

APPLICATION OF TRUETIME FOR WIRELESS PROCESS CONTROL

Duong Minh Duc, Do Trong Hieu

Hanoi University of Science and Technology; duc.duongminh@hust.edu.vn, hieu.dotrong@hust.edu.vn

Abstract - The fast development of the wireless technologies has opened new potential for wireless networked process control system. With wireless network, the scalability of the systems will be extended. In order to analyse and evaluate the efficiency of a wireless networked process control system, a real-time network simulator is required. In this paper, we introduce True Time, an open source Simulink package for simulation of networked control systems. By using True Time, we present an analysis and evaluation of the effects of communication network to the performance of a wireless process control system. Wireless environment is simulated using True Time toolbox. Other parts of the system such as actuator, sensor, controller are real devices.

Key words - wireless networked control system; process control; True Time; PID

1. Introduction

In conventional digital control systems, the outputs of the plant are measured by sensors and sampled at periodic intervals by analog-to-digital (ADC). A control algorithm applied to the samples by the controller, and the result of the control is generated at the output of the controller generally as a zero-order hold (ZOH) signal. There is not any problem, from the theoretical point of view, with this system. However, there are several limitations in system implementation such as:

- The dependence on a single computer of calculating the control algorithm.
- If the plant is physically large, the signals must travel in long distances to the control computer. This causes much noise and disturbance.
- There is difficulty in maintenance and troubleshooting because of long connections.
- Exists less flexibility of the control system when expanding the system or changing the control algorithm.

To overcome these limitations, networked control systems can be considered. There is networked control systems where the sensors, controllers and actuators occupy in different nodes of a network point to overcome the limitations of conventional digital control systems.

Recently, the fast development of the wireless technologies has opened new potential for wireless process control in the field of industrial control [1]. Compared to traditional wired process control, the use of wireless network can permit flexible installations, mobile operation, and rapid deployments [2].

In wireless networked control system, a communication network is the backbone, and the network problems such as network delay, jitter, and losses affect the performance of the system. The control designers have to consider these network problems to guarantee the required performance. In order to analyse and evaluate the wireless networked control system, a real-time network simulator is necessary. True Time [3] is one of the best simulators for

this purpose with various advantages. True Time has been used in various simulation analysis [4-6]. However, in the aforementioned researches, True Time is only used with simulated plants, not with the real plants that limits the application range of True Time. In addition, before implementation of a real networked control system, the effects of network problems to real plants should be analyzed and evaluated carefully, so that the controller can be designed more precisely. Thus, to solve the above problems, in this paper, we propose a testing structure for a networked control system in which communication network environment is simulated using True Time. Other parts of the system such as actuator, sensor, controller are real devices. This structure first will verify the application of True Time with real systems. Second, the evaluation of networked control system can be done.

The rest of the paper is organized as follows. In section 2, the True Time simulator is introduced. The analysis and evaluation of a typical networked control system with True Time is presented in section 3. Section 4 concludes the results and future works.

2. Wireless Control using TrueTime

True Time is a simulator based on Matlab/Simulink which simulates the execution of task in real-time systems, networks (wired or wireless) and dynamic plants. True Time composes some library blocks (Figure 1) and a collection of C++ functions. More details about these blocks and functions can be found in [7].

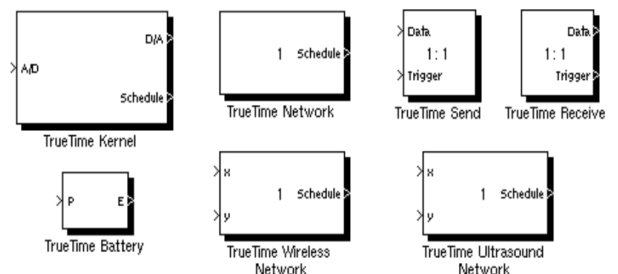


Figure 1. True Time Library

True Time Kernel Block

A block contained in the library is a kernel block, it is a S-function in MATLAB to simulate a real-time CPU with Kernel, A/D and D/A and network interface. Tasks can be cyclic operations to simulate operations such as controllers and I/O tasks. The kernel block is configured through a dialog box with parameters.

True Time Network Block

This kind of block simulates medium access and packet transmission in a local area network. When a node transmits a message, a triggering signal is sent to the network block on the corresponding input channel.

Whenever the transmission of the message is completed, the network block sends a new triggering signal on the output channel corresponding to the receiving node. True Time supports six simple models of networks including CSMA/CD (e.g. Ethernet), CSMA/AMP (e.g. CAN), Round Robin (e.g. Token Bus), FDMA, TDMA (e.g. TTP), and Switched Ethernet. Since the propagation delay is typically very small in a local area network, it is ignored. The network block can be configured through the block mask dialog.

True Time Wireless Network Block

This block has two inputs which allow users to define the relative position of nodes in network to take into account the path-loss of the radio signals. At the beginning, True Time supports two network protocols including IEEE 802.11b/g (WLAN) and IEEE802.15.4 (ZigBee). Some parameters of the network are configured through the block mask dialog.

True Time Battery Block

This block has only one parameter to be configured through the configuration mask, that is the initial power.

True Time Send/ Receive Block

These blocks are used to transmit/ receive messages using the network blocks without using kernel blocks. This permits users to create True Time network simulations without having to initialize kernels, create and install interrupt handlers, etc. In other words, it is possible to create a whole network simulation in Simulink without any M-files at all. The Send/Receive blocks can be configured through the block mask dialogs.

Later in this paper, in order to analyse and evaluate a wireless networked process control system, True Time toolbox will be used to simulate the communication environment. A Wireless Lan Networked Control scheme is constructed whose wireless environment between controller and real process is simulated using True Time.

3. System analysis and evaluation with True Time

In this section we will introduce an example of using True Time to analyse and evaluate a wireless pressure control system. At first, the process model is identified. Then a controller is designed for the system in case of direct connection and wireless connection between the controller and the process. The effect of communication will be analyzed and evaluated.

3.1. System identification

The structure of the controlled system which contains a low-pressure air circuit supported on a panel is shown in Figure 2 and 3. A compressed air source with 40 psi pressure and filter supplies compressed air to the system. There are two separate branches which provide air for the process and for valve control.

The process branch (Branch 2) includes a variable area flow meter, an orifice block with changeable orifice plates, differential and point of measure pressure sensors, a pneumatically operated control valve, and a regulator. The process air flow is discharged to atmosphere via manual

diffused outlets. Users can switch an air receiver tank in and out of the circuit.

The valve control branch (Branch 1) includes a regulator and an electrically operated current to pressure input converter. This branch is used to regulate the pneumatic control valve. The pressure transmitters provide signal conditioning function for the sensors and a linear differential pressure sensor output is given by the Differential Pressure Transmitter. For diagnostic reason, some pressure gauges are installed to indicate the pressures around the system.

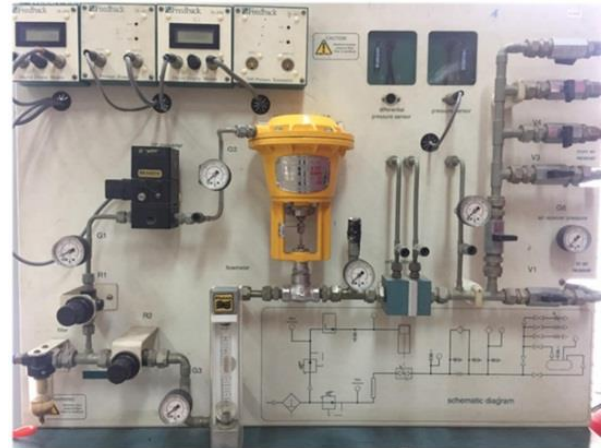


Figure 2. Pressure control experiment

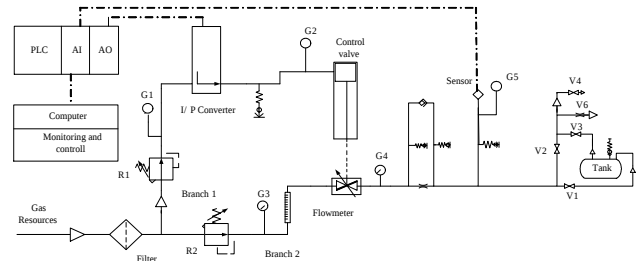


Figure 3. Pressure control schema

In this paper, the PLC Compact Logix 1769-L32E (Allen Bradley – Rockwell Automation PLC) is used as the pressure controller. Processor, power supply, inputs, outputs and other units are realized in the form of modules. Here is some basic information about this PLC:

- User Memory: 750 KB.
- Optional Compact Flash: 64/128 MB.
- Communication ports: Ether Net/IP & RS232 (DF1 or ASCII).

The transfer function of the process will be obtained by using step response identification [8]. The method includes the following steps:

- Keeping the process running to its steady state.
- Changing single step input variable.
- Collecting input and output data at the same time
- Graphical calculating model parameters from process reaction curve.

The graphical working out for determination of the parameters for a first order with dead-time model is shown in Figure 4. The transfer function of the system is as

follows, with $U(s)$ denoting the input and $Y(s)$ denoting the output, both expressed in Laplace transform:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{k}{Ts+1} e^{-Ls} \quad (1)$$

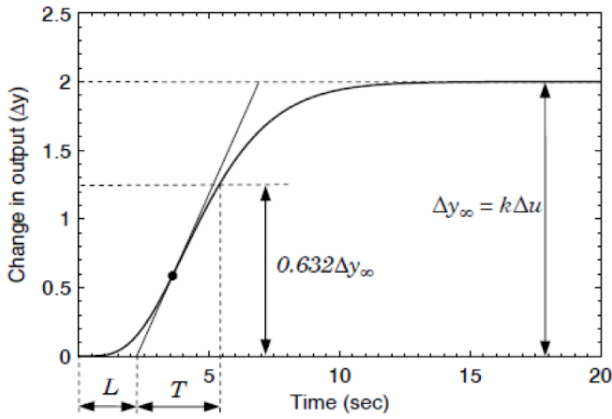


Figure 4. Graphical identification

Figure 5 shows the air flow pressure in process branch when the input signal (valve opening in percentage range) is 30%. The response in this figure can be approximated to the curve in Figure 4. The identification process is then performed using graphical method as described above and one gets:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1.1}{42s+1} e^{-0.5s} \quad (2)$$

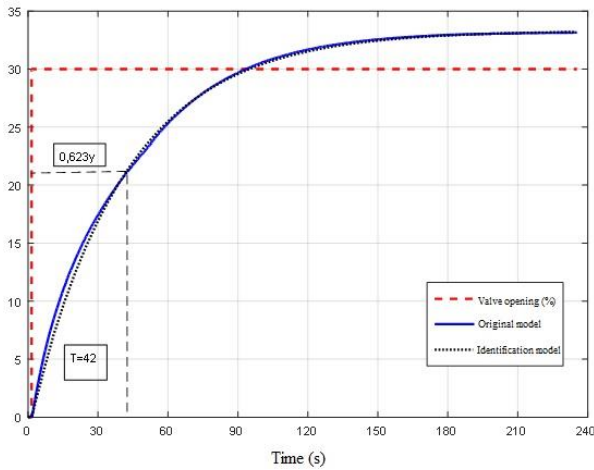


Figure 5. Input and output data of the system

3.2. Direct connection between the controller and the process

A PID controller is implemented in PLC to perform a wired control scheme as in Figure 6.

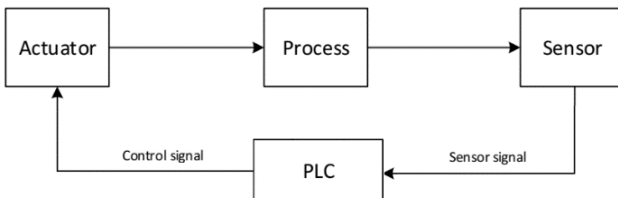


Figure 6. Wired control system

The parameters of the PID controller were originally calculated using Ziegler–Nichols method, then after being well tuned by the trial and error method to achieve a response time of about 20s and an over-shooting less than 5%, these parameters are provided by $K_p = 58$, $T_i = 0.1677$, $K_D = 20$. The performance of the controller is shown in Figure 7 and the input signal is shown in Figure 8.

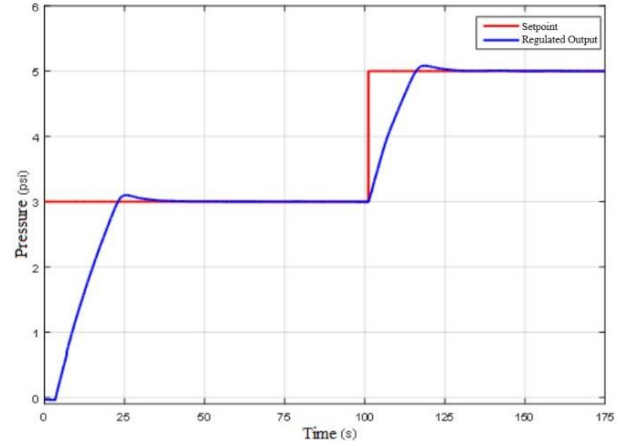


Figure 7. Closed-loop response of wired process control

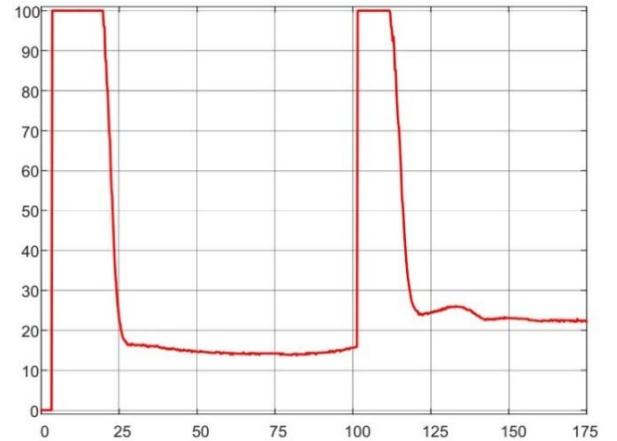


Figure 8. Valve opening (%)

In the following section, we will study the control scheme in wireless environment based on True Time.

3.3. Wireless connection between the controller and the process

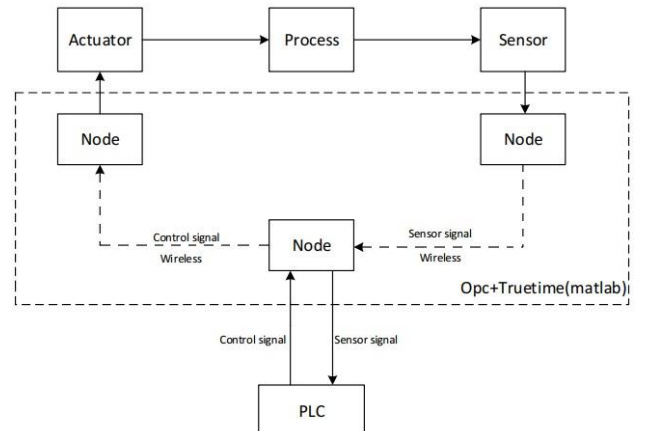


Figure 9. Data transmission model

In this section, in order to analyse and evaluate a wireless networked process control system, we will use True Time to simulate the communication environment between the controller and real plant. A Wireless Networked Control scheme is constructed as in Figure 9. Wireless environment is simulated using True Time toolbox. Different from several previous works, other parts of the system such as actuator, process, sensor, controller (PLC) are real devices. In order to connect True Time/Matlab with PLC, OPC (Object Link and Embedding for Process Control) is used. Each equipment (actuator, pressure sensor, PLC) is then attached to a node

in wireless network.

Figure 10 shows the control scheme realized in True Time/Matlab. Three nodes (actuator node, sensor node and controller) are created using Kernel blocks. These nodes communicate through a Wireless Network block which simulates the wireless environment. The sensor node receives the data from the pressure sensor (through OPC) then send it to the controller node. This controller node then communicates with PLC and receives the control signal, which is later transmitted to actuator node associated with real actuator.

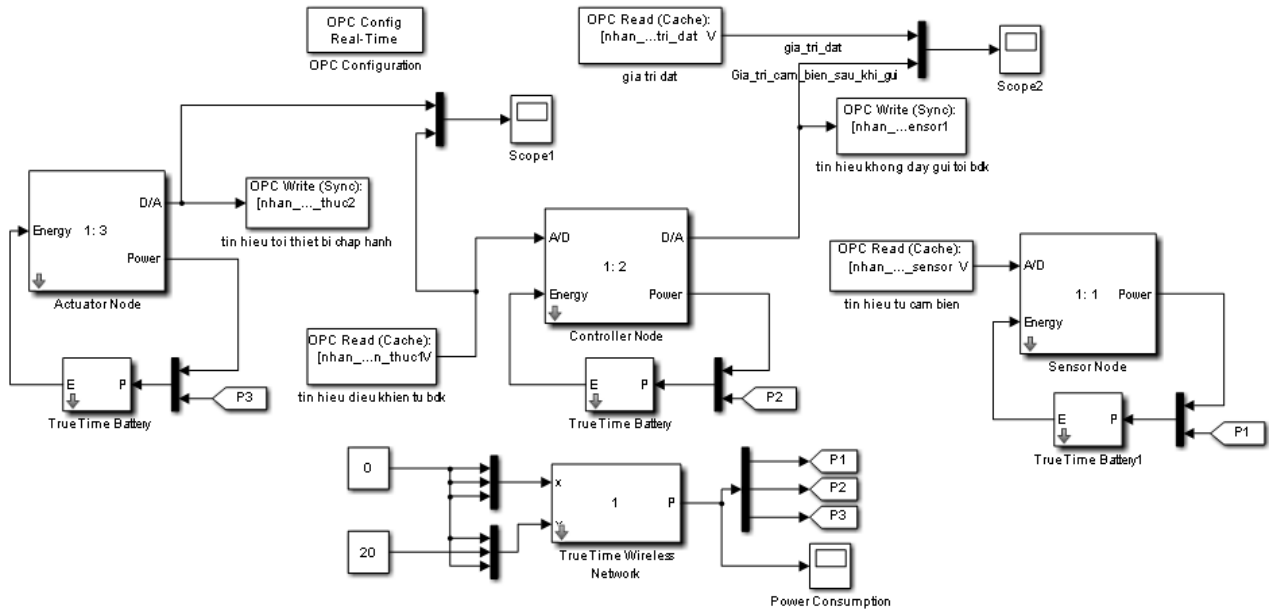


Figure 10. Block scheme of the configuration

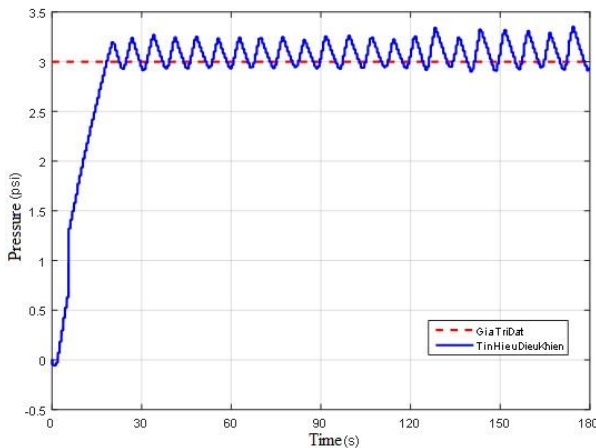


Figure 11. Wireless control first test

At first, the wireless control system is tested with the PID parameters calculated in Section 3.2. As seen in the graph in Figure 11, the closed-loop response oscillates around the set point. The reason is when we design the controller in section 3.2, the effects of communication environment have not been considered. It is found that the communication environment can degrade the system performance. Therefore, it is necessary to re-calibrate the parameters of the controller for the case of wireless networks.

The new parameters of PID controller are chosen as:

$$K_p = 12, K_i = 0.1, K_d = 20$$

With this controller, we get the results as in Figure 12 and Figure 13. The settling time is about 23s and the overshoot is 5% less than 10% as required. The control valve acts not as smooth as in wired control case but it is acceptable.

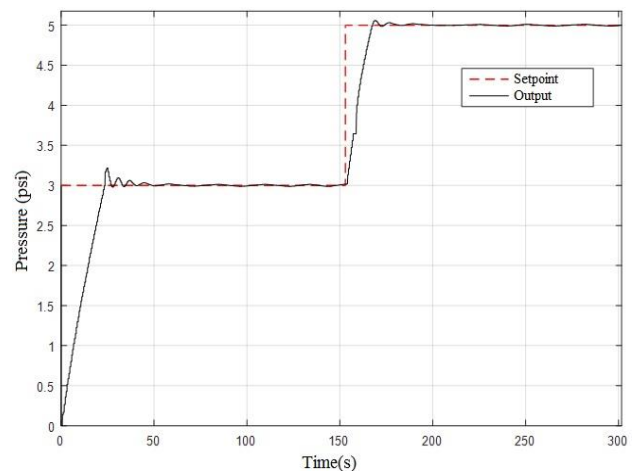


Figure 12. Closed-loop response of wireless process control

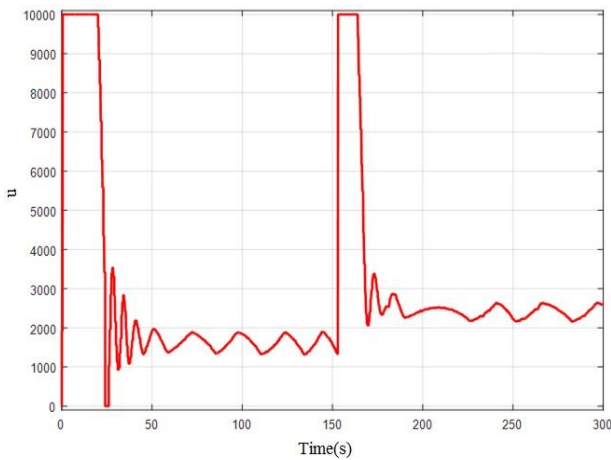


Figure 13. Valve opening (%) of wireless process control

4. Conclusion

This paper deals with an implementation of wireless process control schema using True Time, which is an open source toolbox in Matlab. The simulation and experimental results show a promising approach for studying wireless process control. The toolbox is easy to understand and can be considered as an interesting simulation tool for researcher in wireless network domain. Our future works will focus on application of some advanced control

methods in wireless process industry.

REFERENCES

- [1] D. Caro, "Wireless Networks for Industrial Automation" ISA Press, 2004.
- [2] R. A. Gupta and M-Y. Chow, "Networked control system: overview and research trends", *IEEE Trans on Industrial Electronics*, Vol 57, No. 7, July 2010, pp. 2527-2535.
- [3] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, K. ErikÅrzén, "How Does Control Timing Affect Performance? Analysis and Simulation of Timing Using Jitterbug and TrueTime", *IEEE Control Systems Magazine*, 23:3, pp. 16–30, June 2003.
- [4] J. Fu, C. Xu, B. Kan, and H. Wang, "Design and Simulation of Control Algorithms for WINCS", *Wireless Communications, Networking and Mobile Computing*, Beijing, Sept. 2009, pp. 1-4.
- [5] L. Havet, A. Guenard, F. Lion, "Samovar: An evaluation framework for real time applications deployment over WSANs", *2010 IEEE Conference on Emerging Technologies and Factory Automation*, Bilbao, Nov. 2010, pp. 1-8.
- [6] N.P. Aung, Z. M. Naing, M. M. M. Wai, and H. M. Tun, "Simulation of Wireless Networked Control System Using TrueTime and Matlab", *International Journal of Scientific and Technology Research*, Vol. 5, Issue 06, June 2016, pp 31-35.
- [7] A. Cervin, D. Henriksson, M. Ohlin: "*TRUETIME2.0 – Reference Manual*". Department of Automatic Control, Lund University, Sweden, April 2016.
- [8] Thomas Marlin, "*Process Control: Designing Processes and Control Systems for Dynamic Performance*", 2nd Edition, McGraw-Hill, 2015.

(The Board of Editors received the paper on 10/9/2019, its review was completed on 10/02/2020)