Distributed Control of Wide Area Power System

August 24, 2015

1 Introduction

In this experiment, security attacks are simulated on the communication links of a wide- area power system, disrupting the flow of measurement data from the machines and control input to the machines. This temporary disconnect in the network may cause the dynamic system performance to deteriorate or even cause instability.The aim of the study is to find a sensitivity plot which identifies the scenarios which would result in instability of the system or deterioration in the system performance and to determine the regions of safe operation.

2 Test Bed Architecture

2.1 Power System Simulation

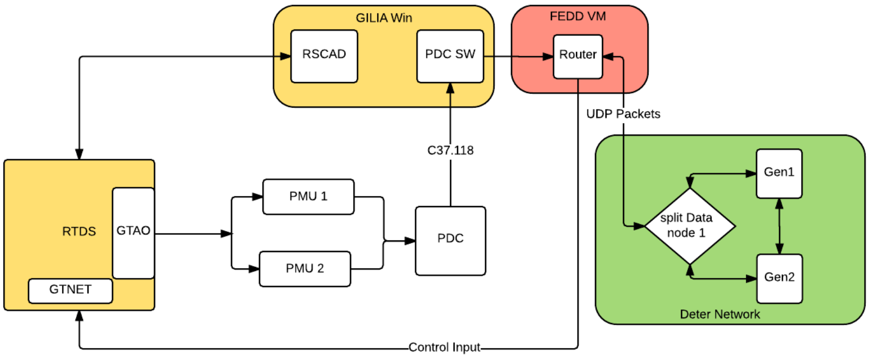
2.1.1 RTDS HIL Model

A multi machine wide area power system model is designed in RSCAD. The model is loaded on to the RTDS (Real Time Digital Simulator), an Hardware in loop simulator, to run real time simulations. The generator states are measured from the internal bus with a PMU (Phasor Measurement Unit) during the simulations. The PMUs are connected to the ports in GTAO(Giga-Transceiver Analogue Output) Card of the RTDS.

The PMU measures the bus voltage, phase angle and frequency. These measurement data from all the PMUs are then forwarded to the PDC(Phasor Data Concentrator). The PDC combines the data from all the PMUs based on their synchronized timestamp. PDC communicates the measurement signal as an IEEE C37.118 message to the base station (computer) over the ethernet as UDP datagrams. In order to implement the control the PDC measurement data is forwarded to the controllers present at the DETER nodes through the Internet using UDP communication.

2.1.2 MATLAB Model

For testing purposes a six machine wide area power system model was designed in MATLAB.The states of the machine are directly measured and communicated to the network controller. In order to make the software simulation closer to the real time simulation a first order hold is implemented in the closed loop control signal (i.e. the value of the control input from the previous time sample is used until a new sample arrives).



Two machine test bed architecture

Figure 1: The Test bed architecture

2.2 Network Structure

The PDC measurement data is received at the base station (GILIA machine) using the PDC Assistant software. The measurement data is transmitted at a 60Hz data frequency in the Local Area Network. This data message is forwarded to the FEDD virtual machine in GILIA through NAT(Network Address Translator). The FEDD VM acts as gateway node to communicate with the DETER Network. A federated experiment is created with the FEDD VM as an external desktop node and all the other nodes are present within the DETER network. Each deter node represents a individual machine.

The FEDD node routes the received data messages to the deter node where the IEEE C37.118 message is decoded based on the initial configuration message. The standard IEEE message is decoded to obtain the phase angle and frequency data from each machine. The states of each machine is forwarded to individual nodes which resembles the machines. All machine nodes are interconnected with each other, in order to share/communicate the states with all the machines, in a full graph network structure. A LQR controller is designed considering the linearized model of the power system and accommodating the delays induced by the data transfer in the network. The feedback gains of each machine calculated from the LQR design are fed into the the nodes and each node individually generates the control signal with the measurement data of states of all the machines.

The calculated control inputs are then routed back to the FEDD VM. The FEDD VM transmits the control inputs back to the RTDS Model by sending them as UDP packets to the GTNET card using socket protocol. In case of the MATLAB simulations the control inputs are forwarded directly to the matlab program.

3 Mathematical Model and Control Design

3.1 Power System Model

Let us consider a power system network with n generators distributed into multiple areas. The Internal voltage of any ith generator is given by $\hat{E}_i = E_i \angle \delta_i, i = 1 : n$. Each synchronous generator may be modelled by a set a third order algebraic differential equations,

$$
\begin{aligned}
\dot{\delta_i} &= \omega_s \\
2H_i \dot{\omega}_i &= P_{mi} - D_i(\omega_i - \omega_s) - P_i^G \\
\tau_i \dot{E_i} &= -\frac{x_{di}}{x_{di}^{'}} + \frac{x_{di} - x_{di}^{'}}{x_{di}^{'}} V_i cos(\delta_i - theta_i) + \hat{E_{Fi}} \\
P_i^G &= \frac{E_i V_i}{x_{di}^{'}} \sin(\delta_i - \theta_i)
\end{aligned}
\tag{1}
$$

where, the states δ_i, ω_i and E_i are respectively, the generator phase angle(radians), angular velocity of the rotor(rad/sec) and quadrature-axis internal emf; ω_s is the synchronous frequency of 120π radians/sec; P_i^G is the active power produced by the ith generator; $2H_i$ is the inertia constant(seconds), D_i is the generator damping, $x_{di}^{'}$ is the direct axis transient reactance; P_{mi} is the mechanical power input to the ith turbine(MW). τ_i is the excitation time constant(seconds);x_{di}, x'_{di} and x_{qi} are the direct-axis salient reactance, direct axis transient reactance and quadrature axis salient reactance (all in ohms), respectively. The control variable is the field voltage E_{Fi}, which is the variation of $\hat{E_{Fi}}$ from the equilibrium point.
Firstly to linearize the non-linear swing equations,consider a small perturbation over an existing equilibrium $(\delta_{i0}, \omega_{i0}), i = 1 : n$. The overall linearized small signal model of the power system can be expressed as

$$
\begin{bmatrix} \Delta\dot{\delta}(t) \\ 2H\Delta\dot{\omega}(t) \\ T\Delta \dot{E(t)} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ -L & -D & -P \\ K & 0 & J \end{bmatrix} \begin{bmatrix} \Delta\delta(t) \\ \Delta\omega(t) \\ \Delta E(t) \end{bmatrix} + \begin{bmatrix} 00 \\ I0 \\ 0I \end{bmatrix} \begin{bmatrix} \Delta P_m \\ \Delta E_F \end{bmatrix} \tag{2}
$$

where, $\Delta\delta = col(\Delta\delta_1, \ldots, \Delta\delta_n)$, $\Delta\omega = col(\Delta\omega_1, \ldots, \Delta\omega_n)$, $H = diag(H_1, \ldots, H_2)$, $\Delta P_m = col(\Delta P_{m1}, \ldots, \Delta P_{mn})$, $\Delta E_F = col(\Delta E_{F1}, \ldots, \Delta E_{Fn})$ and $T = diag(\tau_1, \ldots, \tau_n)$ .The turbine mechanical power ΔP_m typically has a much lower bandwidth than needed for oscillation damping. Therefore for wide area control designs, ΔP_m is treated as zero, and The continuous time state space model of the linearized small signal system can be represented by (3). The control input ΔE_F is designed using PMU data feedback.

$$
\dot{X}(t) = A_c X(t) + B_c U(t) \tag{3}
$$

where A_c - Continuous time state Matrix B_c - Continuous time Input matrix and X is the state vector. The practical implementation of a power system controller in a distributed network would be in the discrete-time.The states from the power system are measured using the PMUs at a sampling rate of one sample per τ seconds. The sampling interval of the states is τ seconds. The discrete time model of the system is given by (4)

$$
\begin{aligned}
X[k+1] &= A_d X[k] + B_d U[k] \\
A_d &= e^{A_c \tau} \\
B_d &= \int_0^{t=\tau} e^{A_c t} dt B_c
\end{aligned}
\tag{4}
$$

where, A_d is the discrete-time state matrix and B_d is the discrete-time input matrix. The discrete time state space model is given by (4).
Since the controller is being implemented on a distributed network, the states of all the machines at time instant k are not readily available at time $t = k$. The states of ith machine arrives to every other controller node after a random communication delay. The maximum delay threshold of a network is a characteristic of the communication network. Let us assume h to be the maximum delay threshold of our network. This implies that in ideal conditions when all links are operational the states of the machines reach every other controller node within h seconds. The control input cannot be evaluated until all the states are available, hence the control input for time instant k will be available only at $t = k + h$. The control input for the period $t = [k, k+h)$ will be the control input calculated at the previous time instant, $U[k-1]$. The effective control input $u_i(t)$ for a single sampling time duration $k < t < k+1$ is represented by

$$
u_i[t] = \begin{cases} u_i[k-1] & , k < t < k+h \\ u_i[k] & , k+h < t < k+1 \end{cases}
\tag{5}
$$

The effective discrete time state space model can be written as,

$$
\begin{gathered}
X[k+1] = AX[k] + B_1 U[k-1] + B_2 U[k] \\
B_1 = \int_0^{t=h} e^{At} B_c dt \; B_2 = \int_h^{t=\tau} e^{At} B_c dt
\end{gathered}
\tag{6}
$$

By defining an augmented state $Z[k] = [X[k] \; U[k-1]]^T$, (6) can be written as:

$$
\begin{aligned}
Z[k+1] &= \begin{bmatrix} A & B_1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X[k] \\ U[k-1] \end{bmatrix} + \begin{bmatrix} B_2 \\ I \end{bmatrix} \\
&= \hat{A} Z[k] + \hat{B} U[k]
\end{aligned}
\tag{7}
$$

A Linear Quadratic Regulator(LQR) controller is designed to control the discrete time system. The LQR optimizes the quadratic cost function J,

$$
J = \sum_{k=0}^{\infty} Z_k^T Q Z_k + U_k^T R U_k
\tag{8}
$$

where Q and R are the design parameters. The control input $U[k]$ from the control law can be written as,

$$
U[k] = KZ[k] = K_0 X[k] + L_0 U[k-1]
\tag{9}
$$

where K is the feedback gain resulting from the LQR control design.

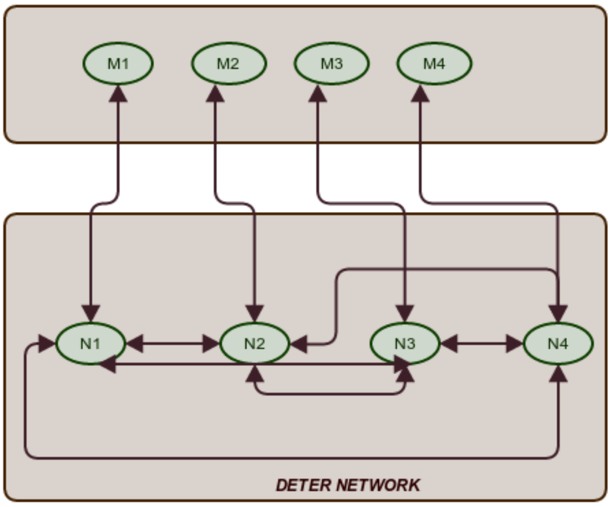


Figure 2: Network Topology

3.2 Network Implementation

This LQR controller is implemented in a distributed network system. The states from the machines are transmitted to the deter nodes and each node evaluates the control input for the respective machine and transmits back to the machine. In order to evaluate the control input all the nodes require the states from all the other machines at every sampling instant.Hence the network will be a full graph structure where all nodes are connected with each other, the graph for a n machine model can be denoted by G. A full graph will have all the elements in the matrix to be 1.when there is a link between nodes i and j down/broken the $G[i,j] = G[j,i] = 0$

$$G = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}_{nxn} \tag{10}$$

The control input of the ith machine at any time instant l , $u_i(l)$ given by , is evaluated by using the measured states of all the machines.

$$u_i[k] = (K_{0i} \circ G_i)X[k] + L_{0i}U[k-1] \tag{11}$$

where, K_{0i},L_{0i} and G_i are the ith row of the state feedback matrix K_0, the gain matrix L_0 corresponding to the previous input and Graph matrix G respectively.

4 Simulation and Testing

A number of simulations can be run in the test bed to study the stability and dynamics of the power system when there is a security attack (denial of service) on the links connecting the generator nodes.The variables considered in the test scenarios are

1. duration of attack
2. Location of attack (link)
3. time identity (i.e. the time of attack after the start of simulation).

A denial of service attack is simulated by using the capabilities of deter network, by flooding the link with enormous amount of data. A typical attack lasts for a specific number of time samples either on one or multiple links. During an attack the graph of the network is not a full graph and can be represented by G_A, if the links between nodes m and n are attacked the elements $G_A(m,n) = G_A(n,m) = 0$. For example if link between node 2 and node 3 is attacked the network graph of a 4 machine system will be,

$$G_A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}_{4x4} \tag{12}$$

We have 3 different mitigation strategies to reduce the effects of the loss of data due to attacks on the links.The strategies are

1. Strategy 1: Skip strategy
2. Strategy 2: Zero Order Hold
3. Strategy 3: State Estimation

below is a list of scenarios,

4.1 Test Scenarios

4.2 Scenario-1: Ideal Case (No Attack)

Under ideal conditions, all the communication links between the deter nodes are functional and the states of all the machines reach every other machine within the maximum delay threshold h. The deter nodes then evaluate the control input based on states of the machines and the control input at previous time instant as shown in (11).

4.2.1 Scenario-2 : Attack on single link

In this case a denial of service attack is simulated on each link individually. The network graph will now have two zero elements. The attack is simulated for a specific number of time samples, n, where $n = 1, 2, 3, 4, 5$. The control input $u_i[k]$ of ith machine can be written as,

where,K_{0i} and L_{0i} are the ith rows of the feedback gain matrices corresponding to the states of the machines and the previous control inputs,G_{Ai} is the graph of the network during attack.

4.2.2 Scenario-2 : Attack on multiple links

The Denial of service attack is simulated on multiple links simultaneously. The network graph will have multiple zero elements. The attack is simulated for n time samples where $n = 1, 2, 3, 4, 5$.

4.2.3 Scenario-3 : Attack on links at different time identities

Performing both the Scenario 1 and Scenario 2 at different time instants of the simulation.(e.g. at the start of the simulation, towards the end of the simulation where the system reaches steady state, at half the settling time).

4.3 Mitigation Strategies

4.3.1 Strategy 1: Skip Strategy

According to skip strategy the states of the machines are assumed to be zero if they are not available to the node before the threshold delay time, h. The control input is calculated with zero value for the missing states and can be written as,

$$u_i[k] = (K_{0i} \circ G_{Ai})X[k] + L_{0i}u_i[k-1] \tag{13}$$

4.3.2 Strategy 2: Zero Order Hold

In case of strategy 2 the states of the machines at previous instant of time are stored at each node. If state $x_i[k]$, the state of ith machine at time instant k, does not arrive within the threshold h, then $x_i[k-1]$ is used to evaluate the control input.(i.e.) $x_i[k] = x_i[k-1]$. If the state of ith machine is not available for n consecutive time samples then $x_i[k] = x_i[k-n]$, where $n = 1, 2, 3, 4, 5$. The control input evaluated at the node can be represented as,

$$u_i[k+n] = (K_{0i} \circ G_{Ai})X[k+n] + L_{0i}u_i[k-1+n] + K_{0mi}x_m[k-n] \tag{14}$$

4.3.3 Strategy 3: State Estimation

In case of strategy 3 the states of the machines and previous control input values are stored at each node. If state $x_m[k]$, the state of mth machine at time instant k, does not arrive within the threshold h, then $x_m[k]$ is estimated by solving the discrete time state space of the dynamic system in (7). The control input is then evaluated by using this estimated state value for the missing state. The control input can be written as,

$$u_i[k] = (K_{0i} \circ G_{Ai})X[k] + L_{0i}u_i[k-1] + K_{0mi}\hat{x_m}[k] \tag{15}$$

where K_{0mi} is the (i, m)th element of the feedback gain matrix K_0 and $\hat{x_m}[k]$ is the estimated value of the state of mth machine. This strategy would reduce the difference between the ideal case control input and estimated control input during a security attack.

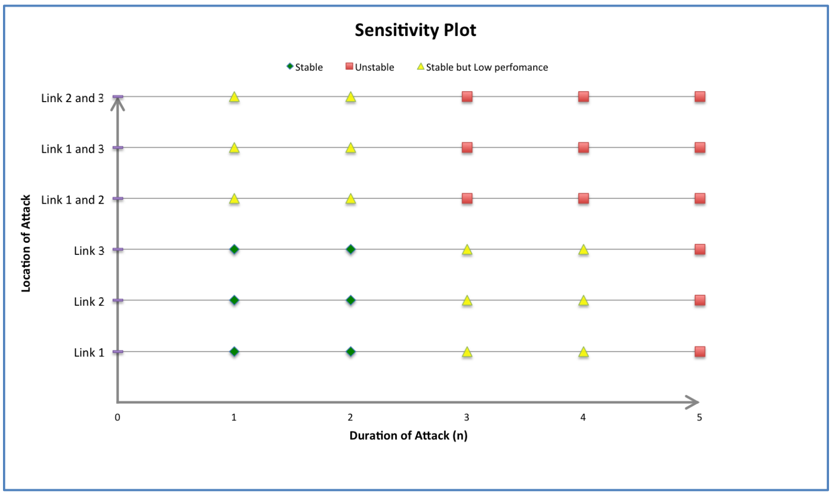


Figure 3: Plot illustrating the stability region

4.4 Senstivity Plot

With the results from running the simulations for all the above test cases we will be able to identify the cases/scenarios when the system is stable, unstable and the performance is degraded. With this information a sensitivity plot as shown in fig(3) for different time identities can be plotted. These plots can be used to identify the stability regions.