the estimation is completed and allow the user to select with state variable to display as output. An overall view of the GUI is shown in Figure 10 at the end of this section. In order to implement the aforementioned features, the following components were added to the interphase. Their functions are explained below:

Browse buttons

The browse button allows the user to select the input data from a directory path. When the button is pushed, 2 windows will be opened sequentially to select the `.cfg` file in the first place and the `.dat` afterwards. Once the files have been selected, we read the data and store it in the `saved_data.mat`.

Feedback box

The feedback box shows information along this process to the user. Its content is initialized to give instructions about how to browse the files. Once the .cfg is loaded, an 'OK' is displayed and when all the read is completed, the frame provides essential information read from both files, such as number total of samples, frequency of the system, and the sampling rate. Figure 6 shows the initial display as well as its content when the data has been read.



```
Press 'Browse' to load data
1) Select .CFG file
2) Select .DAT file

This program assumes COMTRADE format
```

(a) Initial content

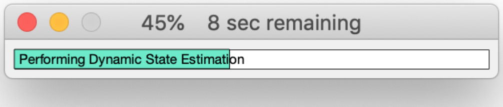
```
Loading configuration file... OK
Loading voltage data... OK
Total samples: 4700
Frequency: 60 Hz
Sampling rate: 4.799985e+03
Ready to display input!
```

(b) Feedback after read is successfully completed

Figure 6: Feedback Text Box

Compute Estimation button

The callback function of this button executes the master file. Since the computation can take relatively some time, a feedback progress bar was also implemented after pushing the button to provide the user with computation timing information. An example of this progress bar is shown in Figure 7. Both estimated time to complete the estimation and the progress made so far are displayed. When the estimation is completed, the graph of confidence level is updated with the



45% 8 sec remaining

Performing Dynamic State Estimation

Figure 7: Progress Bar feature

Select Options menu

This feature allows the user to select the estimated state variable to display. Since there are 15 state variables, there are 15 possible options, as can be observed in Figure 8.

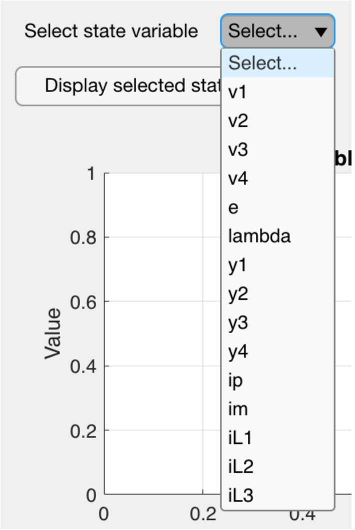
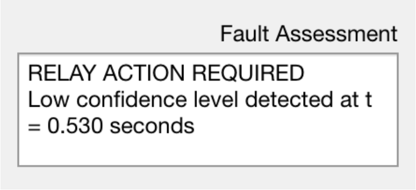


Figure 8: Select Options menu

Relay Actions text box

Its main function is to provide fault assessment and it is updated in Compute Estimation button's callback after the dynamic state estimation is computed. It alerts the user if a low confidence level was detected and relay actions are required:



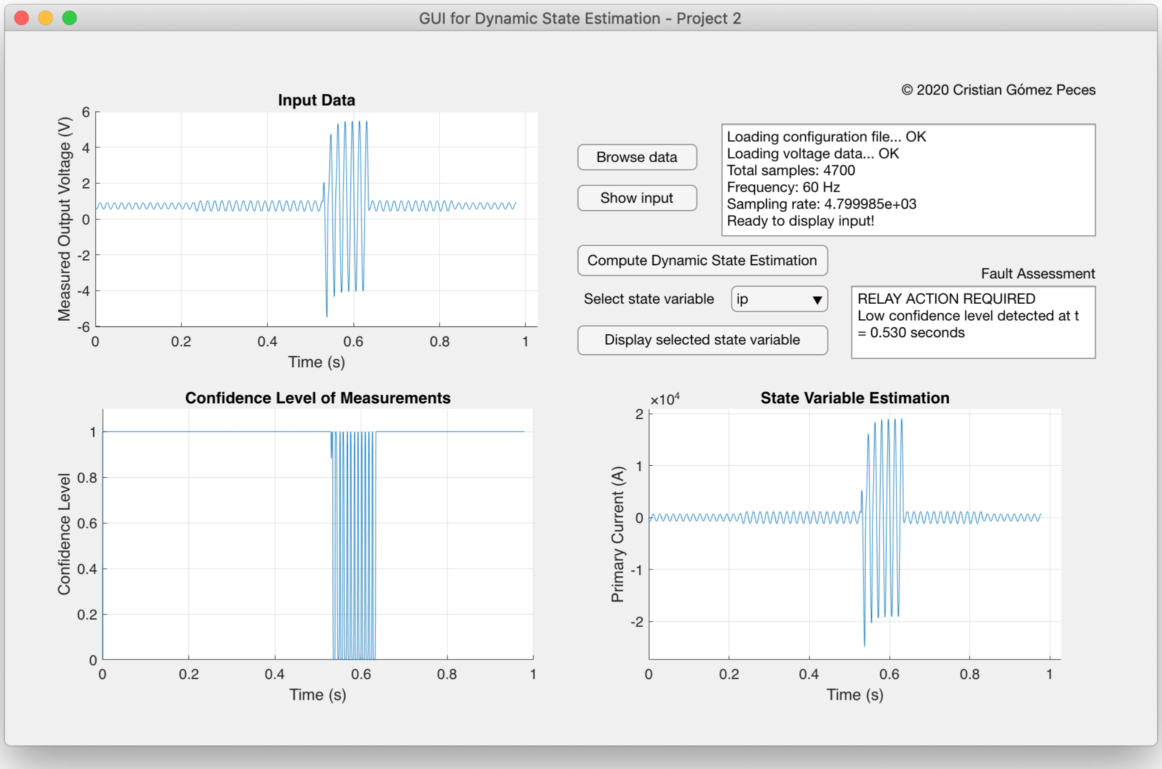
Fault Assessment

RELAY ACTION REQUIRED
Low confidence level detected at t = 0.530 seconds

Figure 9: Fault Assessment display

Plots

The GUI incorporates 3 different plot spaces. The chart on the top-left corner can display the output voltage acquired from the measurements. At the bottom-left corner, the confidence level is displayed after the dynamic state estimation is computed. The last plot on the bottom-right corner allows the user to display any of the state variables estimated along time. Figure 10 shows a general overview of the GUI.


GUI for Dynamic State Estimation - Project 2
© 2020 Cristian Gómez Peces
Input Data
Measured Output Voltage (V)
6
4
2
0
-2
-4
-6
0
0.2
0.4
0.6
0.8
1
Time (s)
Browse data
Show input
Compute Dynamic State Estimation
Select state variable
ip
Display selected state variable
Fault Assessment
Confidence Level of Measurements
Confidence Level
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
Time (s)
State Variable Estimation
×10^4
Primary Current (A)
2
1
0
-1
-2
0
0.2
0.4
0.6
0.8
1
Time (s)

```
Loading configuration file... OK
Loading voltage data... OK
Total samples: 4700
Frequency: 60 Hz
Sampling rate: 4.799985e+03
Ready to display input!
```

```
RELAY ACTION REQUIRED
Low confidence level detected at t
= 0.530 seconds
```

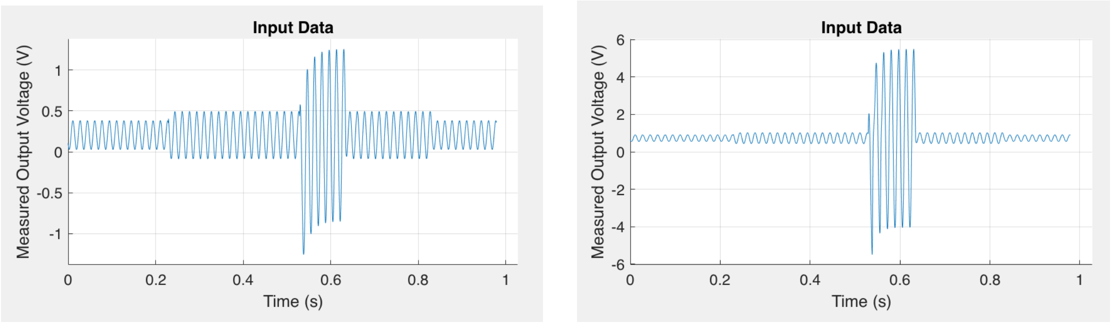
Figure 10: Graphical User Interphase

4 Results

In order to determine the effectiveness of this tool, 2 events that involved faults were tested. All the states along the samples available were determined, as well as the measurements residues and the confidence level. The primary current error and the flux linkage of the current transformer were also evaluated to test estimating capabilities, errors and get insights. Hereby the results for both events are presented.

4 .1 Input data

What we read as input data is the measurement of the output voltage v_{out}. Figure 11 shows the voltage read from the files in COMTRADE format.



(a) Event 1 output voltage

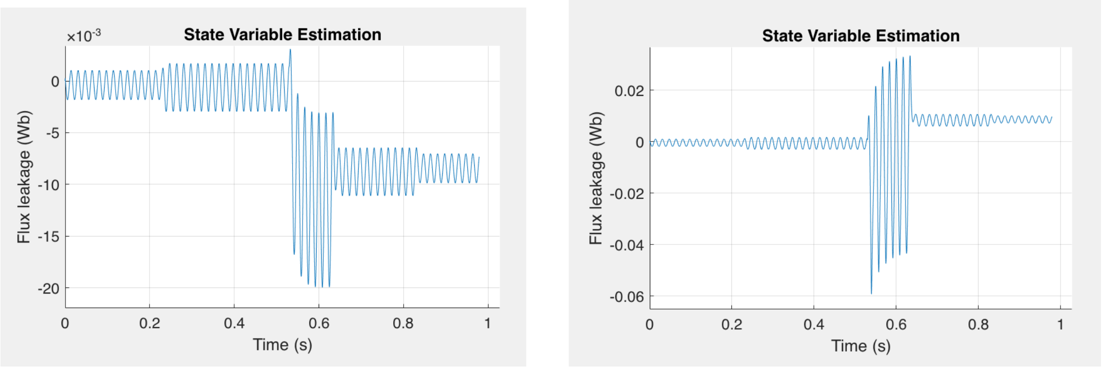
(b) Event 2 output voltage

Figure 11: Output voltages recorded for both events

We observe that both events present abrupt changes around half of a second from when the measures recording starts. There is also an increase in magnitude at around 0.2 seconds that does not necessarily imply a fault, since it is not as abrupt as the aforementioned. We also notice that the voltage recorded for event 2 is significantly higher, so this suggests that in case of a fault, currents will be way higher and confidence level can drop down more significantly.

4 .2 Flux linkage

The resulting estimated flux linkage for both events is shown on Figure 12.



(a) Event 1 flux linkage

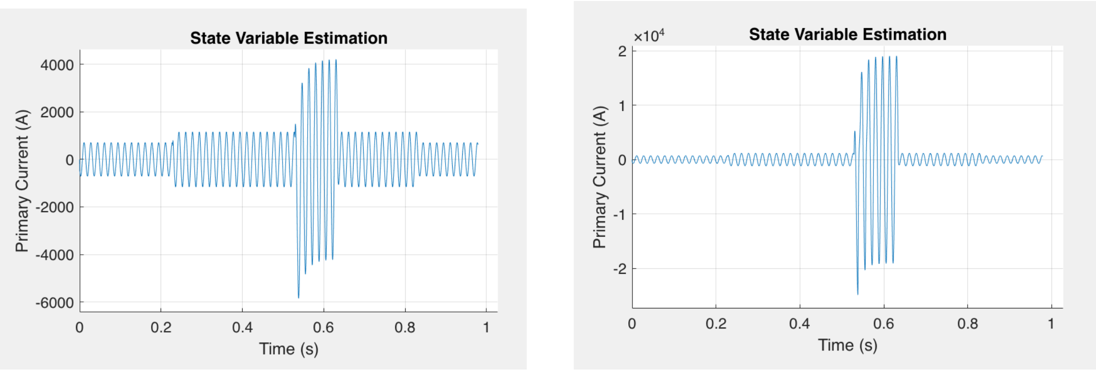
(b) Event 2 flux linkage

Figure 12: Estimated Flux linkage for both events

We observe that the flux experiments an increase with the initial increase of v_{out} at 0.2 seconds, but when the fault actually occurs, the reactions are different depending on the event we are dealing with: while the flux decreases in event 1, we see an increase of flux linkage in event 2.

4 .3 Estimated primary current through transformer

The estimated primary current that flows through the transformer is recorded in Figure 13 for both events.



(a) Event 1 Primary current

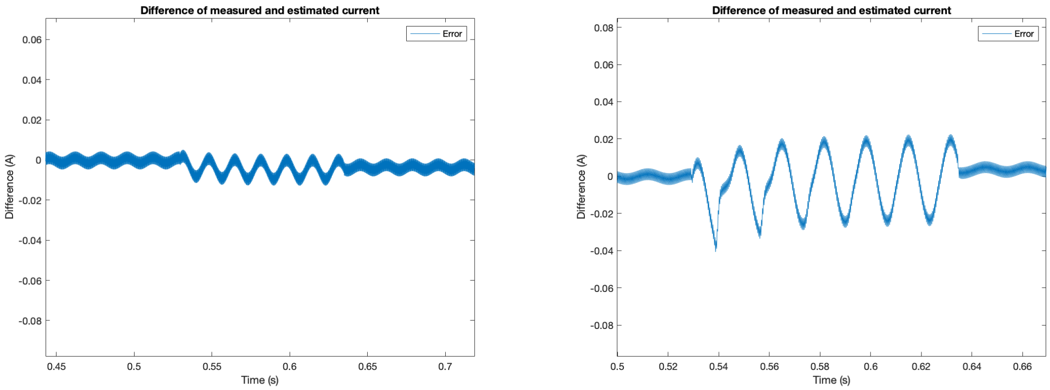
(b) Event 2 Primary current

Figure 13: Estimated primary current for both events

We observe how the current increases around 0.2 seconds, and also how it spikes at the half of the period of study. The second event primary current reaches an amplitude of 20,000 A, around 5 times greater that for event 1. This suggests that the event 2 presents a fault substantially more dangerous than event 1.

4 .3.1 Error of primary current

The error of the primary current, computed as $\epsilon = i_p/n - i_b^m$ (where n is the transformer ratio) is shown in Figure 14.



(a) Event 1 error

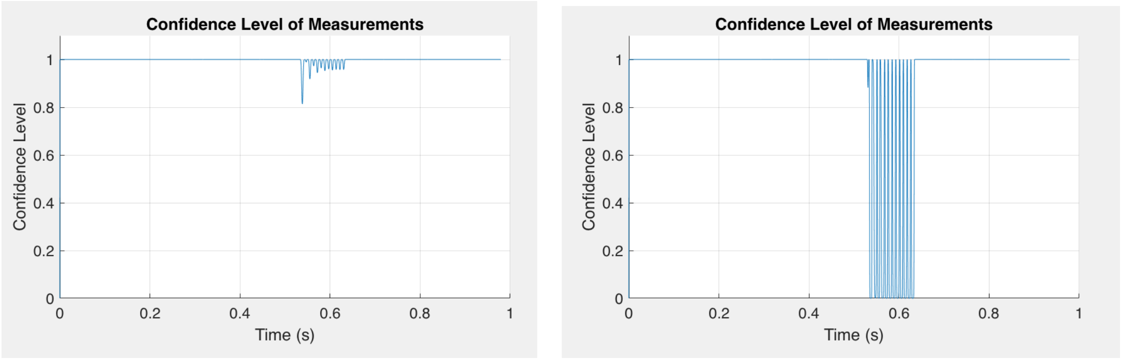
(b) Event 2 error

Figure 14: Computed primary current errors for both events

While we notice a change of differences at around 0.5 seconds, we see that event 2 presents a significant higher difference and that it follows a particular waveform, essentially due to the lack of error correction. This error that appears only in event 2 prevents the error waveform from being sinusoidal. A probable reason that explains this effect is the large magnitude values of the primary current flowing through the transformer. Since we are dealing with a non-ideal transformer, it presents saturation and also hysteresis, which is what is causing the non-linearity that we are observing in the initial transient [5]. Hence, we are indeed observing a resulting error from the saturation of current transformers.

4 .4 Confidence Level

Given the computation of the measures vector in terms of the state, the confidence level is computed as the squared sum of residual for each of the measurements. The resulting confidence level along time is shown of Figure 15.



(a) Event 1 confidence level

(b) Event 2 confidence level

Figure 15: Resulting confidence level for both events

As we have been intuiting, both events present a fault at t=0.536 and t=0.530 seconds, respectively. Those are the first points in time in which the confidence level drops under the threshold of 90% (significance level of 10%). There are 2 key takeaways from results. The first observation is that the confidence level does not drop until 0.5 s in both cases. That means that the increase of output voltage we saw at around 0.2 cannot be considered a fault. Secondly, due to the errors introduced by the core saturation on event 2, we get higher residues for this particular event, leading the confidence level to actually reach 0% of confidence.

4 .5 Relay actions

Following the criteria discussed in section 2.2, since the confidence levels drop in both events under 90%, it means that a fault has occurred, and we require the relay to take actions to minimize the impact of the fault in both cases at the time that has been previously specified.

5 Conclusions

This document has presented a model to perform a dynamic state estimation from an instrumentation channel. A correct estimation is essential to detect faults and indicate the relays when and how to act on the system. Four goals have been achieved. First, a graphical user interphase has been created to read data, make computations on the background and display results to the user in an intuitive manner. Second, a mathematical model has been successfully developed to write the measurements in terms of the state variables. To do so, additional degrees of freedom were introduced by including actual, derived, virtual and pseudo measurements. Thirdly, an iterative algorithm based on the Gauss-Newton method was applied to perform the dynamic state estimation. This method requires to obtain the jacobian of the measurement equations and that was not possible given the fact that some equations present derivatives due to the circuit dynamics. To operate with these equations, we applied the trapezoidal integration so the equations can be rewritten to obtain a new measurement model that ultimately can be used in the algorithm, which has been explained in detail as well. Once the estimation is perform, the goodness of fit was evaluated based on the Chi-Square test using a significance level of 10%.

Lastly, we have proved the validity of this methodology by analyzing 2 different fault events. For both cases, all the states variables, measurements and its residues were determined in order to compute the confidence level, the error between the primary current measure and estimation, and the flux linkage in the transformer. We have seen that both events present a fault around 0.5 seconds after the measurement recording started, as the confidence level drops below 90%. However, event 2 presents higher fault currents that lead to an observable takeaway: due to the high magnitude of current flows through the transformer when a fault occurs, it is possible to observe how the transformer reaches its saturation point: when the difference of estimated and measured primary current is computed, non-linearities due to saturation appear on the initial transient right after the fault occurs. This error introduced by the CT saturation curve leads to higher residues that reduce significantly the confidence level. While event 1 drops to no more that 80% of confidence, event 2 actually reaches 0% of confidence level.

All in all, these experiments have proven this tool to be useful for fault analysis given the parameters of the model and output voltage measurements. Results suggest that the error introduced by the transformer saturation can be corrected taking into account the measured and estimated primary current.

References

[1] *S. Del Prete, A. Delle Femine, D. Gallo, C. Landi, M. Luiso.* Implementation of a distributed Stand Alone Merging Unit. Dept. of Engineering, University of Campania Luigi Vanvitelli, Aversa (CE), Italy (2018).

[2] *M. Gurbiel, P. Komarnicki, Z. A. Styczynski, F. W. Gatzen, C. Dzienis.* Merging Unit Accuracy Testing. SM IEEE 2009.

[3] *Stenbakken, T.; Nelson, M.; Zhou and V. Centeno.* Reference Values for Dynamic Calibration of PMUs. Proceedings of the 41st Hawaii International Conference on System Sciences, pp1-6, 7-10 Jan. 2008.

[4] *A. P. Sakis Meliopoulos and George J. Cokkinides.* Power System Relaying, Theory and Applications.

[5] *A.P. Meliopoulos, George Cokkinides, Jiahao Xie and Yuan Kong.* Instrumentation Error Correction within Merging Units. School of Electrical and Computer Engineering, Georgia Institute of Technology