A Filtered Predictive PI Controller for WirelessHART Networked Control System ⋆

Sabo Miya Hassan, Rosdiazli Ibrahim, Nordin Saad, Vijanth Sagayan Asirvadam and Tran Duc Chung

Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS, 32610, Seri Iskandar, Perak, Malaysia (e-mail: hsmiya2010@gmail.com, rosdiazli@utp.edu.my)

Abstract: In recent years, increasing attention has been paid towards applying wireless technology for control. This is due to its advantages of flexibility, scalability, use of fewer cables and overall reduced operational cost compared to its wired counterpart. However, the technology is often affected by stochastic delay and high frequency noise. PIDs are ill-equipped to deal with these problems while model based controllers such as dead-time compensators (DTCs) like Smith predictor and internal model controllers (IMCs) are complex and require exact plant model for implementation. Thus, predictive PI (PPI) controller being a settlement between the PIDs and the model based controllers is a good candidate. The PPI retains simplicity of the PID, it has the ability to predict long time delay and can be used even with model mismatch. However, the PPI is severely affected by high frequency noise. Therefore, this paper proposes a Filtered PPI controller that can be used even in the presence of high frequency noise. Simulation and experimental results proved the viability of the proposed method.

Keywords: WirelessHART control, filter, PPI controller, variable network delay, disturbance

1. INTRODUCTION

It is evident that the coming on board of the two industrial wireless standards: WirelessHART and ISA100 Wireless (Chen et al., 2010b; Fadel et al., 2015) has triggered researchers both in the industrial and academic sectors to look into the possibility of applying them to not only monitoring, but also for control (De Biasi et al., 2008; Ferrari et al., 2013; Santos et al., 2015; Hassan et al., 2016b; Chung et al., 2016a; Blevins et al., 2015, 2016). While the WirelessHART is based on the traditional Highway Addressable Remote Transducer (HART) protocol developed by HART Communication Foundation (HCF), the ISA100 Wireless is developed by the International Society of Automation (ISA). The two standards have both been ratified by the International Electrotechnical Commission (IEC) to specifically take care of the industrial need for wireless monitoring and control applications. Key features of the standards is that they are based on the IEEE 802.15.4 physical layer and both support mesh topology network which makes them highly reliable. Furthermore, they both transmit on the 2.4GHz Industrial Scientific and Medical (ISM) radio frequency band. In terms of security, the two standards ensure protection through payload encryption and message authentication. This is achieved through the use of Advanced Industry Standard (AES)-128 cipher keys (Petersen and Carlsen, 2011). It should be noted that there is already close to forty million HART compliant devices already installed globally (Olenewa, 2013; Hassan et al., 2017), thus the standard has the potential of being the leader in wireless monitoring and control applications. The difference between the two industrial standards and those before them is that, previous standards such as Bluetooth, Wi-Fi and ZigBee did not satisfy the industrial requirements of reliability, scalability and device interoperability (Chen et al., 2010a,b).

The merits of wireless technology in the industry includes significant reduction in cabling and installation, maintenance, reconfiguration, commissioning and decommissioning times. Improved reliability and flexibility are some the obtainable benefits if mesh wireless standard is used in the industry. Another advantage of wireless is that it can be deployed in remote and or hazardous environments. Unlike its wired counterpart, the technology could easily be deployed in mobile and rotating platforms (Ikram and Thornhill, 2010; Winter et al., 2016). However, despite these advantages offered by this promising technology, its usage in the industry is not devoid of challenges (Ikram and Thornhill, 2010). For example, using wireless transmitters in the wireless network introduces stochastic delay into the control loop. Other challenges are noise due to interference in the communication path and uncertainties such as packet loss due to packet drop-out (Blevins et al., 2014, 2016; Chung et al., 2016b). Other factors that could potentially degrade network control performance are, approximation of higher order systems with lower ones and process dead-time. The latter may occur due to several factors such as transportation time of material from sensor to

⋆ This work was supported by Universiti Teknologi PETRONAS under grant No. Y-UTP 0153AA-A74

actuator and verse versa, communication and computation delays, and time lag accumulation of a serially connected dynamic systems (Wang et al., 2014).

It is difficult to control processes characterised by long dead-time and variable network delay with standard feedback controllers such as PID. Although at the lowest control level the PIDs are the most employed (Larsson and Hägglund, 2011, 2012), their usage extends to even higher level control. The patronage enjoyed by PIDs is mainly due to the key features of simplicity and ease of tuning manually. The derivative term of the PID is usually turned off due to its sensitivity to noise. Thus, the PI controller is the most commonly used among the PID variants (Huba, 2013). Consequently, when PIDs are used in environments such as WirelessHART, the result is oscillation and instability because the PIDs are limited in gain (Tan et al., 2010). Although adding a derivative action to PI controller to achieve PID leads to phase advance hence predictive capability (see Larsson and Hägglund (2012)), this does not work well when long dead-time and high frequency noise are involved due to the earlier stated reason of noise sensitivity. Furthermore, the predictive mechanism of the PID is not suitable when considering non minimum phase.

To improve the performance of closed loop systems with long dead-time, the use of DTCs and MPCs such as Smith predictors and IMCs have been proposed by researchers (Normey-Rico and Camacho, 2008; Jha et al., 2014). The drawbacks of DTCs and MPCs is that they require precise model of the system, hence complex to develop and implement (Hassan et al., 2016b; Shinskey, 2001; Tan et al., 2010).

In order to avoid the complexity of the DTCs, MPCs and the poor performance of the PIDs, a special DTC; the PPI controller was proposed in 1992 by Hägglund (Larsson, 2011). For applications involving long process dead-time and/or stochastic delays, the PPI can be a settlement between the simple but poor performance PID and the complex MPC (Larsson and Hägglund, 2012). This controller is adopted for wireless networked control systems (WNCS) characterized by stochastic network delay and long process dead-time in our previous work (Hassan et al., 2016a). The advantages of the PPI is that in the design, model mismatch is accommodated, it can deal with integrating processes and has fewer tunable parameters compared to other model based compensators (Larsson and Hägglund, 2012). Furthermore, while both the Smith predictor and IMC controllers need systematic experimentation to identify process parameters for the design of controller, the PPI controller does not require this identification process. Hence, the parameters of the controller can be manually tuned. In comparison, with the PID controller, prediction in PPI controller is possible even with long dead-time without amplifying high frequency noise. Thus, the PPI is expected to give faster responses than the PID if the dead-time is long.

To demonstrate the similarities of the PID with the PPI in terms of simplicity, consider for example controlling a first order plus dead time (FOPDT) system commonly used to represent practical systems, five and four parameters are required to be tuned if Smith predictor and IMC are used respectively. On the other hand, three parameters are

Table 1. Comparison of tunable parameters

Control Type	Parameters					
	Model			Controller		
PID	-	-	-	K_c	T_i	K_d
PPI	-	-	L_p	K_c	T_i	-
Smith Predictor	K	T	L_p	K_c	T_i	-
IMC	K	T	L_p	-	T_{cl}	-

required to be tuned for both PID and PPI controllers as shown in Table 1. As seen from the table, while both IMC and Smith predictor requires full knowledge of the model parameters, the PPI only requires the estimate of the delay in the system. The other two parameters of the PPI can be tuned the same way as tuning PI controller (Johansson, 2001).

High frequency measurement noise can degenerate the performance of control system by generating control activity that may lead to wearing of the actuator (Larsson and Hägglund, 2011; Segovia et al., 2014). Other effects of the noise apart from the wearing of the actuator are heat dissipation, acoustic sound, increase production cost and a reduction in overall control precision (Huba and Belai, 2014). Thus, DTCs, MPCs and PIDs can be designed to have good load regulation and robustness. However, when this high frequency measurement noise is involved, the load regulation capability need to be backed by additional filtering. This is to curtail straining of the actuator by large signals due to undesired control activity caused by the noise. The typical PPI controller adopted for wireless environment is not immune from this effect. Thus, this work proposes an improvement to the PPI structure by incorporating a filter into the design. The filter if appropriately chosen will improve a closed loop control performance. Thus, this work proposes an improvement to the PPI structure by incorporating a filter into the design.The filter if appropriately chosen will improve a closed loop control performance (Huba, 2015). This implies additional parameter to be tuned (filter time constant). Despite the additional parameter, the FPPI can still retains its comparative advantages over both PID and MPCs.

The rest of the paper is structured as follows: related work is given in Section 2 while Section 3 gives the detailed explanation on the FPPI design. Section 4 gives the procedure for network delay estimation and plant model selection and WirelessHART hardware in the loop simulation (WH-HILS). Section 5 presents result discussion while the last section draws conclusion.

2. RELATED WORK

Several works have been reported towards extending the application of WirelessHART technology for control. Majority of this works are geared towards solving the problems of delay and packet drop-out in the control network. The use of exponentially weighted moving average filter to take care of packet drop-out and wired link delay has been reported in Chung et al. (2016b). Although the filter has the potential of being used alongside DTC technique, it has not yet been used to compensate for long dead-time delayed networks. The use of setpoint weighting technique has been proposed for wireless environment in Hassan et al. (2016b). This technique has the potential of improving significantly the control performance even with long

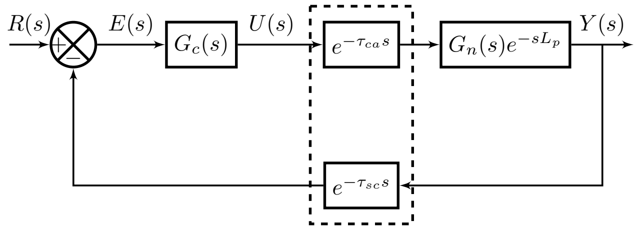


Fig. 1. Network delay representation in a single loop WNCS

dead-time. However, implementing the setpoint weighting function could be costly since model of the system is desirable although not compulsory. PID-plus algorithm was proposed by Blevins et al. (2014) to be used in a wireless environment. The drawback of this technique is that it is still limited by the gain limitation of the traditional PID if used for process with long dead-time.

Several works regarding delay compensation techniques such as IMCs, DTCs and PPI controller have been reported in the literature. A comprehensive survey of several DTCs is can be found in Normey-Rico and Camacho (2008). A new predictive PI controoller with additional filtering was proposed in Ribić and Mataušek (2012). Here, the prediction is achieved through the use of disturbance observer (DO). Comparison between two degree of freedom (2DOF) PI controller and a filtered PI based on predictive disturbance observer (PDO) approach was presented in Huba (2013). In that work, it was shown that enhanced loop performance can be achieved with the PDO based PPI. Arousi et al. (2008) proposed adding a prediction algorithm to the traditional PID algorithm. Here, a noise filter is considered as part of the design parameter. However, only one simulation example is considered with a short dead-time of less than 1s. In Larsson and Hägglund (2012), a comprehensive comparison between PPI controller and robust PID is presented. The authors highlighted the potential of using measurement filter for PPI controller. In De Biasi et al. (2008), the PPI controller was used alongside typical PID to simulate for control of WirelessHART network subject to clock drift. However, this work is based purely on simulation hence, there was no reference to neither real-time delay nor noise.

3. FILTERED PPI CONTROLLER DESIGN FOR WirelessHART NETWORK

3.1 Typical Wireless Networked Control Structure

A single loop WNCS characterized by network delay can be represented as shown in Fig. 1. From the figure, total network delay (τ_N) is given as

$$\tau_N = \tau_{ca} + \tau_{sc} \tag{1}$$

where τ_{ca} is the controller-to-actuator delay and τ_{sc} is the sensor-to-controller delay. Assuming commutativity between the terms in the loop, the process dead-time (L_p) can be added to the network delay to obtain the total loop delay as

$$L = \tau_N + L_p \tag{2}$$

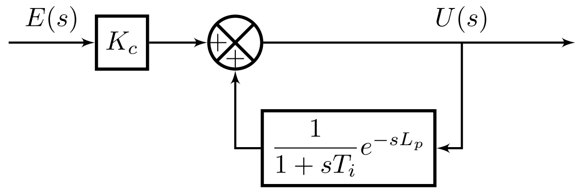


Fig. 2. Implementation of a typical PPI controller

3.2 Typical PPI Structure

Assuming the plant's transfer function in Fig. 1 is a FOPDT given as

$$G_p(s) = \frac{K}{1+Ts}e^{-sL_p} \tag{3}$$

where $G_n(s) = K/(1+Ts)$. The plant parameters K, L_p and T are the process gain, dead-time and time constant respectively.

The transfer function of the closed loop sans the wireless network is given as

$$G_o(s) = \frac{G_p(s)G_c(s)}{1+G_p(s)G_c(s)} \tag{4}$$

Thus from (4), controller $G_c(s)$ is obtained as

$$G_c(s) = \frac{1}{G_p(s)}\frac{G_o(s)}{1-G_o(s)} \tag{5}$$

Define the desired closed loop transfer function in (4) as

$$G_o(s) = \frac{1}{1+sT}e^{-sL_p} \tag{6}$$

Using (3) and (6) in (5), the controller is expressed as

$$G_c(s) = \frac{1+sT}{K(1+sT-e^{-sL_p})} \tag{7}$$

Expressing $G_c(s)$ of (7) in terms its input-output relationship we have

$$(1+sT-e^{-sL_p})U(s) = \frac{1}{K}(1+sT)E(s) \tag{8}$$

where $E(s)$ is the error and $U(s)$ is the control signal. Thus, (8) can be expressed as follows:

$$U(s) = \frac{1}{K}\left(1+\frac{1}{sT}\right)E(s) - \frac{1}{sT}\left(1-e^{-sL_p}\right)U(s) \tag{9}$$

Observing (9), it can be seen that a PI controller acts upon$E(s)$ and the prediction is accomplished through low-pass filtering $U(s)$. Hence, the PPI control action. It should be noted that, if the delay term $L_p = 0$, the controller is a simple PI with gain $K_c = 1/K$ and time constant $T_i = T$. On the other hand when $L_p > 0$, (9) can be written as

$$U(s) = K_cE(s) + \frac{1}{1+T_is}e^{-sL_p}U(s) \tag{10}$$

The block diagram implementation of (10) is shown in Fig. 2. Furthermore, (10) can be written in the form of transfer function and be factored in to cascade of predictor and PI controllers i.e. $G_c(s) = C_{PI}(s)C_{pred}(s)$ as follows:

$$G_c(s) = K_c\left(1+\frac{1}{T_is}\right)\left(\frac{1}{1+\frac{1}{T_is}(1-e^{-sL_p})}\right) \tag{11}$$

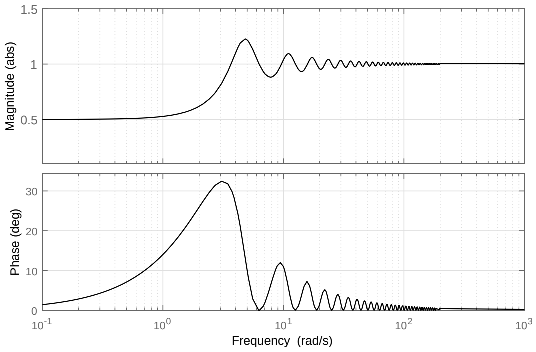


Fig. 3. Predictor frequency response

Thus, the right hand side of (11) can be fragmented into two components and be written in form of the following equations:

$$C_{PI}(s) = K_c\Big(1 + \frac{1}{T_i s}\Big) \tag{12}$$

and

$$C_{pred}(s) = \frac{1}{1 + \frac{1}{T_i s}(1 - e^{-sL_p})} \tag{13}$$

where (12) is the PI controller and (13) is the predictor.

The relationship between plant and controller parameters is not always $K = 1/K_c$ and $T = T_i$, they are actually related as $K = \alpha/K_c$ and $T = \beta T_i$, where α and β are tunable parameters. For the first order system as discussed earlier, these constants are chosen to be unity each.

3.3 Prediction in PPI

Consider the predictor given in (13), its behaviour is governed by the ratio L_p/T_i. Additionally, the predictor contains only left hand plane poles for all values of T_i. Therefore, Taylor series expansion of equation (13) for small value of s gives

$$C_{pred}(s) \approx \frac{1}{1 + \frac{L_p}{T_i}}\left(1 + \frac{1}{2}\frac{\left(\frac{L_p}{T_i}\right)^2}{\frac{L_p}{T_i}} T_i s + ...\right) \tag{14}$$

Thus, the static gain of the predictor is given as

$$C_{pred}(0) = \frac{1}{1 + \frac{L_p}{T_i}} \tag{15}$$

Furthermore, the PPI controller is only equal to PI-Smith predictor controller for certain parameter values where model matching is necessary. Thus, the performance of the PPI can be improved even without considering model matching. The frequency plot of the predictor (see Fig. 3) shows that for high frequencies, the gain of the predictor approaches unity while the phase advance falls rapidly to zero.

3.4 Wireless Network PPI Structure

Consider the process in Fig. 1 now with the wireless network characterised by both network delay τ_N and process dead-time L_p, the total loop delay is given in (2). If the controller $G_c(s)$ is the PPI controller of Fig. 2, the delay L_p is substituted with the total loop delay L, the

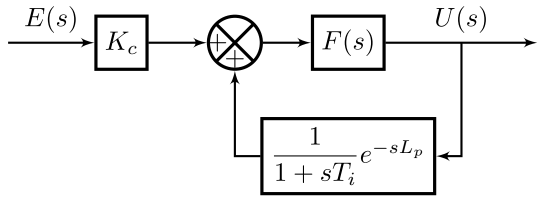


Fig. 4. Implementation of FPPI controller

PPI controller for the wireless systems can be expressed as (16).

$$U(s) = K_c E(s) + \frac{1}{1 + T_i s} e^{-sL} U(s) \tag{16}$$

The controller $G_c(s)$ will now be written in terms of the loop delay L as

$$G_c(s) = K_c\Big(1 + \frac{1}{T_i s}\Big)\left(\frac{1}{1 + \frac{1}{T_i s}(1 - e^{-sL})}\right) \tag{17}$$

3.5 Proposed Filtered PPI (FPPI) Controller Structure

Consider the FPPI structure shown in Fig. 4, the transfer function is given as

$$G_{cf}(s) = \frac{U(s)}{E(s)} = \frac{K_c F(s)}{1 - \frac{1}{1 + sT_i} e^{-sL} F(s)} \tag{18}$$

where $F(s)$ is a filter transfer function. Thus, (18) can be expressed as

$$U(s) = \Big(K_c E(s) + \frac{1}{1 + sT_i} e^{-sL} U(s)\Big) F(s) \tag{19}$$

From (19), it can be seen that the PPI control action is passed through a filter $F(s)$ to achieve the control action of the FPPI controller. This implies that both the error signal and prediction term are filtered to achieve better performance. The FPPI as implemented in the wireless network is shown in Fig. 5. The filter structure will be discussed in the following section.

3.6 Filter Structure

Consider the predictor of the PPI controller given in (13), its gain approaches unity at high frequencies (Larsson and Hägglund, 2012; Åström and Hägglund, 2006). This clearly indicates that the first order measurement filter should be used for the PPI controller. This point has also been corroborated by Larsson and Hägglund (2009). Consequently, the following filter structure is used:

$$F(s) = \frac{1}{1 + sT_f},\ T_f > 0 \tag{20}$$

where the filter time constant $T_f = \epsilon L$ and $\epsilon > 0$.

The digital implementation of the filter for $k \in \{1, 2, 3, ...\}$ can be achieved by using the following recursive relationships:

for $k = 1$,

$$y(1) = u(1) \tag{21}$$

for $k > 1$,

$$y(k) = (1 - \gamma)y(k - 1) + \gamma u(k) \tag{22}$$

where $\gamma = \frac{h}{T_f + h}$ is the filter constant, h is the sampling period and should be chosen to be $h \le \frac{T_f}{5}$.

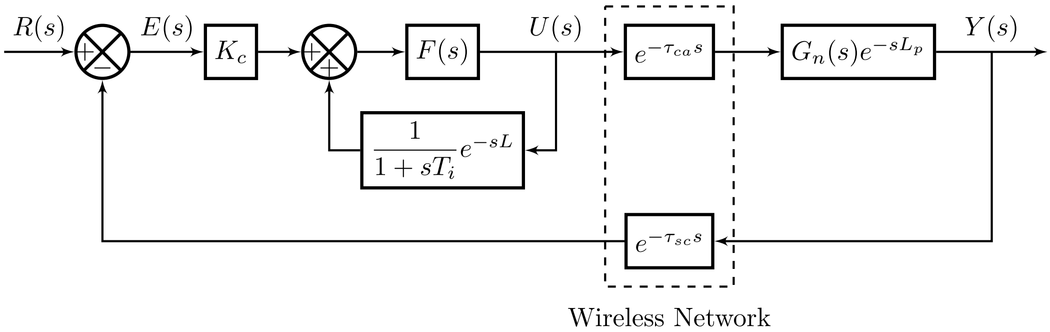


Fig. 5. Wireless networked FPPI control structure

The filter in (20) is also referred to as exponentially weighted moving average filter. In this kind of filter, the weighting factor γ decreases exponentially as the time progresses by allocating higher weights at the beginning.

3.7 Robustness and Stability Analysis

For the Robustness analysis of the proposed approach, the extended complementary sensitivity function and extended sensitivity function methods proposed by Larsson and Hägglund (2009) will be considered. In this method, which was also adopted by same authors in Larsson and Hägglund (2012), the model uncertainties were divided into dead-time and non dead-time based. The robustness computation is established on the open loop transfer function.

If (19) is the control action of FPPI i.e., $G_{cf}(s)$, the total delay in Fig. 5 is given as (2). Under nominal conditions, the entire process model inclusive of network delay can be expressed as

$$G(s) = G_n(s)e^{-sL} \tag{23}$$

where, $G_n(s)$ is the delay free process.

Consider some deviation from nominal conditions where there is variation in both process, dead-time and network induced delays. Assume that the delay error is $\Delta L \in [\Delta L_{min}, \Delta L_{max}]$. Assume also that the multiplicative uncertainty of the process $G(s)$ is $\Delta G(s)$, the entire process together with uncertainties if assumed to be norm bounded can be written as

$$\hat{G}(s) = G_n(s)\left(1 + \frac{\Delta G(s)}{G_n(s)}\right)e^{-s(L+\Delta L)} \tag{24}$$

If the inverse multiplicative uncertainty is considered, the process model can as well be written as

$$\hat{G}(s) = G_n(s)\left(1 + \frac{\Delta G(s)}{G_n(s)}\right)^{-1} e^{-s(L+\Delta L)} \tag{25}$$

With the controller $G_{cf}(s)$, the nominal open loop in the frequency domain $G_{cf}(i\omega)\hat{G}(i\omega)$ is thus assumed to be stable and norm bounded. Therefore, robust stability condition based on inverse multiplicative uncertainty can be given as

$$\left|1 + G_{cf}(i\omega)G_n(i\omega)\left(1 + \frac{\Delta G(i\omega)}{G_n(i\omega)}\right)^{-1} e^{-i\omega(L+\Delta L)}\right| > 0$$

the above inequality can be written as

$$\left|1 + \frac{\Delta G(i\omega)}{G_n(i\omega)} + G_{cf}(i\omega)G_n(i\omega)e^{-i\omega(L+\Delta L)}\right| > 0, \quad \forall \Delta L \in [\Delta L_{min}, \Delta L_{max}], \Delta G(i\omega), \omega \tag{26}$$

Equation (26) can be expressed in the following form since $\Delta G(i\omega)$ can take any direction in the complex plane:

$$\left|1 + G_{cf}(i\omega)G_n(i\omega)e^{-i\omega(L+\Delta L)}\right| - \left|\frac{\Delta G(i\omega)}{G_n(i\omega)}\right| > 0 \tag{27}$$

The extended sensitivity function is thus defined as the inverse of the first term in (27) in the s domain as follows

$$S(s, \Delta L) = \frac{1}{1 + G_{cf}(s)G(s)e^{-s\Delta L}} \tag{28}$$

Thus, robust stability condition derived from both (27) and (28) is

$$\left\|\frac{\Delta G(s)}{G_n(s)}S(s, \Delta L)\right\|_\infty < 1, \; \Delta L \in [\Delta L_{min}, \Delta L_{max}] \tag{29}$$

In the same way, if multiplicative uncertainty is considered, the extended sensitivity function can be written as

$$\left|\frac{1 + G_{cf}(i\omega)G(i\omega)e^{-i\omega(\Delta L)}}{G_{cf}(i\omega)G(i\omega)e^{-i\omega(\Delta L)}}\right| > \left|\frac{\Delta G(i\omega)}{G_n(i\omega)}\right| \tag{30}$$

Thus, defining the extended complementary sensitivity function $T(s, \Delta L)$ as the inverse of left hand side of (30) in s domain we have

$$T(s, \Delta L) = \frac{G_{cf}(s)G(s)e^{-s\Delta L}}{1 + G_{cf}(s)G(s)e^{-s\Delta L}} \tag{31}$$

Therefore, the condition for robust stability can be given as

$$\left\|\frac{\Delta G(s)}{G_n(s)}T(s, \Delta L)\right\|_\infty < 1, \; \Delta L \in [\Delta L_{min}, \Delta L_{max}] \tag{32}$$

4. NETWORK DELAY ESTIMATION AND MODEL SELECTION

4.1 Delay Estimation

To measure the induced by the wireless network, an experiment using the Linear Technology Smart Mesh WirelessHART network development kit was conducted. In the experiment, round trip network delays for communication between a wireless mote and a wireless gateway were measured in the same way to the methods reported by Huang et al. (2014) and Santos et al. (2015). These delays are retrieved by the gateway using the command

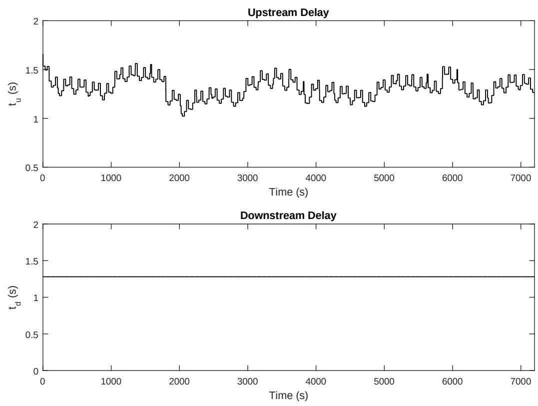


Fig. 6. Upstream and downstream network delay profile

Table 2. Network delay statistics

Delay Type	Statistics (s)			
	Max	Min	Avg.	Std. dev.
Upstream (t_u)	1.6600	1.0220	1.3094	0.1040
Downstream (t_d)	1.2800	1.2800	1.2800	0.0000

getLatency MACaddress where *MACaddress* is the media access control (MAC) address of the motes connected to the gateway (Linear Technology, 2015). Each delay is measured using timestamps on communication messages. This is based on the difference between the received timestamps generated at the gateway and the sent timestamps embedded on the arrival message. Even though each network device has its own internal clock and it experiences drift during operation, network-wide time synchronization prescribed by the WirelessHART standard ensures all devices have the same time reference as the gateway with an accuracy of less than 1ms. This makes the delay measurement from gateway possible. The experimental network delays as recorded from the gateway are shown in Fig. 6. The average values of the upstream delay (τ_{sc}) is 1.3094s and that of the downstream delay (τ_{ca}) is 1.2800s as shown in the statistical information Table 2. The variation of upstream delay over time is due to the communication between the mote and the gateway. Furthermore, being the host in the network, the gateway is always in active state, hence constant downstream delay. On the other hand, the motes will enter idle state after completing communication cycle with other motes or the gateway. This is to reduce power consumption and to prolong motes' battery life. In the idle state, the motes will have lower processing capability, thus resulting in longer signal processing time. This contributes to the upstream delays variation. The packet transmission refresh rate used is 4s to avoid battery drain of the motes.

4.2 Simulation with WH-HILS

To validate the controller design, the use of WH-HILS, a process in the loop simulator developed in our laboratory is employed. The scheme just like many other Hardware-in-the loop simulators (HILS), allows for diagnostics as well as new control strategies to be tested before being deployed in the actual plant. this will save cost and ensure non interference of the plants' operation. Fig. 7 shows the block diagram of the WH-HILS. This approach has been existing for decades. For some recent applications of this technique see Ogan (2015) and Sheng and Sun (2016). As compared to the traditional pure simulation, it has the advantage of using real hardware for simulation. By using real devices, inaccuracies in their models can be minimized, thus simulation results will be more realistic. In this work, the WH-HILS is used for simulation of WirelessHART network control system (WHNCS). The simulator consists of a computer, a gateway and several wireless nodes as shown in Fig. 7. As seen from the figure, the gateway is connected to the computer running MATLAB software using LAN interface. The software is used to simulate virtual process plants given real-time network induced delays from the gateway. For interfacing the MATLAB with gateway, Python program is used. The built-in MATLAB-Python libraries allows it to call and execute Python functions.

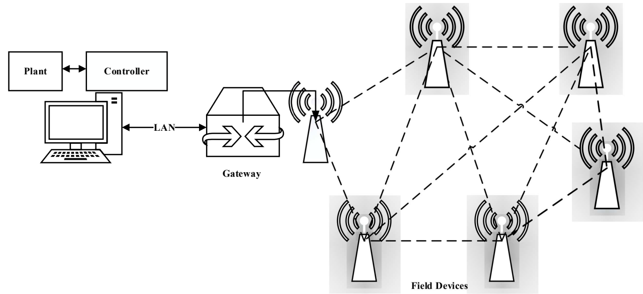


Fig. 7. WirelessHART HILS set-up

For the simulation, the motes are configured through serial interface with a computer. Once this is done, upon powered up, each mote searches and connects to the network automatically through the gateway. In Simulink environment, once the simulation is started, the real-time delays information are obtained from the WirelessHART gateway (see Section 4.1) as they occur and are directly fed into the variable time delay blocks. This is to simulate for the upstream and downstream delays. The real-time sync block is used to synchronise the simulation of the model to the real-time clock. Here, it should be noted that the mesh network formed by the gateway and its motes are similar to the industrial one as both gateway and motes are WirelessHART certified. The experimental set up for this process is shown in Fig. 8. From the set-up, each of the five motes (nodes) is placed on a pilot process plant.

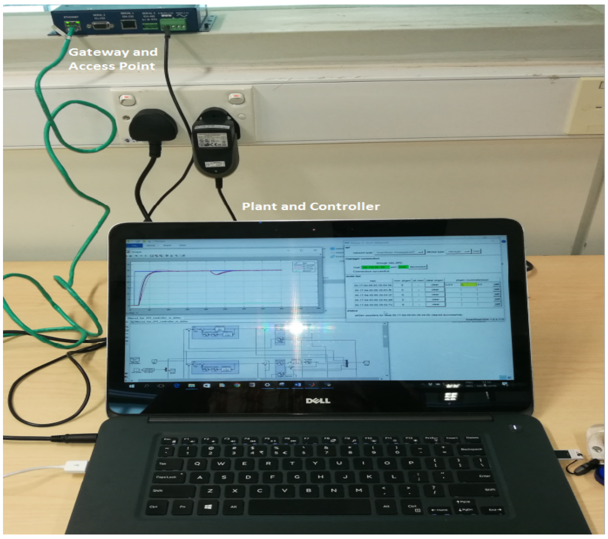
4.3 Model Selection and Controller Parameters

In this work, process models that represent the behaviour and dynamics of practical plants in the industry are used. For example, the first order model used is a thermal chamber practical model reported in Tan et al. (2010). The remaining three models are those representing second, third and fourth order processes respectively. The models are presented in the following equations:

$$G_1(s) = \frac{8}{1 + 9.13s} e^{-10s} \tag{33}$$

$$G_2(s) = \frac{1}{(1 + s)^2} e^{-5s} \tag{34}$$

$$G_3(s) = \frac{1}{(1 + s)^3} e^{-5s} \tag{35}$$



(a)

(b)

Fig. 8. (a) Gateway/access point and plant/controller Simulated in MATLAB ; (b) Location of the network motes (1, 2, 3, 4 and 5) and Access Point (AP) in the laboratory environment

$$G_4(s) = \frac{1}{(1+s)(1+0.5s)(1+0.25s)(1+0.125s)} e^{-5s} \tag{36}$$

For the second and higher order models, the controller design is based on the reduced first order model approximation of the respective plants. This is because the PPI controller design can be done without necessarily considering model matching. The controller parameters for all the models are given in Table 3.

5. RESULTS AND DISCUSSION

In this section, results with the proposed approach will be compared with those of the unfiltered PPI and optimised PI controllers. The effect of change in filter time constant T_f for the first order system will also be considered.

Table 3. Parameters of the controllers

Parameter	$G_1(s)$	$G_2(s)$	$G_3(s)$	$G_4(s)$
K_c	0.125	0.8	0.73	0.95
T_i	9.31	1.3	2	1.5
L_p	10	5	5	5
T_f	3	2	3	4
K_p	0.037	0.271	0.291	0.280
K_i	0.0047	0.0777	0.0721	0.0793

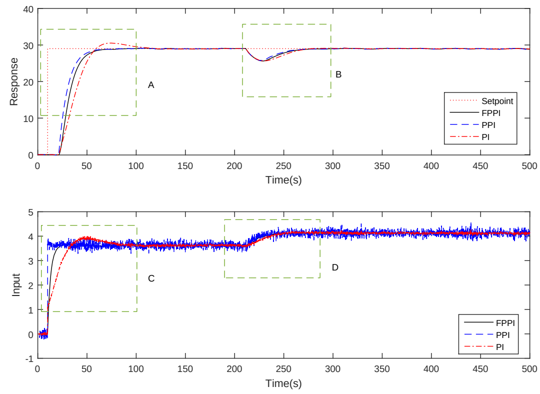


Fig. 9. First order plant response to various controllers

Table 4. First order plant performance

Parameter	FPPI	PPI	PI
Rise Time (s)	20.9327	19.1734	26.7045
Settling Time (s)	256.4493	252.5637	262.6026
Overshoot (%)	0.9469	1.1105	5.5411
Undershoot (%)	0.1492	0.3027	0.1024
IAE	8353.9	7401.7	9996.2

Furthermore, parametric modelling error for both the first and second order models will be considered.

5.1 First Order Process

Parameters of the controllers used for comparison are shown in Table 3. The performance of the plant with three controllers is shown in Fig. 9 and numerical results displayed in Table 4. The plant, being thermal chamber model, is simulated to a step signal of magnitude $29°C$ and a unit step disturbance with a 0.5 magnitude is applied at input of the plant at 200s. At the output, a white noise signal of power 0.1 is injected to simulate for possible noise scenario. Observing the response signal of Fig. 9 and Table 4, it can be clearly observed that the proposed approach is better than both the PPI and PI controllers in terms of both overshoot and undershoot. Although a bit slower than the PPI controller in terms of both rise and settling times, the approach still outperformed those of the PI controller. When observing the input (i.e. control) signal, that of the proposed approach has smoother signal compared to both the PPI and PI. This will ensure less stress on the actuator. The proposed approach recovered quicker than the PI controller. Again its a bit slower than the PPI due to the filter effect. The same pattern of result is also observed for the integral absolute error (IAE).

To observe the effect of the filter time constant, T_f on control performance of the first order plant, the simulation was done using values of T_f = 1, 2, 3, 4, 5 and 6s. From

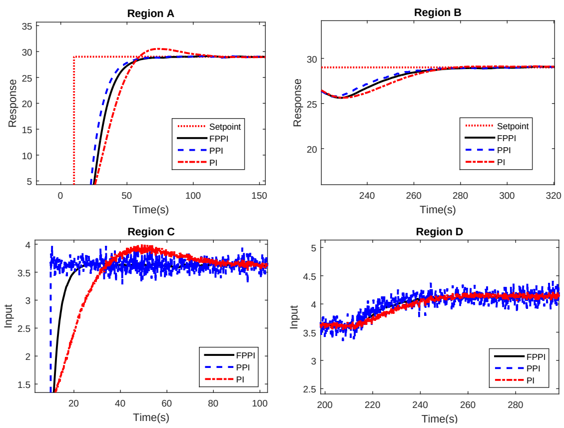


Fig. 10. Zoomed view of Fig. 9 for regions A, B, C and D

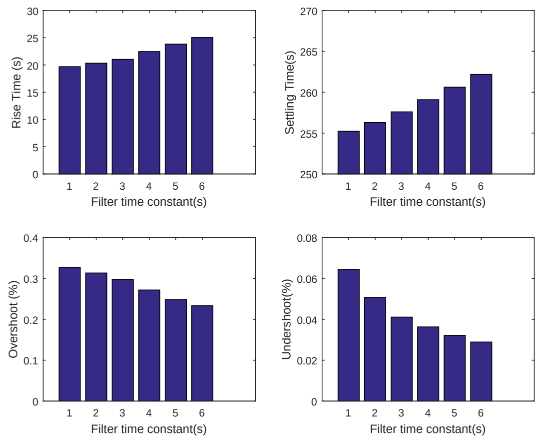


Fig. 11. Effect of the filter time constant on the rise time, settling time, overshoot and undershoot of first order plant

Fig. 11, it can be seen that increasing T_f slows down the response while improving the performance by lowering both overshoot and undershoot.

To analyse the sensitivity of the proposed approach to model mismatch, parametric modelling error due to mismatch in the model parameter (in this case model gain K) is considered. The plant is simulated to both 10% increase and decrease in the model gain. The result is compared to that of PI controller and presented in Figs. 12, 13 and Table 5. From Fig. 12, it can be seen that despite the perturbation in model gain, the proposed controller still outperformed the PI in-terms of rise time, overshoot, undershoot, and IAE (see Table 5 and Fig. 13). However, the proposed approach settled slower at 26.98s as compared to 23.82s of the PI controller. The proposed approach also recovered from the effect of disturbance faster than the PI. This can be seen by observing the zoomed view of Region B in Fig. 13. The smoothness of the control signal with the proposed approach compared to the PI can also be seen in the zoomed view of Region D.

5.2 Second Order Process

In a similar fashion to the first order system, parameters of the various controllers compared are shown in Table 3.

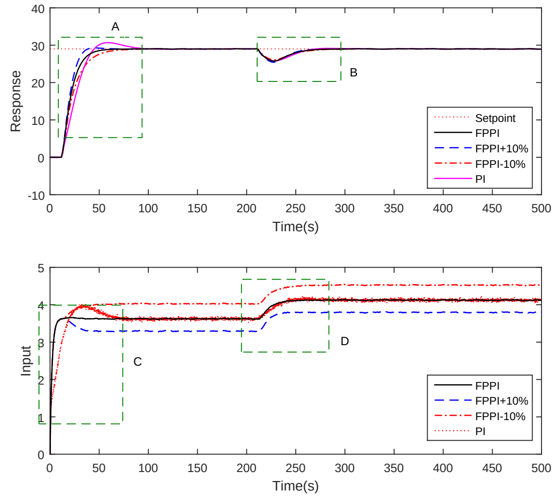


Fig. 12. Sensitivity of the proposed controller to parametric modeling error (10% increase and decrease) in gain of first order plant

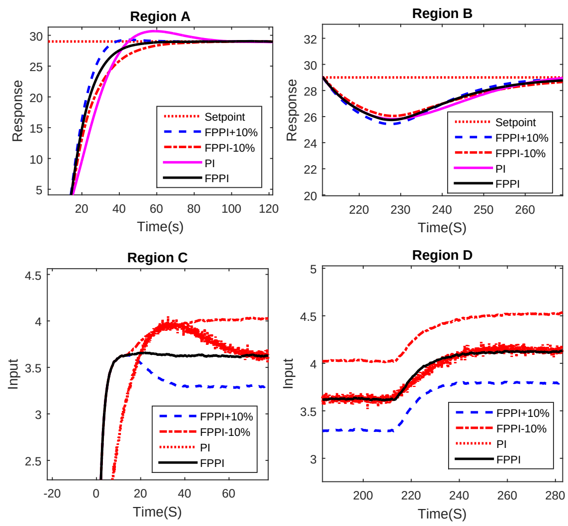


Fig. 13. Zoomed view of Fig. 12 for regions A, B, C and D

Table 5. First order plant performance with model mismatch

Parameter	FPPI	+10% gain	-10% gain	PI
Rise Time (s)	20.340	15.510	26.980	23.820
Settling Time (s)	256.270	252.920	259.950	257.740
Overshoot (%)	0.320	1.170	0.280	5.910
IAE	8011	7474	8765	9375

Performance of the plant with the compared controllers is shown in Fig. 14. The plant is simulated to a unit step signal with disturbance of magnitude 0.5 injected at 200s. Again a white noise signal of power 0.001 is injected at the output. In the figure, regions of interest A, B, C and D are zoomed and further highlighted in Fig. 15. Numerical results of the performance is shown in Table 5. It can be seen by observing the regions of interest that, while PPI controller is severely affected by the noise, the PI controller is not without overshoot. As expected, the response with proposed approach is smoother with overshoot of just

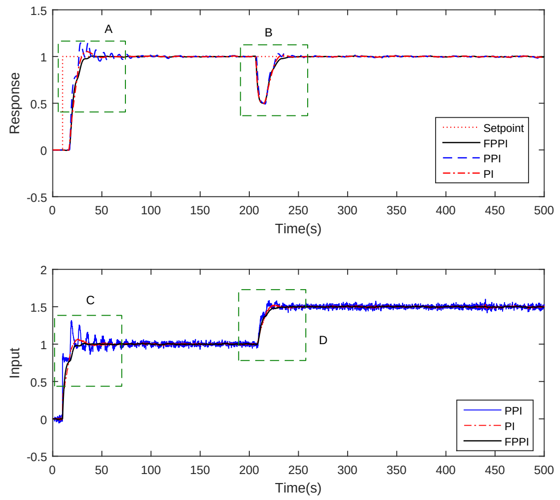


Fig. 14. Second order plant response to various controllers

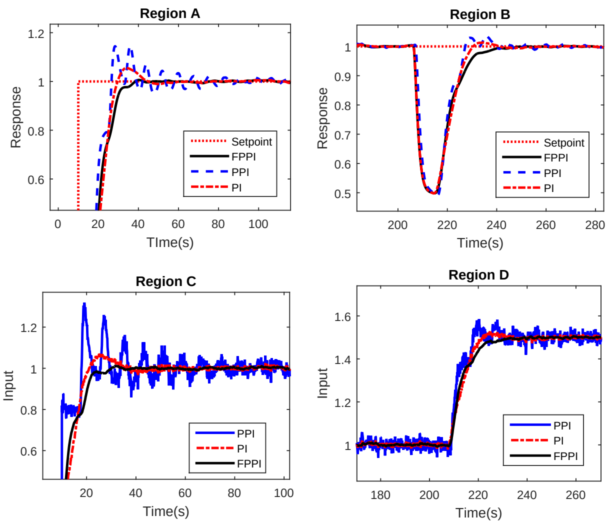


Fig. 15. Zoomed view of Fig. 14 for regions A, B, C and D

Table 6. Second order plant performance

Parameter	FPPI	PPI	PI
Rise Time (s)	10.753	8.3008	9.2581
Settling Time (s)	232.1800	238.2203	228.5845
Overshoot (%)	1.1430	15.3040	5.9200
Undershoot (%)	0.4260	0.6630	0.3038
IAE	221.9950	222.0220	221.5270

around 1% compared to those of PPI and PI with around 15% and 6% respectively. The speed of response of the three controllers are 8.3, 9.3 and 10.8s for the PPI, PI, and FPPI controllers respectively. The slow response of the FPPI is due to filtering effect.

The sensitivity analysis of the proposed controller for second order system is done in a similar way to that of first order system (i.e., by considering 10% increase and decrease in gain). In the same way as the previous plant, despite parametric modelling error, the proposed method produces response with overshoot range of 0.96-1.65% against 12.97% of the PI controller (see Fig. 16 and

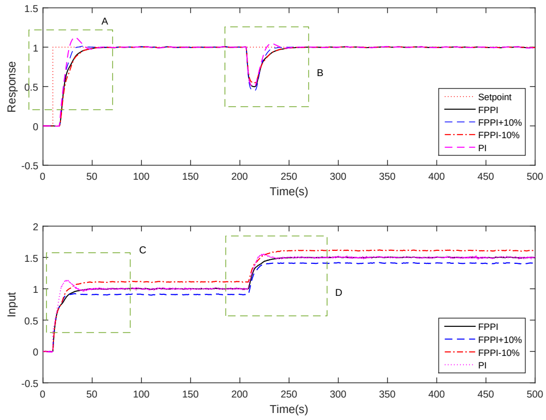


Fig. 16. Sensitivity of the proposed controller to parametric modeling error (10% increase and decrease) in gain of second order plant

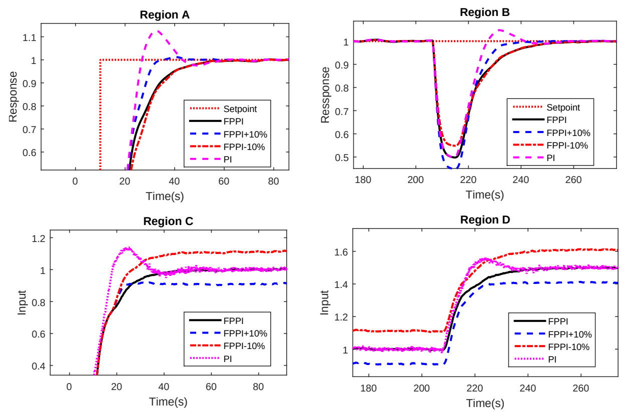


Fig. 17. Zoomed view of Fig. 16 for regions A, B, C and D

Table 7. Second order plant performance with model mismatch

Parameter	FPPI	+10% gain	-10% gain	PI
Rise Time (s)	15.8045	10.3546	16.6866	7.8519
Settling Time (s)	243.489	231.518	242.322	238.527
Overshoot (%)	0.9877	1.6586	0.959	12.969
Undershoot (%)	0.3361	0.3700	0.3026	0.3862
IAE	250.679	226.913	251.348	220.933

Table 7). In the same way, when regions of interests A, B, C and D of Fig. 16 are zoomed in Fig. 17, the robustness of the proposed approach to the mismatches can clearly be seen. Regions A and B shows that the approach responded to step change with little overshoot and can recover from the effect of disturbance in the same way compared to the PI controller. Regions C and D of the figures also showed the effect of the filter on the control signal. The proposed method is smoother compared to the PI.

5.3 Third Order Process

Just as in the case of first and second order systems, parameters of the various controllers compared for the third order system are shown in Table 3. Performance of the plant with the compared controllers is shown in

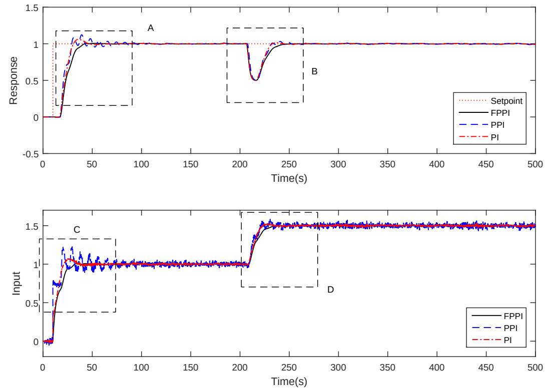


Fig. 18. Third order plant response to various controllers

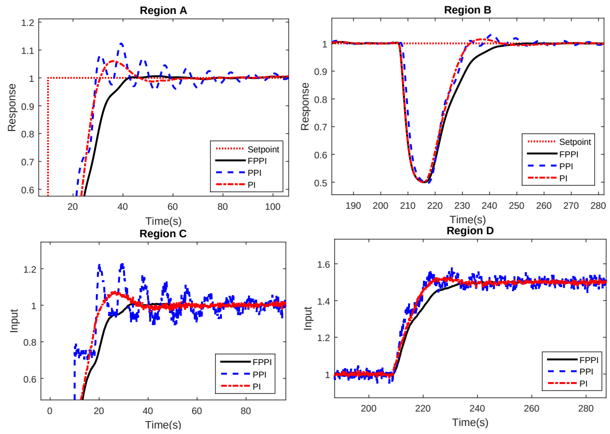


Fig. 19. Zoomed view of Fig. 18 for regions A, B, C and D

Table 8. Third order plant performance

Parameter	FPPI	PPI	PI
Rise Time (s)	13.6855	9.7077	9.8187
Settling Time (s)	239.5497	250.5176	238.7395
Overshoot (%)	1.1184	13.0862	6.6878
Undershoot (%)	0.2779	0.5181	0.3005
IAE	259.6113	242.0804	234.7453

Fig. 18. The plant is simulated to a unit step signal with disturbance of magnitude 0.5 injected at 200s. The injected noise at the output is similar to that of second order plant. In the figure, regions of interest A, B, C and D are zoomed and further highlighted in Fig. 19. Numerical results of the performance is shown in Table 8. The same observation is made here as the second order system regarding the regions of interest. While PPI controller is severely affected by the noise, the PI controller is not without overshoot. Similarly, the response with proposed approach is smoother with overshoot of just around 1.1% compared to those of PPI and PI with around 13% and 6.7% respectively. The speed of response of the three controllers in increasing order are 9.7, 9.8 and 13.6s for the PPI, PI and FPPI controllers respectively. Just as observed in the second order plant, the slow response of the proposed approach is due to the filter effect.

Table 9. Fourth order plant performance

Parameter	FPPI	PPI	PI
Rise Time (s)	9.9675	3.266	8.5755
Settling Time (s)	231.3864	235.4382	232.5069
Overshoot (%)	1.2085	25.4543	6.4535
Undershoot (%)	0.3339	0.8261	0.4959
IAE	259.6113	242.0804	234.7453

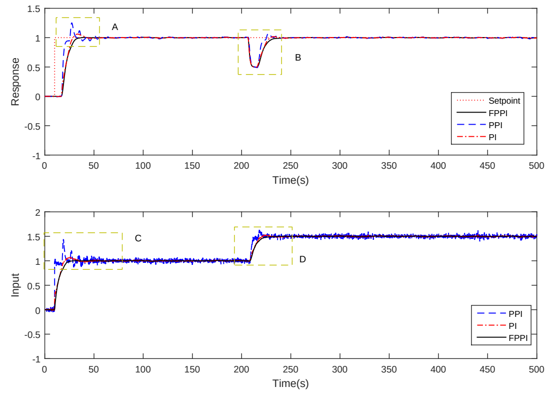


Fig. 20. Fourth order plant response to various controllers

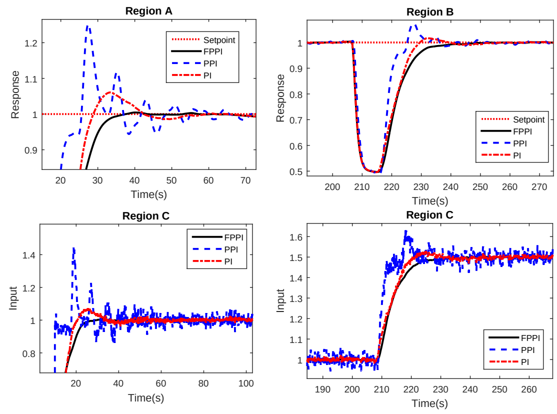


Fig. 21. Zoomed view of Fig. 20 for regions A, B, C and D

5.4 Fourth Order Process

To further ensure that the proposed method can handle complex processes, a fourth order plant is also considered. Similar values of signal are used for this plant as those used for both second and third order plants. The simulation result are presented in Fig. 20 while the regions of interest A, B, C and D in that figure are zoomed in Fig. 21. The numerical results for the first figure are presented in Table 9. For this plant, similar pattern of results with second and third order plants are observed. The lowest overshoot of 1.2% was recorded for the proposed method while those for PPI and PI stood at 6.5% and 25.5% respectively. Again, the effect of the noise on the PPI is evident from the oscillatory nature of its response in both figures.

5.5 Real-time Simulation with WH-HILS

In order to validate the effectiveness of our design, a real-time simulation of the first order model (33) was conducted with the WH-HILS using real-time delay obtained from

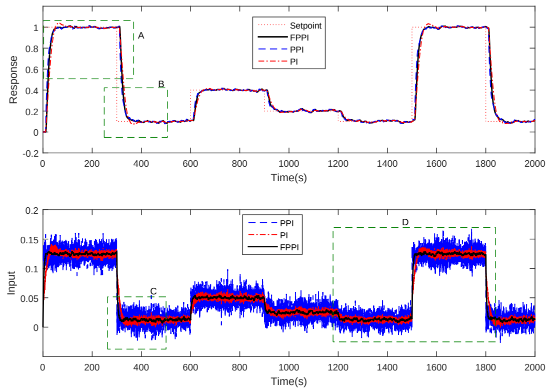


Fig. 22. Response of the first order plant to various controllers with the WH-HILS

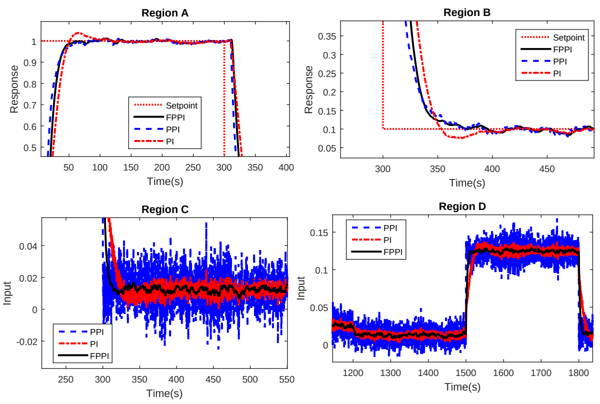


Fig. 23. Zoomed view of Fig. 22 for regions A, B, C and D

the mote with mac address '00170D000030045B' and the results are shown in Fig 22. The zoomed in for the regions of interest are shown in Fig. 23. The real-time delay profile as captured from the network is shown in Fig. 24. The experiment was conducted for the period of 2000s. The experimental set-up is shown in Fig. 8. From Fig. 22, it is observed that the proposed approach outperforms both PI and PPI controllers in terms of noise attenuation and good setpoint tracking. By closely observing Fig. 23, the ability of the proposed approach to attenuate the effect of the noise on the control signal (regions C and D) compared the other controllers is visible.

6. CONCLUSION

In this paper, FPPI controller for use in wireless environment is presented. The robustness of the design to model mismatch and its ability to suppress the effect of measurement noise has been demonstrated in both pure simulation and the hardware in the loop simulation. A likely drawback of the design is that the filter slows the process a little. However even with that effect, the performance of the controller is favourable. Future work will focus on making the design adaptive to the variation in the network delay and practical implementation on an actual plant.

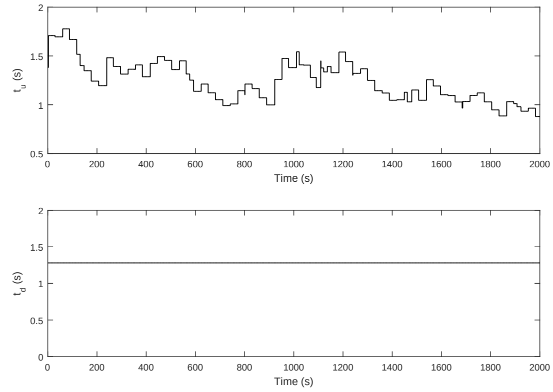


Fig. 24. Real-time Delay of simulation with WH-HILS

ACKNOWLEDGEMENTS

Authors acknowledge the anonymous reviewers for their helpful comments. The authors will also like to thank Mr. Kishore Bingi for his invaluable time and suggestions.

REFERENCES

Arousi, F., Schmitz, U., Bars, R., and Haber, R. (2008). Robust predictive PI controller based on first-order dead time model. In *Proceedings of the 17th World Congress The International Federation of Automatic Control*.

Åström, K.J. and Hägglund, T. (2006). *Advanced PID control*. ISA-The Instrumentation, Systems and Automation Society.

Blevins, T., Chen, D., Han, S., Nixon, M., and Wojsznis, W. (2015). Process Control over Real-Time Wireless Sensor and Actuator Networks. In *2015 IEEE 17th International Conference on High Performance Computing and Communications (HPCC)*, 1186–1191. IEEE.

Blevins, T., Nixon, M., and Wojsznis, W. (2014). PID control using wireless measurements. In *2014 American Control Conference*, 790–795. IEEE.

Blevins, T., Wojsznis, W.K., Nixon, M.J., and Roach, B. (2016). Wireless model predictive control applied for dividing wall column control. In *Event-based Control, Communication, and Signal Processing (EBCCSP), 2016 Second International Conference on*, 1–7. IEEE.

Chen, D., Nixon, M., and Mok, A. (2010a). Future of Wireless and the WirelessHART Standard. In *WirelessHART*, 227–243. Springer.

Chen, D., Nixon, M., and Mok, A. (2010b). Why WirelessHART. In *WirelessHART*, 195–199. Springer.

Chung, T.D., Ibrahim, R., Asirvadam, V.S., Saad, N., and Hassan, S.M. (2016a). Internal model control for industrial wireless plant subject to measured round trip delays. In *Instrumentation and Measurement Technology Conference Proceedings (I2MTC), 2016 IEEE International*, 1–6. IEEE.

Chung, T.D., Ibrahim, R.B., Asirvadam, V.S., Saad, N.B., and Hassan, S.M. (2016b). Adopting EWMA Filter on a Fast Sampling Wired Link Contention in WirelessHART Control System. *IEEE Transactions on Instrumentation and Measurement*, 65(4), 836–845.

De Biasi, M., Snickars, C., Landernäs, K., and Isaksson, A.J. (2008). Simulation of Process Control with

WirelessHART Networks Subject to Clock Drift. In *COMPSAC*, 1355–1360.

Fadel, E., Gungor, V., Nassef, L., Akkari, N., Maik, M.A., Almasri, S., and Akyildiz, I.F. (2015). A survey on wireless sensor networks for smart grid. *Computer Communications*, 71, 22–33.

Ferrari, P., Flammini, A., Rizzi, M., and Sisinni, E. (2013). Improving simulation of wireless networked control systems based on WirelessHART. *Computer Standards & Interfaces*, 35(6), 605–615.

Hassan, S.M., Ibrahim, R., Bingi, K., Chung, T.D., and Saad, N. (2017). Application of Wireless Technology for Control: A WirelessHART Perspective. *Procedia Computer Science*, 105, 240–247.

Hassan, S.M., Ibrahim, R., Saad, N., Asirvadam, V.S., and Chung, T.D. (2016a). Predictive PI controller for wireless control system with variable network delay and disturbance. In *Robotics and Manufacturing Automation (ROMA), 2016 2nd IEEE International Symposium on*, 1–6. IEEE.

Hassan, S.M., Ibrahim, R., Saad, N., Asirvadam, V.S., and Chung, T.D. (2016b). Setpoint weighted WirelessHART networked control of process plant. In *Instrumentation and Measurement Technology Conference Proceedings (I2MTC), 2016 IEEE International*, 1–6. IEEE.

Huang, G., Akopian, D., and Chen, C.P. (2014). Measurement and characterization of channel delays for broadband power line communications. *IEEE Transactions on Instrumentation and Measurement*, 63(11), 2583–2590.

Huba, M. (2013). Comparing 2DOF PI and predictive disturbance observer based filtered PI control. *Journal of Process Control*, 23(10), 1379–1400.

Huba, M. (2015). Filter choice for an effective measurement noise attenuation in PI and PID controllers. In *Mechatronics (ICM), 2015 IEEE International Conference on*, 46–51. IEEE.

Huba, M. and Belai, I. (2014). Experimental evaluation of a DO-FPID controller with different filtering properties. *IFAC Proceedings Volumes*, 47(3), 198–203.

Ikram, W. and Thornhill, N.F. (2010). Wireless communication in process automation: a survey of opportunities, requirements, concerns and challenges. In *Control 2010, UKACC International Conference on*, 1–6. IET.

Jha, S.K., Yadav, A.K., Gaur, P., Gupta, J.R.P., and Parthasarathy, H. (2014). Robust and optimal control analysis of sun seeker system. *Journal of Control Engineering and Applied Informatics*, 16(1), 70–79.

Johansson, P. (2001). An Adaptive PPI Controller. *MSc Theses, Lund University*.

Larsson, P. and Hägglund, T. (2012). Comparison between robust PID and predictive PI controllers with constrained control signal noise sensitivity. *IFAC Proceedings Volumes*, 45(3), 175–180.

Larsson, P.O. (2011). Optimization of low-level controllers and high-level polymer grade changes. *PhD Theses, Lund University*.

Larsson, P.O. and Hägglund, T. (2009). Robustness margins separating process dynamics uncertainties. In *Control Conference (ECC), 2009 European*, 543–548. IEEE.

Larsson, P.O. and Hägglund, T. (2011). Control signal constraints and filter order selection for PI and PID controllers. In *Proceedings of the 2011 American Control Conference*, 4994–4999. IEEE.

Linear Technology (2015). SmartMesh WirelessHART User's Guide. `http://www.linear.com/docs/41887`. Online; accessed 29 January 2016.

Normey-Rico, J.E. and Camacho, E.F. (2008). Dead-time compensators: A survey. *Control engineering practice*, 16(4), 407–428.

Ogan, R.T. (2015). Hardware-in-the-Loop Simulation. In *Modeling and Simulation in the Systems Engineering Life Cycle*, 167–173. Springer.

Olenewa, J. (2013). *Guide to wireless communications*. Cengage Learning.

Petersen, S. and Carlsen, S. (2011). WirelessHART versus ISA100. 11a: the format war hits the factory floor. *IEEE Industrial Electronics Magazine*, 5(4), 23–34.

Ribić, A.I. and Mataušek, M.R. (2012). A New Predictive PI Controller with Additonal Filtering. *IFAC Proceedings Volumes*, 45(3), 489–494.

Santos, A., Lopes, D., César, J., Luciano, L., Neto, A., Guedes, L.A., and Silva, I. (2015). Assessment of WirelessHART networks in closed-loop control system. In *Industrial Technology (ICIT), 2015 IEEE International Conference on*, 2172–2177. IEEE.

Segovia, V.R., Hägglund, T., and Åström, K.J. (2014). Measurement noise filtering for PID controllers. *Journal of Process Control*, 24(4), 299–313.

Sheng, S. and Sun, C. (2016). An adaptive attitude tracking control approach for an unmanned helicopter with parametric uncertainties and measurement noises. *International Journal of Control, Automation and Systems*, 14(1), 217–228.

Shinskey, F. (2001). PID-deadtime control of distributed processes. *Control engineering practice*, 9(11), 1177–1183.

Tan, K.K., Tang, K.Z., Su, Y., Lee, T.H., and Hang, C.C. (2010). Deadtime compensation via setpoint variation. *Journal of Process Control*, 20(7), 848–859.

Wang, H., Tian, Y., and Christov, N. (2014). Event-Triggered Observer based Control of Networked Visual Servoing Control Systems. *Journal of Control Engineering and Applied Informatics*, 16(1), 22–30.

Winter, J., Pereira, C., Netto, J., Souza, F., Muller, I., and Catunda, S. (2016). Analysis of a radio physical layer fault in WirelessHART networks. In *Instrumentation and Measurement Technology Conference Proceedings (I2MTC), 2016 IEEE International*, 1–5. IEEE.