Application of Ultra-Local Models in Automatic Generation Control with Co-Simulation of Communication Delay

A thesis submitted by

Wenjie Han

In partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Tufts University

May 2018

Adviser: Alex M. Stankovic

Abstract

This paper applies the ultra-local models (also labeled as intelligent PID controllers) to Automatic Generation Control (AGC) in a Multi-Area power system (New England 39 Bus) in the presence of communication delay. Given the fast development of smart grid in the past years, power system and communication network are coupled much more tightly than before. Therefore, the independent study of the two domains or simplistic modeling the communication delay in the power system simulation environment is no longer effective and convincing. In our work, the power system model containing AGC is built in Simulink/Matlab, and the communication network is co-simulated in Network Simulator 2 (NS2) through PiccSIM, a simulation platform for (wireless/wired) networked control systems.

A careful comparison is made between the performance of intelligent-P (iP) controller and the conventional PI controller under the co-simulation environment of AGC with time-varying wired communication delay. The results show the two controllers are largely equivalent for relatively small frequency deviations. However, intelligent-P controller performs better than conventional PI controller under more severe system variations, such as large communication delay. The avoidance of integration in the intelligent-P controller makes anti-windup algorithm unnecessary, which enhances the applicability and the simplicity of controller tuning.

In addition, the co-simulation results also demonstrate that the system is more sensitive to the communication delay of control signals from the area control center to the distributed generators than to the delay of tie-line active power measurements. This conclusion is very illuminating that

the distributed control mechanism becomes a promising alternative for the control mechanism applied widely now. Instead of a one control center per regulating area, every generator participating in AGC could have its own local controller. The distributed generator controllers can communicate with each other and share the needed information. This fully-distributed automatic generation control (AGC) is more tolerant of communication delay, and thus more robust and stable under various conditions.

Acknowledgement

First, I would like to express my deep appreciation to my advisor, Professor Alex Stankovic for his continuous support and encouragement. Thanks for his time, patience and understanding. Whenever I met difficulties in my research, he always gave inspiring suggestions and steered me back to the right track. It is such a great honor to work with him.

My gratitude also goes to the Power Electronics and Systems, Electric Energy Processing Lab in the Department of Electrical and Computer Engineering, Tufts University. My special thanks go to Gang Wang and Stephen Gemme for installing the co-simulation platform, PiccSIM, and their instructions on simulations.

I would also like to thank Professor YeonWook Choe in the Department of Measurement and Control Engineering, Pukyong National University, South Korea. Thanks for his detailed explaining of intelligent PID controller, and sharing the controller models. Thanks to Dr. Mikael Björkbom, who is the developer of PiccSIM, for taking the time out of a busy schedule to answer my questions.

Finally, I'd like to express my sincere thanks to my parents and friends for giving me unconditional support and love through all my study.

TABLE OF CONTENTS

1. INTRODUCTION **1**

2. BASICS OF AUTOMATIC GENERATION CONTROL (AGC) **4**

2.1 AGC IN A SINGLE AREA 5

2.2 AGC IN MULTI-AREA SYSTEM 7

3. OVERVIEW OF INTELLIGENT PID CONTROLLER **9**

3.1 BASICS OF ULTRA-LOCAL MODEL AND INTELLIGENT PID CONTROLLER 10

3.2 ONLINE ESTIMATION OF F 11

3.3 CONNECTIONS BETWEEN CONVENTIONAL AND INTELLIGENT CONTROLLERS 13

3.4 NUMERICAL DERIVATION (ND) METHOD 16

4. NS2 AND THE CO-SIMULATION PLATFORM PICCSIM **18**

4.1 NETWORK SIMULATOR 2 18

4.2 PICCSIM 20

4.2.1 Architecture of PiccSIM *20*

4.2.2 Time Synchronization of PiccSIM *23*

5. CO-SIMULATION MODEL **24**

5.1 POWER SYSTEM SIMULATION MODEL 25

5.1.1 New England 39 Bus Power System *25*

5.1.2 AGC – Classic PI Controller *28*

5.1.3 AGC – Intelligent-P Controller *30*

5.2 COMMUNICATION NETWORK 31

5.3 SIMULATION SCENARIOS 35

6. SIMULATION RESULTS AND ANALYSIS **36**

7. CONCLUSION AND FUTURE WORK **45**

BIBLIOGRAPHY **47**

1. Introduction

With the rapid development of new power technologies, the communication infrastructure has been widely deployed for monitoring, control, and operation in a power grid. Therefore, the power system has become strongly coupled with the information and communication technology (ICT). The interdependency of the two different domains requires a co-simulation platform to analyze the physical and ICT processes as a single entity. However, a major challenge of co-simulation is the fundamentally different modeling and simulation concepts of different domains, such as time synchronization and event handling, [1].

Paper [1] summarizes the related work in this research area, shown in Table I. The EPOCHS (Electric Power and Communication Synchronizing Simulator) package in [3] pioneered the efforts to build a co-simulation tool. EPOCHS consists of three simulators: PSLF (GE's Positive Sequence Load Flow) for large-scale power system stability analysis, PSCAD/EMTDC for transient protection simulation, and NS 2 (Network Simulator 2) for communication network modeling. For time synchronization, EPOCHS applies a time-stepped synchronization mechanism with fixed synchronization points, which leads to a tradeoff between precision and efficiency. GECO (Global Event-Driven Co-simulation Framework) proposed in [2], also integrating PSLF and NS 2, used a global event-driven mechanism. The simulator's fidelity and accuracy are enhanced by a global event scheduler, which maintains a global event queue to arrange the event sequence. In paper [4], Simulink and OPNET (Optimized Network Engineering Tool) were integrated to operate together, and different communication network scales were considered with both stationary and mobile nodes.

PiccSIM [5] is utilized in this thesis to integrate the well-known and powerful tools, Simulink and NS2, into one tool set. This co-simulation platform allows the study of all layers and interactions of large and complicated networked control systems.

	Recent Focus	Power System Simulator	Network Simulator	Simulation Frameworks	Time Strategy	Scalability	Execution Mode
GECO	PMU-based WAMPAC, high voltage grid	PSLF	NS-2	Ad-hoc (TCL linking)	Global event-driven	Large systems	NA
INSPIRE	WAMPAC, high voltage grid	DIgSILENT PowerFactory	OPNET Modeler	IEEE 1516-2010 (HLA evolved)	Dynamic time stepped	Large systems	NA
EPOCHS	Multi-agent protection and control systems	PSCAD/ EMTDC, PSLF	NS-2	IEEE 1516-2000 (HLA)	Fixed time stepped	Large systems	NA
ADEVS	WAMPAC	ADEVS	NS-2	Ad-hoc (inte-grated in NS-2)	DEVS	Large systems	NA
VPNET	WAMPAC	VTB	OPNET Modeler	Ad-hoc (Sockets)	Time stepped	Small systems	NA
GridSim	WAMPAC	Powertech TSAT	GridStat	Ad-hoc	Fixed time stepped	Components can be distributed	Real-time + HIL
PowerNet	Controlling power devices	Modelica	NS-2	Ad-hoc (Unix named pipes)	Time stepped	Small systems	NA
Bergmann	Evaluation of DERs and VPPs	NETOMAC	NS-2	Ad-hoc (JNI)	Time stepped	Small systems	Close to real-time
Babazadaeh	WAMC, HVDC, low voltage/mid voltage Grid	OPAL-RT	OPNET SITL	Ad-hoc, emulated, sockets	Real-time	Small (HIL), medium (emulated)	Real-time
Greenbench	Cyber security, low voltage grid	PSCAD	OMNeT ++	Ad-hoc (IPC)	Global event-driven	Tested for small systems	NA

Table I: Overview of State-of-the Art Co-Simulation Approaches [1]

As the prototypical system-level automatic closed-loop power system control over the communication system [6], automatic generation control (AGC) is very sensitive to the connecting configurations of the two systems. This integration in turn has a significant effect on the stability of the overall system. For the satisfactory operation of a power system, AGC is applied to regulate frequency to the nominal value and to maintain the interchange power flows between control areas at the scheduled values. Since the supervisory control and data acquisition (SCADA) system is widely used in the power grid now, measurements of the frequency deviation and tie-line power flows can be transmitted to the control center (one per area) and processed. Then the control signal is sent to generators to regulate the turbine-governor units.

Most research on AGC disregarded the communication delay of tie-line power measurements and generator control signals. Recently, several papers [7]–[9] studied the effect of communication delay on AGC, but the delay is either simulated in Matlab or deduced from an idealized mathematical model. Therefore, the simulation results have some limitations in practice. The co-simulation platform PiccSIM used in our work provides more practical and realistic results than the tests with idealized delay patterns.

The PID control design plays a critical role in the performance of AGC. However, the design is quite difficult when the system is not known well, which is often encountered in practice - for example, smart grid with strong penetration of renewable generations. In addition, the practical robustness of conventional PID controllers can prove questionable - e.g., the windup effect of the integral part when a large change occurs in setpoints. We aim to show that the ultra-local or intelligent PID (i-PID) controller proposed recently in [10] has the potential to be applied to some systems where the unknown parts might be nonlinear and/or time-varying without explicit modeling requirements ("model-free control"). The main tool of model-free control is online parameter estimation in a simple control structure. The intelligent PID controller has following desirable features [11]:

(i) No need for preliminary parameter estimation,

(ii) Simple structure as PID controller,

(iii) Simple adjustment for complex systems,

(iv) No need to readjust parameters of the controller when the characteristics of the system are changed.

Thus the ultra-local PID controller is an interesting alternative for the conventional PID controller.

This thesis is constructed in the following way. Chapter 2 reviews the basics of automatic generation control (AGC) both for single area and multi-area power systems. Chapter 3 introduces the ultra-local or intelligent PID controller, and explains the relationship between the conventional PID controller and intelligent PID controller. In section 3.4, a brief introduction to the numerical derivation (ND) method is made. Chapter 4 introduces NS2 and the co-simulation platform PiccSIM. Chapter 5 describes our co-simulation model, the New England 39 Bus system and the communication network settings. In Chapter 6, we studied six scenarios of different operation states and made a detailed comparison between the performance of conventional and intelligent PID controller. Finally, conclusions and future work are presented in the last Chapter.

2. Basics of Automatic Generation Control (AGC)

Frequency deviation in a power system originates from the imbalance between electrical loads and the total real power supplied by generators. In today's interconnected large-scale power systems, areas can exchange active power based on a prearranged schedule to satisfy the demand. The primary objective of automatic generation control (AGC) is to regulate the grid frequency to the nominal value (60Hz in US) and maintain the area-to-area interchange power flows at the scheduled values by adjusting the power output of selected generators. This control mechanism is also referred to as Load Frequency Control (LFC). The second objective is to achieve economic allocation of the power generation variations.

2.1 AGC in a Single Area

There exist two control loops for a single control area: primary control and supplementary control. Primary control relies on the local governor control to ensure balance. But for the system with only primary control, a load change will result in a steady state frequency offset. Restoration of the system frequency to the nominal value requires the participation of the supplementary control through adjusting the load reference setpoint, [12]. The control loop described above is shown in Figure 1. It should be noted that all the values in Figure 1 are per-unit values, so the rotor speed ω_r is equal to the system frequency.

The value of R in Figure 1 determines the steady-state speed versus power output or gate/valve position characteristics. The ratio of speed deviation $\Delta\omega_r$ or frequency deviation Δf to the change of gate/valve position ΔY or power output ΔP is equal to R. The value of R is also referred to as speed regulation or droop. Figure 2 shows the steady-state speed droop characteristics with only primary control. R can also be expressed in percent as:

$$Percent\ \mathrm{R} = \frac{percent\ \text{speed or frequency change}}{percent\ \text{power output change}} \times 100$$

$$= \left(\frac{\omega_{NL} - \omega_{FL}}{\omega_0}\right) \times 100$$

Where ω_{NL} and ω_{FL} are the steady-state speed with no load and full load, respectively. ω_0 is the nominal speed. For example, a 2% regulation or droop means a 2% frequency deviation can cause 100% power output or gate/valve position change.

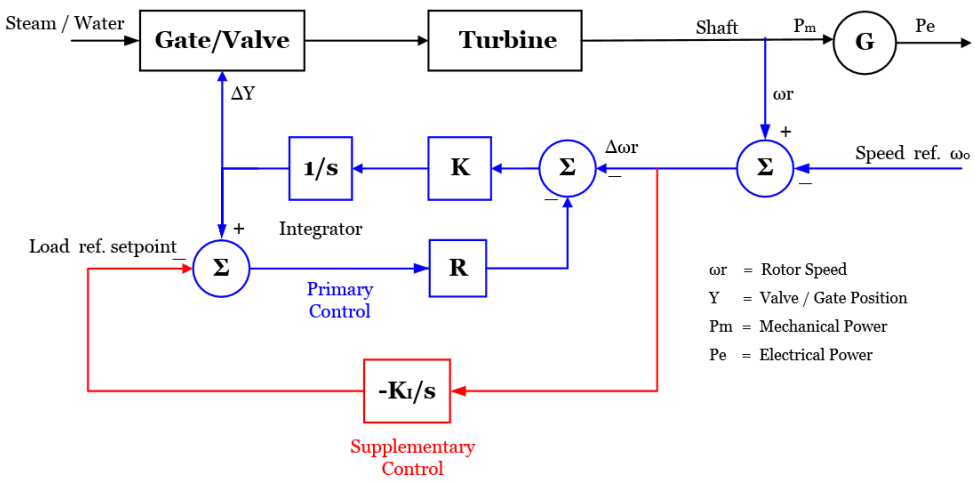


Figure 1. AGC for a single area

When a sudden load change occurs, primary control responses faster than supplementary control. As the steady-state droop characteristics show, primary control can't regulate the frequency back to the nominal value. It is the supplementary control that functions to restore the frequency to the nominal value by changing the load reference setpoint. In practice, this regulation is realized by operating the speed-changer motor, and the effect is shown in Figure 3.

It can be seen that through the setpoint adjustment, the single steady-state speed droop characteristics become a family of parallel characteristics for different changer-motor settings (load reference setpoints). For characteristics of A, at f=60 Hz (nominal value), the corresponding power output is zero, while characteristics B results in 50% output, and characteristics C results in 100%. Therefore, the power output of at a given speed (frequency) can be adjusted to any desired value by adjusting the load reference setpoint through actuation of the speed-changer motor.

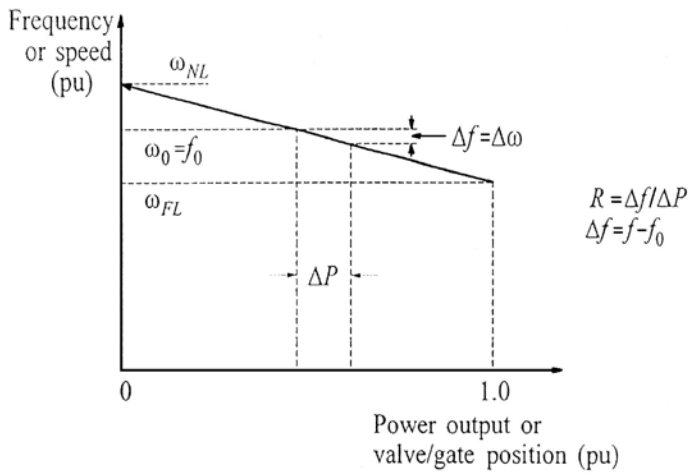


Figure 2. Ideal Steady-State Characteristics of a governor with speed droop [12]

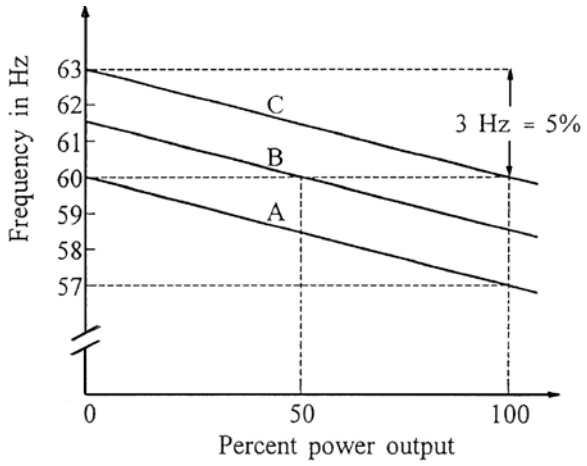


Figure 3. Effect of speed-changer setting on governor characteristics [12]

2.2 AGC in Multi-area System

In multi-area power systems, each area monitors its frequency deviation and the tie-line active power flows. The measurements are then gathered in the control center to calculate the Area Control Error (ACE) signal, which serves to realize the coordination of different control areas. ACE for area m is calculated as:

$$ACE_m = \sum_{n \in A_m} (P_{mn} - P_{mn}^{sch}) + \beta_m * \Delta f_m \quad , \qquad (1)$$

where A_m denotes all areas connected to area m, P_{mn} is the actual power interchange from area m to its connected areas, while P_{mn}^{sch} stands for the scheduled power flow. β_m is the bias factor for area m and Δf_m is the frequency deviation.

The ACE signal is then processed through an integrator or a PI controller in the control center, and the output is sent to generators to regulate the turbine governor unit, shown in Figure 4. In North America, the control signals are fed to the generator units typically, about once every 2-4s. Assuming overall stability, the integral control action ensures the ACE signal is zero in steady-state, so both the frequency and interchange active power flows are restored to the nominal frequency or scheduled values.

Under normal conditions, when each area is able to carry out the load change in its own area, steady-state corrective action of AGC is confined to the area where the load change occurs. Meanwhile, area-to-area tie-line active power flows are still kept as the scheduled values, and the system frequency stays at the nominal value.

Under abnormal conditions, one or more areas become unable to respond to the area-wise load changes, which is often encountered when the increase of demand excesses the supplying ability of the generating units. In such an event, other area assistance must be allowed. So the interarea power exchange should be allowed to deviate from the pre-disturbance schedule values. The participation of a specified area in AGC is proportion to its available regulating capacity relative to the overall system.

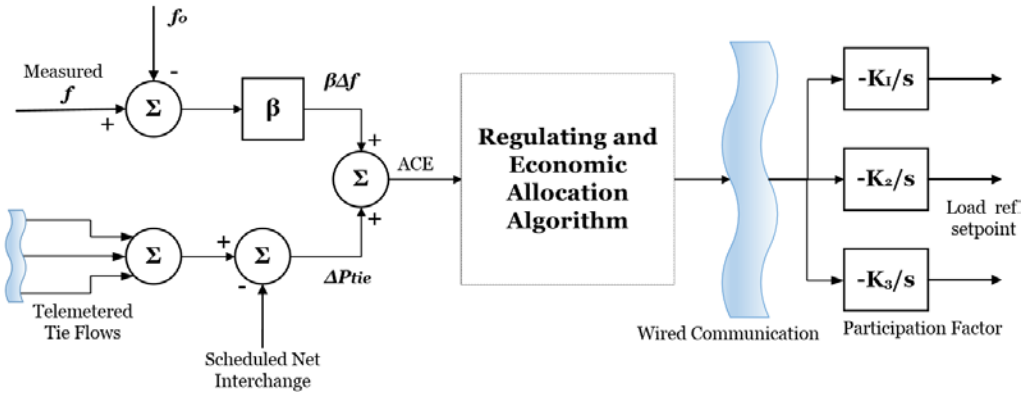


Figure 4. AGC for Multi-area

As shown in Figure 4, the economic allocation of load change to each generator is accomplished by using the *base points* and *participation factors* (PFs). The allocation algorithm is performed once every 5 minutes to update the generation plant based points and participation factors, [13]. To make sure that every generator works economically, the allocation follows the principle of equal incremental costs. The base point represents the most economical operation mode of a specified generator, and the participation factor is the rate of the output change for a single generator with respect to the overall load change. The new desired output for each generator is calculated as follows, [12]:

$$P_{desired} = P_{base\ \text{point}} + PF(\Delta P_{total})$$

Sum of all participation factors should be equal to 1.

3. Overview of Intelligent PID Controller

PID controller is a control loop feedback mechanism commonly used in industrial control systems. It continuously calculates the error signal $e(t)$ as the difference between a desired value

and a measured process variable, and applies a correction based on the proportional, integral, and derivative terms. Although PID controllers are applicable to many control problems, and often perform satisfactorily without any improvements or only coarse tuning, they can perform poorly in some applications. One common problem is integral windup. Following a large change in setpoint, the integral term can accumulate an error larger than the maximal value for the regulation variable (windup), thus the system overshoots and continues to increase until this accumulated error is reset (unwound).

3.1 Basics of Ultra-Local Model and Intelligent PID Controller

Model free control and the corresponding intelligent PID controllers (iPIDs) are presented in a unified manner for the first time in [10]. The main tool is online parameter estimation and identification approach. In many engineering fields, the plant to be controlled can be approximately described with a finite-dimensional ordinary differential equation - in the SISO example,

$$E(t, y, \dot{y}, ..., y^{(v)}, u, \dot{u}, ..., u^{(l)}) = 0, \qquad (2)$$

where E is a sufficiently smooth function of its arguments. $y(t)$ and $u(t)$ are the output and input signal respectively. The implicit function theorem yields locally,

$$y^{(n)} = f(t, y, \dot{y}, ..., y^{(n-1)}, y^{(n+1)}, ..., y^{(v)}, u, \dot{u}, ..., u^{(l)}), \qquad (3)$$

The unknown complex function f is replaced by an *ultra-local model*

$$y^{(n)} = F + \alpha u\,, \qquad (4)$$

Where $y^{(n)}$ is the derivative of order $n \geq 1$ of y. The existing examples show that n is usually 1 or seldom 2. α is a non-physical parameter, and it servers to scale the input $u(t)$ and $y^{(n)}$ to the same magnitude. F subsumes all the poorly known part of the control system as well as the disturbances, delays and noise, without the need to make any distinction between them.

Assume $n = 1$ in Equation (4), and close the loop via *intelligent-PID* controller,

$$u = \frac{\dot{y}^* - F + K_P e + K_I \int e + K_D \dot{e},}{\alpha} \tag{5}$$

where y^* is the output reference trajectory, and $e = y - y^*$ is the tracking error. Combing Equation (4) and (5) yields,

$$\dot{e} - K_P e - K_I \int e - K_D \dot{e} = 0. \tag{6}$$

Note that F does not appear in Equation (6) any more. Therefore, the tuning of coefficients K_P, K_I, and K_D becomes straightforward for obtaining a good tracking of y^*. This is a major benefit comparing with the conventional PID controllers. We need to adjust K_P, K_I, and K_D such that,

$$\lim_{t \to \infty} e(t) = 0. \tag{7}$$

3.2 Online Estimation of F

In Equation (5), F should be continuously estimated online and in [10], several explicit techniques are mentioned.

Here, three estimation methods are shown:

(i) With Equation (4), $n=1$, based on Laplace transformation, we can get

$$sY = \frac{F}{s} + \alpha U + y_0, \tag{8}$$

Where y_0 is the initial condition corresponding the time interval. In order to get rid of y_0, multiply both sides by $\frac{d}{ds}$,

$$Y + s\frac{dY}{ds} = -\frac{F}{s^2} + \alpha\frac{dU}{ds}, \tag{9}$$

Multiplying both sides by s^{-2} to smooth the noise yields Equation (10) in time domain,

$$\hat{F} = -\frac{6}{L^3}\int_{t-L}^{t}((L-2\sigma)y(\sigma) + \alpha\sigma(L-\sigma)u(\sigma))d\sigma\ , \tag{10}$$

where L is quite small and the value of L depends on the sampling interval and the intensity of noise.

(ii) For n=1, close the loop with the intelligent-Proportional controllers (similar to Equation (5)). It yields,

$$\hat{F} = \frac{1}{L}\left[\int_{t-L}^{t}(y^* - \alpha u + K_P e)d\sigma\right]. \tag{11}$$

(iii) In our work, F is estimated from the input u(t) and the derivative of the output y(t). When n=1,

$$\hat{F}(t) = \dot{y}(t) - \alpha u(t-h) \tag{12}$$

where h is a sampling interval. It is clear that *h* should be very small so that the estimation of F becomes close enough to the unknown real value. Based on this estimation technique, the

relationship between the intelligent PID controller and the conventional PID controller can be deduced.

3.3 Connections Between Conventional and Intelligent Controllers

(a) PI and intelligent-P

Combining Equation (12) and Equation (5), an intelligent-P controller is given by,

$$u(t) = \frac{-\dot{e}(t)}{\alpha} + \frac{K_P}{\alpha} e(t) + u(t-h), \tag{13}$$

The corresponding block diagram is shown in Figure 5.

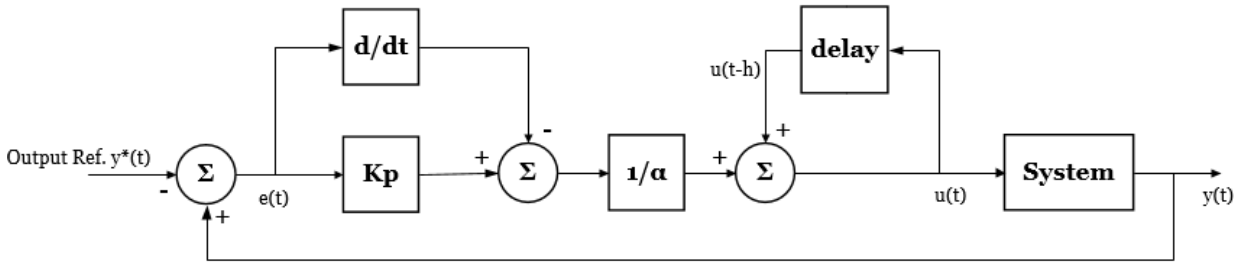


Figure 5. Intelligent-P Controller

A conventional continuous-time PI controller is shown as Equation (14),

$$u(t) = k_p e(t) + k_i \int e(\tau) d\tau \,, \tag{14}$$

A crude sampling of the integral $\int e(\tau) d\tau$ through a Riemann sum $I(t)$ leads to,

$$\int e(\tau)d\tau \approx I(t) = I(t-h) + he(t), \qquad (15)$$

Where h is the sampling interval. Then combining Equation (14) and (15) yields the discrete form of Equation (14),

$$u(t) = k_p e(t) + k_i I(t) = k_p e(t) + k_i I(t-h) + k_i he(t), \qquad (16)$$

Combing Equation (16) with

$$u(t-h) = k_p e(t-h) + k_i I(t-h), \qquad (17)$$

yields

$$u(t) = k_p (e(t) - e(t-h)) + u(t-h) + k_i he(t), \qquad (18)$$

In Equation (13), replace $\dot{e}(t)$ by $\frac{e(t)-e(t-h)}{h}$,

$$u(t) = -\frac{e(t)-e(t-h)}{\alpha h} + \frac{K_P}{\alpha} e(t) + u(t-h), \qquad (19)$$

Comparing Equation (18) and Equation (19), they are identical if

$$k_p = -\frac{1}{\alpha h}, \quad k_i = \frac{K_P}{\alpha h} \ . \qquad (20)$$

It should be emphasized that Equation (20) does not hold for continuous intelligent-P controller and conventional PI controller. This equivalence is strictly related to the sampling interval h.

(b) PID and intelligent-PD

The standard form of a continuous conventional PID controller is shown in Equation (21),

$$u(t) = k_p e(t) + k_i \int e(\tau)\, d\tau + k_d \dot{e}(t), \tag{21}$$

The velocity form of Equation (21) reads,

$$\dot{u}(t) = k_p \dot{e}(t) + k_i e(t) + k_d \ddot{e}(t), \tag{22}$$

Combining $\dot{u}(t) = \dfrac{u(t) - u(t-h)}{h}$ yields,

$$u(t) = u(t-h) + hk_p \dot{e}(t) + hk_i e(t) + hk_d \ddot{e}(t), \tag{23}$$

For $n = 2$ in Equation (4), and *intelligent-PD* controller is

$$u(t) = \frac{\ddot{y}^* - \hat{F} + K_P e + K_D \dot{e}}{\alpha}, \tag{24}$$

Similar to Equation (12), assuming $\hat{F} = \ddot{y}\ (t) - \alpha u(t-h)$, then

$$u(t) = -\frac{\ddot{e}(t)}{\alpha} + u(t-h) + \frac{K_P e(t) + K_D \dot{e}(t)}{\alpha}, \tag{25}$$

It is obvious that Equation (23) and (25) are identical when

$$k_d = -\frac{1}{\alpha h},\ \ k_i = \frac{K_P}{\alpha h},\ \ k_p = \frac{K_D}{\alpha h}\ . \tag{26}$$

Other correspondences between the conventional and intelligent PID controller are described in Table II.

		iP	iPD	iPI	iPID
PI	k_p	$-1/\alpha h$			
	k_i	$K_P/\alpha h$			
PID	k_p		$K_D/\alpha h$		
	k_i		$K_P/\alpha h$		
	k_d		$-1/\alpha h$		
PI^2	k_p			$-1/\alpha h$	
	k_i			$K_P/\alpha h$	
	k_{ii}			$K_I/\alpha h$	
PI^2D	k_p				$K_D/\alpha h$
	k_i				$K_P/\alpha h$
	k_{ii}				$K_I/\alpha h$
	k_d				$-1/\alpha h$

Table II: Correspondence between the gains of sampled classic and intelligent controllers [10].

3.4 Numerical derivation (ND) method

The intelligent-P controller shown in Equation (13) contains a derivative term. When the feedback signal is extremely noisy, the derivative calculation may introduce instability or important tracking errors. To avoid this kind of problem, numerical derivation (ND) method is applied in our work. This solution has already played an important role in model based nonlinear control and in signal processing, [14].

For a noisy feedback signal $e(t)$, the Taylor expansion at t=0 yields,

$$e(t) \approx p(t) = a_0 + a_1 t \ , \qquad (27)$$

where the first two items in the Taylor expansion are used to estimate $e(t)$. Using Laplace Transformation on both sides of Equation (27) and multiplying by Laplace operator *s*, it yields,

$$sE(s) = a_0 + \frac{a_1}{s}\ , \tag{28}$$

To eliminate a_0, multiply both sides by $s^{-2}\frac{d}{ds}$,

$$s^{-2}E(s) + s^{-1}\frac{dE(s)}{ds} = -\frac{a_1}{s^4}, \tag{29}$$

Equation (29) corresponds to Equation (30) in time domain by Inverse Laplace Transformation,

$$\hat{e}(0) = a_1 = -\frac{3!}{T^3}\int_0^T (T-2t)e(t)\,dt, \tag{30}$$

where T>0, is a quite small time interval window. This window is sliding in order to get the estimate at each time instant, [15].

Here is an example to show the de-noising effect of the numerical derivation method. Numerical derivative method and standard derivative block in Simulink are applied to calculate the first order derivative of a sine wave. The noise is assumed as a normally (Gaussian) distributed random signal, with zero mean and variance equal to 0.1. Window width T in Equation (30) is 50ms, and the sample time is 1ms. The result is shown in Figure 6.

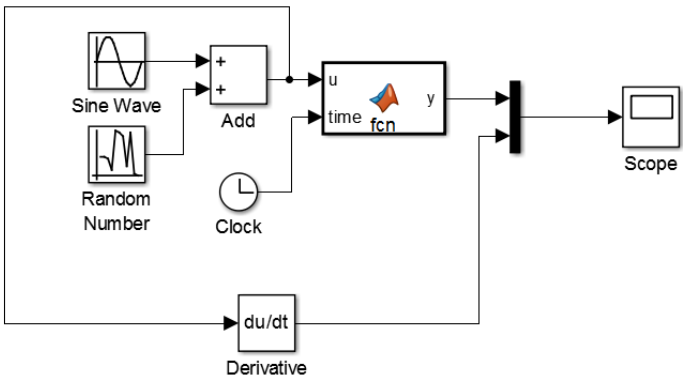


Figure 6. (a) Comparison of Two Derivative Methods

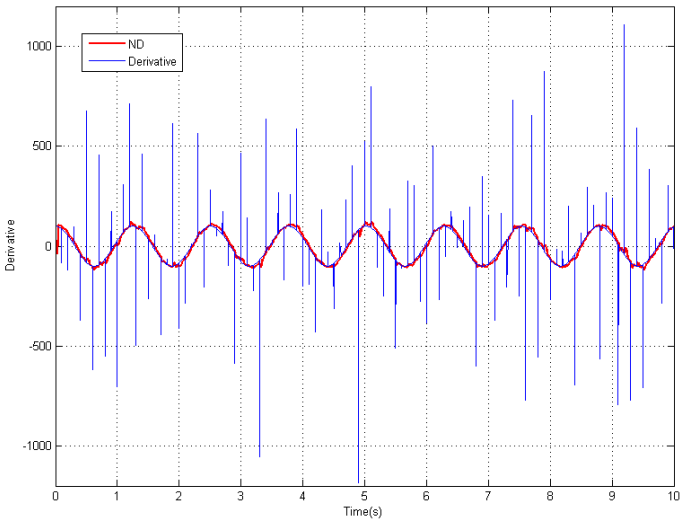


Figure 6. (b) Simulation Results of Two Derivative Methods

It is clear in Figure 6. (b) that the derivative calculated by numerical derivative method is less influenced by the noise. The de-noising effect of ND method is thus demonstrated in this simulation.

4. NS2 and the Co-Simulation Platform PiccSIM

4.1 Network Simulator 2

Network Simulator (Version 2), known as NS2, is a widely used open-source communication network simulator operating in Linux environment. It is an event-driven simulation tool, and provides substantial support for simulation of multicast protocols and IP protocols, such as UDP (User Datagram Protocol) and TCP (Transmission Control Protocol) over wired and wireless (local and satellite) networks. NS2 has many advantages that make it a powerful tool, such as

support for several algorithms in routing and queueing as well as the capability of graphically detailing network traffic.

NS2 began as a variant of the REAL network simulator, which was developed by UC Berkeley and Cornell University in 1989. In the past decades, NS2 has been developed and improved significantly. Up to today, NS2 is available on several different platforms such as FreeBSD, Linux, SunOS and Solaris.

NS2 consists of two languages: C++ and Object-oriented Tool Command Language (OTcl). C++ is generally used to define, write and use different algorithms and commands to process large sets of data. OTcl sets up the simulation by assembling and configuring the objects as well as scheduling discrete events, [16]. A framework called OTcl linkage links the OTcl codes to the implementation by C++, so users can write and compile new network protocols or models in C++, and manipulate them in OTcl. The basic architecture of NS 2 is presented in Figure 7.

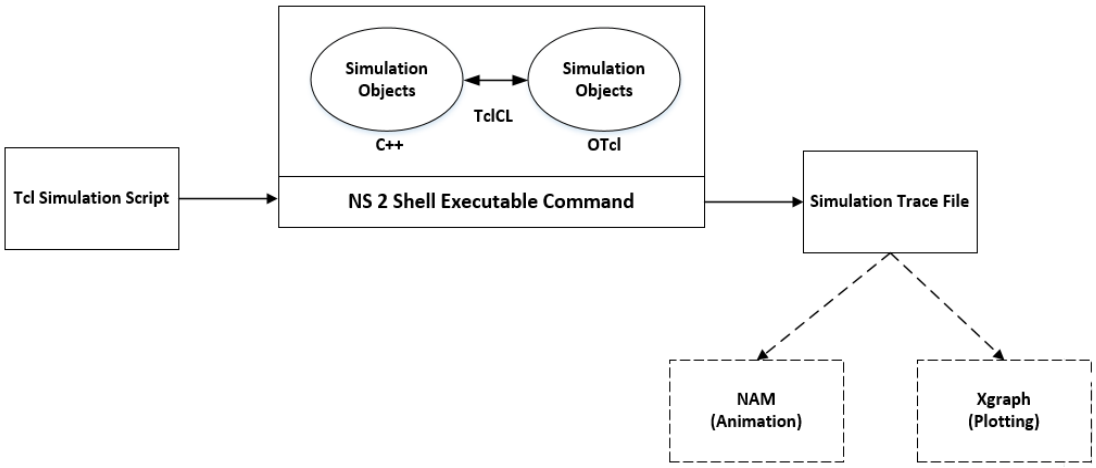


Figure 7. Basic Architecture of Network Simulator 2

As shown in Figure 7, NS 2 provides users with executable command whose input argument is the name of a Tcl simulation script. In most cases, a simulation trace file will be created at the

start of the simulation process, and the trace file can be used to plot graphs and calculate statistical results, such as the end-to-end delay, jitter, packet loss and throughput. NAM in Figure 7 is a Tcl based animation tool for viewing network simulation traces and real world packet traces. It is mainly intended as a companion animator to the ns simulator.

4.2 PiccSIM

4.2.1 Architecture of PiccSIM

PiccSIM stands for *Platform for integrated communications and control design, simulation, implementation and modeling*, [17]. It is a network and control co-simulation platform developed by Wireless Sensor Systems group in Aalto University, Finland. PiccSIM consists of a control design and simulation tool, Matlab/Simulink, and a network simulation tool, NS2. The detailed analysis and extended application notes for PiccSIM are described in [18] and [19], where a sensor network-based indoor building automation simulation case is studied.

The hardware requirement of this co-simulation platform consists of one computer running Matlab with the toolchain, and another computer running NS 2 on Linux. The embedded toolchain in Matlab coordinates the control simulator and network simulator through XML (Extensible Markup Language) formatted messages. The communication between the two computers is shown in Figure 8. The NS 2 computer runs a java program called NS2_command to listen to incoming TCP connections, control NS 2 and reply status back to the toolchain. PiccSIM toolchain uses a Java program, NS2_submitter to send control commands to the NS 2

computer. In addition, NS2_submitter is also used to send NS2 network configuration scripts (TCL file) from Toolchain to NS2 computer, [17].

The toolchain starts Matlab and NS 2 at the same moment with a button click on a graphical user interface, shown in Figure 9. The toolchain first sends the TCL file to NS 2 computer and starts the network simulator. Then it starts the Matlab/Simulink model simulation. When the co-simulation is finished, the Matlab computer can retrieve network simulation results from the NS 2 computer.

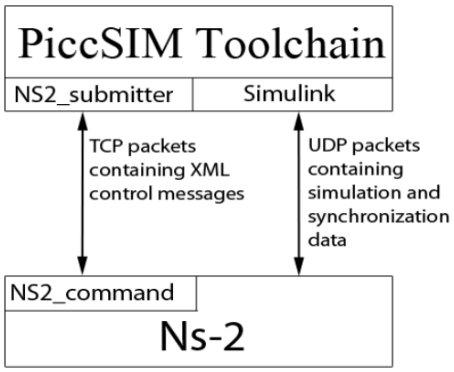


Figure 8. Communication Structure of Toolchain [17]

It can be concluded that PiccSIM Toolchain ties the design, implementation and simulation into one framework, and all the functions mentioned are available through a graphical user interface (GUI), presented in Figure 9.

The simulation models panel in Figure 9 allows users to create and load target Simlink model and the ns-2 network file (TCL file). By clicking the network setting button, network configurations can be set in a graphical window, where we can drag nodes and create links easily. The TCL codes can be generated automatically after saving the graphical network configuration.

Some important communication settings such as the node positons, network protocols and other simulation parameters, can also be set in this graphical environment. However, this function still has some unpredictable bugs on occasion, so we usually create our TCL file independently instead of using this code generation tool.

The version 0.80 of PiccSIM available to researchers currently does not include functions like automatic code generation and controller tuning. The Simulation panel includes two buttons for starting and stopping the simulation. They can start and stop the Simulink simulation and NS 2 simultaneously. The Get ns-2 results button can show some statistical results like delay and packet drop rate based on the trace files. Last but not least, before we start the simulation, the ns-2 IP must be set as the IP address of the NS 2 computer.

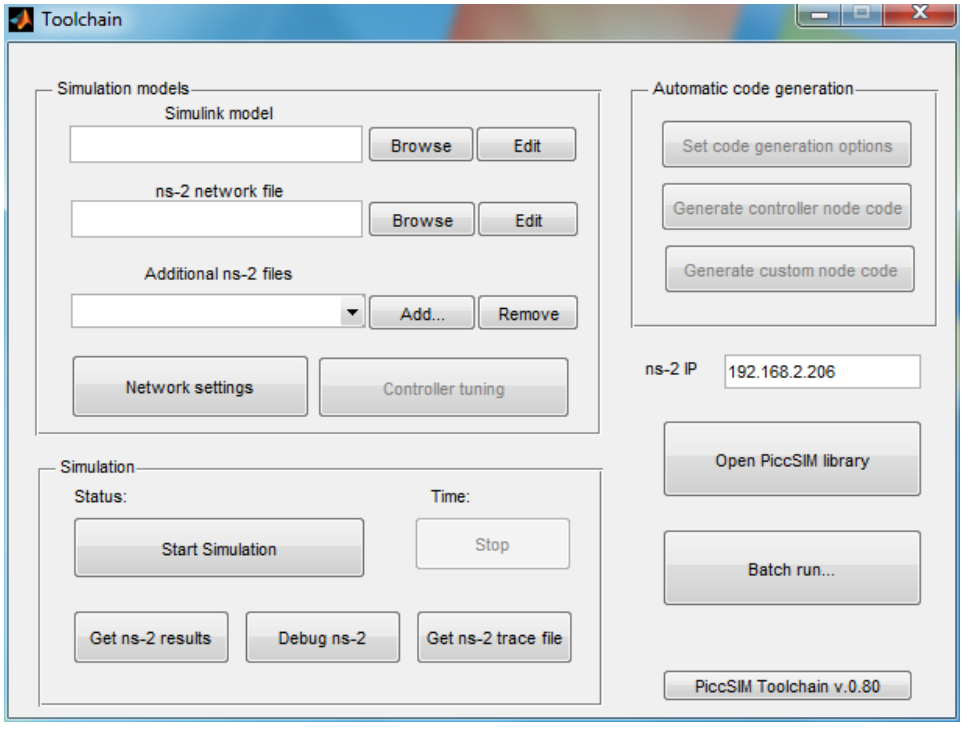


Figure 9. Toolchain Graphical User Interface

The PiccSIM library, presented in Figure 10, is accessed by clicking the Open PiccSIM library button. The library contains blocks for sending and receiving packets over the simulated network. The target Simulink model must contain these blocks for sending and receiving signals. The library also includes time synchronization mechanism block, which must exist in the Simulink model as well. The details of nodes settings can be found PiccSIM Manual.

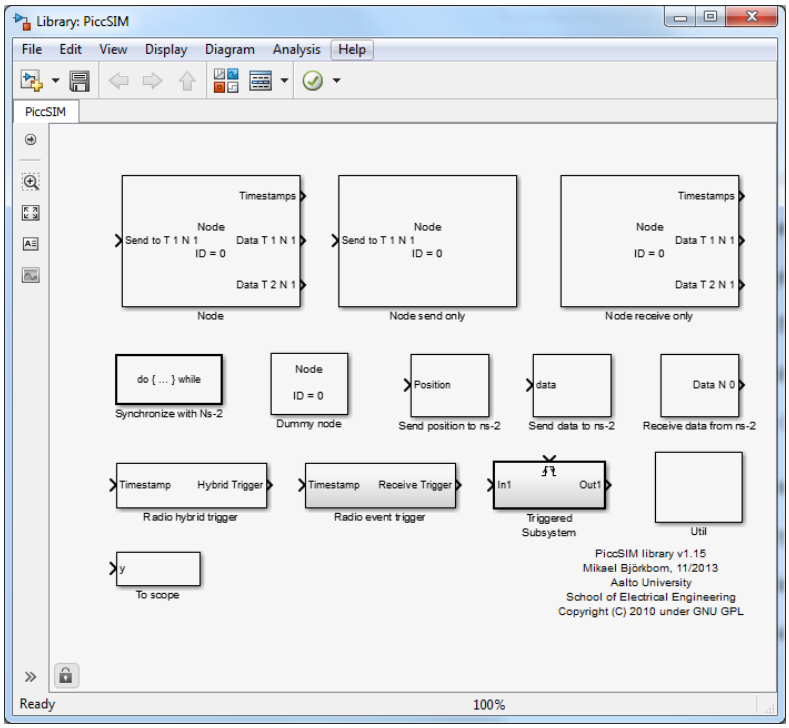


Figure 10. PiccSIM Library

4.2.2 Time Synchronization of PiccSIM

In the co-simulation between two different domains, time synchronization is extremely important. Even a slight difference can make the results inaccurate. PiccSIM solves this problem by modifying the network simulator scheduler to keep pace with the Simulink model, presented in

Figure 11. The benefits of this modification are the simulations do not need to run in real-time, so the simulations take less time but get more accurate results, [17].

The time synchronization process works as follows: Simulink sends NS 2 an UDP packet, containing the current simulation time in milliseconds. NS 2 then simulates the communication network up to that time, replies to the Simulink and waits for the next synchronization packet. After receiving the reply from NS2, Simulink will advance one step in simulation and send a new UDP packet to NS 2. In order to avoid interference, time synchronization UDP has a dedicated UDP port, 22301, which should be stated in the TCL file. In addition to the synchronization packets, the simulation data packets are also exchanged between Simulink and NS 2 through UDP ports 223XX.

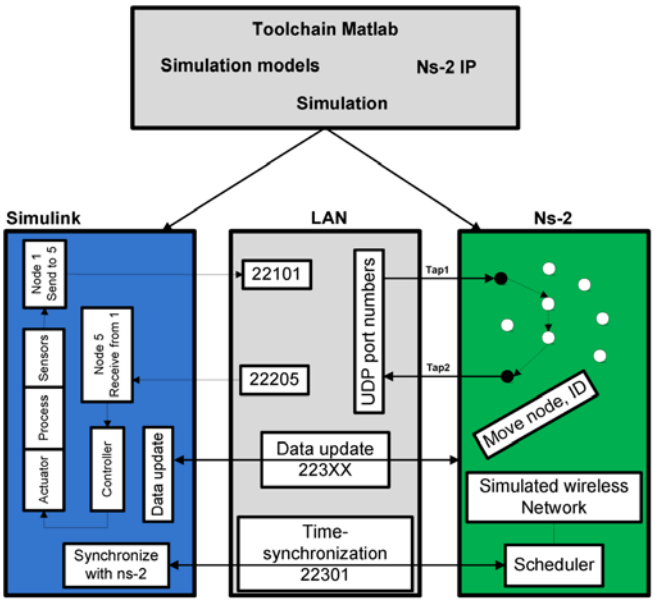


Figure 11. PiccSIM Time Synchronization

5. Co-Simulation Model

5.1 Power System Simulation Model

The power system model used in this paper is the New England 39 bus system (NE39 Bus), which contains 39 buses, 19 loads, and 10 generators. The Simulink model of this power system is presented in Figure 12.

5.1.1 New England 39 Bus Power System

General concepts of this model are shown as follows:

- Three phase PI section line is used to represent the transmission line. Positive- and zero-sequence resistance, inductance, capacitance, and line length can be set in this block.
- All of the loads are represented by three phase parallel RLC load (Y Grounded) in PQ type so that both the active and reactive demands can be specialized.
- Three phase transformer blocks are used to simulate transformers. Y-Y configuration is set for interconnecting transformers, and Δ-Yg configuration is applied for generator transformers.
- Generators are all modeled by three phase synchronous generators in dq rotor reference frame.
- Stream turbine and governor blocks are used for all generators.
- Each generator is equipped with a Multi-Band Power System Stabilizer (MBPSS). Mode operation is set as simplified setting (IEEE Std 421.5).

Each generator in Figure 12 represents an aggregation of a large number of generators. This system is widely used for small signal stability studies and dynamic stability analysis, [20].

All the generators in Figure 12 have a same value of nominal power, 1000MVA. Except for G1, all other generators are connected to the NE 39 Bus System via a Δ-Yg three-phase transformer to increase the output voltage to the transmission line-line voltage (Vrms). The nominal line-line voltage of G1 is 345KV, equal to the nominal line-line voltage of Bus 39, so G1 can be connected to the power grid directly. The nominal line-line voltages (Vrms) of all generators and buses are shown in Table III. Based on the geographical distribution, the whole system is divided into 3 control areas, with 4 transmission lines connecting each other. Power measurement devices are deployed over the transmission lines to monitor the are-to-area active power flows.

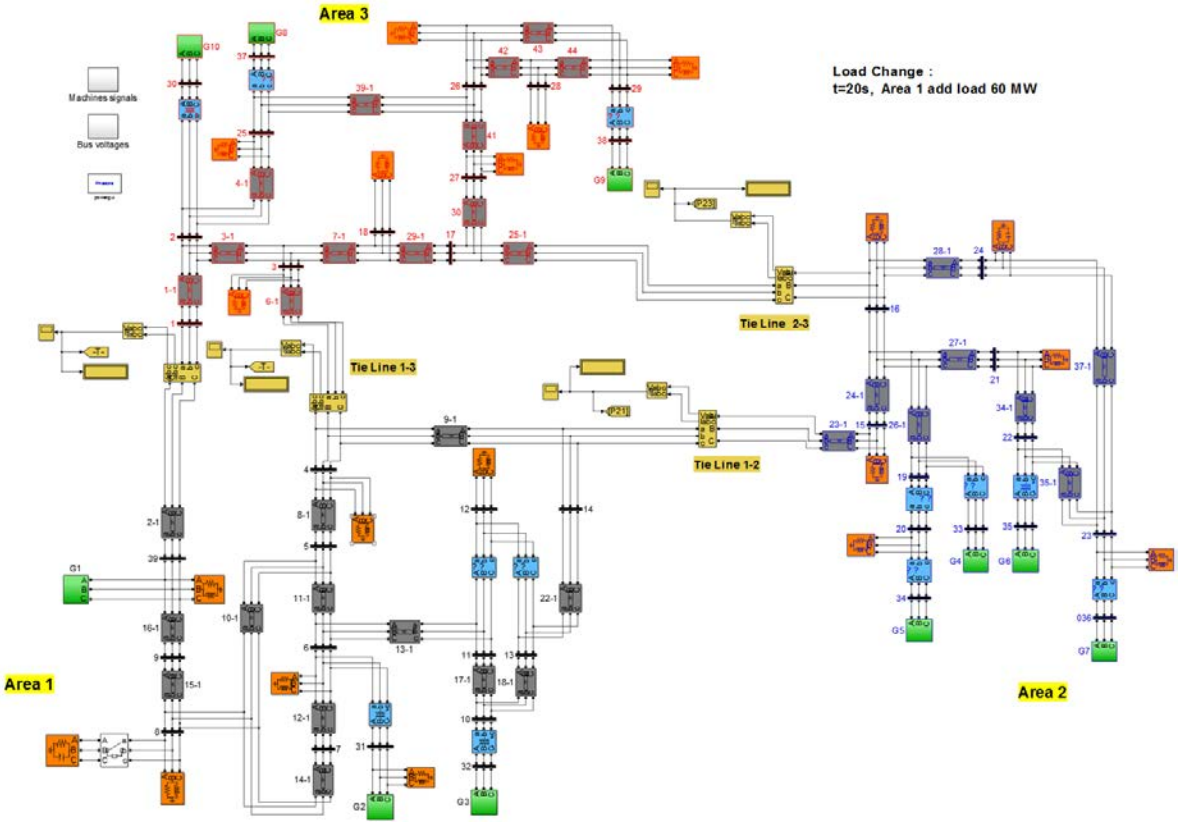


Figure 12. NE 39 Bus System

Generators	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
Nominal Voltage (KV)	345	22	22	22	22	22	22	22	22	22

Table III (a): Nominal Line-Line Voltage of Generators

Buses	Bus 1 - Bus 11	Bus 12	Bus 13 - Bus 29	Bus 30 - Bus 38	Bus 39
Nominal Voltage (KV)	345	230	345	22	345

Table III (b): Nominal Line-Line Voltage of Buses

As shown in Figure 4, for AGC implemented in a multi-area power system, the area control error (ACE), $ACE = \Delta P + \beta \Delta f$, is computed based on the measurements of tie-line active power exchange and frequency deviation. Since frequency is a global signal that varies little throughout the power system, frequency values can be collected locally. Therefore, the power system needs dedicated communication channels to transfer the measurement signals of tie-line power flows from the measurement devices to the control center, and to send the control signals from the control center to distributed generators.

Although wireless communication has advanced rapidly [21], considering the requirement of the reliable long distance transmission, the wired communication is more suitable in AGC application today. Our work studied the effect of wired communication delays on the AGC performance. The co-simulation is implemented on the platform, PiccSIM, with NE 39 Bus model built in Matlab/Simulink (Figure 12), and communication network simulated in NS 2.

The control center is also referred to as the dispatch center. Information pertaining to frequency, tie-line flows, and unit MW loadings is transferred to this location, where the control signals to generators are determined by a digital computer. The normal form of the control signals is raise

or lower pulses of varying lengths. The control unit in generation plants then changes the reference setpoints up or down in proportion to the pulse length, [12]. All control areas are regulated in a similar manner, but they are independent to each other. In other words, AGC in the multi-area system is “area-wise decentralized”.

The participation factor (PF) of each generator is shown in Table IV. The values are determined based on the operation state of steady state when there is no load change or the power generation allowance. In practice, the participation factors are updated continuously based on an economic allocation algorithm, such as the equal incremental principle.

Area 1	Area 2	Area 3
$PF_{G1} = 0$	$PF_{G4} = 0$	$PF_{G8} = 0.25$
$PF_{G2} = 0.7$	$PF_{G5} = 0.6$	$PF_{G9} = 0$
$PF_{G3} = 0.3$	$PF_{G6} = 0$	$PF_{G10} = 0.75$
	$PF_{G7} = 0.4$	

Table IV. Participation Factors of Generators

5.1.2 AGC – Classic PI Controller

This thesis studied the event of a sudden load change at t=20s, and AGC is enabled at the same moment. A classic PI controller is applied at the control center and its design in Simulink is shown in Figure 13.

The scheduled net interchange active power for the three control areas are shown as:

- Area 1 Net Interchange Power : -380.76MW (Import)
- Area 2 Net Interchange Power : 315.76MW (Export)
- Area 3 Net Interchange Power : 65MW (Export)

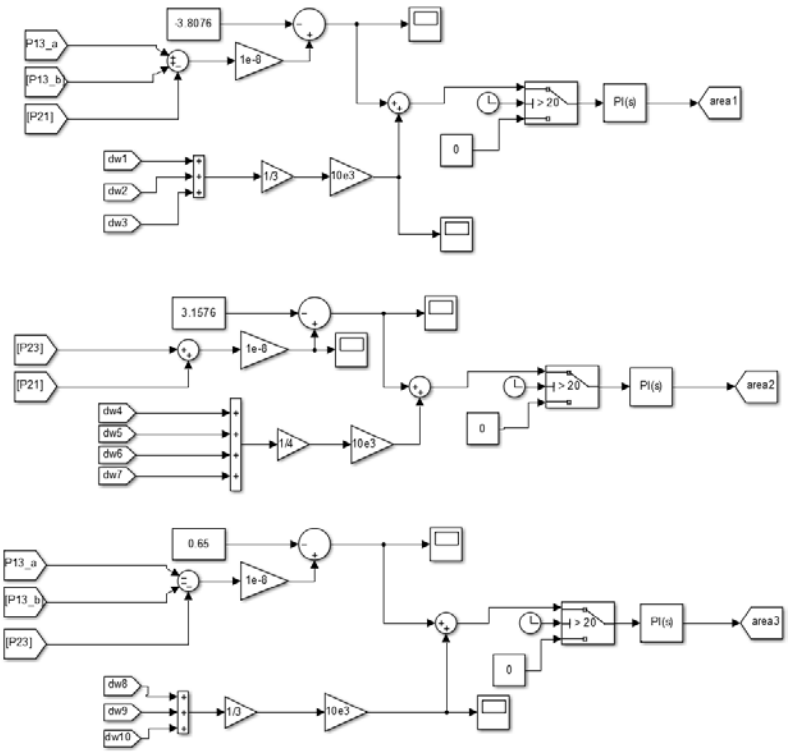


Figure 13. PI Controller for AGC

The tie-line power measurements signals P_{13_a} , P_{13_b} , P_{23} , P_{21} corresponds to the Power Measurements 1 to 4 respectively, shown in Figure 15. The frequency deviations are collected from all generators in each control area, and a simple average calculation is applied for de-noising. As mentioned before, the frequency deviation can be monitored locally. But in this model, the synchronous machine block in Simulink already provides the measurement of rotor speed deviation, which is equal to the frequency deviation in per unit value (p.u.). Therefore, the ACE signal is calculated based on the average of rotor speed deviations of all generators in a single control area. The bias factor (β) is set as 1000 for all three areas. This value plays a critical role to balance the importance of frequency deviation and interchange power deviation in

ACE, since the magnitude of frequency deviation is often too small, compared with the power deviation.

The parameters of the PI controllers for three control areas are shown in Table V.

Parameters	Proportional	Integral
Area 1	-0.01	-0.005
Area 2	-0.01	-0.01
Area 3	-0.02	-0.005

Table V. Parameters for PI Controllers

5.1.3 AGC – Intelligent-P Controller

Based on the discussion in 3.3 (a), intelligent-P controller and conventional PI controller are identical when appropriate parameters are set. In this model, $h = 10ms$ and the corresponding intelligent-P controller in Simulink is presented in Figure 14.

The constant 0 is the reference trajectory of ACE signal, and $y(t)$ is the practical time-varying ACE signal. Like the conventional PI controller, intelligent-P controllers are also enabled at t=20s. The design is very similar to Figure 5 with k_p and α determined from Equation (20). Therefore, the controller designs in Figure 13 and 14 are equivalent, and their performances under a sudden load change event are tested and compared in this thesis.

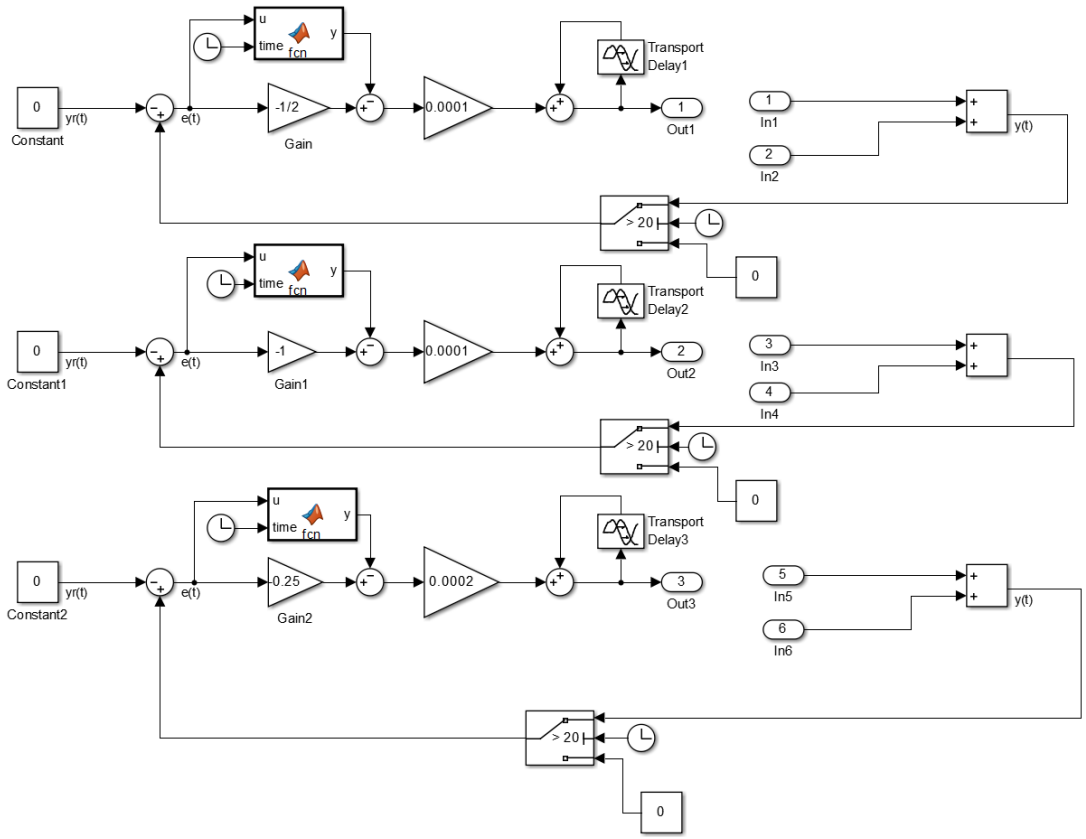


Figure 14. Intelligent-P Controller for AGC

5.2 Communication Network

As shown in Figure 15, there exist 7 links in the communication network. Link 1 to link 4 are the transmission channels to transfer the tie-line flow measurements signals to the control center, and link 5 to link 7 are channels to send control signals from the control center to each generator. In a specified control area, the distribution of generators is relatively concentrated. So in this model, we used one link to represent the transmission channels from the control center to all generators in one control area to simplify the network settings.

The communication network is simulated in NS 2. The corresponding parameters are set in the TCL file with send interval=50ms, packet size varying from 1000 to 3000 bytes, bandwidth=0.03Mb and CBR (Constant Bit Rate) over UDP (User Datagram Protocol) agents.

Take link1 as an example. Node 0 is the tie-line flow measurement device location shown in Figure 15 as Power Measurement 1. Node 4 is the receiving end at the control center. The link is duplex, which means the data can be transmitted in binary directions. Bandwidth is 0.03Mb, delay time is set as 500ms, and the queue mechanism is DropTail. In this mechanism, each packet is treated identically. When the queue is filled to its maximum capacity (queue limit), the newly incoming packets will be dropped until queue have sufficient space to accept incoming traffic. The loss of packets causes an enter delay which decreases the network throughput, and thus increases its congestion window. Many other mechanism such as RED (Random Early Discard), FQ (Fair Queuing) and SFQ (Stochastic Fair Queuing) are also available options.

It should be noted that the real delay is not a constant as we set but a variable depending on the network traffic congestion during the simulation, which will be explained in the following discussion.

```
$ns duplex-link $n(0) $n(4) 0.03Mb 500ms DropTail

set null_0 [new Agent/Null]
$ns attach-agent $n(4) $null_0
set udp_0 [new Agent/UDP]
$ns attach-agent $n(0) $udp_0

set cbr_0 [new Application/Traffic/CBR]
$cbr_0 set interval_ 50ms
$cbr_0 set packet_size_ 1000
$cbr_0 attach-agent $udp_0

$ns connect $udp_0 $null_0
$udp_0 set fid_ 4
$ns queue-limit $n(0) $n(4) 20
```

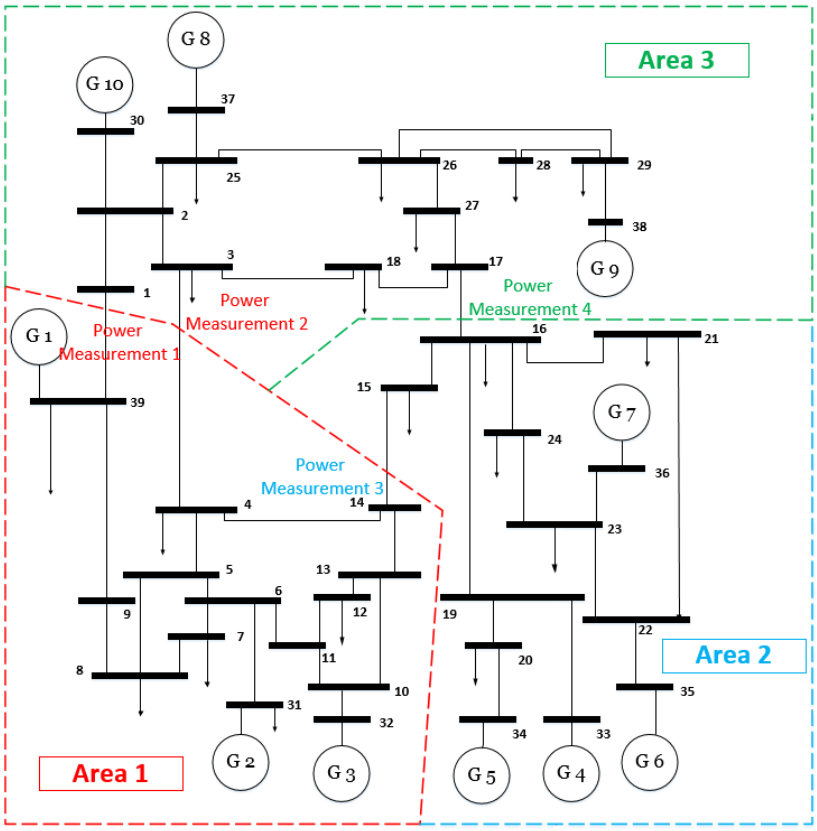


Figure 15. (a) Tie-Line Power Measurement Locations

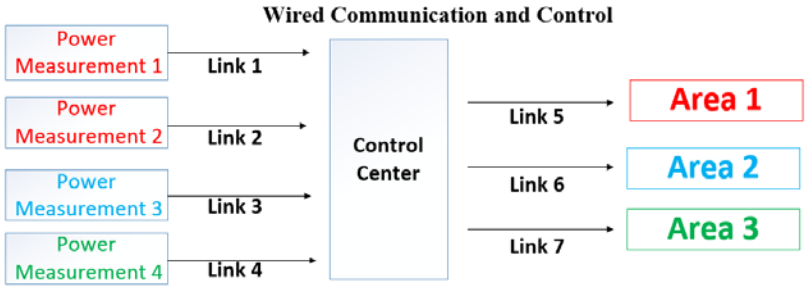


Figure 15. (b) Communication Links Configuration

Two kinds of communication applications are involved in NS 2, CBR over UDP and FTP (File Transfer Protocol) over TCP (Transmission Control Protocol). TCP is a dynamic reliable congestion protocol, which is used to provide reliable transmission of packets from one host to

another by sending acknowledgements on proper transfer or loss of packets. Thus TCP requires bi-directional links in order for acknowledgements to return to the source. File Transfer Protocol (FTP) is a standard mechanism provided by the Internet for transferring files from one host to another. FTP contains both data transfer and control information, [22]. Therefore, FTP uses the services of the TCP.

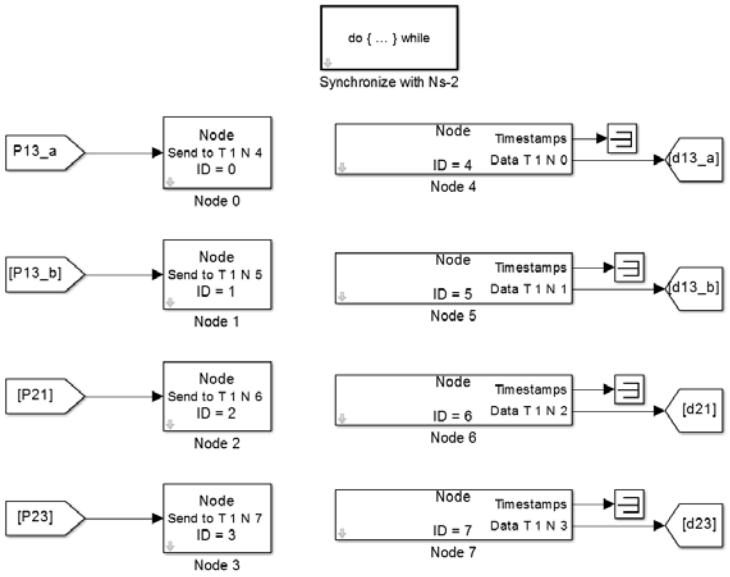


Figure 16. Communication Settings in Simulink

Constant Bit Rate (CBR) means that the rate of the output data is a constant. CBR is useful for streaming multimedia content on limited capacity, and it can take advantage of all of the capacity. The User datagram Protocol (UDP) is a main protocol of the Internet protocol suite. UDP helps the host to send messages in the form of datagrams to another host without requirement for channel transmission setup. UDP provides an unreliable service, and the datagrams may arrive out of order, appear duplicated, or go missing without notice. UDP assumes that error checking and correction is either not necessary or performed in the application, avoiding the overhead of

such processing at the network interface level, [22]. UDP is often applied in time-sensitive applications, and that's why UDP is more suitable than TCP in our co-simulation research.

In the TCL file used in our model, a UDP agent is attached to node 0 as a sender, and a Null agent is attached to node 4 as a receiver. The packets are sent from node 0, and Null agent frees the packets received. The CBR traffic generator is attached to the UDP agent. `$udp_0 set fid_ 4` means the flow ID (fid) for data transferred in link 1 is marked as 4 in the trace file. `$ns queue-limit $n(0) $n(4) 20` sets the queue length limit to be 20.

In this thesis, CBR over UDP is applied since the requirement of real time data exchange between NS 2 and Matlab/Simulink. The communication settings in Matlab/Simulink are shown in Figure 16.

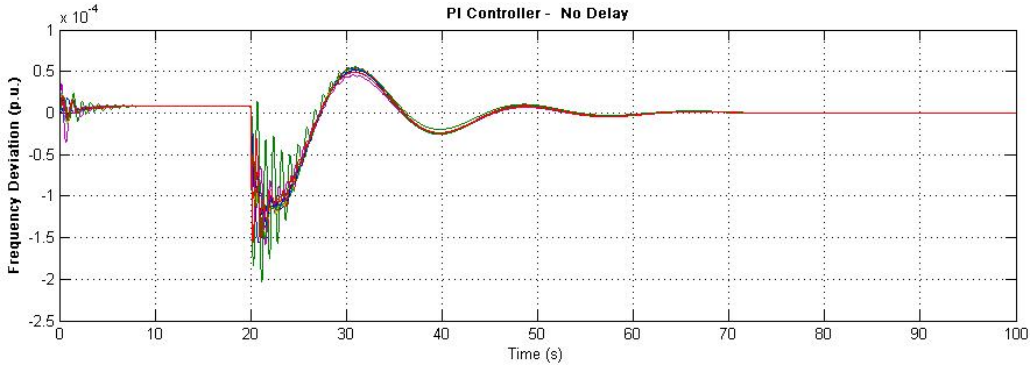
5.3 Simulation Scenarios

The event of interest consists of a sudden load increase of 0.06 p.u. (60MW) occurring at Bus 8 in area 1 at t=20s, and AGC is enabled at the same moment. In this thesis, six different scenarios are considered, and compared with the scenario where the communication delay is ignored. The first two scenarios consider the delay of tie-line power flow measurements and the delay from the control center to distributed generators separately. Four other scenarios simulate the communication delay in all links. The performance of conventional PI controller and intelligent-P controller are compared, and the influence of delay over the tie-line power measurements as well as delay over the control signals to generators is analyzed in detail-.

Considering the instability the differentiator may introduce, estimation of Equation (30) is applied with T=50ms, and the sample time is 1ms, presented as the Matlab function block in Figure 14. In Equation (30), $\hat{\dot{e}}(0) = a_1 = -\frac{3!}{T^3}\int_0^T (T-2t)e(t)\,dt$, the first order derivative of $e(t)$ at t=0 is estimated by the value of $e(t)$ in the time interval [0,T]. However, in practice, the system is always real-time and causal, it is impossible to get measurements ahead of the current moment. So Equation (30) is transformed to a different format so that we can estimate $\dot{e}(t_x)$ based on the value of $e(t)$ in the past time interval $[t_x\text{-T}, t_x]$.

6. Simulation Results and Analysis

This chapter presents the co-simulation results: frequency deviation for all 10 generators when PI and intelligent-P controller are applied respectively under 6 different scenarios as mentioned before, and the variant delay time over the 7 communication links. The performances of PI and intelligent-P controller are compared, and the effects of delay occurring on different links are studied as well.



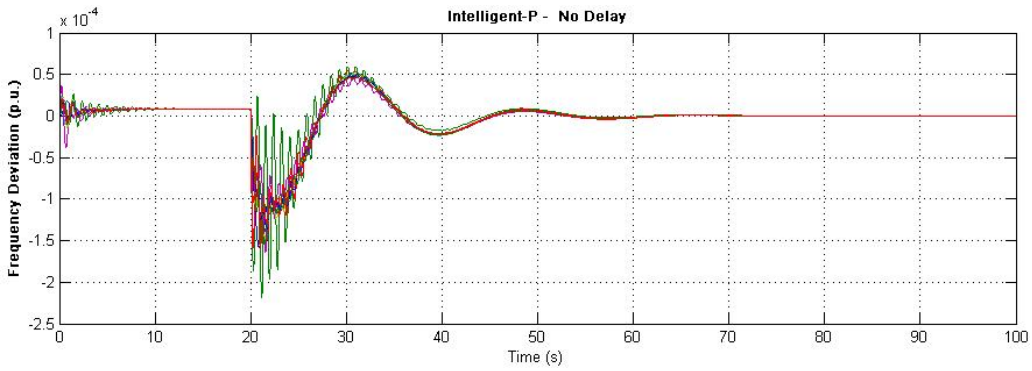


Figure 17. No Communication Time Delay

Figure 17 shows the frequency deviation of 10 generators with no communication time delay both when PI controller and intelligent-P controller are applied at the control center. The sudden load change occurs when t=20s at Bus 8 in area 1, and at the same time, AGC is enabled. It can be seen that the frequency begins to decrease at once. Then owing to the function of AGC, the setpoints of generators in area 1 participating in AGC are increased, which means the steady-state speed droop characteristics are adjusted until the generators can supply enough output power with frequency maintaining at the nominal value. In Figure 17, the settling time needs about 60s, and the oscillation of the frequency deviation is limited in $[-2.5*10^{-4}, 0.5*10^{-4}]$ (p.u.). It can also be seen that, under this scenario, PI controller and intelligent-P controller perform equivalently.

Figure 18 and 19 consider the delay deriving from area-to-area power flow measurement devices to the control center and from the control center to control areas, separately. Because of the limit of bandwidth, delays are not constant but variable depending on the network traffic congestion. The average delays of the four links to transfer tie-line flow measurement signals and of the three links from the control center to generators are Link1=7.010s, Link2=6.335s, Link3=6.334s, Link4=6.336s, and Link5=1.519s, Link6=1.385s, Link7=1.982s. The corresponding figures of

delay versus time are presented in Figure 20. Clearly, the communication delay has a non-negligible influence on the performance of AGC.

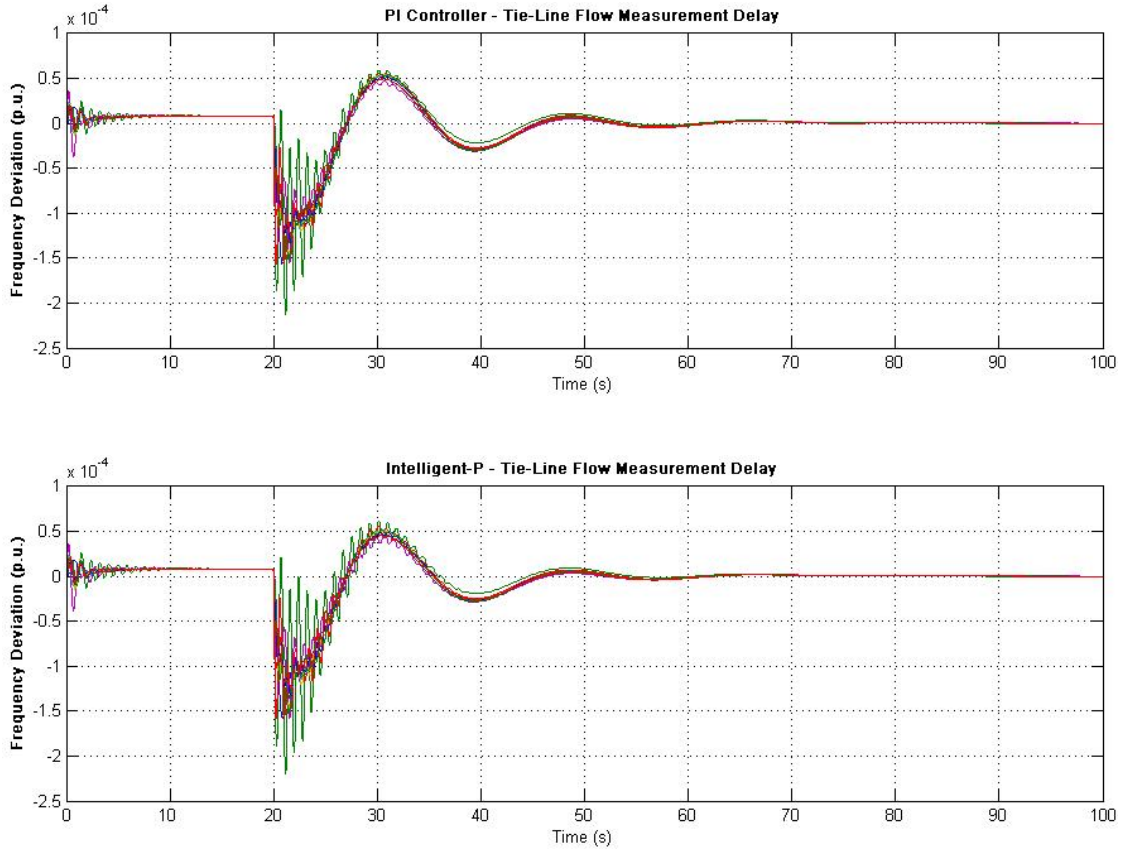


Figure 18. Tie-Line Flow Measurements Delay

Comparing Figure 18 and 19, the system is obviously much more sensitive to the delay from control center to generators, where even 1s ~ 2s time delay can result in long settling time and large frequency overshoot. This difference of delay sensitivity will be analyzed later, and based on this conclusion, a new control mechanism of AGC can be proposed. Secondly, intelligent-P controller performs nearly the same as PI controller when the resulting frequency oscillation is small in Figure 18, and slightly better in Figure 19 with larger frequency oscillation. This improvement is quite helpful in practice since large frequency oscillation can result in instability of the whole power system because of protection-driven generator tripping.

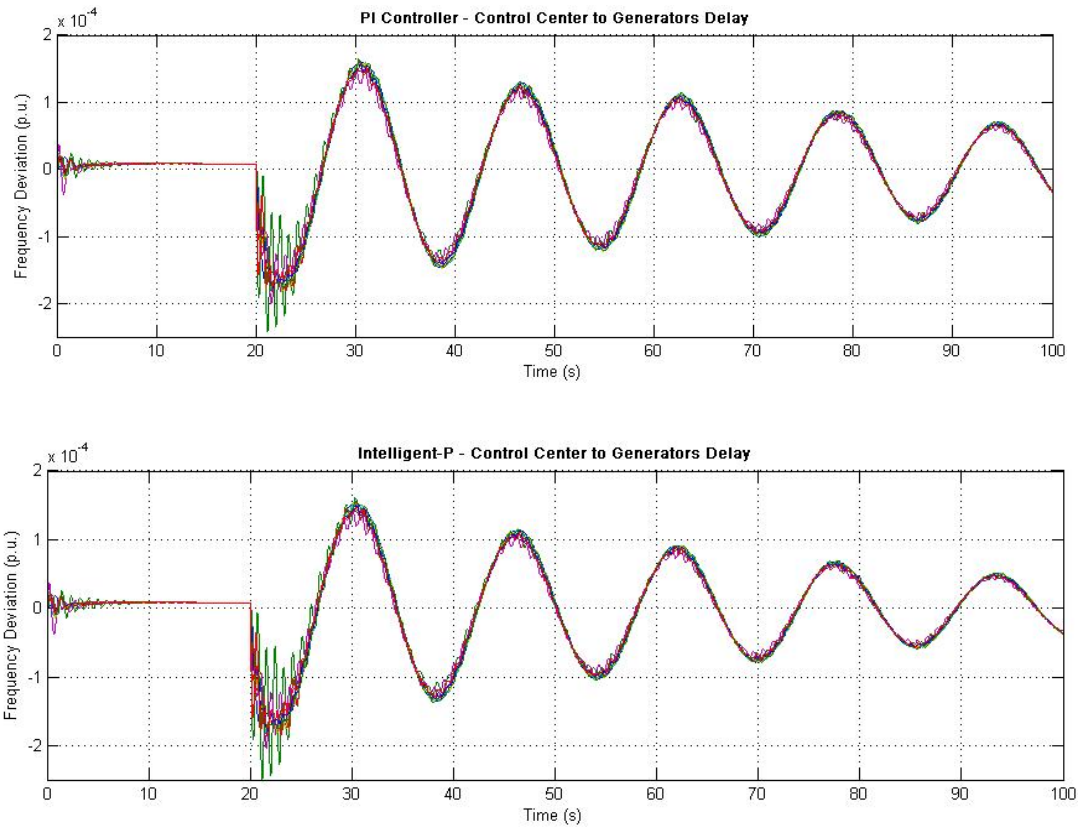


Figure 19. Control Center to Generators Delay

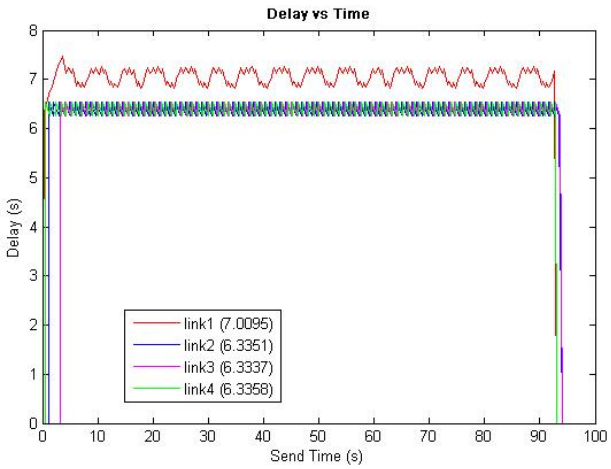


Figure 20. (a) Delay of Power Measurements

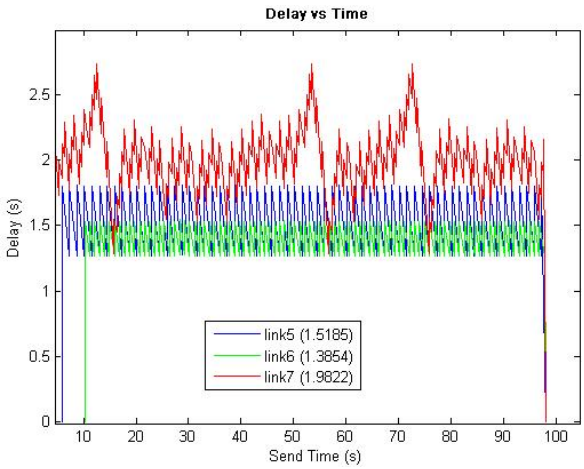


Figure 20. (b) Delay of Control Signals

In Figure 20, the delay of all 7 links keeps to be zero when the co-simulation is just started. This is because the staring moments of CBR are set differently on purpose in the TCL file in order to avoid the network traffic getting congested suddenly, shown below.

```
#Start up the sources
$ns at 0 "$cbr_0 start"
$ns at 1.1 "$cbr_1 start"
$ns at 3.2 "$cbr_2 start"
$ns at 0.5 "$cbr_3 start"
$ns at 5.9 "$cbr_4 start"
$ns at 10.2 "$cbr_5 start"
$ns at 2.5 "$cbr_6 start"
```

Besides, the delay close to the end of co-simulation (t=100s) is also shown as zero in Figure 20. It should be noted that the horizontal axis is send time. Because of the transmission delay, the packets sent close to t=100s have not been received yet when the simulation ended. Therefore, the delay is assumed zero under this circumstance. When calculating the average delay, this part should be ignored.

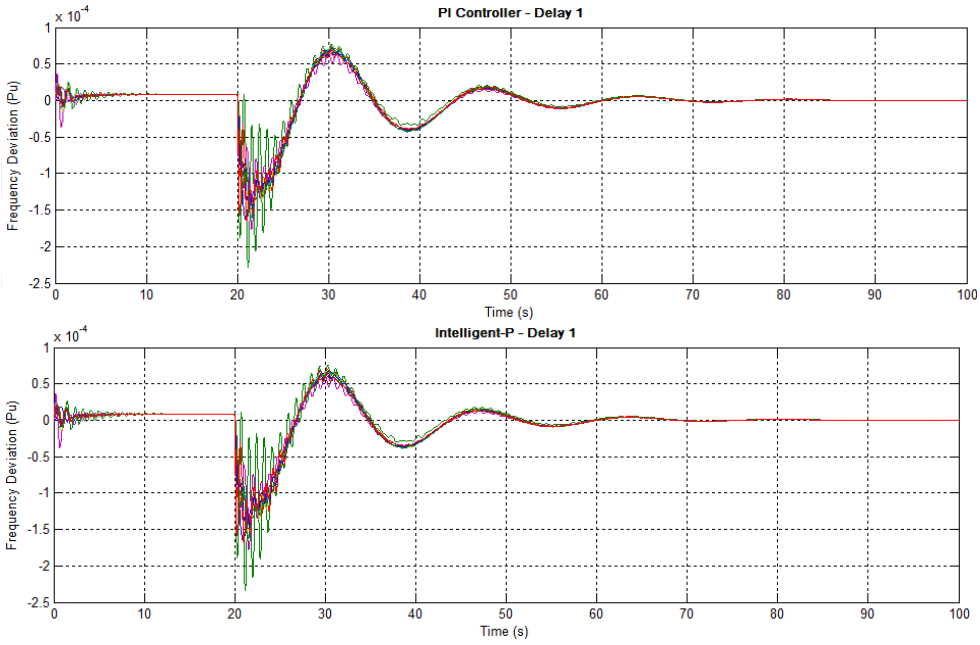


Figure 21. Communication Time Delay 1

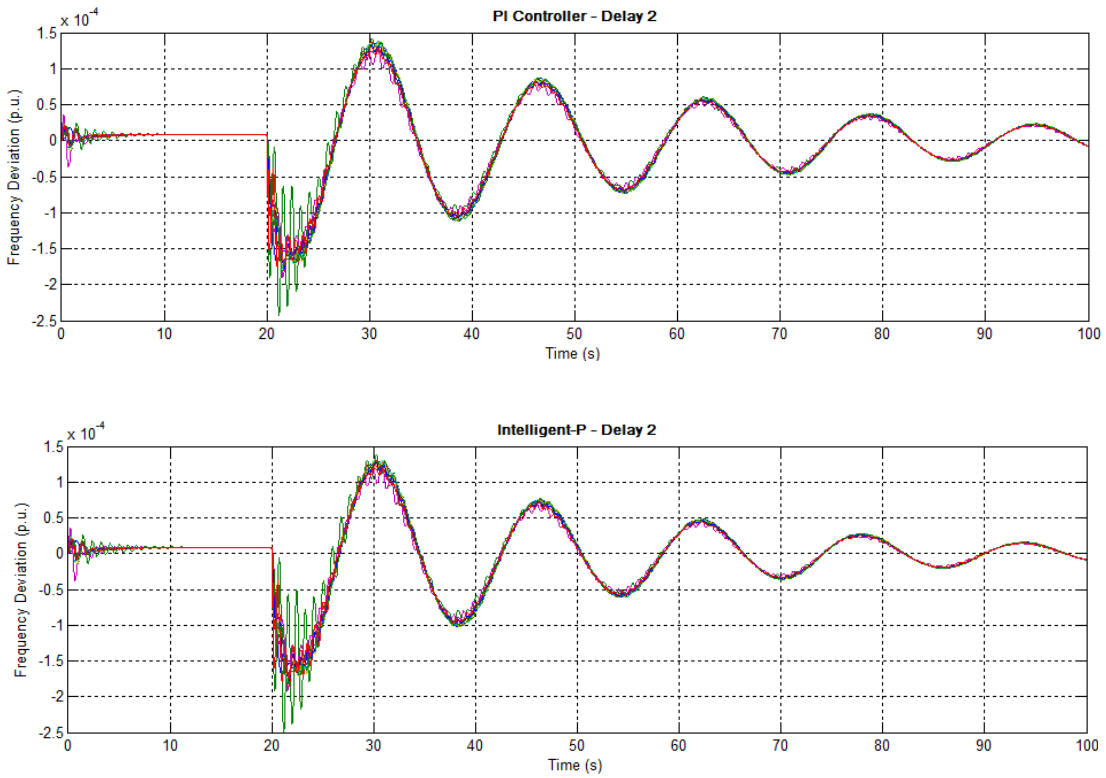


Figure 22. Communication Time Delay 2

Links	1	2	3	4	5	6	7
Delay 1 (s)	5.523	3.763	3.565	5.001	0.529	0.395	0.987
Delay 2 (s)	1.053	0.397	0.397	0.396	1.518	1.187	1.635
Delay 3 (s)	5.523	4.357	4.158	4.357	1.775	1.880	2.230
Delay 4 (s)	5.523	5.347	4.554	4.852	1.766	1.583	1.982
Loss Rate	4.71%	8.61%	8.48%	8.61%	8.27%	7.91%	4.61%

Table VI. Average Delay and Loss Rate

Figure 21-24 simulate the communication delay in all links, and Figure 25 shows the delay of all links for the four scenarios. The average delay and packets loss rate are shown in Table VI.

The system can track the load change, and has an acceptable performance in Figure 21 and Figure 22. The system maintains stable with communication delay of all links. The performance

of AGC in Figure 21 is better than Figure 22. It can be seen in Table VI the communication delay of the control signals (Link 5 to 7) in the Communication Delay 2 is obviously greater than the Communication Delay 1. Since this closed-loop control system is more sensitive to the delay of the control signals, the performance of Communication Delay 2 becomes worse than Communication Delay 1 with larger frequency deviation and longer settling time, even though its delay of power measurements is much smaller than Communication Delay 1.

However, it can be seen in Figure 23 that large communication delays can make the system unstable. For large frequency variations, intelligent-P controller still performs better than the PI controller like mentioned above. Looking at Figure 24, the frequency deviation with the PI controller settles to a nearly constant oscillation while for the intelligent-P controller, the system remains stable though it may need a long settling time to reach the steady state.

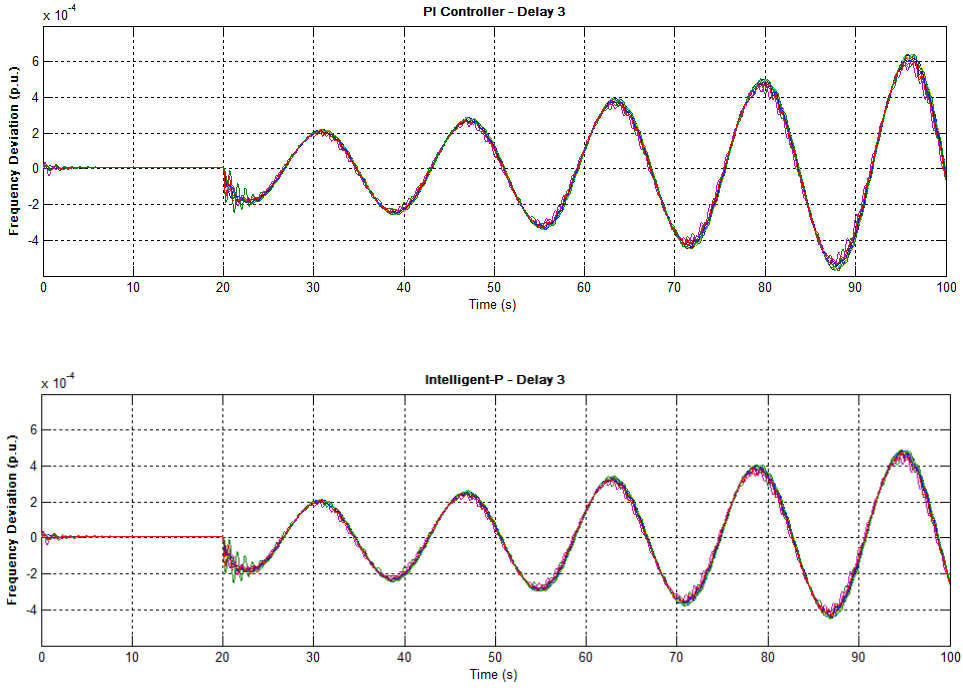


Figure 23. Communication Time Delay 3

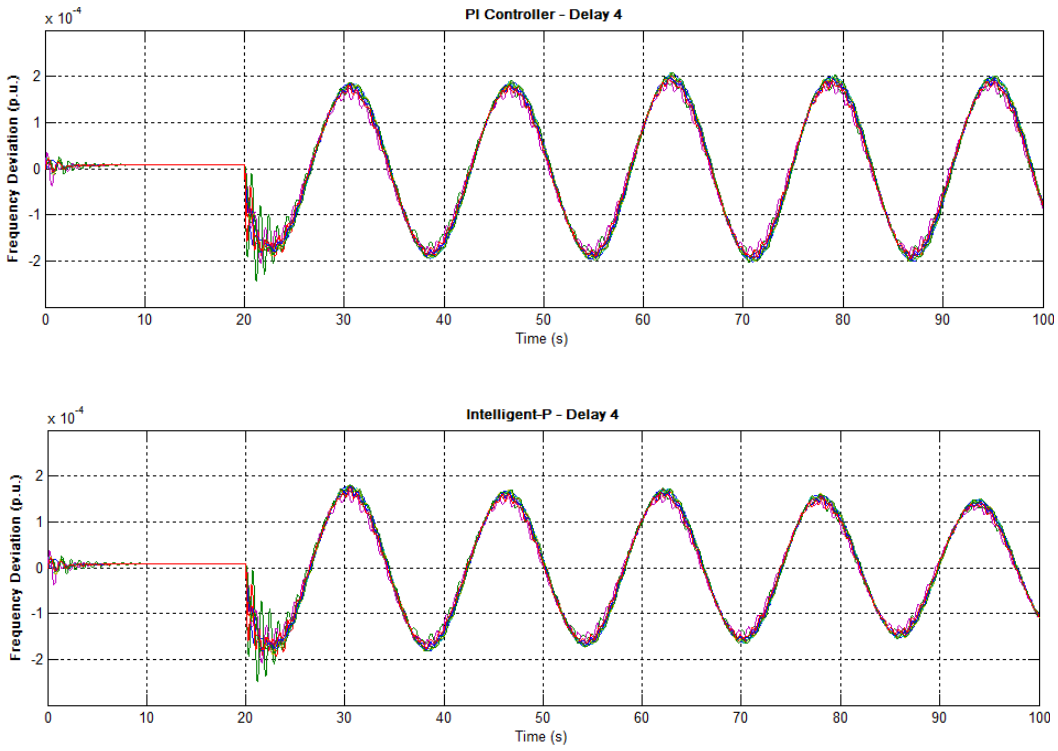
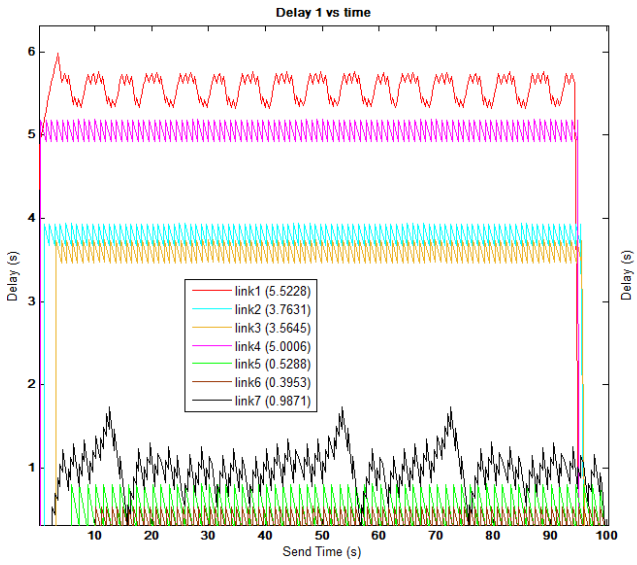


Figure 24. Communication Time Delay 4

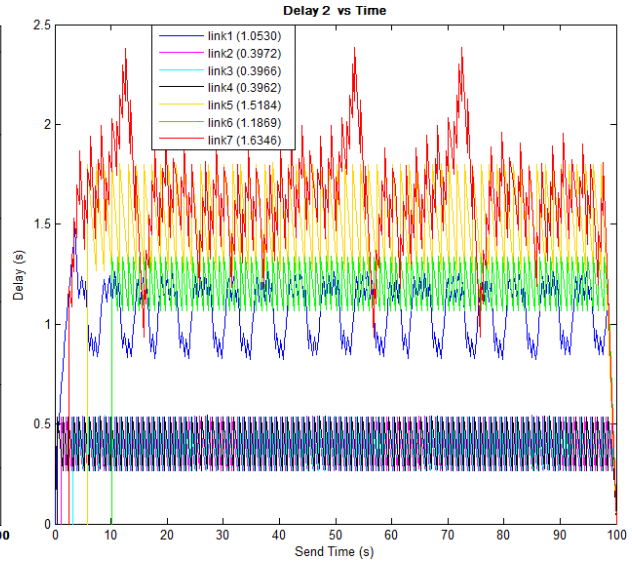
In other words, intelligent-P controller increases the delay margin of the conventional automatic generation control. This is quite useful under fault conditions in the communication channels, which allows the system operator to set up a larger upper bound of the communication delay. This property of the intelligent-P controller helps to lower the strict requirements of communication infrastructure, so the construction cost of the network can be reduced largely.

Figures 21 to 24 also confirm the conclusion that AGC is more sensitive to delay from control center to generators than to delay of tie-line power measurements. The area control signal ($ACE = \beta\Delta f + \Delta P_{tie}$) contains two parts, and the value of bias factor β is adjusted so that the frequency deviation and tie-line power deviation have balanced importance. Therefore, delay of

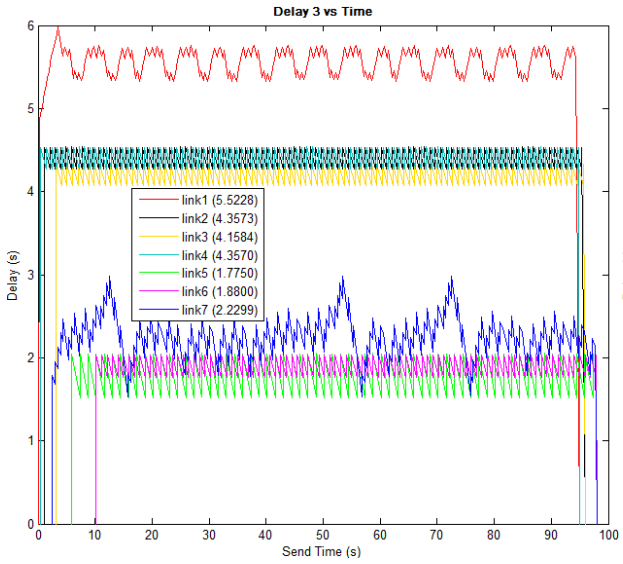
tie-line flow measurements can be compensated by the accurate frequency deviation measurements to some extent.



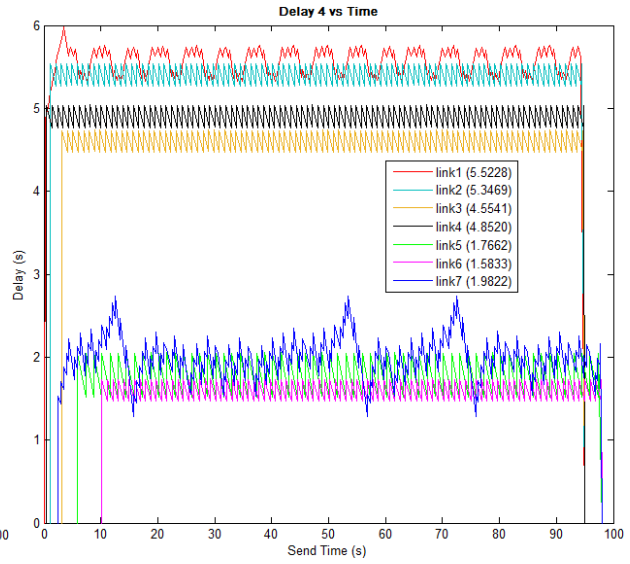
(a) Communication Delay 1



(b) Communication Delay 2



(a) Communication Delay 3



(b) Communication Delay 4

Figure 25. Link Delay

7. Conclusion and Future Work

This thesis demonstrates the need to include the influence of communication time delay on AGC in a multi-area system. Considering the increasing integration of power system and communication network in the evolution of smart grid, the simulation is implemented on a co-simulation platform PiccSIM, allowing real-time data exchange between Simulink and NS2. Our analysis shows that AGC is more sensitive to delays from control center to generators than to tie-line power measurement transmission delays.

Using a conventional PI controller as a reference, we study ultra-local or intelligent PID controller in AGC application, and compare their performance under different scenarios. When the system undergoes a large disturbance, the intelligent-P controller seems to perform better than the conventional PI controller. An important additional advantage of intelligent-P controller is the avoidance of a pure integrator, and of the need to deploy an anti-windup algorithm.

A possible way to increase the tolerance of the communication delay in AGC is implementing the fully decentralized control, where every single generator could have its own controller locally, instead of the conventional area-centralized control. The distributed controller collects its own frequency deviation and tie-line power deviation, and calculates ACE itself. Therefore, the communication delay of control signals to generators is avoided in this way. However, the tie-line flow measurements would need to be sent to all generators, which increases the communication network complexity as a result.

We comment on a Lyapunov-based stability criterion for AGC [23] which takes into account system delays. However, its application is not straightforward due variations in communication delays that occur in practice. The delay in NS2 is strongly dependent upon the network topology and employed hardware. It is widely expected that future power grids will see an increase in the number of generators participating in frequency control and a concomitant increase in diversity of communication modalities. Thus, it will likely be impractical to base control designs on models that explicitly prescribe network delays.

Bibliography

[1] Müller, Sven Christian, et al. "Interfacing Power System and ICT Simulators: Challenges, State-of-the-Art, and Case Studies." IEEE Transactions on Smart Grid (2016).

[2] Lin, Hua, et al. "GECO: Global event-driven co-simulation framework for interconnected power system and communication network." IEEE Transactions on Smart Grid 3.3 (2012): 1444-1456.

[3] Hopkinson, Kenneth, et al. "EPOCHS: a platform for agent-based electric power and communication simulation built from commercial off-the-shelf components." IEEE Transactions on Power Systems 21.2 (2006): 548-558.

[4] Hasan, Mohammad Shahidul, et al. "Co-simulation of wireless networked control systems over mobile ad hoc network using SIMULINK and OPNET." IET communications 3.8 (2009): 1297-1310.

[5] Nethi, Shekar, et al. "Platform for emulating networked control systems in laboratory environments." World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a. IEEE, 2007.

[6] N. A. S. Initiative et al., “Synchrophasor technology roadmap,” Pacific Northwest National Laboratory, Richland, Washington, 2009.

[7] S. Bhowmik, K. Tomsovic, and A. Bose, “Communication models for third party load frequency control,” IEEE Transactions on Power Systems, vol. 19, no. 1, pp. 543–548, 2004.

[8] L. Jiang, W. Yao, Q. Wu, J. Wen, and S. Cheng, “Delay-dependent stability for load frequency control with constant and time-varying delays,” IEEE Transactions on Power Systems, vol. 27, no. 2, pp. 932–941, 2012.

[9] K. Rahimi and P. Famouri, “Performance enhancement of automatic generation control for a multi-area power system in the presence of communication delay,” in North American Power Symposium (NAPS), 2013. IEEE, 2013, pp. 1–6.

[10] M. Fliess and C. Join, “Model-free control,” International Journal of Control, vol. 86, no. 12, pp. 2228–2252, 2013.

[11] Y. Choe, “Intelligent pid controller and its application to structural vibration mitigation with mr damper,” in Control, Automation, Robotics and Vision (ICARCV), 2016 14th International Conference on. IEEE, 2016, pp. 1–6.

[12] P. Kundur, N. J. Balu, and M. G. Lauby, Power system stability and control. McGraw-hill New York, 1994, vol. 7.

[13] Variani, Maryam Hassani, and Kevin Tomsovic. "Distributed automatic generation control using flatness-based approach for high penetration of wind generation." IEEE Transactions on Power Systems 28.3 (2013): 3002-3009.

[14] Fliess M, Join C (2008) Non-linear estimation is easy. Int J Model Identif Control 4:12–27

[15] H. Thabet, M. Ayadi, and F. Rotella, “Towards an ultra-local model control of two-tank-system,” International Journal of Dynamics and Control, vol. 4, no. 1, pp. 59–66, 2016.

[16] T. Issariyakul and E. Hossain, Introduction to network simulator NS2. Springer Science & Business Media, 2011.

[17] Kohtamaki, Tuomo, et al. "Piccsim toolchain-design, simulation and automatic implementation of wireless networked control systems." Networking, Sensing and Control, 2009. ICNSC'09. International Conference on. IEEE, 2009.

[18] S. Nethi, M. Pohjola, L. Eriksson, and R. J¨antti, “Simulation case studies of wireless networked control systems,” in Proceedings of the 2nd ACM workshop on Performance

monitoring and measurement of heterogeneous wireless and wired networks. ACM, 2007, pp. 100–104.

[19] M. Pohjola, S. Nethi, and R. Jantti, “Wireless control of mobile robot squad with link failure,” in Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008. WiOPT 2008. 6th International Symposium on. IEEE, 2008, pp. 648–656.

[20] A. Moeini, I. Kamwa, P. Brunelle, and G. Sybille, “Open data ieee test systems implemented in simpowersystems for education and research in power grid dynamics and control,” in Power Engineering Conference (UPEC), 2015 50th International Universities. IEEE, 2015, pp. 1–6.

[21] Parikh, Palak P., Mitalkumar G. Kanabar, and Tarlochan S. Sidhu. "Opportunities and challenges of wireless communication technologies for smart grid applications." Power and Energy Society General Meeting, 2010 IEEE. IEEE, 2010.

[22] Issariyakul, Teerawat, and Ekram Hossain. Introduction to network simulator NS2. Springer Science & Business Media, 2011.

[23] M. Wu, Y. He, J.-H. She, and G.-P. Liu, “Delay-dependent criteria for robust stability of time-varying delay systems,” Automatica, vol. 40, no. 8, pp. 1435–1439, 2004.