Indian Journal of Science and Technology, *Vol 8(11), 53613, June 2015*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

TCP/IP Based Control and Automation of Temperature Process

D. Pamela* and M. S. Godwin Premi

[1]Department of Communication Engineering, Sathyabama University, Chennai - 600119, Tamil Nadu, India;
pamela@karunya.edu
[2]Department of Electronics and Control, Sathyabama University, Chennai - 600119, Tamil Nadu, India;
godwinpremi@yahoo.com

Abstract

Objectives: In the present scenario, modern Industrial and Commercial Systems require communication networks to exchange information between spatially distributed communicating equipments. The challenge for researches in this area is to compensate for transmission delay, packet loss and manage network traffic. This paper focuses on design of controllers for one such application where the process and the controller communicate through Ethernet via TCP/IP protocol. **Methods/Analysis:** Different control strategies for temperature control in Networked control System, is analyzed and the performance of Conventional and Model based controllers over the network was compared in terms of performance metrics like settling time and peak-overshoot. The transmission delay was deliberately introduced into the system and the controller performance was analysed for different traffic levels by introducing imaginary nodes in the network. The simulation was done using LabVIEW. **Findings:** The obtained results prove that Model based controllers perform better in networked environment, offering faster settling time and reduced peak overshoots. **Conclusion**: The network delay and traffic introduced into the process disturbs the stability of the system, but the control strategies adopted in this work offers better performance and maintain the stability of the system.

Keywords: IMC Controller, Network Control System, PID, Time Delay, TCP/IP

1. Introduction

The trend of modern Industrial and Commercial Systems require integration of both computing and control of process into different levels of machine operations and information process to reduce cost. The goal of this project is to model a system that uses a network protocol to communicate between the process and the controller and to monitor, compute and control the process parameters via a network protocol. In the recent years, Network Control System has gained attentions for research and development. NCS is a classification of control systems in which the sensors, final control elements, predictors and controllers are connected through communication Networks[1]. The major advantages of NCS are in terms of its cost effectiveness, ease in maintenance and diagnosis, higher level of flexibility and so on. The conventional field-bus networks dedicated for industrial usage are now slowly being replaced with TCP/IP interfaces using LAN and internet. The sensor nodes and the actuators are connected using Ethernet or LAN communicating using TCP/IP protocol[2]. Pondering on the challenges of networked control system, the PID and Internal Model Controller has been designed and analyzed. Communication networks certainly introduce delays, due to restricted bandwidth in the network[3]. The experimental results shows that the model based controller like Internal Model Controller perform better in Networked environment.

---

*Author for correspondence

2. Process Description

2.1 NCS Model for Temperature Process

In general, NCS is a closed-loop model as shown in Figure 1, where some network delays are introduced from sensor to controller and controller to plant. The delay varies according to network load and traffic. The network communication cable used is Ethernet and the protocol used to communicate is TCP/IP[4]. The scenario in the NCS is such that, the temperature process is the client operated from a server at the remote end where, the communication between the client and the server is done through TCP/IP read/write[5]. The client requests the server and the server responds based on the request.

In addition to that networked control systems reduce the design complexity and hence increase the ease in implementation. Without major revisions in their basic structure, NCS can be easily expanded. The complete architecture of NCS model is given in Figure 2. Moreover, it features competent sharing of data between controllers.

2.2 Delay in Network-Based Control System

The inevitable delays in the networks introduced because of processing and transmission, in addition to delays from a sampling period T are basically of three types, they are Sensor to Controller Delay (τ_k^{sc}) Computational delay (τ_k^{c}) and Controller to actuator delay(τ_k^{ca}) are described as follows.

2.2.1 *Sensor to Controller Delay (τ_k^{sc})*

When a sensor transmits a measured variable to a controller, it results in Sensor to Controller Delay (τ_k^{sc}). At time index k, the sensor to controller delay is computed by

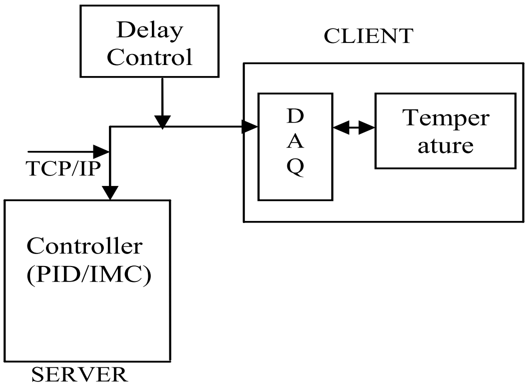


Figure 1. Block diagram of NCS Model for Temperature Process.

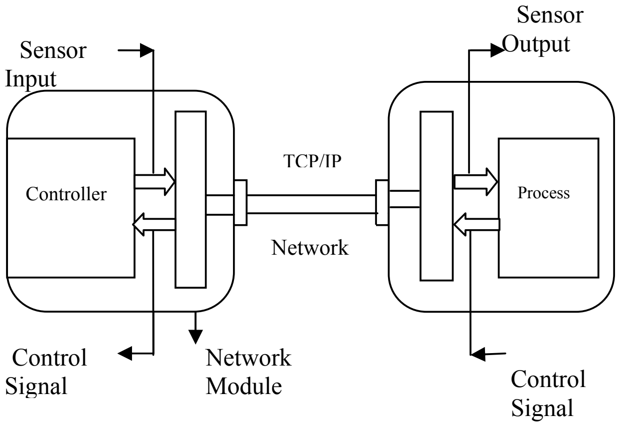


Figure 2. Network control System Architecture.

$$\tau_k^{sc} = t_k^{cs} - t_k^{ss}$$

Where t_k^{cs} and t_k^{ss} are the time instants at which the controller starts to compute the control signal and the sensor starts to measure the output of system, respectively.

2.2.2 *Computational Delay (τ_k^{c})*

The time needed for a controller to compute a control signal based on the received measurement results in Computational delay. This delay is described by

$$\tau_k^{c} = t_k^{cf} - t_k^{c}$$

Where t_k^{cf} is the time instant when the controller finishes computing a control signal

2.2.3 *Controller to Actuator Delay (τ_k^{ca})*

The controller to actuator delay occurs when a controller send its control signal to the actuator. This is defined as

$$\tau_k^{ca} = t_k^{as} - t_k^{cf}$$

where t_k^{as} is the time instant at which the actuator receives the control signal and then it starts to operate. The sum of delays τ_k^{sc}, τ_k^{c} and τ_k^{ca} is referred to as the total control delay $\tau_{k..}$The difference between the sampling time of the sensor and the sampling time of the controller is the time skew[6].

2.3 Temperature Process

The experimental set up for temperature process is shown in Figure 3. LM35 sensor has been used to measure the temperature in the process. The LM35 series are precision integrated-circuit. The LM35 has is calibrated in °Kelvin, which is a major advantage of LM35 over

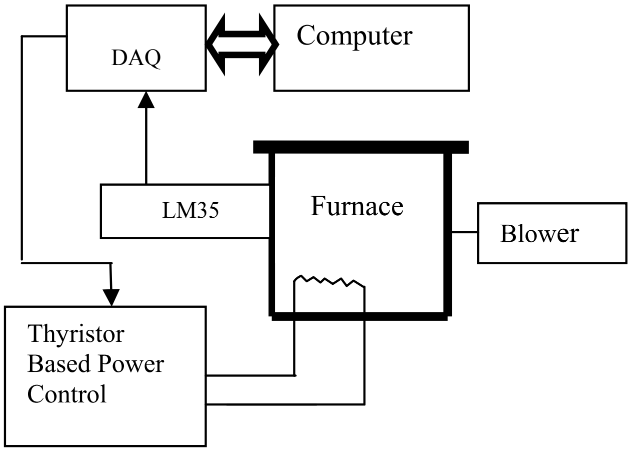


Figure 3. Block Diagram of the Temperature Process.

linear temperature sensors. The External calibration or tuning is not required in LM35. The accuracy of LM35 is ±¼°C at room temperature and ±¾°C over a full -55°C to +150°C temperature range. The desired temperature value and the temperature sensor outputs are compared and the control signal is given to final control element or the actuator, which is a SCR. The figure below describes the temperature process setup.

3. Process Modeling

Asystem model is the conceptual model that describes and represents a system. The temperature process system has been modeled for different operating regions. The model that behaves like the process lies in the region 31°C to 49°C and the transfer function for this region is G1(s) = 9.81e^20s/215s+1. The system modeling was carried out in the laboratory and the process gain, time constant and transport delay was calculated using the experimental data.

4. Controller Design

4.1 Pid Controller

The PID algorithm has operates under three basic modes, the Proportional mode, Integral mode, and the Derivative mode. It is necessary to identify which controller has to be used for the process, before applying the algorithm for the process. Then the parameters or settings for each mode used can be finalised. Three basic algorithms that are commonly used are: P, PI or PID[7]. The measured value is compared with the desired value and the difference between both, which is also called as error, is applied to the controller and the controller gives the appropriate control signal to the actuator[8,9]. Unlike the simple controllers, the PID can adjust process output based on the history and rate of change of the error signal, which gives more accurate and stable control.

$$U(\mathrm{t}) = k(e(t) + \frac{1}{\tau i}\int_0^t e(t)dt + Td.\frac{de(t)}{dt} \qquad (4.1)$$

Where,

U is the control signal

e is the control error(e = r-y).

The reference value is also called the set point. The P-I-D terms. The controller parameters are proportional gain k_p, integral gain k_i and derivative gain k_d.

K_p: **Proportional Gain** - Larger K_p typically means faster response since larger the error.

Ki: **Integral Gain** - Larger Ki means, errors are eliminated quicker.

K_d: **Derivative Gain** - Higher the, K_d overshoot is decreased.

4.2 Internal Model Controller

In the Internal Model Control shown in Figure 4, the control can be achieved only if there is a model that will replicate the characters of the process[10]. The typical internal model controller structure is given below, where Gp is the actual process, Gm refers to the process model, and Gimc is the IMC controller, r, y and y^* refer to the input, output of the actual process and the output of the model of the process respectively and d is the disturbance of the system[11].

Gimc is determined by,

$$\mathrm{Gimc} = 1/\mathrm{Gmm} \qquad (2)$$

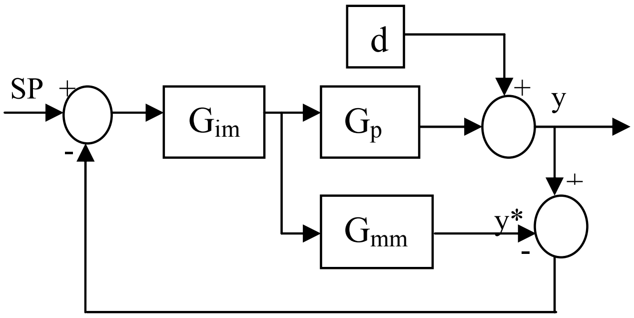


Figure 4. Block diagram of internal model controller.

where Gmm is a transfer function that has minimum phase characteristics. To design a practical IMC controller, Gimc is multiplied by a transfer function of the filter, f(s), where the filter is given by

$$f(s) = 1/(\lambda s + 1)^n \tag{3}$$

where λ is the filter parameter and n is the integer[12,13]. The filter parameter λ is chosen by trial and error method.

The practical IMC controller can be expressed as,

$$G_{imc} = \frac{f(s)}{Gmm} \tag{4}$$

5. TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) is the most primitive communication language or protocol of the Internet. This is the most commonly used communication protocol in a private network (intranet/extranet). This protocol offers a direct access to the Internet, the host computer is provided with TCP/IP program to write and the remote server is provided with TCP/IP program to read[14]. It supports client/server model of communication. In the two functional layers, the higher layer: Transmission Control Protocol manages the assembling and reassembling of packets in the transmitting and receiving ends[15]. The lower layer: Internet Protocol, concentrates on delivering the packets to the right destination by handling the address part. The gateway computer checks the address and forwards the message to the destination. In the industrial environment there will be transmission delay and access delay depending on the network load, scheduling policies, number of nodes and varied protocols[16,17]. The experiment was carried out in the presence of delay that was introduced manually using LabVIEW programming. The delay was introduced by multiplying a random number with the number of active nodes.

6. Results and Discussions

The experimental setup in the laboratory is shown in Figure 5. The results were obtained by connecting the process and the remote controller using an Ethernet cable. The communication between the client (process) and the server (controller) was established through TCP/IP. The controller was designed for two different control algorithms like conventional PI controller and model based IMC controller. The data from the temperature process was acquired using a Data Acquisition Card (DAQ).

Figure 5. Experimental setup for wired network.

The NCS model for temperature process with PI controller, implemented in LabVIEW is shown in Figure 6 and the response of PI Controller in Network is shown in Figure 7. The set point given is 45°C. The settling time is 320 seconds and the overshoot of the response is 17%.

The NCS model for temperature process with Internal Model Controller, implemented in LabVIEW is shown in Figure 8 and the response in network is shown in Figure 9. The set point given is 45°C. The settling time is 280 seconds and the overshoot of the response is 0%.

The NCS model with network induced delay for temperature process implemented in LabVIEW is shown in Figure 10. The response of the system to PI controller is shown in Figure 11. To simulate network traffic 1000 nodes was assumed to be active. Each node introduces a delay of 1msec. The set point given is 45°C. The settling time is 340 seconds and the overshoot of the response is 22.5%.

The response of the system to Internal Model Controller, with network induced delay is shown in Figure 12. The set point given is 45°C.The settling time is 290 seconds and the overshoot of the response is 0%.

The performance analysis of PI Controller and IMC in network with time delay is given in Table 1.IMC gives better performance than PI Controller when comparing the performance metrics like settling time and overshoot.

7. Conclusion and Future Work

The performance of PI controller and Internal Model Controller has been analyzed in networked environment

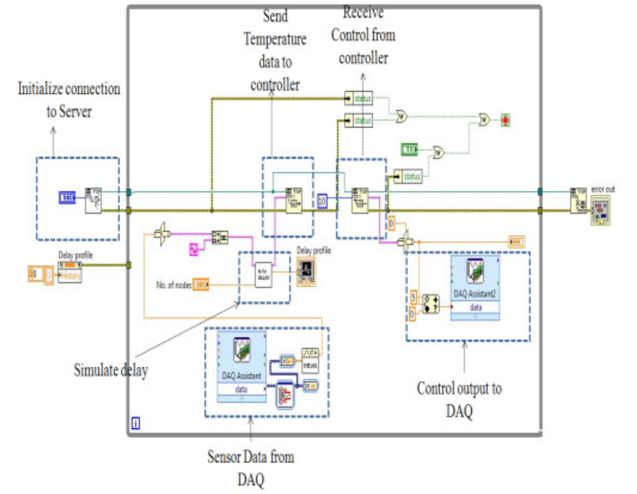


Figure 6. NCS model of temperature process with PI controller.

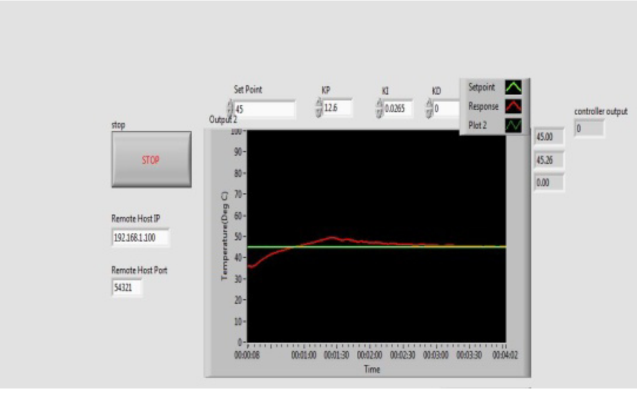

stop
STOP
Remote Host IP
192.168.1.100
Remote Host Port
54321
Set Point 45
KP 12.6
KI 0.0265
KD 0
controller output 0
45.00
45.26
0.00

Figure 7. Response of PI controller in Network.

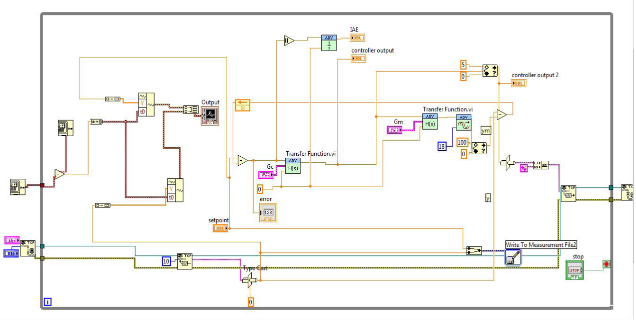


Figure 8. NCS model of temperature process with IMC controller.

with and without network delay and traffic. The model based controllers proves to perform better, tolerating the network delay and traffic. The performance is analysed based on the metrics like peak overshoot and settling time. Internal Model Controller proves to be a better controller choice in networked environment.

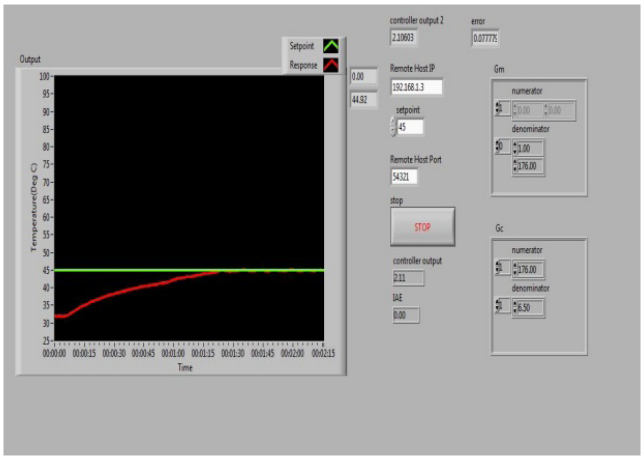

controller output 2
2.10603
error
0.07777
Remote Host IP
192.168.1.3
setpoint
45
Remote Host Port
54321
stop
STOP
controller output
2.11
IAE
0.00
0.00
44.92
Gm
numerator 0.00 0.00
denominator 1.00 176.00
Gc
numerator 176.00
denominator 6.50

Figure 9. IMC Response in Network.

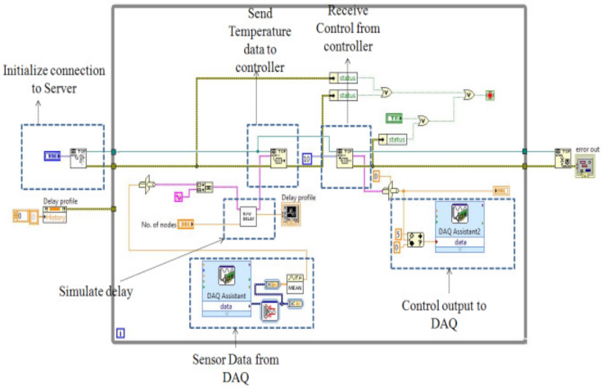


Figure 10. Network with Induced Time Delay.

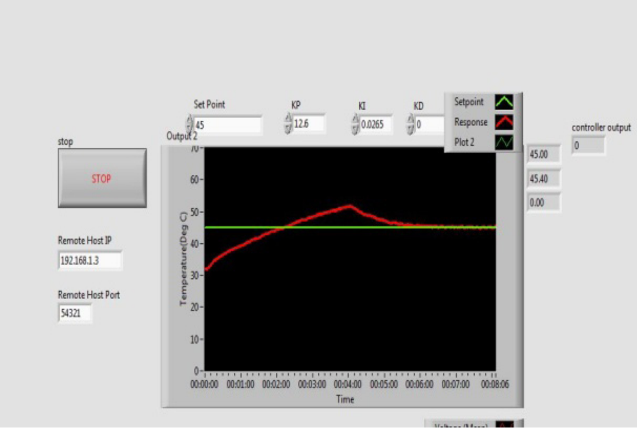
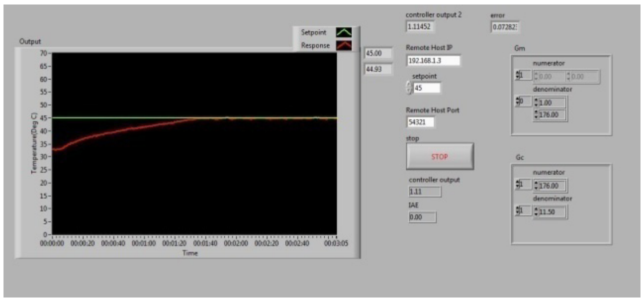

stop
STOP
Remote Host IP
192.168.1.3
Remote Host Port
54321
Set Point 45
KP 12.6
KI 0.0265
KD 0
controller output 0
45.00
45.40
0.00

Figure 11. PI Response in Network with traffic.



45.00
44.93

controller output 2: 1.11452

error: 0.07282

Remote Host IP: 192.168.1.3

setpoint: 45

Remote Host Port: 54321

stop

controller output: 1.11

IAE: 0.00

Gm numerator: 1, 0.00, 0.00
denominator: 0, 1.00, 176.00

Gc numerator: 1, 176.00
denominator: 1, 11.50

Figure 12. IMC Response in Network with Traffic.

Table 1. Performance Analysis of controllers in NCS

Parameter	PI Controller	IMC
Settling Time in (sec) Without Network Delay	320	17%
Settling Time in (sec) With network induced Delay	340	22.5%
Overshoot(%) without Network Delay	280	0%
Overshoot(%) with Network induced Delay	290	0%

8. References

1. Zhao S, Cui Q. Predictive control applied to networked control systems. Frontiers of Model Predictive Control. 2010 Jan; 67–88.
2. Bouchoul F, Mostafai M. From e-Manufacturing to m-Manufacturing. The International Arab Journal of Information Technology. 2008; 5(2):140–7
3. Feng-Li Lain, James Moyne, Dawn Tilburg. Network design considerations for distributed control system. IEEE Transactions on Control Systems Technology. 2011 Mar; 10(2):297–307.
4. TipsuwanY, Chow M-Y. Control methodologies in Network Control System. Control Engineering Practice 11. Elseiver; 2003. p. 1099–111.
5. Shahri ME, Balochian S, Balochian H, Zhang Y. Design of fractional-order PID controllers for time delay systems using differential evolution algorithm. Indian Journal of Science and Technology. 2014 Sep; 7(9):1307–15.
6. Hussain AR. Time delay analysis in networked control system based controller area network. International Conference on Advances in Computer and Information Technology (ACIT 2012); 2012. p. 62–5
7. Coughanowr DR. Process system analysis and control. 2nd ed. McGraw-Hill International; 1991.
8. Aslam F, Kaur G. Comparative analysis of conventional, P, PI, PID and Fuzzy Logic controllers for the Efficient Control of Concentration in CSTR. Int J Comput Appl. 2011; 17(6):12–6.
9. Haugen F. Ziegler-Nichols' closed-loop method. TechTeach; 2010 July 17.
10. Alzohairy TAA. Neural internal model control for tracking non-linear discrete time systems under disturbances. Int J Comput Appl Tech. 2012; 40(6): 19–26.
11. Astrom KJ, Wittenmark B. Adaptive control. 2nd ed. Addison-Wesley; 1995.
12. Rivera DE, Morari M. Skogestad S. Internal model control for PID controller design. 1986.
13. Rajalakshmi.K, Magaiyarkarasi.V. Design of IMC for heat Exchanger. IEEE Conference on Information Communication and Embedded System; 2013. p. 899–903.
14. Pamela D, Godwin Premi MS, Suganshia E. Performance analysis of control strategies for temperature process control in client server model. Int J Appl Eng Res. Forthcoming. 2015; 10(1):81–6.
15. Huang D, He B-G. Dynamic characteristics of mesh based network control sytem under optimal resource circumstances. J Electr Eng. 2012; 63(5):322–7.
16. Wen P, Cao J. Design of high performance networked real time control systems. IEEE Transactions on Control Theory Application. 2007 Mar. p. 1329–35.
17. Dinesh C, Manikanta VV, Rohini HS, Prabhu KR. Real time level control of conical tank and comparison of fuzzy and classical PID controller. Indian Journal of Science and Technology. 2015 Jan; 8(S2):40–4.