Evaluating the Impact of Packet Delay and Loss on a Network Control System in DETERlab

Christine P. Chen
Electrical Engineering and Computer Sciences
UC Berkeley
christinechen@berkeley.edu

Graduate Mentor: **Jia Bai and Xiaowei Li**
Faculty Mentor: **Professor Yuan Xue**

July 31, 2010

TRUST Research Experiences for Undergraduates (TRUST-REU)
in Cyber Security and Trustworthy Systems 2010

Institute for Software Integrated Systems
Vanderbilt University

Evaluating the Impact of Packet Delay and Loss on a Network Control System in DETERlab

Christine P. Chen

Abstract

The objective of my project was to analyze how adding time delay and packet loss will alter the second-order input signal after it has travelled from the digital controller across the wired control network system to the plant output. The study was conducted using the state-feedback UDP network control system that was created by using software implementation. The plant and controller programs were written with the use of socket programming in Java, and the programs were compiled and run by remote accessing into two nodes. The nodes were part of the network topologies that were created in experiments in the DETER Network Security Testbed at UC Berkeley. Using data that was collected from the experiments, the magnitude of the plant output was plotted over time, and the graph results that were obtained allowed for comparative analysis between the ideal plant output in an idealized network that had no added packet loss or time delay parameters with the plant output from an imperfect network that had additional network security vulnerabilities. The results from this paper contribute to data that is being collected in understanding how time delay and packet loss will affect a state-feedback signal of UDP packets that has gone through an internet communications network, and it reveals an efficient way for internet network security monitoring.

Index Terms —Cyber-Physical Systems, Networked Control Systems, User Datagram Protocol, DETER Network Security Testbed, network security, time delay, packet loss

I Introduction

Cyber-physical systems (CPS) research is a developing field that incorporates embedding computation into physical components in order to maintain the systems functionality [1]. The computations that are done are subsequently affected by interactions with the physical system [2]. For networked control systems (NCS), which are closed feedback loops with the data being transmitted through wired or wireless communication systems, the problems of instability in the network arise in the form of time delay, packet loss, and other network vulnerabilities when transmitting the data. The instabilities of the internet network become ever more apparent since the data being transferred is in the form of packets. However, due to the benefits of using NCS, like its flexibility in transmitting data off-site and the low cost in maintaining the system, NCS will continue to be used in transmitting data [3]. As a result, the problems that arise when transmitting data through wired or wireless transmissions must be taken into consideration when designing NCS. In order to design the system to run more effectively, these phenomena must be understood in regards to maintaining the stability of the closed feedback loop.

Understanding the behavior of NCS can be done through the study of collecting data from running real-time systems, by generalizing mathematical models, or by simulating experiments with many different parameters [4]. The third and final approach was weighted most heavily in this project. With more research in this area, it can be better understood how the signal sent through a NCS by the method of packet transmission is affected as the result of the packets traversing through the network.

The objective of this paper is to describe how a platform for signals that are transmitted through the NCS is created. Then, the resulting waveforms that had traversed through the network with different network uncertainty parameters defined will be displayed and analysis of how the varying parameters affected the signal packets will be conducted over how the physical network altered the input data pattern.

The outline of the paper is as follows. In Section II, the method behind setting up the NCS is described, along with elaborating the process of sending and receiving UDP packets and collecting time-stamped data. In Section III, the results of the experiments running different network parameters of packet delay and loss are detailed. Section IV discusses the observations and the interpretation of the collected data. Then, Section V presents the conclusion and further work.

II Methods

The experiments that were run for this project required access to nodes and links of network topologies. These network topologies were created by using the DETER Network Security Testbed, which is a testbed consisting of 400 PCs located at USC ISI and UC Berkeley [6]. The characteristics of the links between the nodes could be configured to the desired bandwidth, packet acceptance rate, latency, and/or loss parameters. The virtual machine boxes that were installed in the nodes ran the Ubuntu804-JDK Operating System image that was created as a modified version of the Ubuntu804-STD Operating System image with the Java Development Kit installed, since the source code compiled on these nodes were written using the Java language. The nodes in this network could be remotely accessed, and, in this case, the remote machines were logged into by using Secure Shell. Then, the code that included the plant algorithm and the User Datagram Protocol (UDP) sender/receiver setup was compiled in node-0 while the code that included the controller algorithm and the UDP sender/receiver setup was compiled in node-1. The plant code was run on node-0, immediately followed by the controller code being run on node-1.

As shown in Figure-1, the files that were run on the two nodes implemented two components: the algorithm for the functions of the plant and controller in a stabilizing, state-feedback model and the UDP protocol for transmission of data through the network.

Both these files included functions of the state-space equations specific to the plant and the controller. The plant in its continuous-time form can be described by

$$\dot{x}_p(t) = ax_p(t) + bu_p(t) \tag{1}$$

$$y_p\left(t\right) = cx_p\left(t\right) + du_p\left(t\right) \tag{2}$$

where t is a continuous time variable. In order to discretize this

$$\dot{x}_p\left(t\right) = ax_p\left(t\right) + bu_p\left(t\right) \tag{3}$$

$$\frac{x_p\left(t+\Delta t\right) - x_p\left(t\right)}{\Delta t} = ax_p\left(t\right) + bu_p\left(t\right) \tag{4}$$

In order to approximate this function, the Δt was chosen to be very small. When it has been discretized, and in its actual implementation, the plant's state-space equations becomes

$$x_p\left(k+1\right) = x_p\left(k\right) + \left[ax_p\left(k\right) + bu_p\left(k\right)\right]\Delta t \tag{5}$$

$$y_p\left(k\right) = cx_p\left(k\right) + du_p\left(k\right) \tag{6}$$

where k is a discrete time variable. The coefficients have the values $a = 0.0, b = 0.5, c = 0.6826, and$ $d = 0.0,$ and $\Delta t = 1.0$ millisecond, a much smaller value than the sampling rate of 50.0 milliseconds. Meanwhile, the discrete controller can be described by the following

$$x_c\left(k+1\right) = a_c x_c\left(k\right) + b_c u_c\left(k\right) \tag{7}$$

$$y_c\left(k\right) = c_c x_c\left(k\right) + d_c u_c\left(k\right) \tag{8}$$

where $a_c = 1.0, b_c = 0.05, c_c = 24.9980,$ and $d_c = 7.994$ are the coefficient's values.

Since both the plant and controller have initial states to be taken into account, the implementation for the code was broken up into the initial state and the following current state. Since UDP utilizes datagram sockets to send and receive packets between the nodes, the Java Class DatagramSocket, which sends packets that are individually addressed [8], was used.

The programming for the Plant consists of three timers from the Java Timer Class. Taking into consideration the choice of the five second period for the sinusoidal signal, Timer-1 implemented the passive sampler, so it took the plant input that is being updated once every 50.0 milliseconds as the current input, and updated x_p and the plant output (y_p) once every millisecond. Timer-2's function was to send the most updated y_p value out to the controller and to update the u_p. This was done every 50.0 milliseconds, and, as such, this discretized the continuous signal. Timer-3 has the socket check to receive packets every millisecond. If a packet is accepted, it is saved in a temporary variable that will be used to update the u_p

value inside Timer-2, as it was said, every 50.0 milliseconds.

The programming for the Controller was implemented with three timers as well. Timer-1 received the input data y_p from the plant once every millisecond. It then updated the x_c, y_c, and u_c values. This was in order to ensure that the packets were able to be received continuously, while maintaining a separate implementation for the sending and receiving function. Timer-2's function was to send the updated y_c value out to the plant once every 50.0 milliseconds. Timer-3's job was to consistently update the time relative to the start time, isolated from both the functions of sending and receiving.

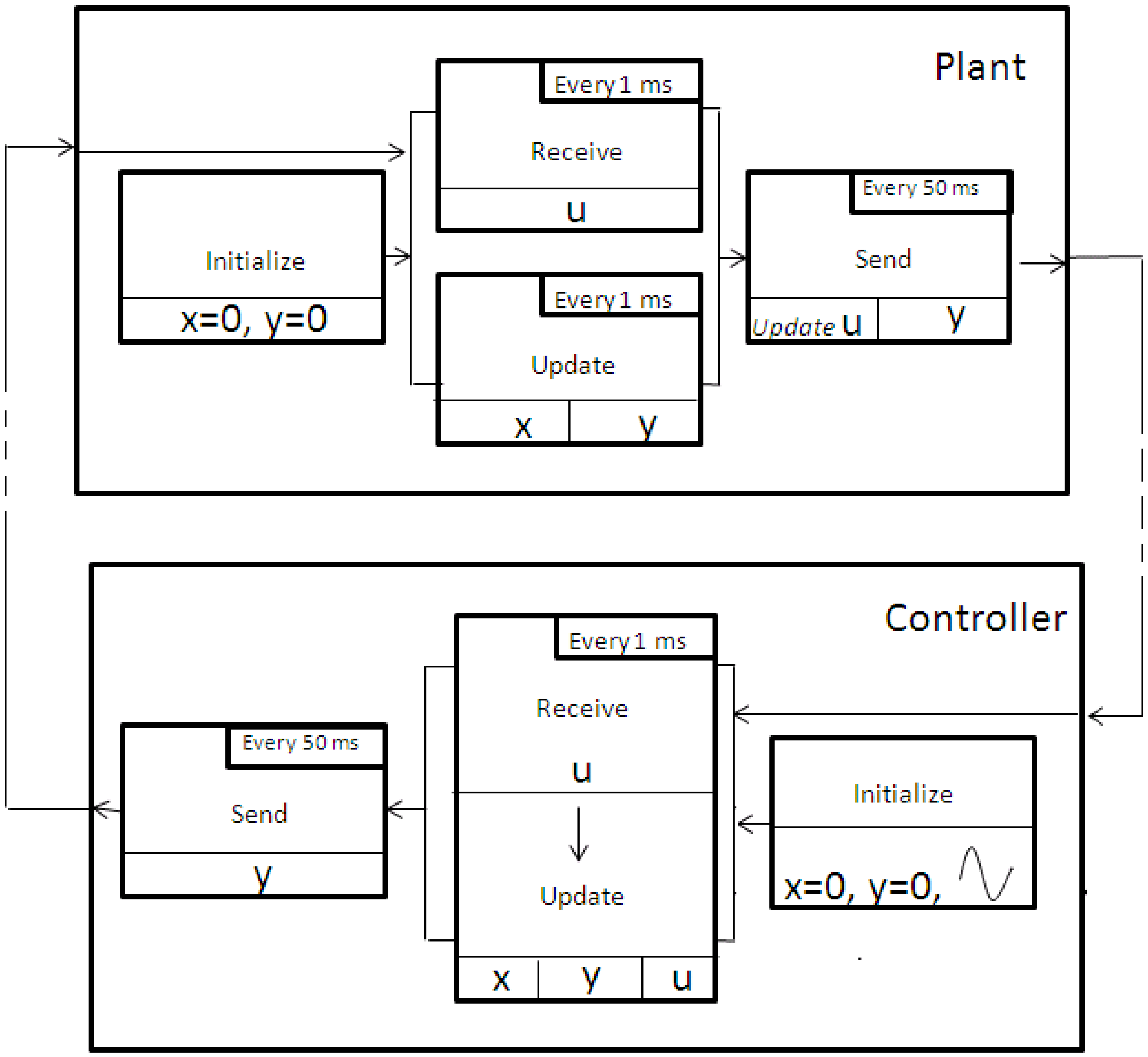

Figure-1: This flowchart depicts the plant and controller model. The instructions inside the block diagrams were how the coding implementation was divided up.

With the plant and controller programs installed in the nodes, experiments with the same two-node topology interconnected by a single link were run with different network settings. First, the experiment with ideal conditions was run on DETER to serve as the control during comparisons, with the bandwidth set to 100.0 Mb and the queue-type to be Drop-Tail [9]. The initial condition of the controller input u_c(t)=sin($\frac{2\pi t}{5}$) served as the reference input, having a period of T=5.0 seconds, and data with the time-stamp in seconds and the magnitude of the plant output y_p(t) were collected to plot the graphs in Microsoft Excel for further analysis. The bandwidth and queue-type continued to be the default setting for all the experiments. Figure-2 shows the state-feedback model with the reference input of the sinusoidal signal and the respective plant output.

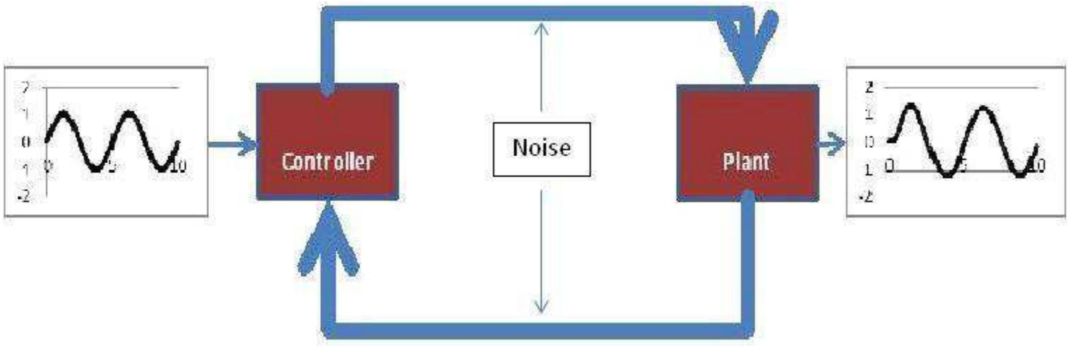

Figure-2: State-feedback model with the sinusoidal input and the output signal

Four experiments were run that included network characteristics of time delay of 50.0 milliseconds, 60.0 milliseconds, 70.0 milliseconds, and 80.0 milliseconds, which are demonstrated in this paper. And four experiments were run with packet loss of 0.3, 0.4, 0.5, and 0.6 displayed in this paper.

III Results

All these experiments were run with a sinusoidal input having a period of 5.0 seconds and for a variable amount of time-span greater than 300.0 seconds. The experiments were done with different time delays set and/or different packet loss.

III.1 Packet Delay

The experiments consisted of variation in setting the parameter of time delay, and the plant output was collected in order to see how the time delay affected the input sinusoidal wave.

For the delay=50.0 ms experiment shown in Figure-5 and the delay=60.0 ms experiment shown in Figure-6, the plant output seemed similar to the original plant output with no delay in Figure-4.

For delay=70.0 ms in Figure-7, it seems to be able to stabilize at a maximum of 20.0 seconds. Since the period was not really altered, there was no big difference if you did not care about the beginning of the data collection.

For delay=80.0 ms in Figure-8, this seemed to be the point where the system could not hold on to its stability, and the graph began to oscillate erratically. The period of T=5.0 seconds was not maintained, and the amplitude ranged from around -3.0 to 3.0 during the first 20.0 seconds. It can be noted in Figure-9 that by the time it was plotted to 260.0 seconds, the graph had oscillated to maximum amplitudes ranging from -30.0 to 30.0, and the period had reduced to around T=2.0 seconds.

III.2 Packet Loss

Experiments were also conducted with different probabilities of packet loss set.

For 0.3 packet loss, which is shown in Figure-10, there was no discernable change in the period or amplitude of the plant output.

For 0.4 packet loss, it can be seen from Figure-11 that the period remained around 5.0 seconds and the amplitude remained around 1.2. However, there were certain instances when the direction of the waveform would go in the reverse direction for around half a period before regaining its stability and continuing the cycle. This would occur at random intervals during the various experiments, and an instance is shown in Figure-12. The sharp dip could be a good indication to show the influence from an occurrence of packet loss.

Then, for 0.5 packet loss, Figure-13 shows that there were substantially greater alterations in the signal waveform. The period was affected considerably during certain intervals of the experiment due to the abrupt switch in the direction of the waveform. The amplitude would then increase from having its peak around 1.2 to a wide range up to the hundreds, as seen from the experiments that were run. In these instances, it would be able to stabilize back to a recognizable waveform with the period of around 5.0 in around a 10.0 second interval, depending on the rate of ascending or descending of the waveform.

Lastly, for 0.6 packet loss that is shown in Figure-14, there were significantly greater alterations in the sinusoidal wave. The period of the waveform could no longer acclimate back to 5.0 seconds after swinging to extreme heights, and the amplitude range was extremely variable, with no constant peak amplitude.

IV Discussion

Partitioning the implementation of the data collection, sending, and receiving into the way described in above methods section was conducive to synchronizing all the activity of sending, receiving, and constantly updating the input, output, and state. Timing is an important issue in a communications system, and even if the data need not be received extremely precisely, a working system should be able to synchronize the sending, receiving, and updating process. In the setup of this network control system, UDP sockets were chosen because this real-time control system is able to drop datagrams, and duplicates will not alter its output by that much. The UDP protocol does not guarantee the packets to arrive in the order it was sent, and, likewise, there is no delay in waiting for acknowledgement [5]. By making the implementation parallel in nature, this eliminates the bottleneck that could arise if the sending and receiving steps were dependent on the other′s completion. By constantly checking to receive packets, there is minimal chance of missing out on receiving the data, and once there is something to send out, on either side, the most updated value will be passed by variable. The time will consistently be updated, and the most updated plant output is saved every fifty milliseconds, along with this time stamp, to be plotted.

For all the experiments, the initial period or so will have some minor aberrations from a perfectly smooth curve, since the feedback is just initializing, and there will be some time before it has found a balance from the plant and controller communications.

IV.1 Packet Delay

Examining the experiment results with the time delay parameter, the 50.0 millisecond delay does not have any apparent aberrations in magnitude and frequency. For the 60.0 millisecond delay, there are no discerning alterations in the first 20.0 seconds. However, as the experiment continues running, it can be seen that the magnitude increased to a range of -4.0 to 4.0, and the period changed to T=2.0 seconds. For the 70.0 millisecond delay, there is a similar phenomenon, and then for the experiment with an 80.0 millisecond delay set, in the later range, past 260.0 seconds, there is a large oscillation of -30.0 to 30.0 that can be seen.

In order to understand how the time delay is related to the period of the input sinusoidal wave, the ratio of the time delay with the period, which is denoted as r_d, was calculated for the four experiments, with respect to the correlating results. For the 50.0 millisecond delay, the ratio of the delay to the period was r_d=0.010, and the period remained at T=5.0 seconds for the extent of the experiment. The amplitude was around 1.2 the entire time as well, which was equivalent to the amplitude of the plant output from the original experiment. For the 60.0 millisecond delay, the ratio of the delay to the period was r_d=0.012, and similar results were seen as in the prior experiment with the 50.0 millisecond delay. In the experiment with the 70.0 millisecond delay, the ratio of the delay to the period was r_d=0.014, and period remained at T=5.0 seconds, but the amplitude did not hold steady, ranging from 1.4 to 0.9. Then, in the experiment with the 80.0 millisecond time delay, the ratio of the delay to the period was r_d=0.016, and the values of both the period and the amplitude had become unstable. The period had become smaller and more erratic, ranging from around 1.7 seconds to 1.6 seconds from peak to peak. The amplitude had begun to grow wider in range as time went on, going to height of 30.0 by 260.0 seconds, as can be seen from Figure-8.

Schenato writes in his paper that there was no significant change when there was time delay [3]. This seems true up to a certain point that is unique with respect to the system and the period of the sinusoidal input to the system. In the case of these experiments, it can be seen that up to 60.0 millisecond time delay, there was no obvious distortion of the waveform or any change in its properties. However, after that instance, the waveform had become unstable, and changes in period and amplitude could be noticed from the plots.

IV.2 Packet Loss

The probability of packet loss arises in systems that have back log. After a certain amount of time delay, with the packet not being sent, it will just be dropped [4]. In the case of DETER, the setting can be made for the probability of packet loss for the given link between two nodes.

Looking at the results of the experiments with the parameters of packet loss set, it can be seen that when the probability of packet loss is 0.4, there are random instances in the plot where the loss of packets being sent to the plant can be seen from the abrupt change in direction of the waveform from ascending to descending or vice versa. The packets with values that would update the sinusoidal wave were lost, and so the waveform updated its current y_p value by reusing its previous packet to update the internal state of the system, until a packet with updated information was finally received. From then on, it would begin acclimating back to the original path of the waveform. From the graph with packet loss set to 0.5, the alterations were greater, with a greater increase in the magnitude. It should be taken into consideration that this real-world system will experience phenomena characteristic of increased uncertainty placed on the system. The characteristics described when there is the parameter of 0.6 packet loss can be seen in sharper contrast in comparison to the 0.5 packet loss. In the particular experiment shown from Figure-14, it starts out with a fairly flat horizontal oscillation, and the sinusoidal waveform cannot be distinguished from the plot. The zero-crossings also become erratic.

From these experiments, a relation between the period of the sinusoidal input with respect to the amount of time delay can be discerned from the data. The characteristic of the sinusoidal wave can be seen when the experiments are run with packet loss as well.

V Conclusion and Further Work

From this research, the two-node state-feedback UDP network control system was created into an internet communications system, and the sinusoidal wave input was fed into the system in order to see how the wired connection with imperfections in the network system changed the signal. Different time delays and probability of packet loss parameters were run in order to discern how the delay and loss affected the sinusoidal wave with the chosen frequency, and determine how the imperfections and the sinusoidal wave were interrelated. The results showed that an approach could remotely monitor the impact of packet loss and time delay variations from the network channel disturbances if the relevant parameters were well-defined. Further work can be done in determining more about the underlying characteristics of how the network system will affect specific waveforms that are sending the packet data, using this research as the foundation on how to build a viable network control system model and utilize it to collect and analyze the network control system output.

List of Supplemental Figures

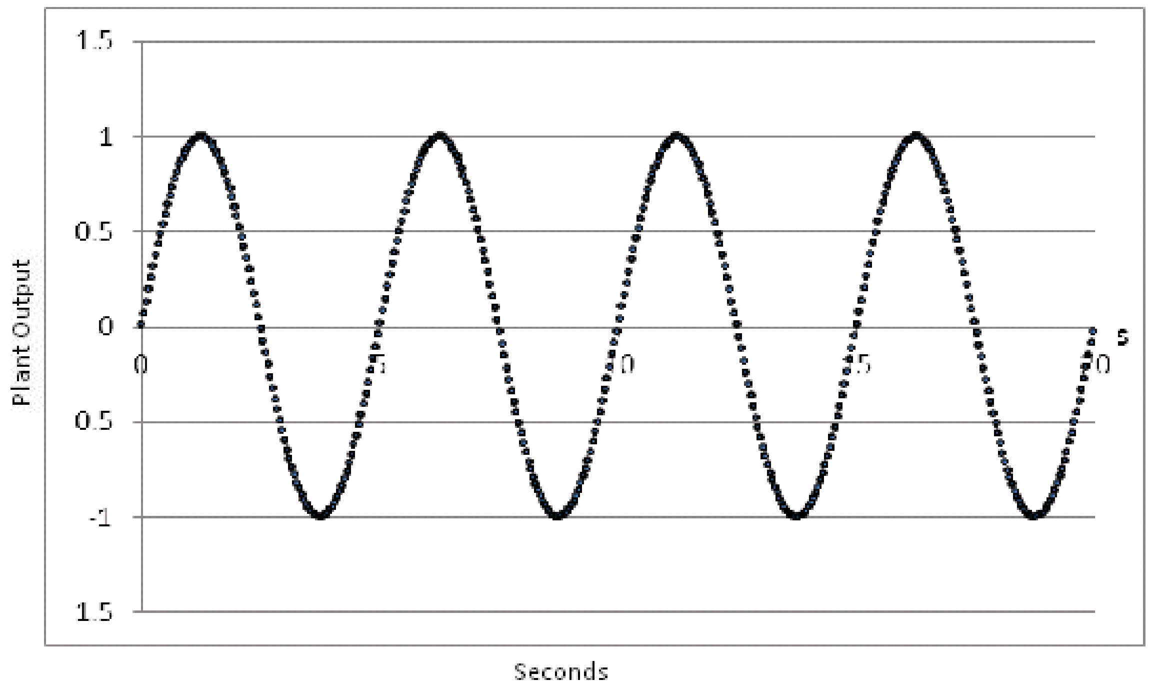

Figure-3: Reference Input Signal vs. Time (s.) Used as an Initial Controller Input

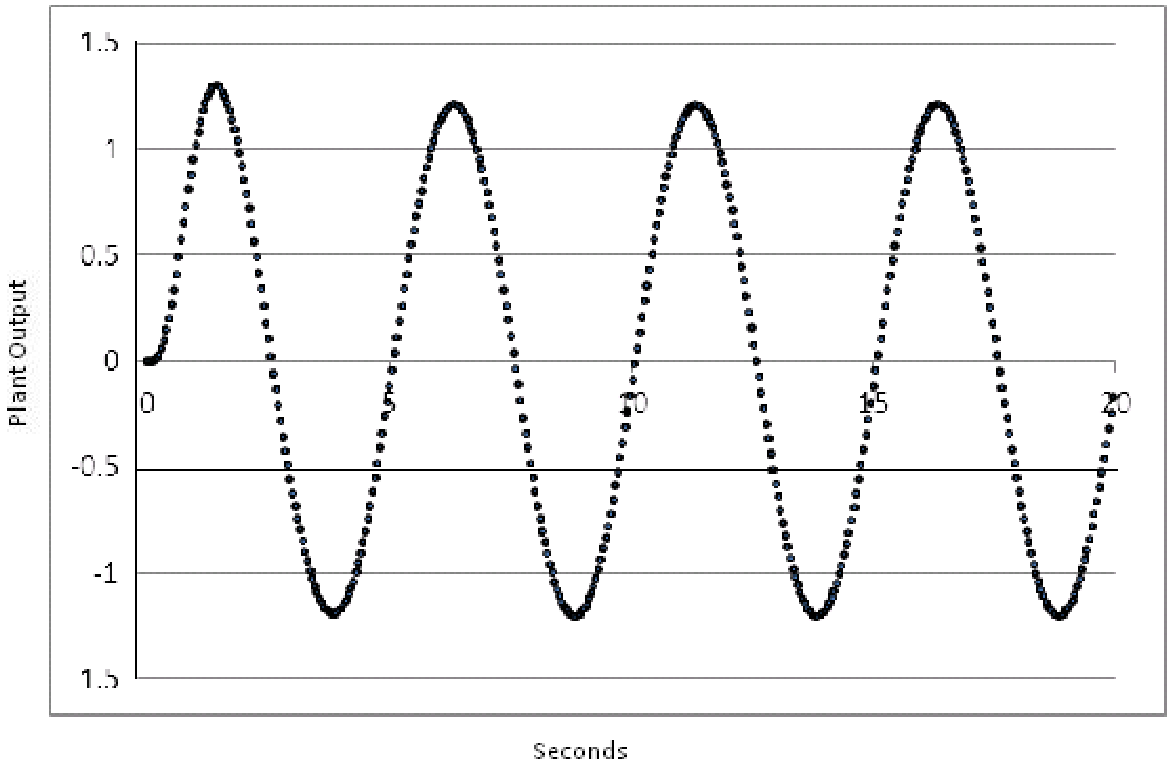

Figure-4: Plant Output Signal vs. Time (s.) in Original Experiment of Ideal Environment

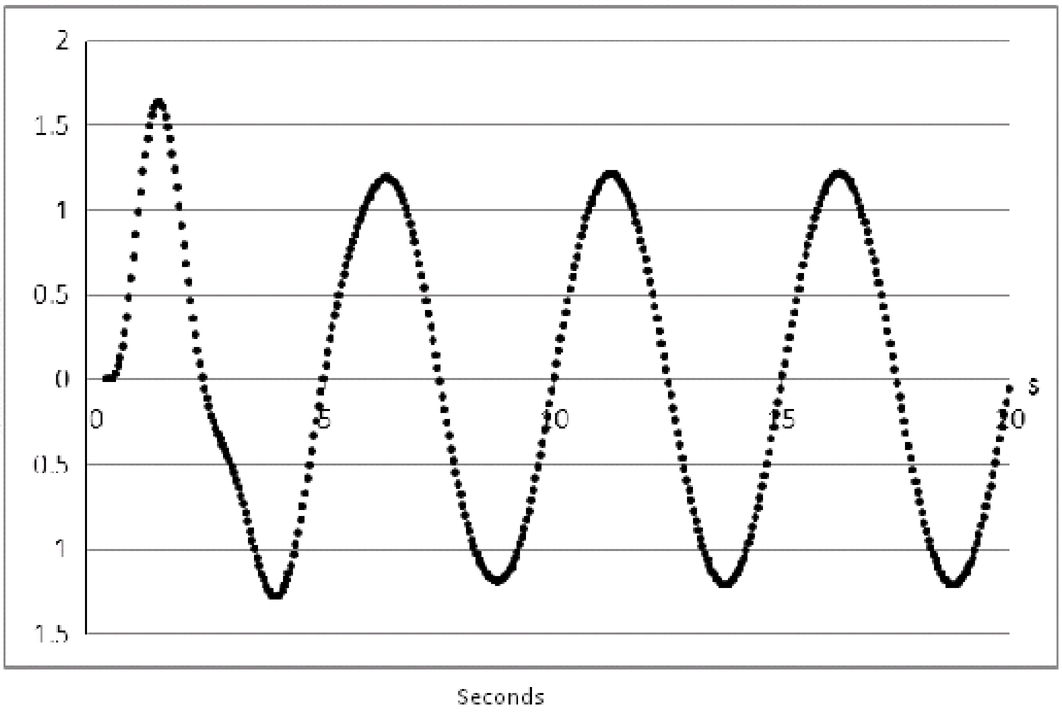

Figure-5: Plant Output Signal vs. Time (s.) for 50.0-ms Delay Experiment

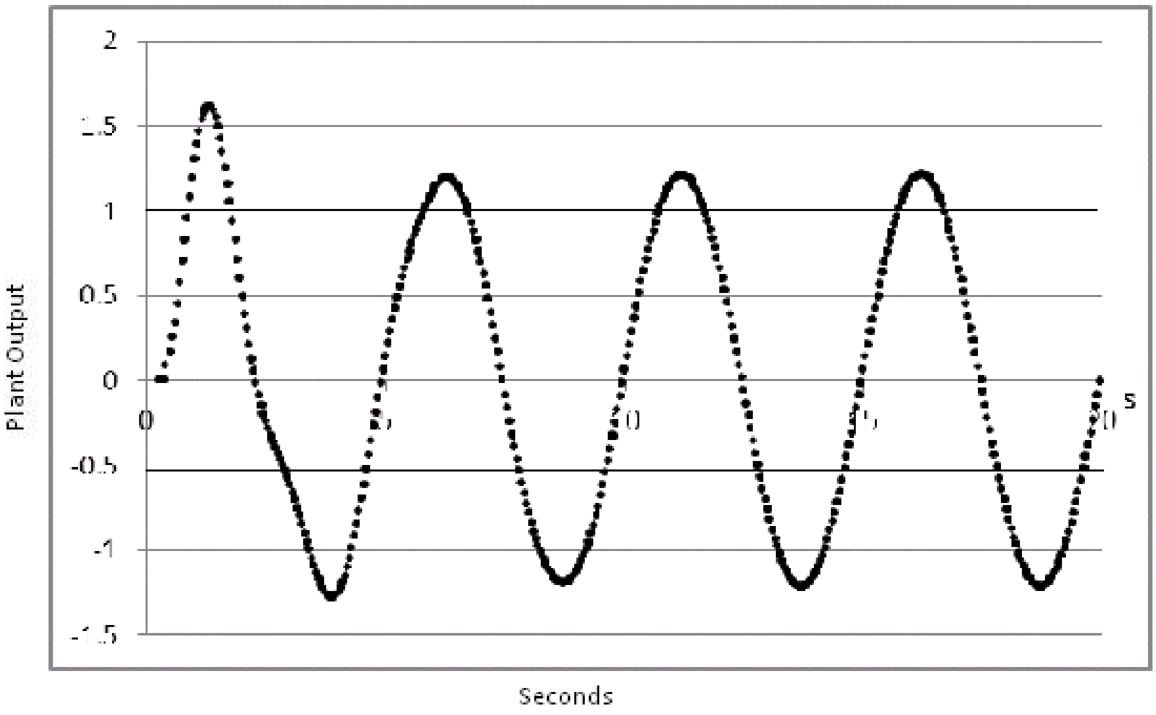

Figure-6: Plant Output Signal vs. Time (s.) for 60.0-ms Delay Experiment

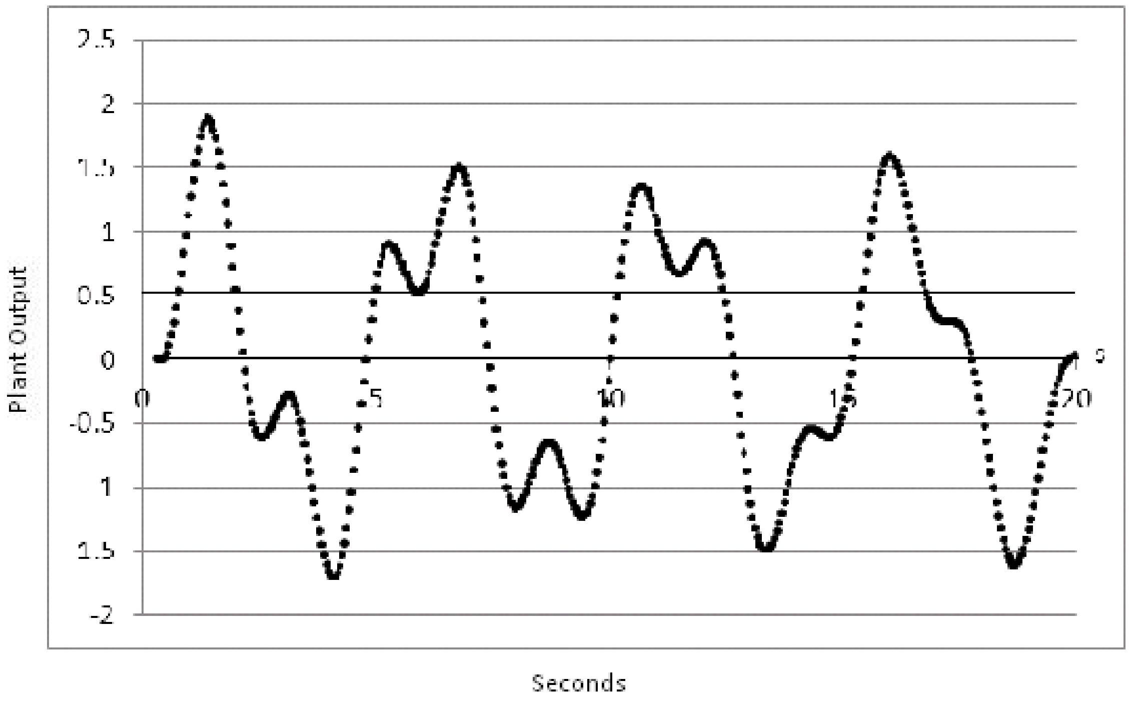

Figure-7: Plant Output Signal vs. Time (s.) for 70.0-ms Delay Experiment

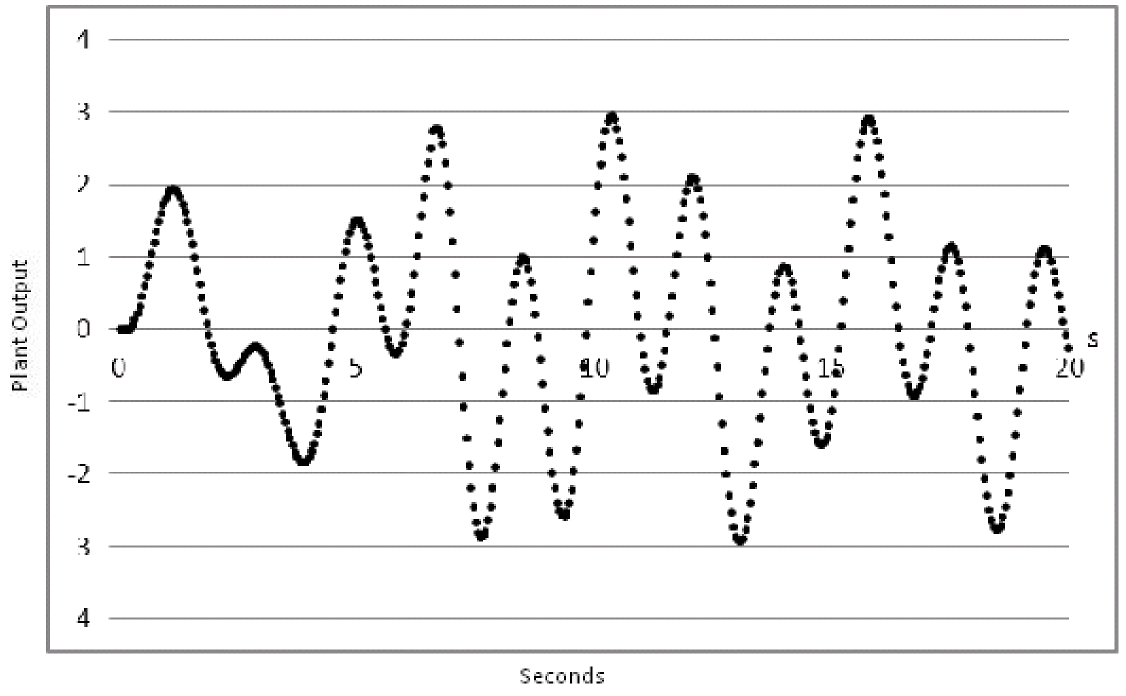

Figure-8: Plant Output Signal vs. Time (s.) for 80.0-ms Delay Experiment

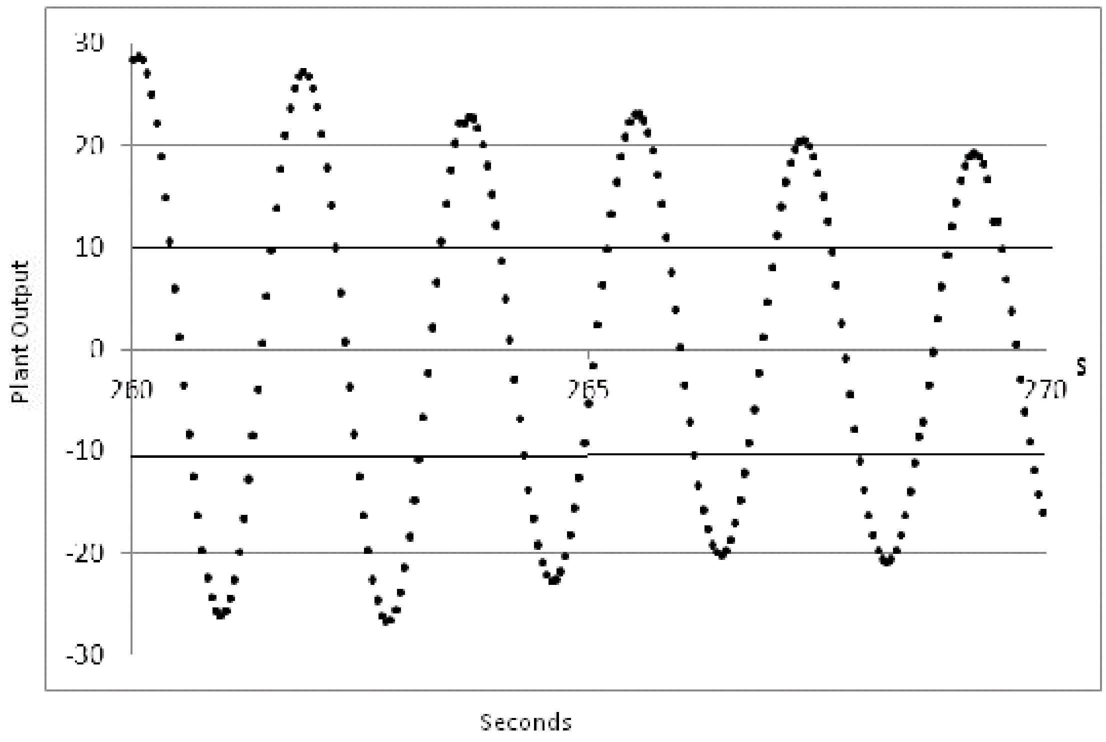

Figure-9: Plant Output Signal vs. Time (s.) for 80.0-ms Delay Experiment showing the increasing amplitude of the oscillations

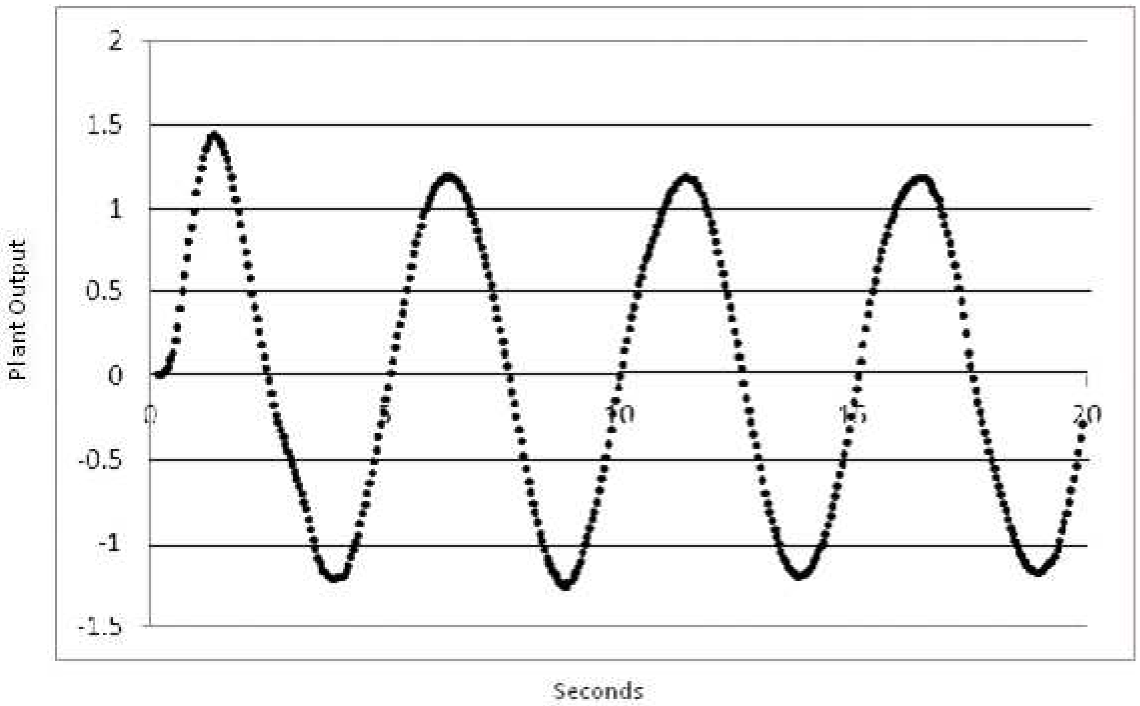

Figure-10: Plant Output Signal vs. Time (s.) for 0.3 Packet Loss Experiment

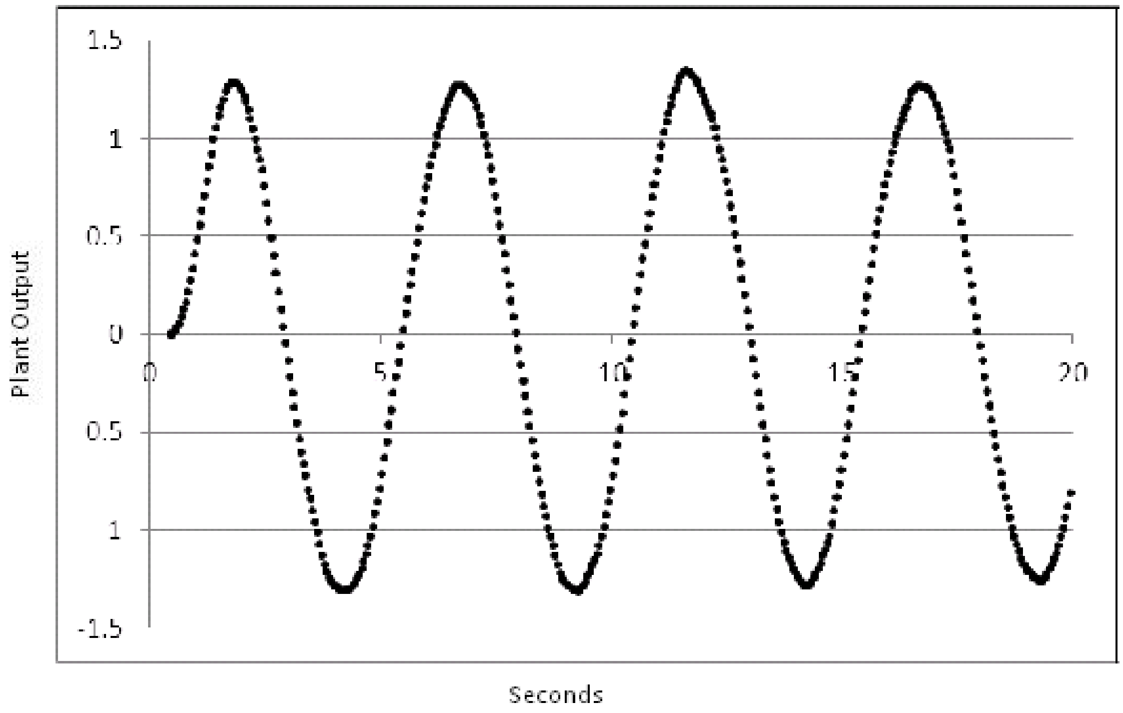

Figure-11: Plant Output Signal vs. Time (s.) for 0.4 Packet Loss Experiment

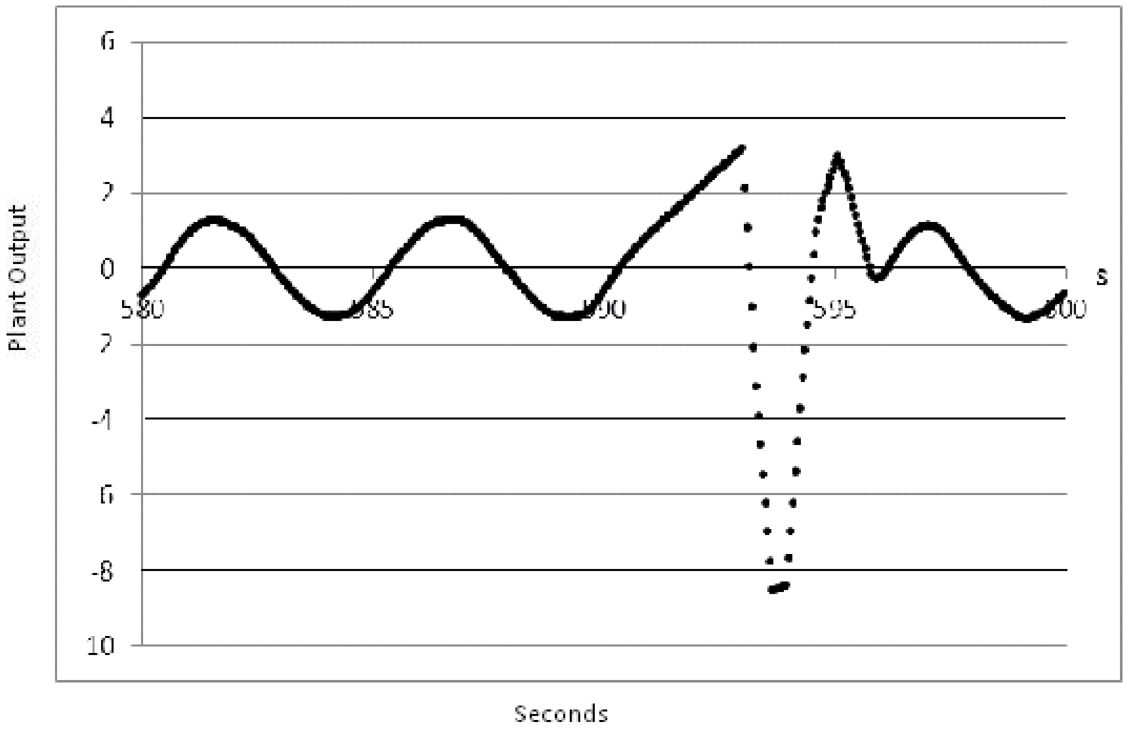

Figure-12: Plant Output Signal vs. Time (s.) for 0.4 Packet Loss Experiment

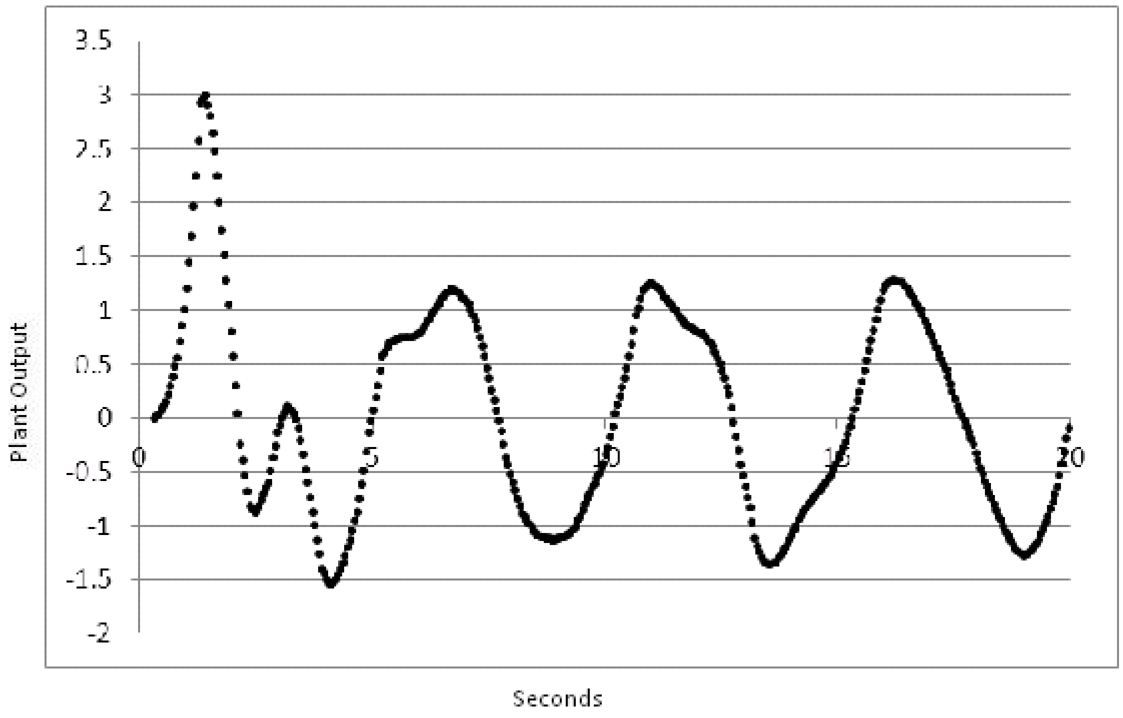

Figure-13: Plant Output Signal vs. Time (s.) for 0.5 Packet Loss Experiment

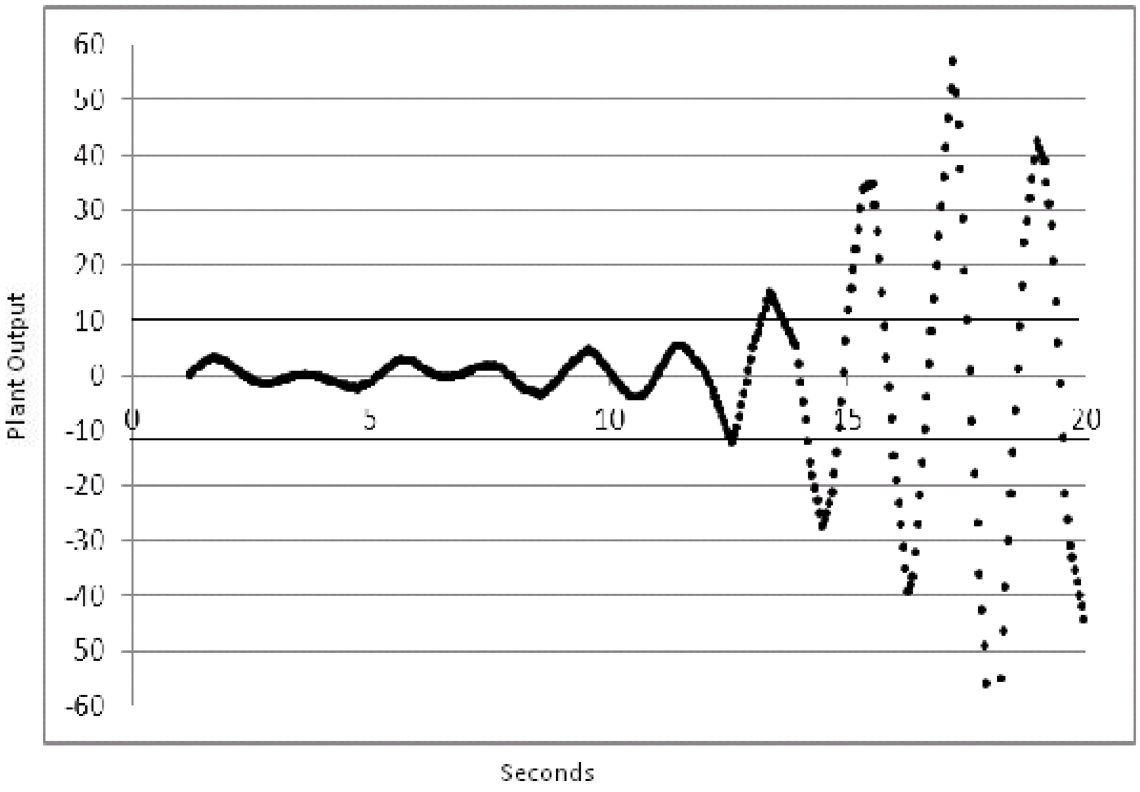

Figure-14: Plant Output Signal vs. Time (s.) for 0.6 Packet Loss Experiment

Acknowledgements

I would like to thank the TRUST program for providing me with the opportunity to conduct this research at Vanderbilt University. I also want to thank Vanderbilt University for hosting the Vanderbilt TRUST-REU students. Thank you to Ph.D students Jia Bai, Xiaowei Li, and Professor Yuan Xue for their help and encouragement in giving me direction on my research. Also special thanks to Dr. Kristen Gates for her guidance in the TRUST-REU program.

List of References

[1] Tabuada, Paulo (2006, October). Cyber-Physical Systems: Position Paper. Symposium conducted at the NSF Workshop on Cyber-Physical Systems, Austin, TX.
[2] Lee, Edward A. (2006, October). Cyber-Physical Systems —Are Computing Foundations Adequate? Symposium conducted at the NSF Workshop on Cyber-Physical Systems, Austin, TX.
[3] L. Schenato, “ Optimal estimation in networked control systems subject to random delay and packet drop,“ IEEE Trans. Automat. Control, vol. 53, no. 5, pp. 1311-1317, June 2008.
[4] K. Lee and S. Chanson, “ Packet Loss Probability for Realtime Wireless Communications“, IEEE Trans. on Vehicular Technology, Vol.51, No.6, pp.1569-75, November 2002.
[5] Kottenstette, Nicholas, Koutsoukos, Xenofon, Hall, Joseph, Sztipanovits, Janos, Antsaklis, Panos (2008, June). Passivity-Based Design of Wireless Networked Control Systems for Robustness to Time-Varying Delays, In International Workshop on CyberPhysical Systems —Challenges and Applications, Santorini Island, Greece.
[6] “DETER Network Security Testbed“ (2010, July 25), (Deterlab based on emulab),
Available: https://www.isi.deterlab.net/index.php3?stayhome=1 (Accessed: 2010, June 7).
[7] Java 2 Platform Std. Ed. V1.4.2 “DatagramSocket“ http://download.oracle.com/docs/cd/E
$17476_0$1/javase/1.4.2/docs/api/java/net/DatagramSocket.html
[8] Emulab.“Emulab Tutorial “ https://users.emulab.net/trac/emulab/wiki/Tutorial