Effect of Loss Probability on Network Scheduling & System Output and Simulation of Wireless Networked Control System Using TrueTime

Mr.Vijay R. Dahake[1]

Asst professor, Department of Electronics & Telecommunication Engineering, Ramrao Adik Institute of Technology, Navi Mumbai, Maharashtra, India. Email: vijay.dahake@rait.ac.in

Mr. Mukesh D. Patil[1]

Professor, Department of Electronics & Telecommunication Engineering, Ramrao Adik Institute of Technology, Navi Mumbai, Maharashtra, India

Mr.Vishwesh A. Vyawahare[2]

Professor, Department of Electronics Engineering, Ramrao Adik Institute of Technology, Navi Mumbai, Maharashtra, India

Abstract— In application like space exploration, under water exploration, robot co-ordination, avionics control systems, and smart process manufacturing delay between Sensor to Controller delay (S to C), Controller to Actuator delay (C to A) and dropouts of packets beyond certain limit is strictly intolerable due to reaction of instability of system under consideration. These are nothing but examples of wired or wireless networked control systems which is overlapping area of study between control, communication and information system. At research platform, one part of prototype of such application can be developed, study and analysis of various delay and stability can be done using TrueTime toolbox which can simulate wired and wireless networked control system. Before few decades use of wired networked & embedded control system with CAN protocol, ProfiBus was in tremendous demand. But such wired system poses drawback like large number of wired connections, difficult maintenance and upgradation. Due to rapid developments in wireless communication and IT technology, such wired system is being rapidly replaced today by wireless networked control system (WNCS) and cyber physical system(CPS). Main contribution of this paper is to explain effect of loss probability on networking scheduling & system output. Also some complex wireless networked control systems is simulated under action of PID controller using IEEE 802.11B/G (WLAN) and IEEE 802.15.4 (ZigBee) in TrueTime.

Keywords—Wireless Networked control system, Cyber physical system, Packet dropout, S to C delay, C to A delay, CAN Protocol, ProfiBus, TrueTime, Sensor, Actuator.

1. Introduction

1.1 Theory and definition of WNCS

Wireless Networked Control Systems (NCS) are completely distributed real time feedback control system in which sensors to controller (S-C) and controller to actuator (C-A) communication i.e., exchange of different signals e.g., feedback signal, control signals take place among system components like sensor, actuator in the form of information packets through wireless network which always have limited bandwidth. Sensor continuously read output of process and due to limited bandwidth of wireless network; sensors do not sent output as it is to the controller through wireless network. Instead, sensor's output is continuously sampled and quantized at fixed rate such that this rate is sufficient to recover the process or system output successfully and then it is sent to controller (S-C) through wireless network. Controller then compares this output with reference input, accordingly generates the control signal and sent this signal to the actuator. Actuator accordingly reacts with & generates actuating signal for the plant so that it will produce desired output. As compared to the point to point system or wired networked system, the use of an WNCS has the advantages of low installation cost, reducing system wiring, simple system diagnosis and easy maintenance [1,2]. However, due to use wireless network for communication between sensors, controller and actuator in feedback loop, some inherent drawbacks s of NCSs, such as bandwidth constraints, packet dropouts and packet delays, will degrade the performance of NCSs or even cause

instability [3,4,].
Survey on quantization, packet dropout and stability is given here. The problem of quantised stage feedback H1 stabilization for LTI systems over data networks with limited network QoS was studied, and a quantised state feedback strategy was developed for global and asymptotical stabilization [5]. The state feedback stabilization problem for a class of NCSs with nonlinear perturbation and state quantisation was investigated, and a quantised asymptotically stabilizing sufficient condition was proposed [6]. A unified framework for controller design for control system with quantisation and time scheduling via an emulation-like approach was developed [7]. Considering the S/C quantisation and data dropout, an output-feedback control for NCSs was designed, which guaranteed the exponentially mean-square stable [8].

Wireless NCS is very vast area. Research of WNCS may be divided into two main types: a) control of network, and b) control through or over network. [9]
a) Control of network:
Under this class researcher can work on the issues of communication network such as routing control, congestion control and communication protocol etc.
b) Control through or over network:
Under this class researcher can pay attention on the design and control of systems that are using a wireless network as a medium for transmission of data to obtain the desired performance.
Control through or over network guarantees the quality of service (QoS- concerned with transmission rates and error rates) and the quality of control (QoC- concerned with the stability of the system subjected to different conditions.). Simultaneous achievements both of QoS and QoC is a major objective of research in NCS. Research on Networked Control System allows us to work on different areas like, a) Methods for design of robust controllers for NCS b) Modelling of time delay in the NCS c) Modelling of packet loss in networked control systems d) Analysis of stability in networked control systems e) Synthesis of Network types and communication protocols [10].

1.2 Types and importance of sampling used in NCS
In WNCS, sensor output signals are continuous time signals need to be sampled due to bandwidth constraint of the wireless network. Choice of sampling shall be decided by considering network traffic, influence of disturbances and computational load [18]. There are two methods of sampling in WNCS to sample the continuous time signals: a) time triggered sampling and b) event triggered sampling. In time triggered sampling, the next sampling instant occurs after the elapse of fixed time interval, regardless of plant state. Time triggered sampling also called as 'Riemann sampling' again may be classified into two categories: Hard Sampling Period and Soft Sampling Period, by considering relationship between sampling period and message delay [19]. The message delay of the hard sampling period is smaller than sampling period. In event triggered sampling, the execution of control task is determined by the occurrence of an event rather than the elapse of a fixed time period as in the time triggered control. Events are triggered only when stability or pre-known control performance are about to lost. [20, 21]. Event triggered sampling can reduce the traffic load of the network with minor control performance degradation. [22].

1.3 Objective and Motivation
Many researchers worked and are working on topic wireless NCS and constantly contributing to this area. Before two decades, research on WNCS was limited, but due to remarkable progress in wireless network, different protocols structure and control system, this topic today has huge potential and one gets chance to work in different areas like control, communication and computer system, this is the main motivation of this research study. Extension of this area is Cyber-physical system (CPS). Research in CPS becomes easy if we worked first in WNCS, because WNCS is background for CPS.

Quality of service or communication (QoS) is concerned with transmission rates and error rates and quality of control (QoC) which is concerned with the stability of the system under different conditions. Simultaneous guarantees and achievement of both QoS and QoC in any application is the main objective of research in WNCS.
Main objective of this paper is to explain how communication happens between sensor to controller and controller to actuator by using wireless network, how simultaneously with less delay and low packet loss (QoS) can be achieved by maintaining stability (QoC) of the given system while doing simulation using TrueTime. There is many software which can be used for simulation of WNCS e.g. PiccSimulator, RTNS, LabNS2.

1.4 Literature survey

Many authors contribute to research in WNCS by using TrueTime toolbox for practical implementation and extend and appreciate work of Dr. Anton Cervin, Dan Henriksson and Martin Ohlin [11]. TrueTime is powerful & popular tool used to do analysis of real time wireless NCS, developed by Dr. Anton Cervin & team at Lund University in year 2002. It is continuously upgraded with addition of new features since year 2002.
In paper [12], author Mojabata Ghodrati and Alireza Saheb design and simulate fuzzy control of NCSs, and TrueTime and LMI toolboxes were used to simulate a fixed delay network. The Takagi-Sugeno (T-S) fuzzy model method was used to design the fuzzy controller.
In paper [13], author L. Farkas and J. Hnat describes how practical simulation of WNCS using TrueTime is helpful to understand application like wirelessly communicating moving robots and automated industries.
In paper [14], author Martin Andersson et al. used different network simulators like TOSSIM, NAB, RTSIM and TrueTime for wireless embedded system. They presented two simulation case studies: a simple communication scenario and a mobile robot soccer game using TrueTime simulator.
Due to Gain range limitation of conventional proportional-integral-derivative (PID) controllers, they are not suitable for application of delayed environment like wireless network. Hence in paper [15], author, Sabo Miya Hassan, Rosdiazli Ibrahim et al. proposed setpoint weighting strategy to improve the performance of the PID in such environments.
In paper [16], author Shayok Mukhopadhyay et al. studied using a real world networked control system platform called the CSOIS Smart Wheel, the effect of modelling the network delay dynamics using non-Gaussian distributions, and compensating for such a delay in closed-loop systems using a FO-PI controller. Also he studied the order of the fractional order proportional integral controller which gives least ITAE, ISE for a particular distribution of the delay.

1.5 Organization of the Paper

.
This paper is organized as follows. Modeling of wireless networked control system using state space analysis by considering delay and packet loss at controller and actuator is explained in section 2. Different simulators for simulation of NCS and different blocks in TrueTime are described in section3. Effect of loss probability of wireless network block in TrueTime on network schedule and system output is investigated for non-minimum phase and unstable system in section 4. Wireless simulation of delay system and third order system is done using TrueTime and MATLAB is given in section 5, Finally, conclusion is given in section 6, followed by references.

2. Modelling of Wireless NCS

Modelling and analysis of linear time invariant plant for wired networked control system and feedback control system is easy task since negligible delay or no delay is involved for communication between sensor to controller (S-C) and controller to actuator (C-A). Modelling of such system can be easily done using following state space analysis equation given by (1) & (2),

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) ---------- (1)$$

$$y(t) = Cx(t) \qquad ------------(2)$$

Bu modelling & analysis of linear time invariant (LTI) plant system in wireless networked control system, delay is involved for communication between sensor to controller τ_{sc}(t) and indicator of packet loss γ(t) at controller must be considered. Also delay involved for communication between controller to actuator τ_{ca}(t) and indicator of packet loss α(t) at actuator must be considered. Due to this modelling & analysis of linear time invariant plant in wireless networked control system increases complexity to great extent. Modelling of LTI plant in wireless NCS can be done with following modified state space equation [17] given in (3) & (4),

$$\frac{dx(t)}{dt} = Ax(t) + B\,u^a(t) -----------(3)$$

where $u^a(t) = \alpha(t)\, u^c[t - \tau_{ca}(\mathrm{t})]$ OR

$$\frac{dx(t)}{dt} = Ax(t) + B\alpha(t)u^c[t - \tau_{ca}(\mathrm{t})] ------(4)$$

$$y(t) = C\gamma(\mathrm{t})x[t - \tau_{sc}\tau(\mathrm{t})] ----------- (5)$$

where,

$$\gamma(t) = \begin{cases} I^{n\times n} & \text{if controller received x(t) from the plant at time t} \\ 0^{n\times n} & otherwise \end{cases}$$

$$\alpha(t) = \begin{cases} I^{m\times m} & if\ actuator\ received\ control\ command\ from\ controller\ in\ time\ t \\ 0^{m\times m} & otherwise \end{cases}$$

with $x(t) \in \mathbb{R}^{n\times n}$ represents system state, $u^c(t) \in \mathbb{R}^{m\times m}$, represents control inputs computed at controller, $u^a(t) \in \mathbb{R}^{m\times m}$, represents control inputs received at actuator and $y(t) \in \mathbb{R}^{n\times n}$ represents output of the plant. Note that $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ denotes system matrices. Here we assume that sum of both network induced delay be bounded i.e.,$\tau_{sc}(t) + \tau_{ca}(t) < bT_s$ [17] where b denotes delay bound and T_s is sampling interval.

3. TrueTime: Simulator for NCS

Networked control system is overlapping subject of research & study between two areas: Information Systems and Feedback Control System as indicated in figure 1. Different simulators used for independent area and overlapping area i.e., NCS are also shown in figure 1.

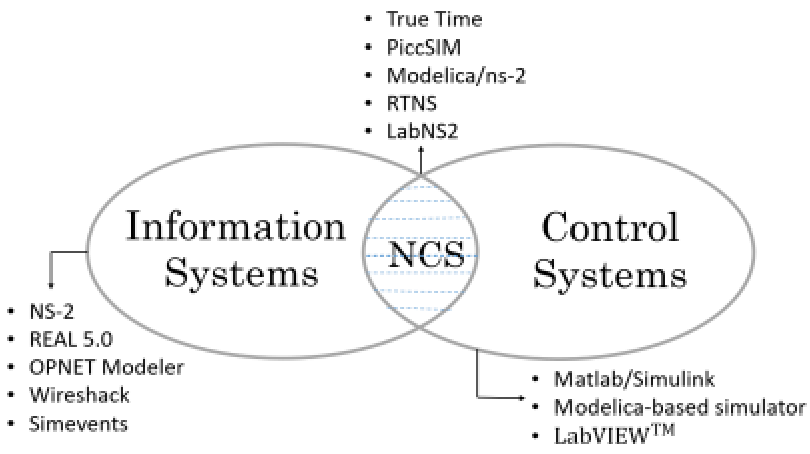

Figure 1: Simulators for NCS [10]

TrueTime is powerful & popular tool used to do analysis of real time wireless NCS, developed by Dr. Anton Cervin & team at Lund University in year 2002. It is continuously upgraded with addition of new features since year 2002. TrueTime toolbox is a MATLAB/Simulink based library of simulation blocks that extends usability of MATLAB/Simulink to simulate networked process control co-simulation of controller task execution in real-time kernels, network transmissions, and continuous plant dynamics. Library of TrueTime contains different blocks like True Time Kernel block, True Time Send block, True Time Receive block, True Time Network block and True Time Wireless Network block as shown in figure 2.

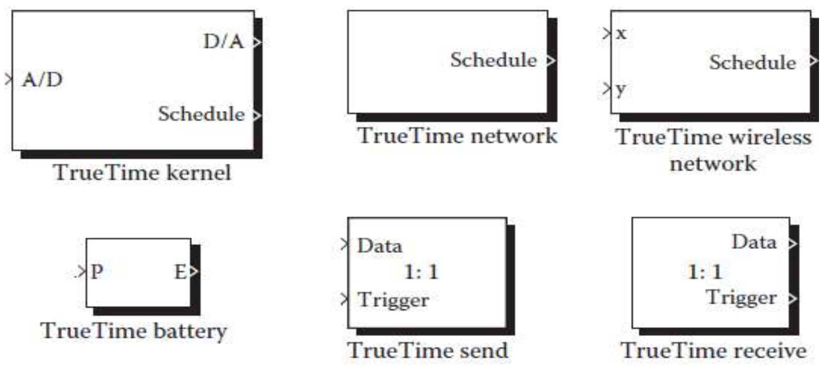

Figure 2: Toolbox of TrueTime [11]

The TrueTime Kernel Network Blocks:

Responsibility of TrueTime kernel block is to look after for I/O and network data acquisition i.e. data processing and calculations. TrueTime kernel block which is brain of every simulink system can be realize by using a control algorithm/logic. In the kernel can be executed several periodic, nonperiodic independent tasks which can cooperate on the same goal. TrueTime software consists of a kernel block and a network block, both variable-step S-functions written in C++. Different blocks in TrueTime blocks are connected with ordinary continuous Simulink blocks in MATLAB to form a real time control system. TrueTime kernel block executes user defined tasks such as interrupts handlers, representing I/O tasks, control algorithms, and communication tasks. The TrueTime kernel block simulates a computer with an event-driven real time kernel, A/D and D/A converters, a network interface, and external interrupt channels.

Execution is defined by user written code functions (C++ functions or m-files in MATLAB) or graphically using ordinary discrete Simulink blocks. Execution time of the any code after simulation may be constant, random or even data-dependent. Generally scheduling policy of the TrueTime block kernel is arbitrary and decided by the user. The dialog box of kernel block in TrueTime is shown in figure 3.

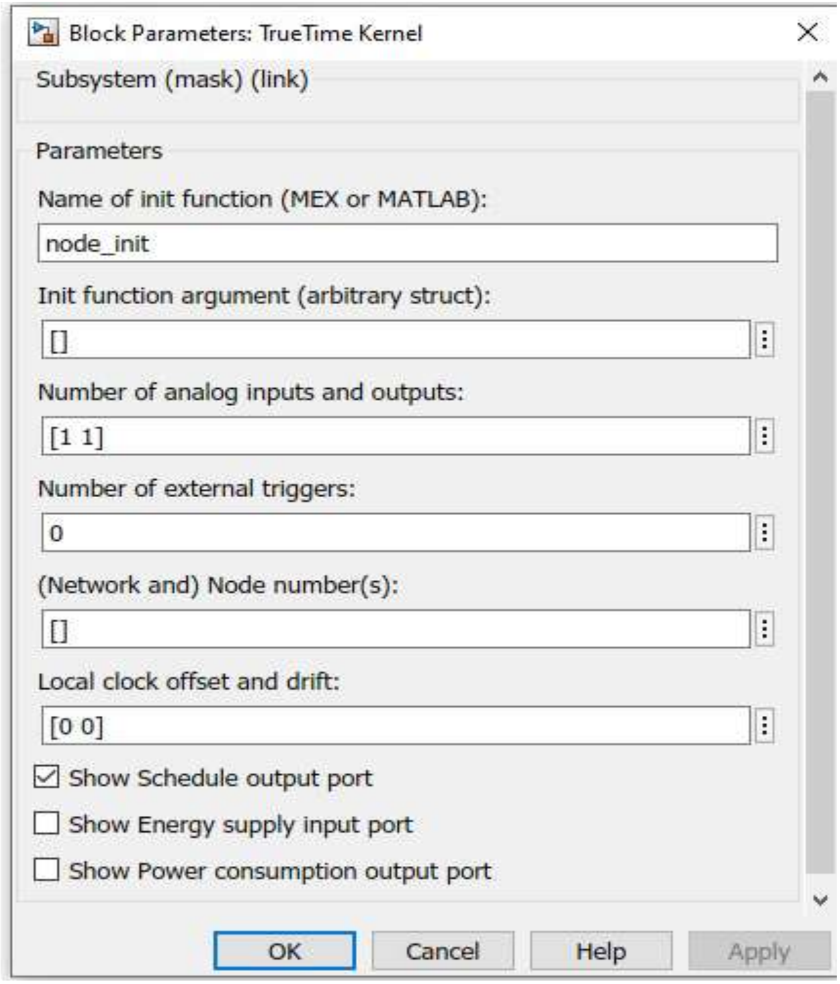

Figure 3: The dialog box of kernel block in TrueTime

The TrueTime Wired/Wireless Network Blocks:

The TrueTime network block simulates medium access and packet transmission (physical and medium access layer) in a local area network. When a node tries to transmit a message, a triggering signal is sent to the network block on the corresponding input channel. When the simulated transmission of the message is finished, the network block sends a new triggering signal on the output channel corresponding to the receiving node. The transmitted message is put in a buffer at the receiving computer node [13].

TrueTime network blocks include generally used wired networks as CSMA/CD (Ethernet), CSMA/AMP (CAN), Round Robin, FDMA, TDMA, Switched Ethernet, FlexRay, PROFINET, NCM. TrueTime Wireless Network blocks include wireless network as IEEE 802.11b/g (WLAN), IEEE 802.15b/g (WLAN). There is also a TrueTime Ultrasound network support in the TrueTime 2.0 beta version. The network blocks are configured by their block mask dialogues as shown in figure 4. There are some common parameters for all networks like the network number, number of nodes, data rate or minimum frame size. The other parameters are network type specific transmit power or receiver signal threshold in wireless networks. There can be a number of network blocks in a model. That is why each network is identified by an id number. A number also identifies the connected nodes. This node id number has to be network specific.

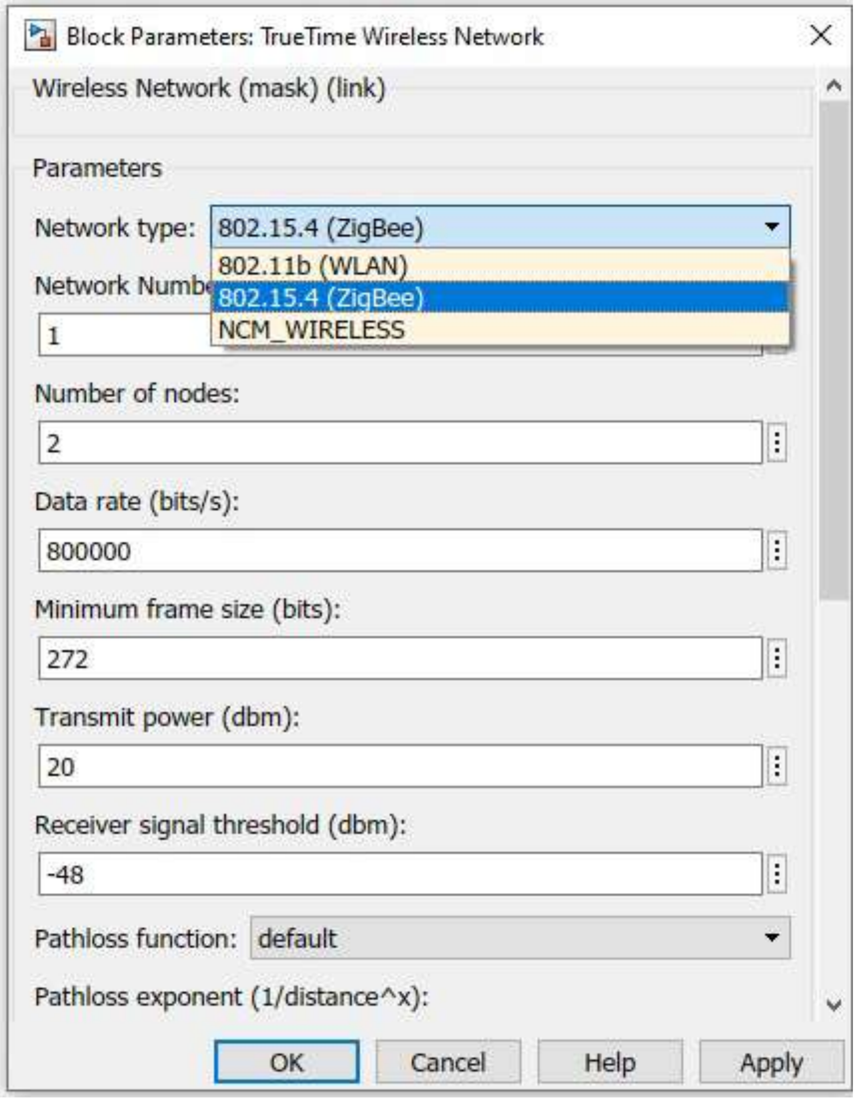

Figure 4: The dialog box of wireless network block in TrueTime

The TrueTime Standalone Network Interface Blocks:

TrueTime Send and TrueTime Receive are standalone network interface blocks which can be used to send messages through the network (using network blocks) without using kernel block. This means that no initialization code or task code must be written. According to [13] the whole network simulation can be created in Simulink without using any m-files, or C++ code. It is possible to mix the standalone blocks with kernel blocks in a simulation. It means that some stations can send messages without m-file task codes (e.g. sensors) and some stations use kernel blocks i.e. controllers.
The Send and Receive blocks are configured through their block mask dialogs which is not shown here. The send block can be time-triggered or event-triggered. The trigger input port can be configured to trigger on raising, falling or either flanks.
The TrueTime Battery Block:
The battery block acts as a power source for the battery enabled kernel blocks. It uses a simple integrator model so it can be both charged and recharged. The only one parameter in its block mask is the initial power. To use the battery, enable the check box in the kernel configuration mask and connect the output of the battery to the E input of the kernel block.

4. Simulation for effect of loss probability in network schedule and system output.

In this section, we will investigate effect of loss probability on network schedule and system output using TrueTime for non-minimum phase and unstable transfer functions. For this transfer function first is simulated using MATLAB [24] with selecting discrete PID controller [23] in ideal or parallel form with sampling 0.01 sec, discrete PID controller is tuned, PID gain parameters K, Ki, Kd [23,24] are updated and K, Ki & Kd parameters are noted and same are used to code init file of Kernel block.

Same transfer functions are simulated using TrueTime under action of PID controller whose simulink diagram is given in figure 5. For wireless simulation using TrueTime, note that same PID parameters K, Ki, Kd are used for coding and developing of m-file to realize Kernel block as discrete PID controller and sampling rate at sensor is kept as 0.01 sec. Output response and wireless network schedule are given and investigated for four different values of loss probability P= 0, 0.3, 0.5, and 0.8 of wireless network in figure 6 to 18. It is observed that for P=0, system provide expected output response because P=0 means implies zero packet dropout at controller and actuator node. As P increases, output of system deteriorates. Network scheduling waveform for P= 0.8, clearly tell us packet dropout in time slots increases and in that time slots output response deteriorates more.

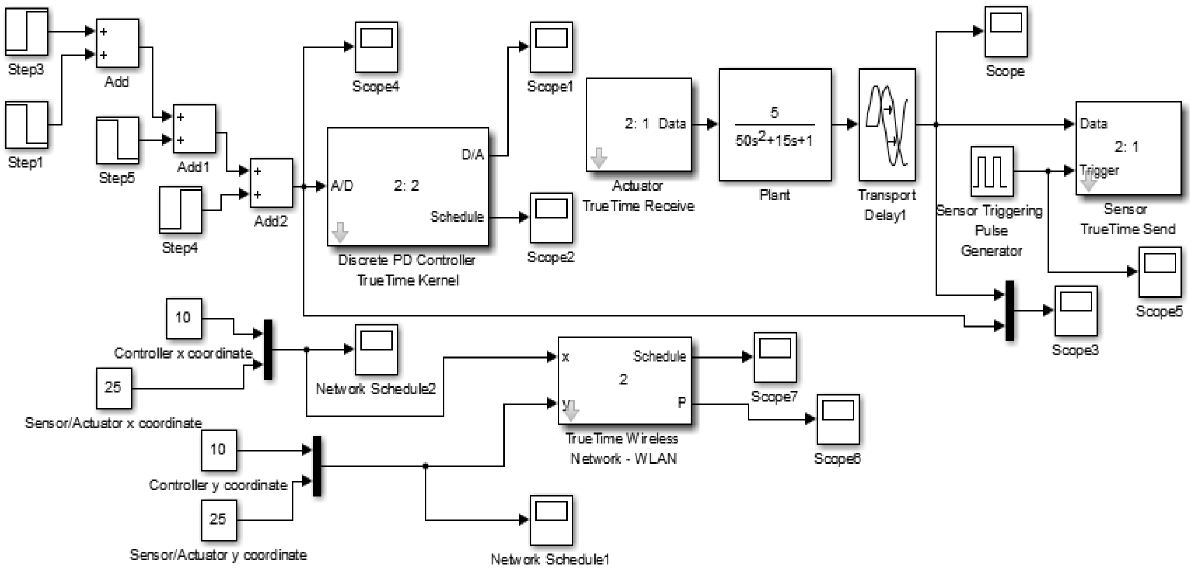

Figure 5: Simulink Diagram Using TrueTime

Following parameters are used for wireless network while doing wireless simulation using TrueTime. Simulink diagram using TrueTime is shown in figure 5.

Type of Wireless Network: 802.11b (WLAN), Data rate: 800,000 Bits per Second

Minimum Frame Size:272 Bits, Transmit Power: 25 dBm

Receiver Signal Threshold: - 48 dBm, Path Loss Exponent: 3.5

ACK Timeout:0.00004 Sec, Error Coding Threshold: 0.03

Loss probability:0, 0.3, 0.5 and 0.8

Development and working

Kernel block in TrueTime is programmed as discrete PID controller by writing initialization code (init file) and function code as PID in MATLAB, because kernel block worked as computer and is responsible for all function like I/O operation, scheduling, A/D and D/A conversion, set up timers, change task attributes etc. Note that one can write C++ code also. No other block in TrueTime requires coding. TrueTime Receive block functions as actuator and TrueTime Send block functions as sensor, both are connected to Plant TF block. Written number 1:1, 1:2, 1:2 written respectively on kernel block, TrueTime Receive and TrueTime Send block indicates that kernel acts as node 1 and TrueTime Receive & TrueTime Send both acts as node 2 and for communication amongst each other they use wireless channel 1 which is 802.11b (WLAN). So first 1 denotes wireless channel and second 1 or 2 denotes node number. Note that sampling rate at sensor node and in init file of discrete PID controller must be equal and preferred value is 0.01 sec.

Output signal of Plant TF block is continuously sampled by sensor block (TrueTime Send), sampled values are sent to the discrete PID controller block (kernel) via wireless network 802.11b (WLAN). Note that this is sensor to controller communication (S to C), delay & packet loss may take here at controller. Depending upon received values from sensor, discrete PID controller reacts & generates signal for actuator, controller send this signal to actuator again via wireless network 802.11b (WLAN). Note that this is controller to actuator communication (C to A), delay & packet loss may also take here at actuator. Actuator accordingly generates the actuating signal to control the plant. We can observe different signal at different blocks using scope, also we can observe network schedule signal.

For plant 1: $G_1(s) = \frac{3-s}{s^2+6\,s+5}$ (non-minimum phase system):

With PID parameters K= 2.6286, K_i = 0.6767, K_d = 0.0623 and loss probability P=0

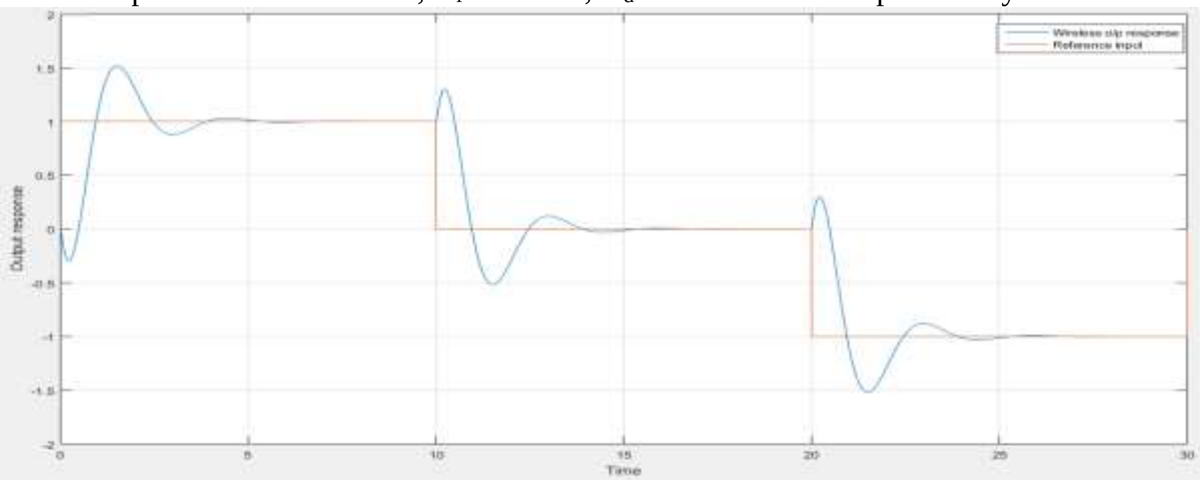

Figure 6: Simulated response using TrueTime for plant 1 with loss probability P = 0

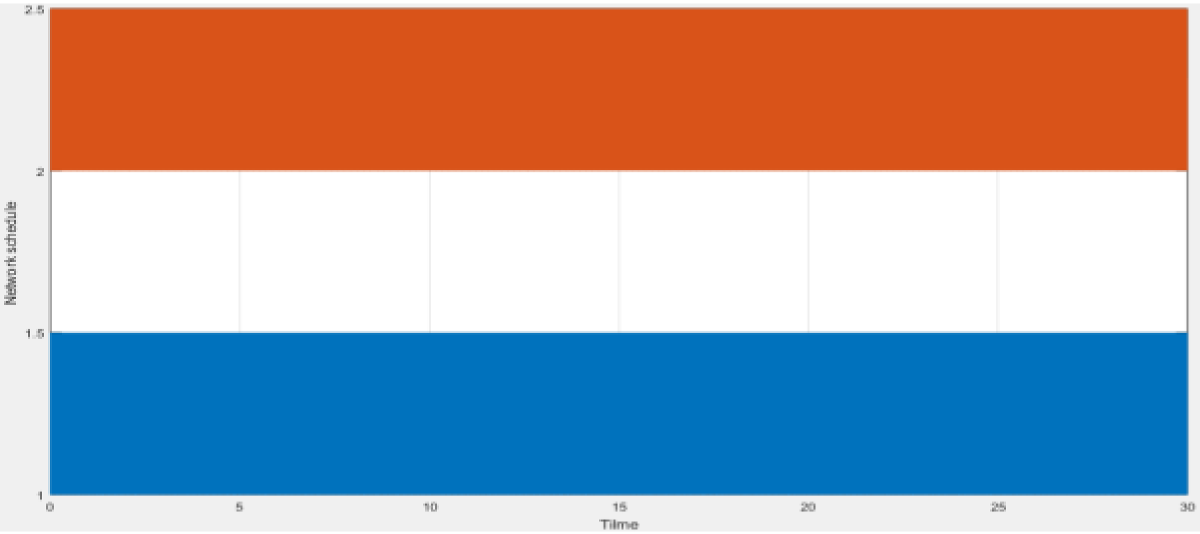

Figure 7: Network schedule waveform for plant 1 with loss probability P = 0

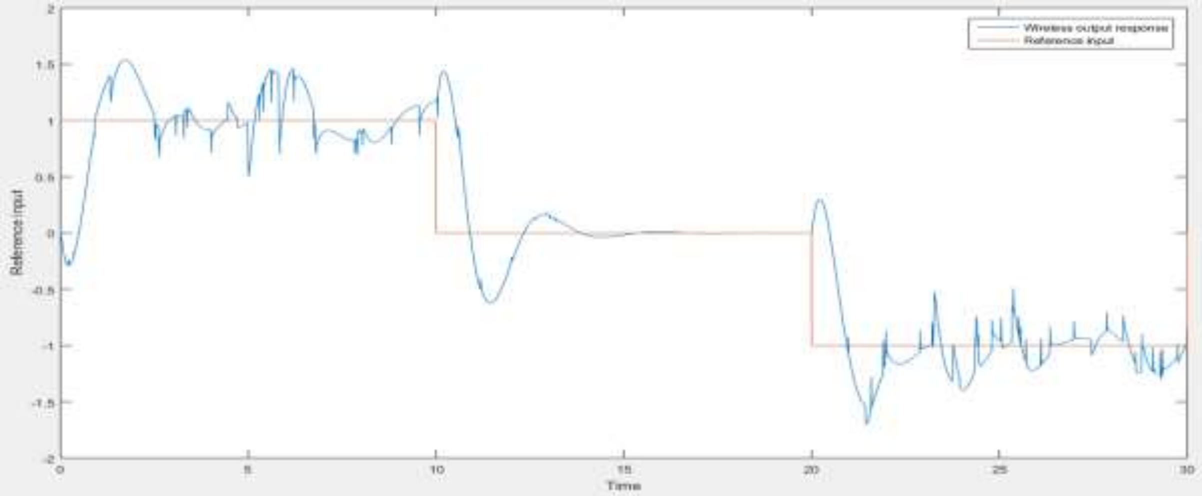

Figure 8: Simulated response using TrueTime for plant 1 with loss probability P = 0.3

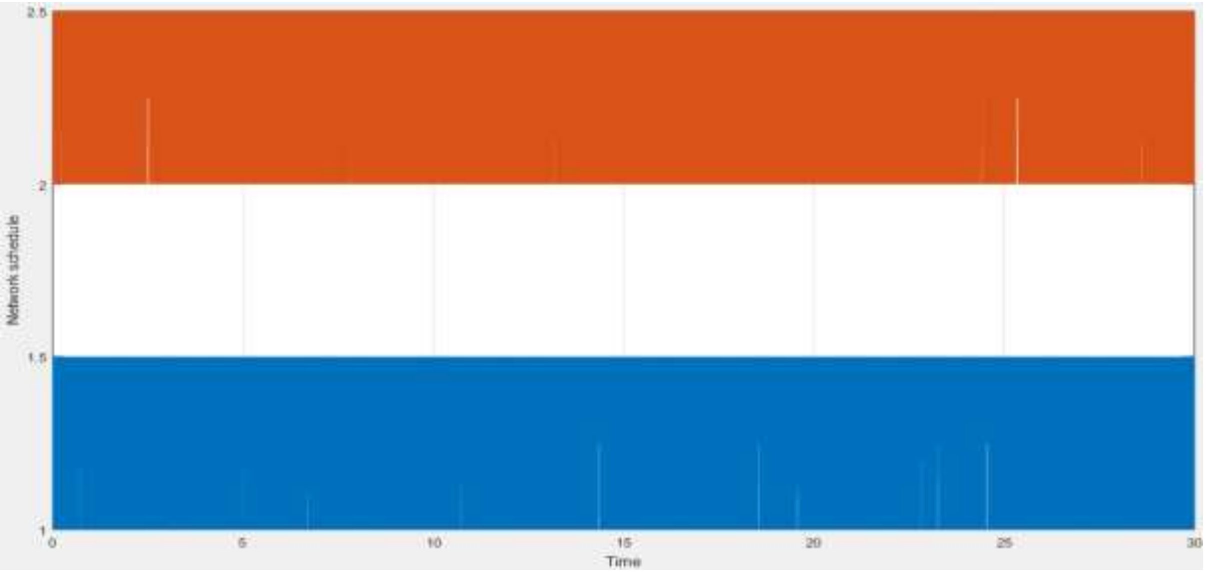

Figure 9: Network schedule waveform for plant 1 with loss probability P = 0.3

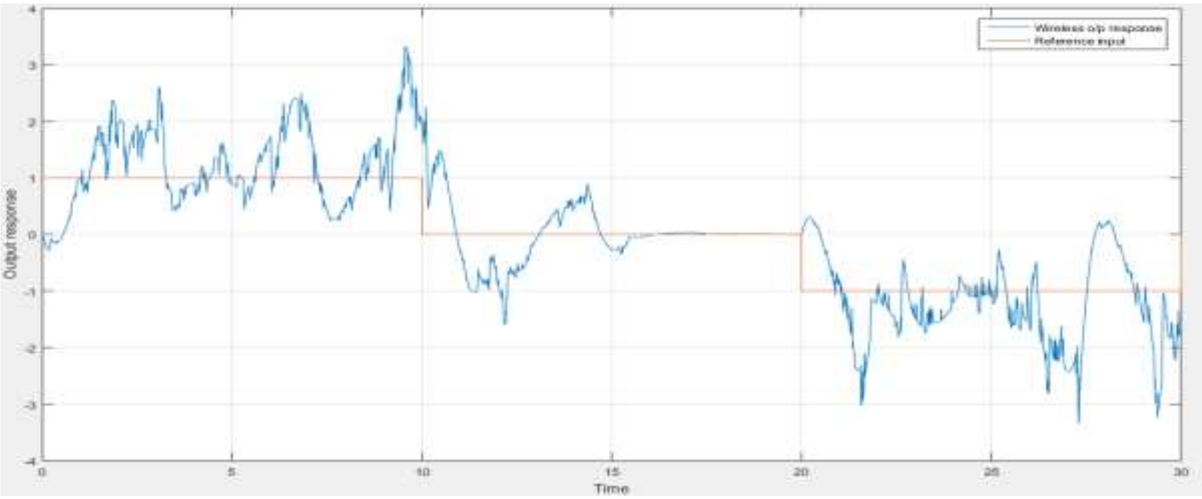

Figure 10: Simulated response using TrueTime for plant 1 with loss probability P = 0.5

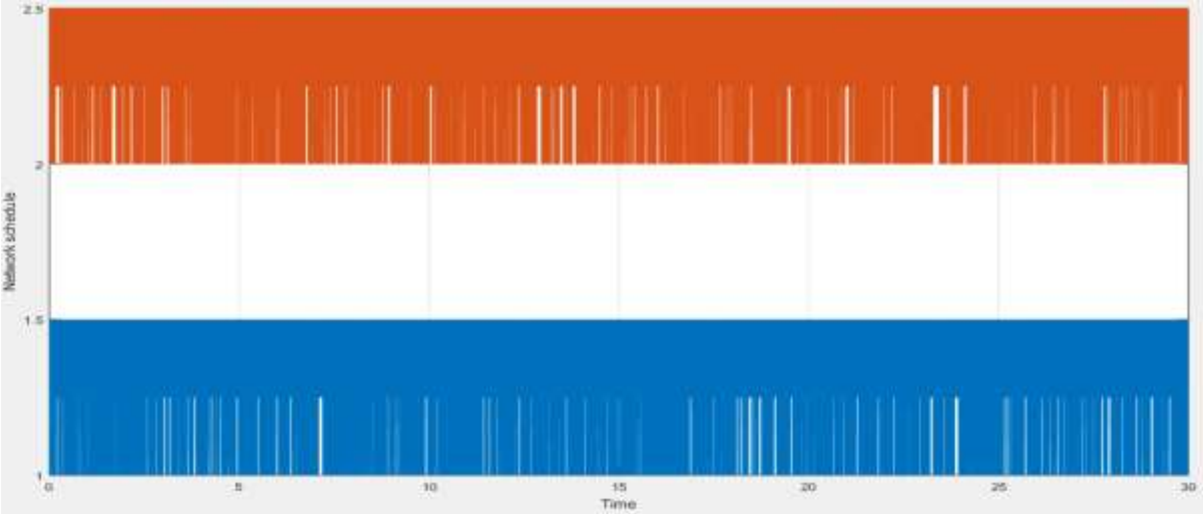

Figure 11: Network schedule waveform for plant 1 with loss probability P = 0.5

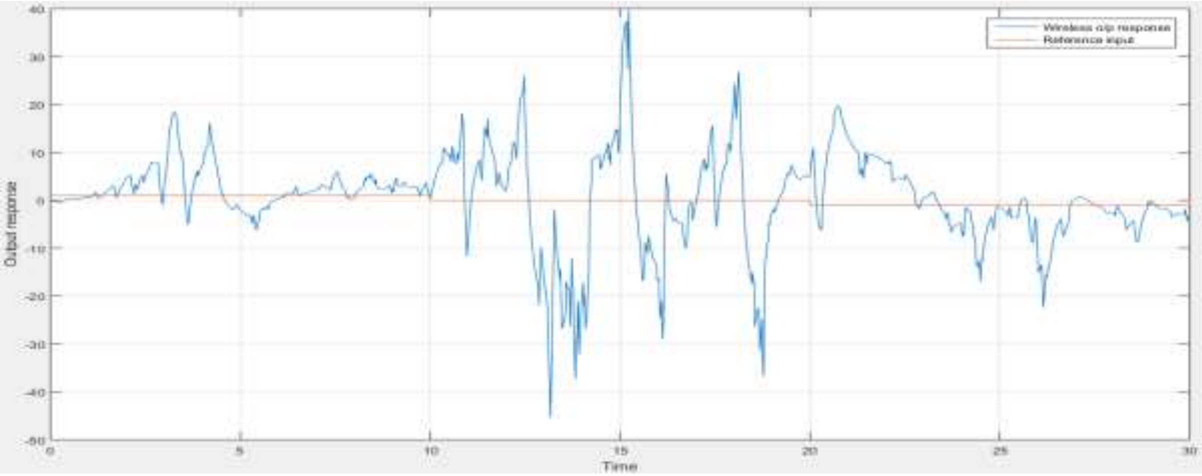

Figure 12: Simulated response using TrueTime for plant 1 with loss probability P = 0.8

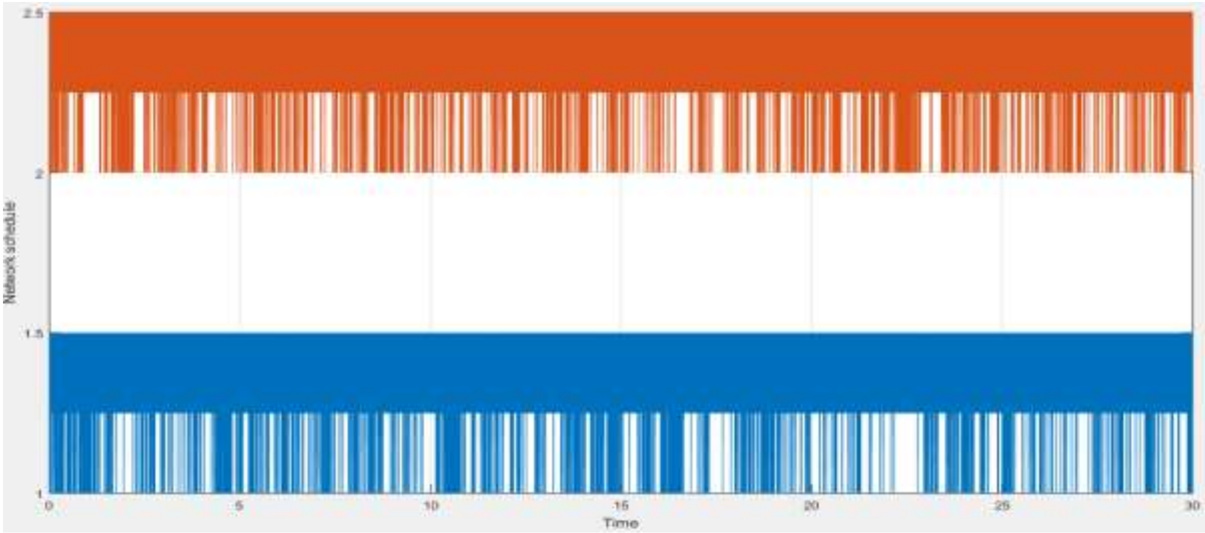

Figure 13: Network schedule waveform for plant 1 with loss probability P = 0.8

For plant 2: $G_2(s) = \frac{3}{s^2 - 5\,s + 6}$ (Unstable system):

With PID parameters K= 70.403, K_i = 0.5222, K_d = 0.1512 and loss probability P=0

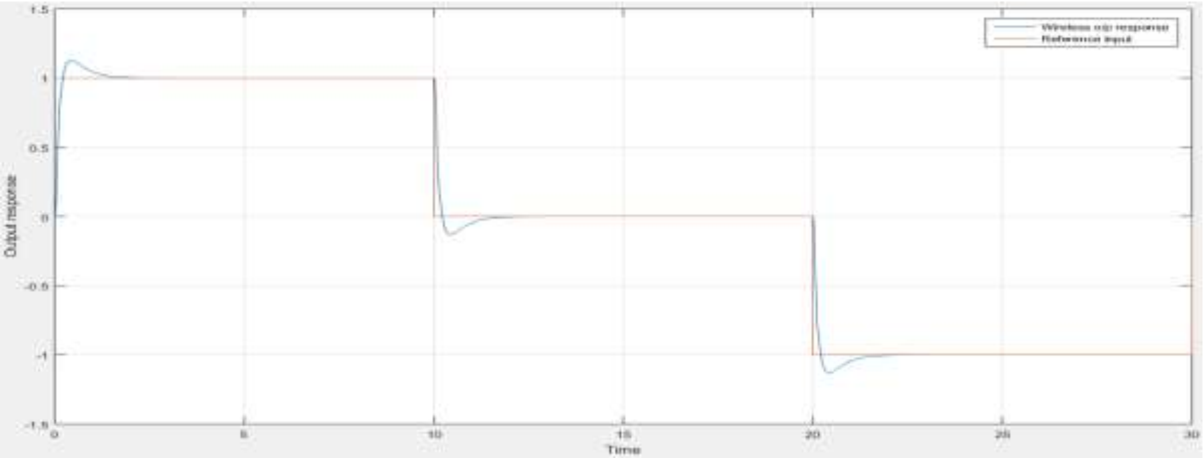

Figure 14: Simulated response using TrueTime for plant 2 with loss probability P = 0

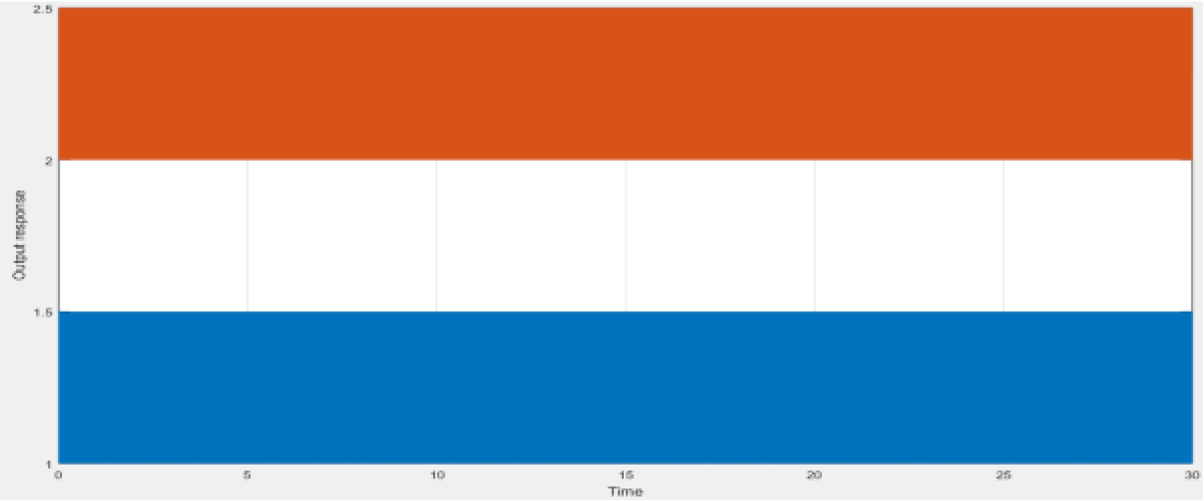

Figure 15: Network schedule waveform for plant 1 with loss probability P = 0

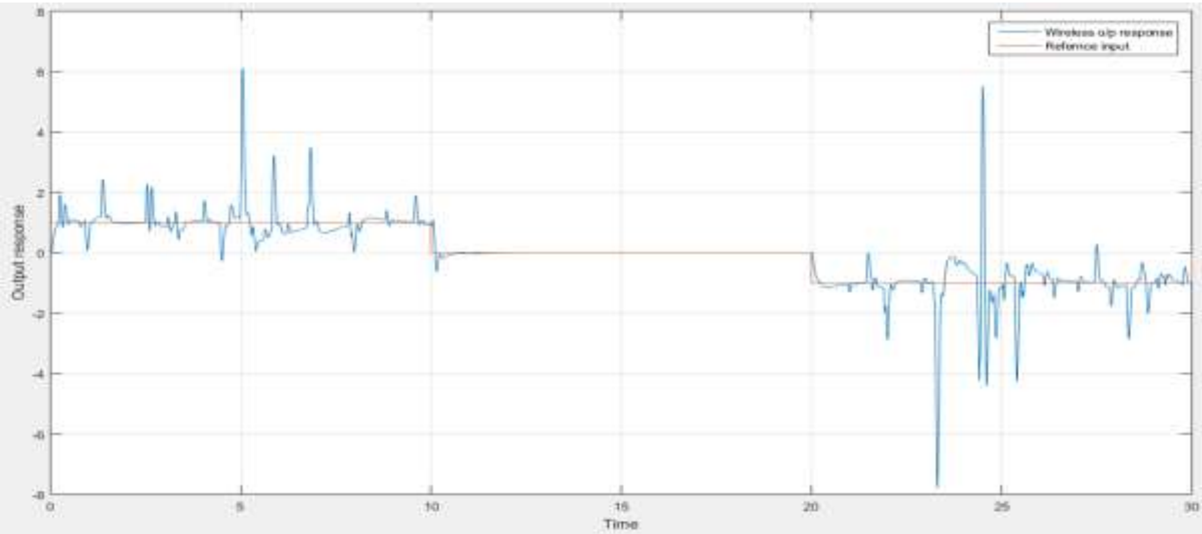

Figure 16: Simulated response using TrueTime for plant 2 with loss probability P = 0.3

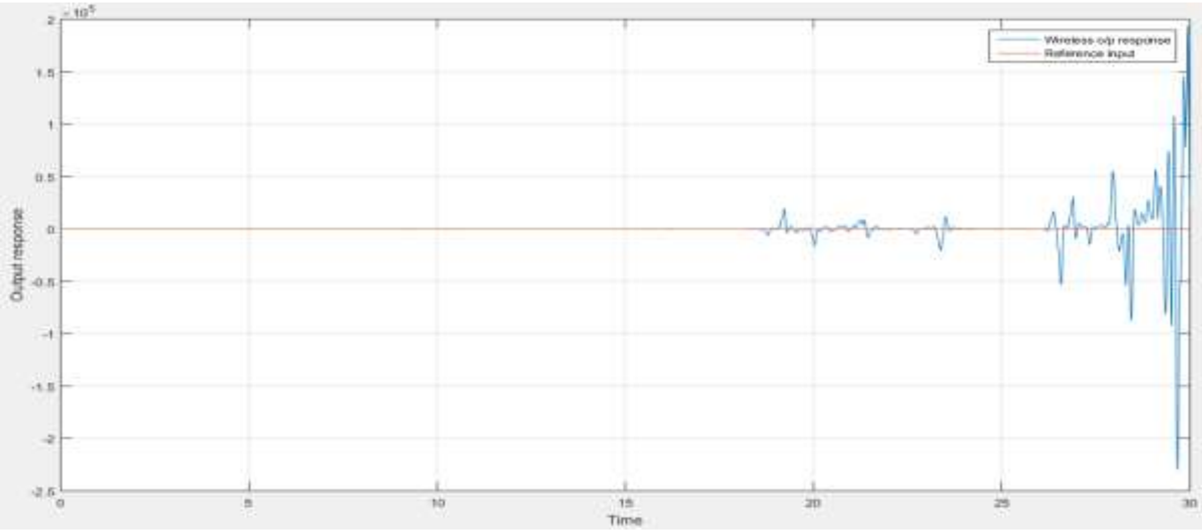

Figure 17: Simulated response using TrueTime for plant 2 with loss probability P = 0.5

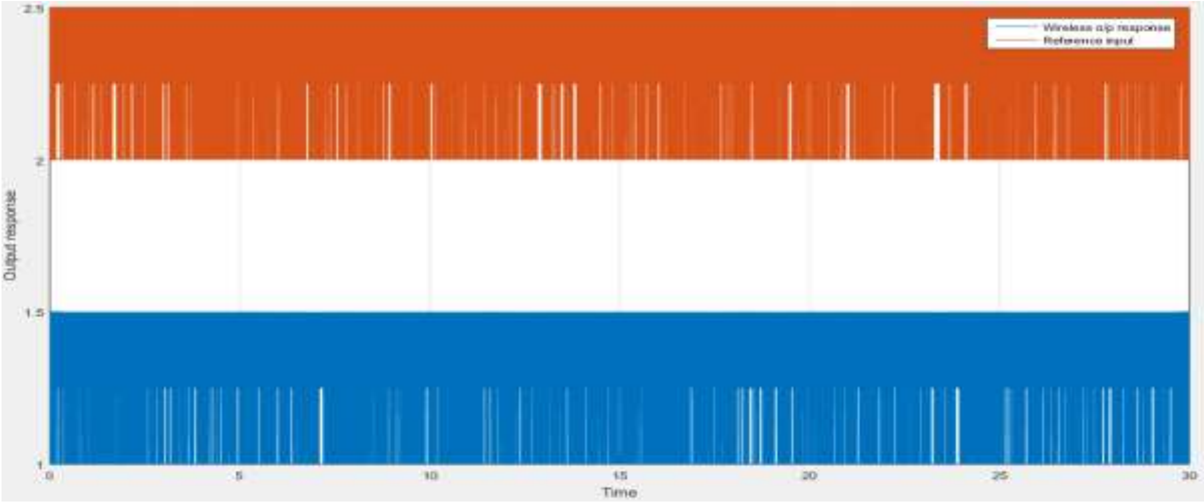

Figure 18: Network schedule waveform for plant 2 with loss probability P = 0.5

5. Wireless simulation for delay system and fourth order system.

For plant 3: $G_3(s) = \frac{0.38*s^2+0.038*s+0.209}{s^4+1.06*s^3+0.56*s^2+0.5*s}$, this plant is simulated using MATLAB as well as TrueTime. PID parameters are taken here from MATLAB simulation. Both responses are shown in figure 18.

With PID parameters K= 17.3826, K_i = 4.6692, K_d = 7.7383 and loss probability P=0

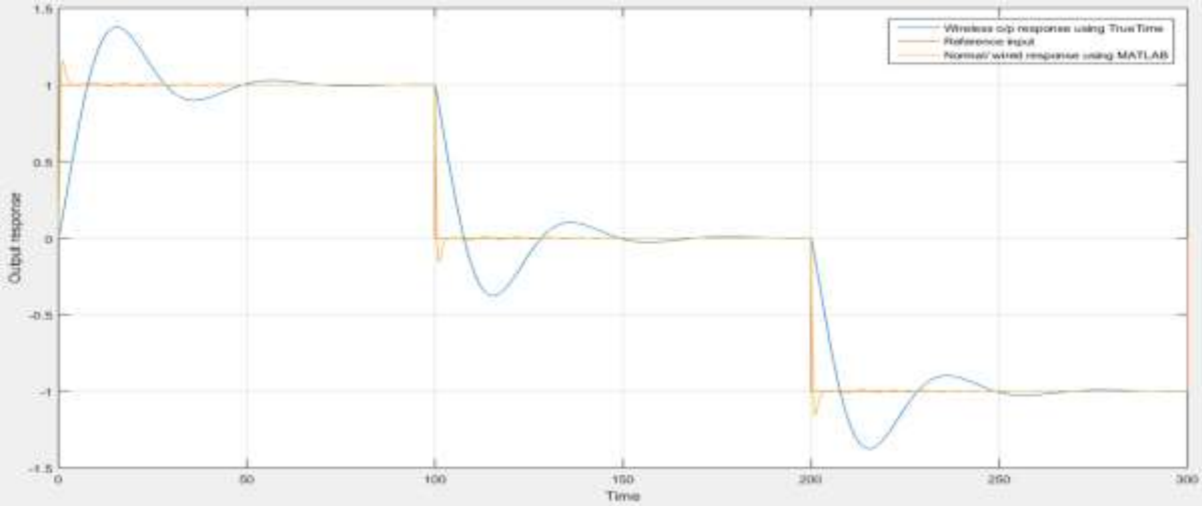

Figure 19: Simulated wireless response using TrueTime and normal response using MATLAB for plant 3.

For plant 4: $G_4(s) = \frac{5}{50\,s^2+15*s+1} e^{-5s}$, this plant is simulated using MATLAB as well as using TrueTime. PID parameters are taken here from MATLAB simulation. Both responses are shown in figure 20.

With PID parameters K= 0.3267, K_i = 0.02083, K_d = 1.0711 and loss probability P=0

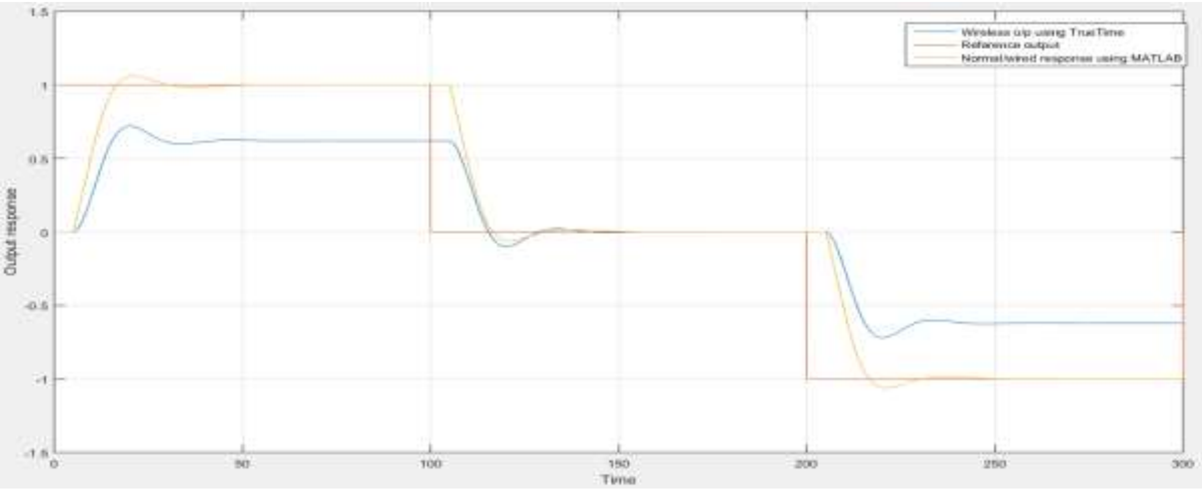

Figure 20: Simulated wireless response using TrueTime and normal response using MATLAB for plant 4.

6. Conclusion

Effect of loss probability of wireless network block in TrueTime on network schedule and system output is investigated for non-minimum phase and unstable system in this paper. Under zero loss probability condition there can be minimum packet dropout at controller and actuator node, hence system provide expected output. With increase in loss probability, possibility of packet dropout at controller and actuator node increases, hence system output deteriorates more from expected output. Also wireless simulation of delay system and fourth order system is done using TrueTime and MATLAB, both responses are as expected.

In wireless networked control system that, when communication between sensor to controller (S-C) and controller to actuator (C-A) takes place through wireless network e.g., 802.11b WLAN, ZigBee then sampling process and sampling rate (h) plays very important role. To get expected output from plant or process, sampling rate (h) used at sensor node and used at PID controller must be same. This very can be well can be experimented using TrueTime which can be considered as a future work.

References:

[1] Wang X, Lemmon MD, “Event-triggering in distributed networked control systems” IEEE Transaction on Automatic Control 56(3):586–601, 2011
[2] Du D, Fei M, “A two-layer networked learning control system using actor-critic neural Network”, Applied Mathematics and Computation 205(1):26–36, 2008
[3] L.A. Montestrque, P.J. Antsaklis, “Stability of model based network control system with time varying transmission time”, IEEE Transaction on Automatic Control, vol.21, no.1, pp.84-99, 2001.
[4] W. Zhang, M.S. Branicky, and S.M. Philips, “Stability of network control system”, IEEE Control System Magzine, vol.21, pp.88-94, 2001
[5] Peng C, Tian Y, “Networked H1 control of linear systems with state quantization”, Information Science 177:5763–5774, 2007
[6] Zhou L, Lu G, “Quantized feedback stabilization for networked control systems with nonlinear perturbation”, International Journal of Innovative Comput Information and Control 6(6):2485–2495
[7] Nesic D, Liberzon D, “A unified framework for design and analysis of networked and quantized control systems. IEEE Transaction on Automatic Control 54(4):732–747, 2009
[8] Niu Y, Jia T et al. “Output-feedback control design for NCSs subject to quantization and Dropout”, Information Science 179(21):3804–3813, 2009
[9] Rachana Ashok Gupta, Mo-Yuen Chow, “Networked Control System: Overview and Research Trends’’ IEEE Transactions on Industrial Electronics, Vol.57. No.7, July 2010. [20] [10] A Ph.D. Thesis by Ncedo Sandiso Mkondweni, “Design and Implementation of Linear Robust Networked Control System” Cape Peninsula University of Technology, Cape Town, South Africa, March 2013.
[11] TrueTime 2.0 Reference Manual by Anton Cervin, Dan Henriksson and Martin Ohlin, Department of Automatic Control, Lund University, Sweden, 2016.
[12] Mojabata Ghodrati and Alireza Saheb, “Design and Simulation of Fuzzy Control in Network Control, IJCSNS International Journal of Computer Science and Network Security, VOL.18 No.9, pp.1-11, September 2018
[13] L. Farkas and J. Hnat, “Simulation of Networked control system using TrueTime” pp. 1-11, 2013
[14] Martin Andersson, Dan Henriksson, Anton Cervin and Karl-Erik Arzen, “Simulation of Wireless Networked Control Systems”, Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005 Seville, Spain, December 12-15, 2005.
[15] Sabo Miya Hassan, Rosdiazli Ibrahim, Nordin Saad, Vijanth Sagayan Asirvadam, Kishore Bingi and Tran Duc Chung, “Fuzzy Adaptive Setpoint Weighting Controller for WirelessHART Networked Control system” http://dx.doi.org/10.5772/intechopen.70179, pp. 1-21, 2017.
[16] Shayok Mukhopadhyay, Yiding Han and YangQuan Chen, “Fractional Order Networked Control Systems and Random Delay Dynamics: A Hardware-In-The-Loop Simulation Study”, American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA, pp. 1-6,2009.
[17] A Text Book by Jagannathan Sarangpani and Hao Xu,” Optimal Networked Control System with MATLAB” Published by Taylor & Francis in 2016.Missouri University of Science and Technology Rolla, Missouri, USA
[18] B. Wittenmark, K. J. Astrom, and K.-E. Arzen, “Computer control: An overview,” IFAC Professional Brief, Tech. Rep., 2002.
[19] K. Arzen, “A simple event-based PID controller,” in *IFAC World Congress*, 1999.
[20] P. Tabuada, “Event-triggered real-time scheduling of stabilizing control tasks,” *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680-1685, 2007.
[21] X. Wang and M. D. Lemmon, “Self-triggered feedback control systems with finite-gain L2 stability,” *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 452-467, 2009.
[22] J. Lunze and D. Lehmann, “A state-feedback approach to event-based control,” *Automatica*, vol. 46, no. 1, pp. 211-215, 2010.

[23] A Text Book by Ogata Katsuhiko, "Modern Control Engineering" Pearson Publication, India, 2013
[24] A Reference Manual of MATLAB R 2016