Original Research Article

Performance Evaluation of Networks using Gain Scheduling PID Networked Control System for Nonlinear Systems

---

ABSTRACT

Aims: Though Networked Control Systems (NCS) have great benefits, such as remote control, low costs, and better flexibility, it also has several drawbacks such as network-induced time delay and packet loss, which may result in performance degradation and system instability in nonlinear systems. This paper proposes a gain scheduling Proportional-Integral-Differential (GS-PID) controller design to address the performance degradation issues associated with NCS

Study design: TrueTime toolbox simulator in MATLAB is used for the NCS simulation and network comparison.

Place and Duration of Study: Department of Electronic and Computer Engineering, Faculty of Nigeria, University of Nigeria, Nsukka Enugu State, Nigeria, between March 2022 and March 2023.

Methodology: The performance of the proposed GS-PID and conventional PID controllers is compared through simulations on three different networks (Ethernet, Controlled Area Network (CAN), and switched ethernet).

Results: The GS-PID controller outperforms the traditional PID controller with improvements in settling time and overshoot ranging from 15% to 22%.

Conclusion: The results of the GS-PID over the conventional PID on these three networks highlight the effectiveness of the proposed approach and show potential in improving system performance.

Keywords: Networked control systems, Gain scheduling, Proportional integral derivative, Ethernet, Controlled area network, Switched ethernet

1. INTRODUCTION

Networked Control System (NCS) has been a major focus of academic and industrial research for many years because it is a particular type of cyber-physical system (CPS) that enables efficiency and remote control [1, 2]. As a result, it has displaced the traditional control system, which has several disadvantages [1]. NCS performance has several problems, such as delays caused by the network, transmission delays, and lost packets [3]. In addition, data transfers between the controller and the remote system will cause network delays in addition to delays in controller processing as long as NCS uses a network [4]. Numerous issues, ranging from the analysis and design of control networks [5] to the design of controllers that can handle the effects of communication, are noted to arise when control systems are implemented over networks. According to [6], one of the most important future directions for control is network control. An overview of NCS, including information on system settings, network delay features, and the outcomes of networked delays, is given in [7].

There has been some research on NCS control systems, but most of it has been on PID and an optimized PID controller [13-19]. In [13], a comparative study regarding PID control (for a nonlinear level control system) was evaluated regarding transient response transmitters using PID fuzzy supervision and PID gain scheduling. A performance evaluation of a PID controller for NCS using Ant Colony Optimization (ACO) and fuzzy logic is presented in [14]. In [15], fuzzy logic (FL) with a PID controller was suggested for the design of the Wireless NCS (WNCS), along with the use of a ZigBee network and the PSO technique for getting the best rules. [16] developed an NCS model with an Ethernet network for a third-order DC motor as the plant. Different traffic patterns on the network were looked at. A delay-dependent GS-PID controller was made to deal with the differences that cause measurement and control to take different amounts of time depending on the load. Based on traditional PID, FL, and Gain scheduling, [17] made a fuzzy PID-like GS control method. The control method was tested with a second-order stepper motor, and Ethernet was used to simulate different loads and randomly packed loss. [18] suggested a controller technique for NCS that makes a GS-based state feedback integral controller with an integral action to deal with disturbances that are not zero. The NCS communication network has a limited number of connections, so [19] made a way to schedule them. They showed a probabilistic algorithm for designing scheduling logic and static state feedback controllers that meet the stability requirements of scheduling logic. [20] showed the earliest deadline first (EDF) scheduling method and chose the best network (out of five networks) for it to avoid delays.

In this work, we investigated the behavior of a nonlinear model controlled by a gain-scheduling PID controller in MATLAB, with data transmission over a communication network in TrueTime Simulator, to address the issue of network-induced delay in control systems. Three different communication networks-Ethernet, CAN, and Switched Ethernet are compared to observe how well the system works compared to a conventional PID controller. The paper is organized as follows. In section 2, the nonlinear system model and the system model formulation are described. Then, the networked control system for the work which includes the induced delay, the TrueTime simulator, the PID controller design, and the gain scheduling scheme was presented. In section 3, the simulation results and comparison for the different networks were given. Finally, section 4 summaries the research work by discussing the model performance with respect to the three different networks.

2. METHODOLOGY

2.1 Nonlinear System Model

Figure 1 shows the schematic block diagram of the proposed NCS for a nonlinear model with induced network delay. The NCS schedules the transmission of Input control from GS-PID and Output measurement (Amplitude) from the nonlinear model to meet the desired Amplitude, Ar. Using the gain scheduling technique, the weights of the PID controller are optimally tuned to minimize the error difference between the measured output (amplitude) and the desired amplitude.

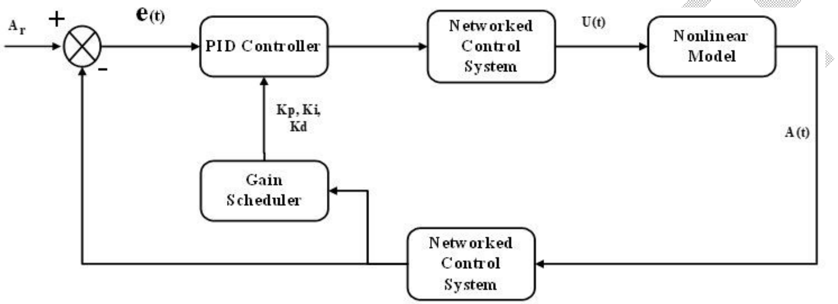


Fig. 1. The Networked Control System with GS-PID schematic

2.1.1 System model formation

Consider a continuous time-invariant nonlinear system that produces an amplitude, x(t) with a given control input, u(t) as defined in Eq. (1). The nonlinear system is assumed to operate without disturbance.

$$u(t) = \dddot{x}(t) + 3\ddot{x}(t) + 3\dot{x}(t) + x(t)$$

The nonlinear model is reformulated into state-space [21] as:

$$\dot{x}(t) = A\vec{x}(t) + Bu(t)$$
$$y(t) = Cx(t)$$

The continuous time-invariant of Eq. (2) and Eq. (3) must be approximated by a discrete-time system because the nonlinear system is computer-based. This discrete-time system is described as follows [21]:

$$x(k+1) = A_d x(k) + B_d u(k)$$
$$y(k) = Cx(k)$$

Where $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -3 & -3 \end{bmatrix}$, $B = [0 \quad 0 \quad 1]^T$, $C = [1 \quad 0 \quad 0]$, $\vec{x}(t) = [x_1 \quad x_2 \quad x_3]^T$, $A_d = e^{A.Ts} \approx (I + A.Ts)$, $B_d = B.Ts$, and Ts is the sampling time.

In order to test the NCS of the nonlinear model shown in Figure 1 over a communication network, the nonlinear model, PID controller with a gain scheduler, and NCS model are developed in MATLAB/SIMULINK as illustrated in Figure 2. The NCS makes use of the Truetime kernel and Truetime blocks designed by [22]. The SIMULINK model's respective node communication schedules are the interference node, Controller node, and System node. By utilizing Truetime Send and Truetime Receive, the schedule transmits and receives signals between the three nodes. The Interference node schedules interferences of data transmission between the three communication networks (Ethernet, CAN, and Switch Ethernet) and the Controller and the System model. As depicted in Figure 3, the system node receives an input signal via the Truetime Receive (Actuator) and outputs a signal via the Truetime Send (Sensor).

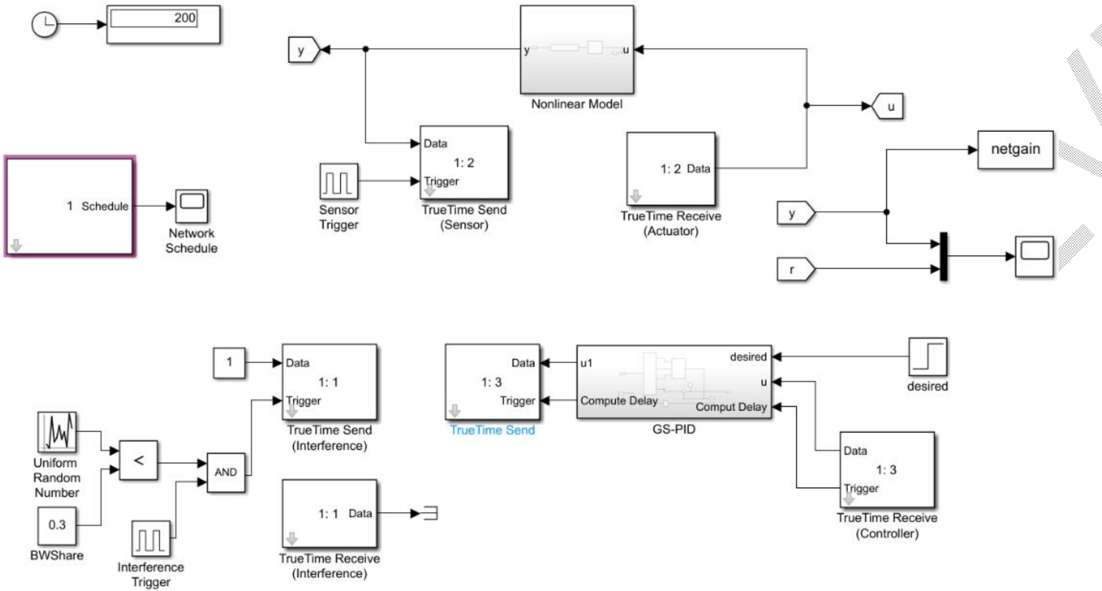


Fig. 2. Simulink model of the nonlinear system with GS-PID Networked control system

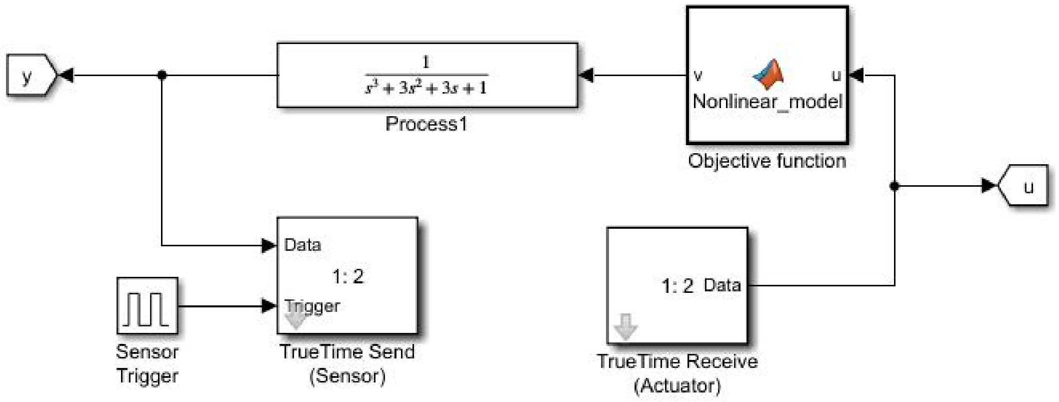


Fig. 3. The nonlinear system Simulink model

2.2 Networked Control System

2.2.1 Networked induced delay

A communication network facilitates the data transmission between the nonlinear system output of Eq. (1) and the controller and the transfer of data between the nonlinear system model and the controller. Therefore, the nonlinear system of Eq. (1) will encounter a network-induced delay, as stated in Eq. (6), because networks are involved in data transmission [23, 24]. This delay could seriously impair the performance of the NCS and even cause systemic instability [25].

$$x(k+1) = A_d x(k) + B_d u(k - \hat{\tau})$$

Where $\hat{\tau}$ is the total delay, τ per sampling time, Ts. According to definitions in Eq. (7) [25, 26], the total delay is calculated as the product of the controller to the nonlinear system (or actuator) delay, τ_{ca}, and the controller to nonlinear system output measurement (or sensor) delay, τ_{sc}.

$$\tau = \tau_{ca} + \tau_{sc}$$

2.2.2 TrueTime simulator

As shown in Figure 4, the nonlinear system defined in Eq. (1) is controlled in real time by the TrueTime [22] MATLAB/Simulink simulator. It enables the simulation of network transmissions, model dynamics, and the execution of controller tasks in real-time kernels. The block library contains the Network block, which allows nodes to communicate over a simulated network; the TrueTime Kernel block, which models a real-time kernel and performs user-defined tasks and interrupts handlers; and a few standalone interface blocks. The kernel block, a representation of a real-time kernel, executes a user-defined task and an interrupt handler. Kernel blocks, also known as nodes, can communicate over simulated networks that are either wireless or wired due to the various network blocks. As shown in Figure 5, three different networks—Ethernet, Control Area Network (CAN), and Switched Ethernet—are chosen from the network scheduler. The masked scheduler enabled network parameters such as processing time delays, data transmission rate, frame overhead, and loss probability to be specified [20]

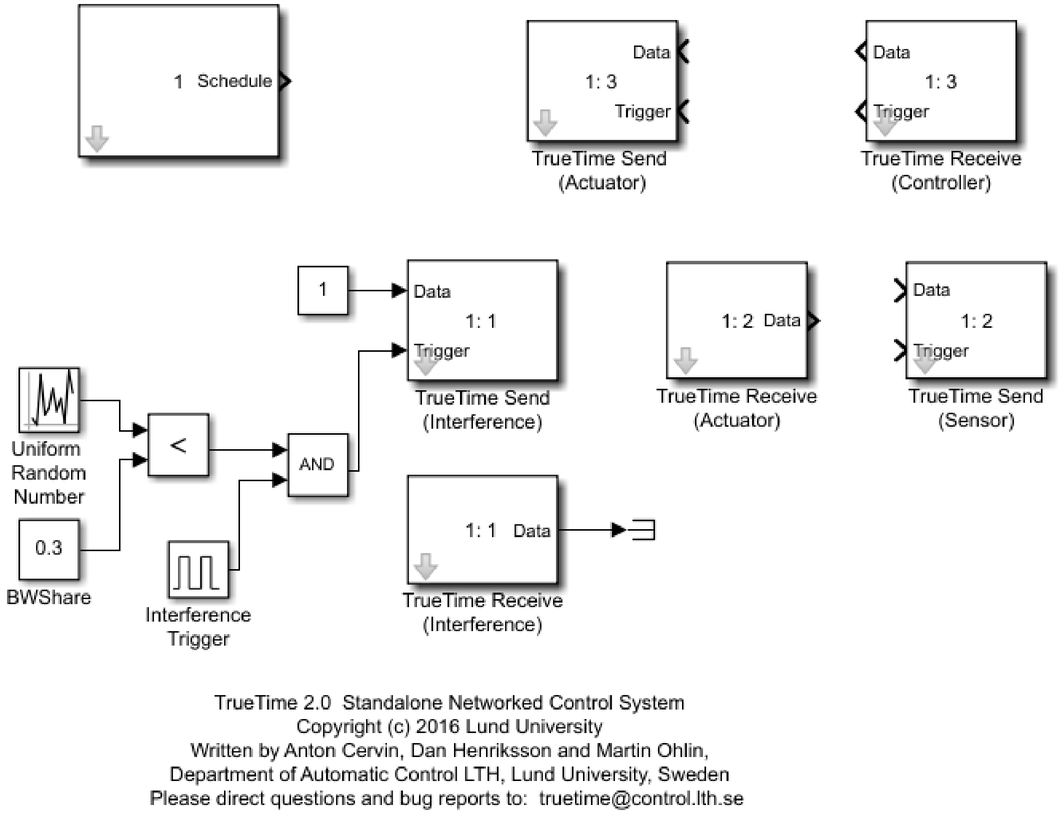
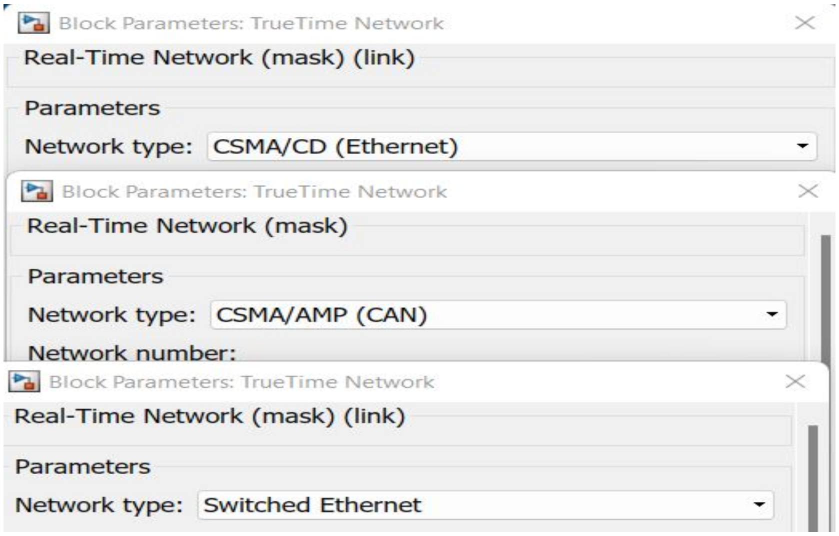


Fig. 4. Truetime standalone Networked Control System [23]


Block Parameters: TrueTime Network
Real-Time Network (mask) (link)
Parameters
Network type: CSMA/CD (Ethernet)

Block Parameters: TrueTime Network
Real-Time Network (mask)
Parameters
Network type: CSMA/AMP (CAN)
Network number:

Block Parameters: TrueTime Network
Real-Time Network (mask) (link)
Parameters
Network type: Switched Ethernet

Fig. 5. Truetime block Library

2.2.3 PID controller design

As shown in Figure 6, the MATLAB/SIMULINK-modeled PID controller generates the required input parameter for the nonlinear system model by ensuring that the error between the desired amplitude and the measured amplitude from the nonlinear system model is close to zero. The PID is delay independent [20] and is derived from the discrete model Eq. (6) as defined in Eq. (8).

$$u(k) = K_P e(t_K) + K_i \int e(t_K) + K_D \frac{d(e(t_K))}{dt_K}$$

The proportional gain, integral gain, and derivative gain are denoted by K_P, K_i, and K_D, respectively. The gain scheduler is used to optimize the tuning of these PID gains. The error term $e(t)$ is defined in Eq. (9).

$$e(t) = A_r - A$$

Where A_r is the desired amplitude and A is the measured amplitude from the nonlinear model.

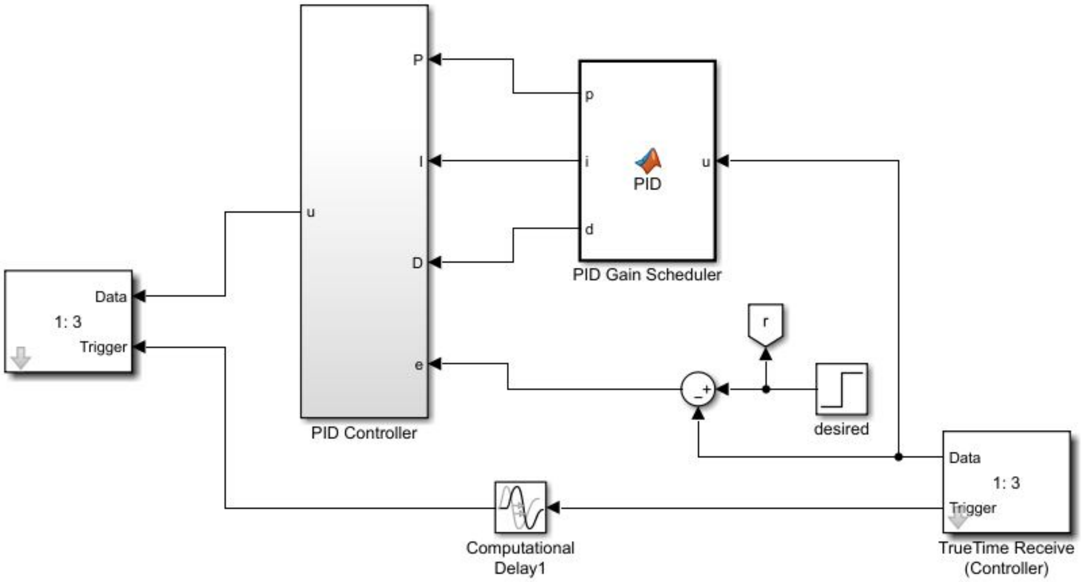


Fig. 6. Proposed Simulink Gain Scheduling PID control technique

2.2.4 Gain scheduling design

Gain scheduling (GS), as shown in Figure 6, is a strategy for dealing with nonlinear systems with time-varying or operating condition-dependent control requirements. It is highly beneficial for reducing the adverse effects of parameter variations concerning the dynamics of the process. The primary advantage of applying gain scheduling is swiftly modifying the controller's parameters in reaction to process modifications [27]. However, the rate at which secondary measurements react to changes in the process is one of the inherent limitations of GS [28]. The most significant disadvantage of GS controller design is that changing from one set of controller parameters to another may not be seamless. As a result, GS may cause oscillatory behavior or even control system instability. In actuality, the controller can utilize the parameter set nearest to the intermediate set - point when the model works among two points. GS may have an adverse effect on the system's overall control efficacy [29]. Table 1 shows the gain scheduler that produces the optimal performance of the proposed NCS in MATLAB.

Table 1. GS-PID MATLAB script Algorithm

%Algorithm: PID Gain scheduling

```
function [Kp, Ki] = PI(u)
if u < 1
    Kp=u;
    Ki=0.15;
    Kd=0;
elseif u==1
    Kp=1/u;
    Ki=1/u/2.5;
    Kd=0;
else
    Kp=1/u;
    Ki=1/u/5.5;
    Kd=0;
end
```

3. RESULTS AND DISCUSSION

The NCS created in the previous section is designed and simulated using initialized parameters from Table 2 in the SIMULINK library of MATLAB 2021a on an MSI CROSSHAIR 15, i7 11th Gen, 16GB Ram, and RTX 3050 GPU system for three communication networks: Ethernet, CAN, and Switched Ethernet. In addition, the data performance for these three communication networks is exported to the Python libraries MATPLOTLIB and SEABORN for improved analysis and visualization.

Table 2. NCS Setup parameters

Parameters	Values
Initial state, $x(0)$	0
Sample time, Ts	0.001 sec
Time delay, $\tau_{ca}=\tau_{sc}$	0.002 sec
Conventional Proportional gain, Kp	0.15
Conventional Integral gain, Ki	1
Conventional Derivative gain, Kd	0
GS-PID gains (Kp, Ki, Kd)	Automatically computed by the gain scheduling algorithm
Minimum frame size	80 bits
Switch memory	80,000 bits
Data rate	8,000 bits/sec

3.1 Ethernet Network

As shown in Figure 7, the output (Amplitude) of the nonlinear model developed in (1) was simulated to track a reference amplitude with PID controller and GS-PID controller using an ethernet network as the communication network with respect to 200 seconds simulation time. At zero initial conditions, the PID controller, denoted by a red dashed line, and the GS-PID controller, denoted by a green dashed line, started tracking the reference (blue line) of 1 amplitude. The PID-controlled nonlinear model has a 70% overshoot, a peak time of 10 seconds, and a settling time of 113 seconds, whereas the GS-PID model has a 45% overshoot, a peak time of 8 seconds, and a settling time of 56 seconds. Based on simulation results, the PID with GS showed superior tracking of the reference amplitude than the conventional PID.

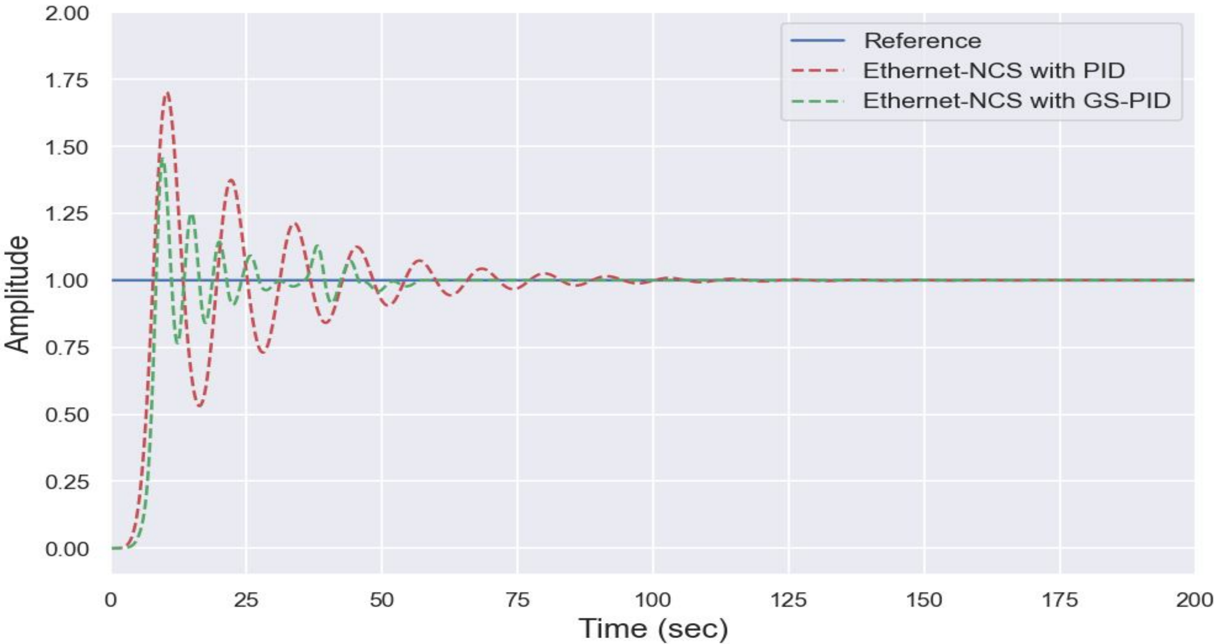


Fig. 7. Simulation Output of NCS Amplitude using Ethernet communication network

Figure 8 shows the control input for a conventional PID, represented by a red dashed line, and a PID control with GS, shown as a green dashed line, with respect to a simulation time of 200 seconds. The GS-PID control input oscillates less and settles at 80 seconds, while the PID control input oscillates more and finally stabilizes at an amplitude of 1 at 100 seconds. Figure 9 shows the Ethernet communication network schedule of a three-node system (Interference, controller, and system). The Ethernet schedule with PID is shown on the left side of the result, while the schedule with GS-PID is shown on the right. According to the simulation analysis, the schedule with PID has a broad range of scheduling intervals for the controller and system nodes. In contrast, the schedule with GS-PID has a smaller range of scheduling intervals.

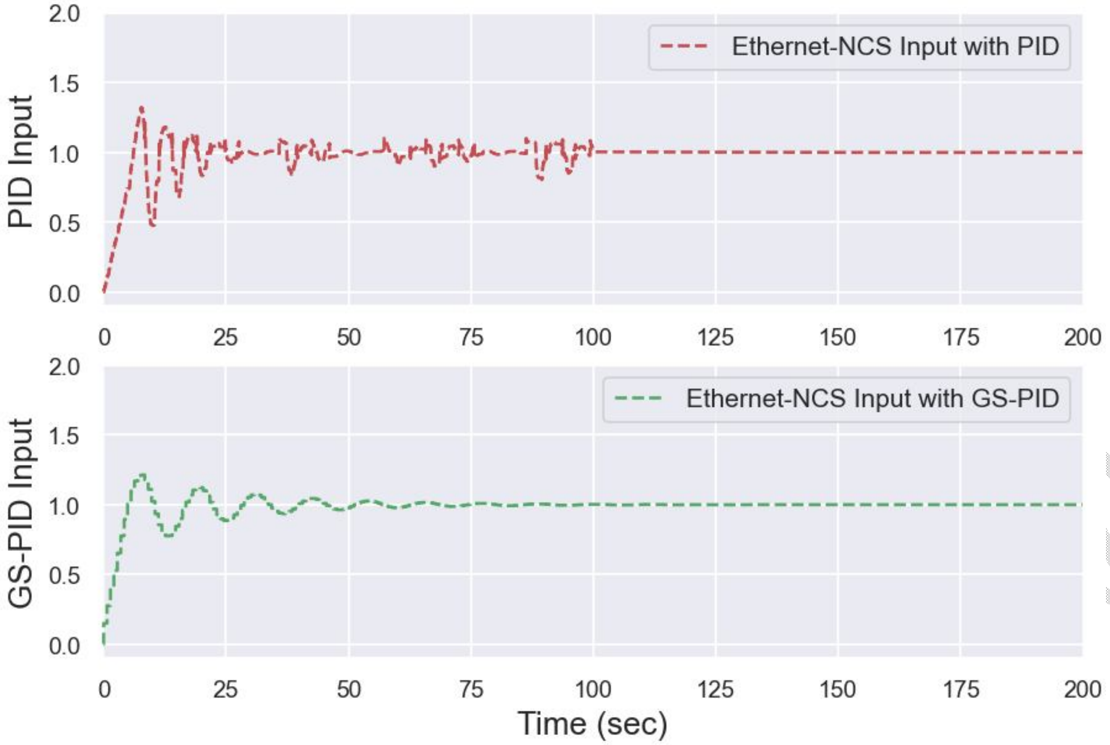


Fig. 8. Simulation of NCS control Input using Ethernet with respect to time

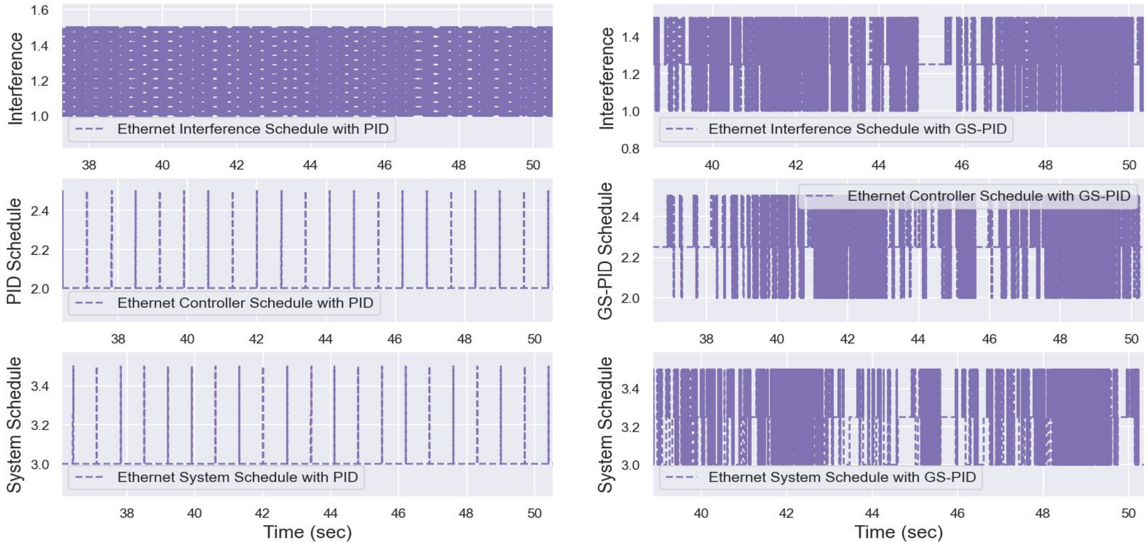


Fig. 9. The Ethernet communication Network Schedule

3.2 Controlled Area Network (CAN)

As shown in Figure 10, the reference tracking performance of the model Amplitude with PID (red dashed line) and GS-PID (green dashed line) controllers are simulated using the CAN network at 200 seconds of simulation time. According to the analysis, the PID has a 60% overshoot, a peak time of 11 seconds, and a settling time of 80seconds, whereas the GS-PID has a 45% overshoot, a peak time of 30 seconds, and a settling time of 8 seconds. The PID with GS demonstrated superior tracking of the reference amplitude compared to the conventional PID.

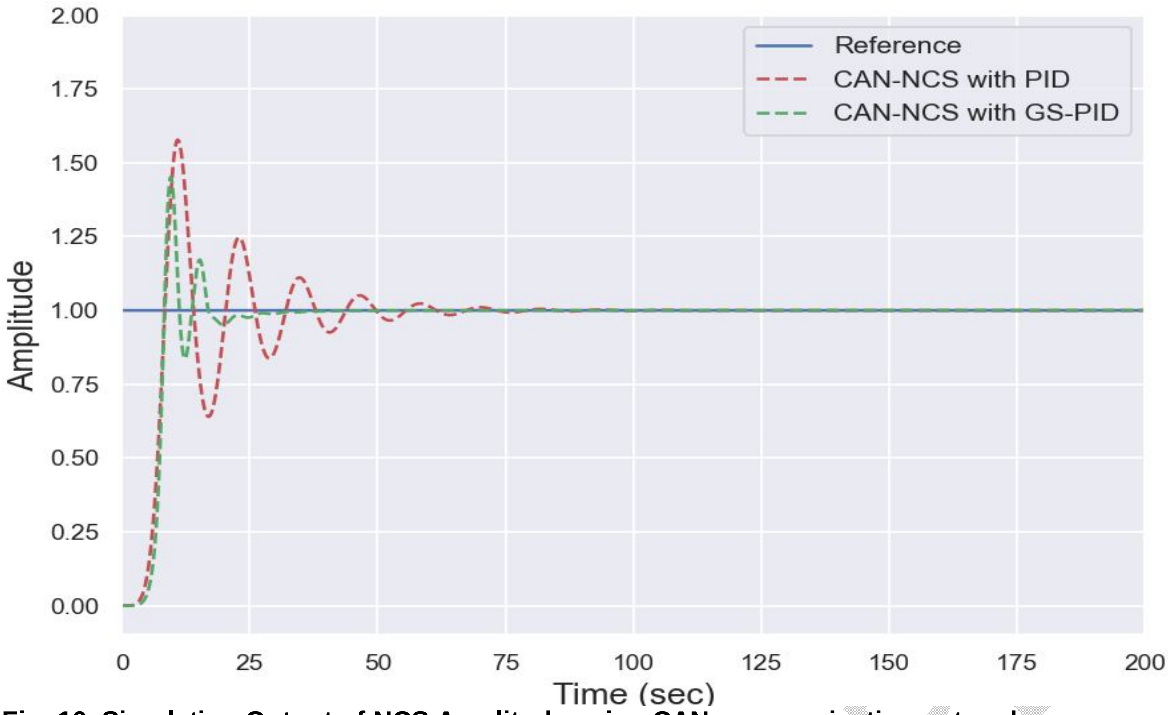


Fig. 10. Simulation Output of NCS Amplitude using CAN communication network

As shown in Figure 11, the PID control input (red dashed line) and the GS-PID control input (green dashed line) performances are simulated using a CAN communication network over a 200-second simulation time. The PID controller stabilized at 60 seconds during the initial sample simulation time, while the GS-PID controller, which depicted irregular oscillations, stabilized at 20 seconds. The CAN communication network schedule of Interference, Controller, and Nonlinear system nodes for conventional PID (left) and GS-PID (right) are simulation with respect to sample time, as shown in Figure 12. The CAN schedule with PID has a broad range of uniform scheduling intervals. In contrast, the GS-PID schedule has a very compact time interval.

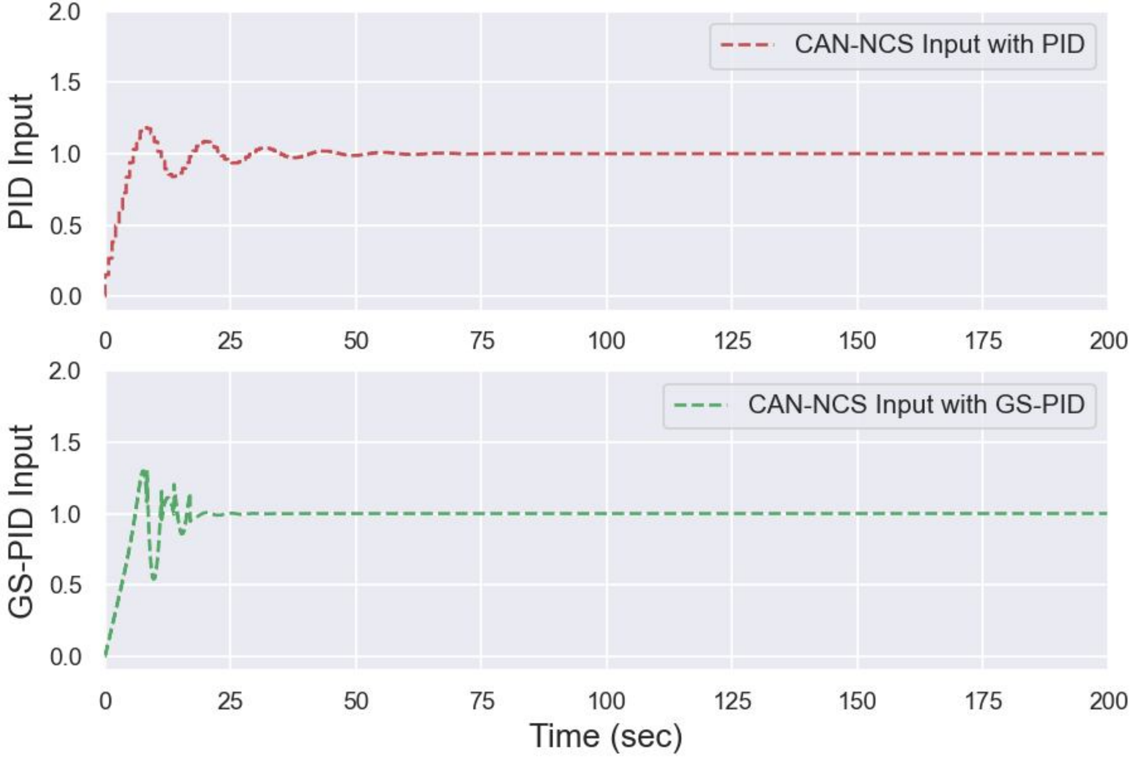


Fig. 11. Simulation of NCS control Input using CAN with respect to time

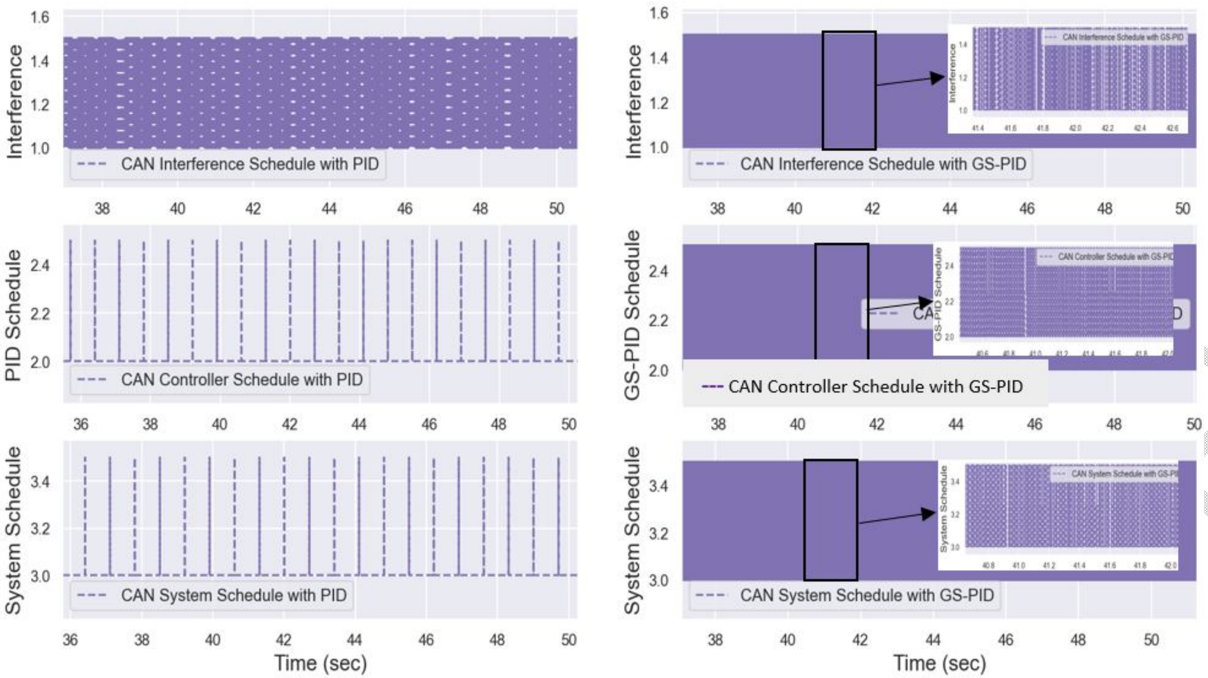


Fig. 12. **The Ethernet communication Network Schedule**

3.3 Switched Ethernet

As shown in Figure 13, the reference tracking performance of the Amplitude with PID (red dashed line) and GS-PID (green dashed line) controllers is simulated using a switched ethernet network at a simulation time of 200 seconds. From the simulation results, the PID has a 65% overshoot, a peak time of 10 seconds, and a settling time of 100 seconds, while the GS-PID has a 43% overshoot, a peak time of 25 seconds, and a settling time of 8 seconds. When the performance of these two controllers was compared, the PID with gain scheduling outperformed the conventional PID for switched Ethernet by 22% with an earlier settling time.

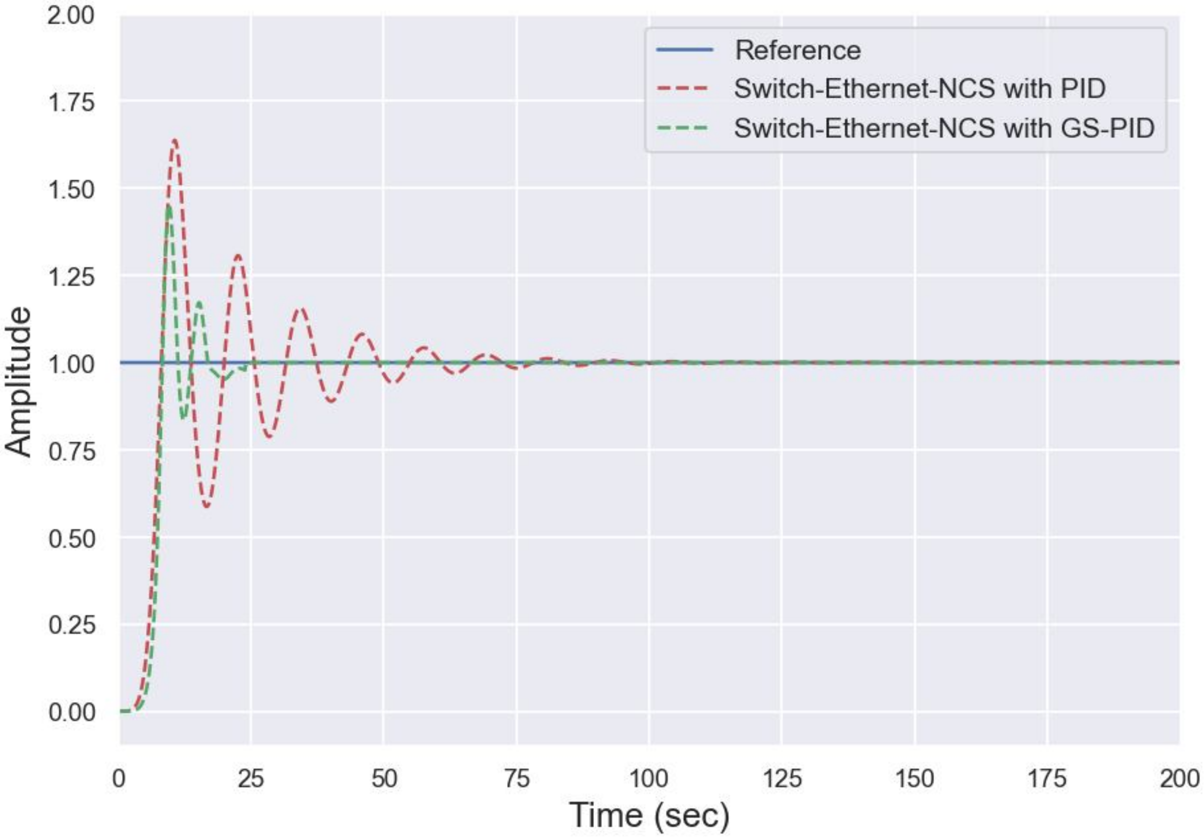


Fig. 13. Simulation Output of NCS Amplitude using Switch Ethernet communication network

The PID (red dash) and GS-PID (green dash) control input performances are simulated using a Switched Ethernet communication network at 200 seconds simulation time, as shown in Figure 14. The PID controller with smooth oscillations stabilized in 65 seconds, while the GS-PID controller with irregular oscillations stabilized in about 25 seconds. As shown in Figure 15, the switched ethernet communication network schedule performed differently for the three nodes (Interference, controller, and system) regarding simulation time. The switched Ethernet with PID is on the left, with a packed schedule for the Interference and wide scheduling interval ranges for the Controller and Systems nodes, respectively. At the same time, the Switch Ethernet network nodes for GS-PID have a more compact scheduling interval.

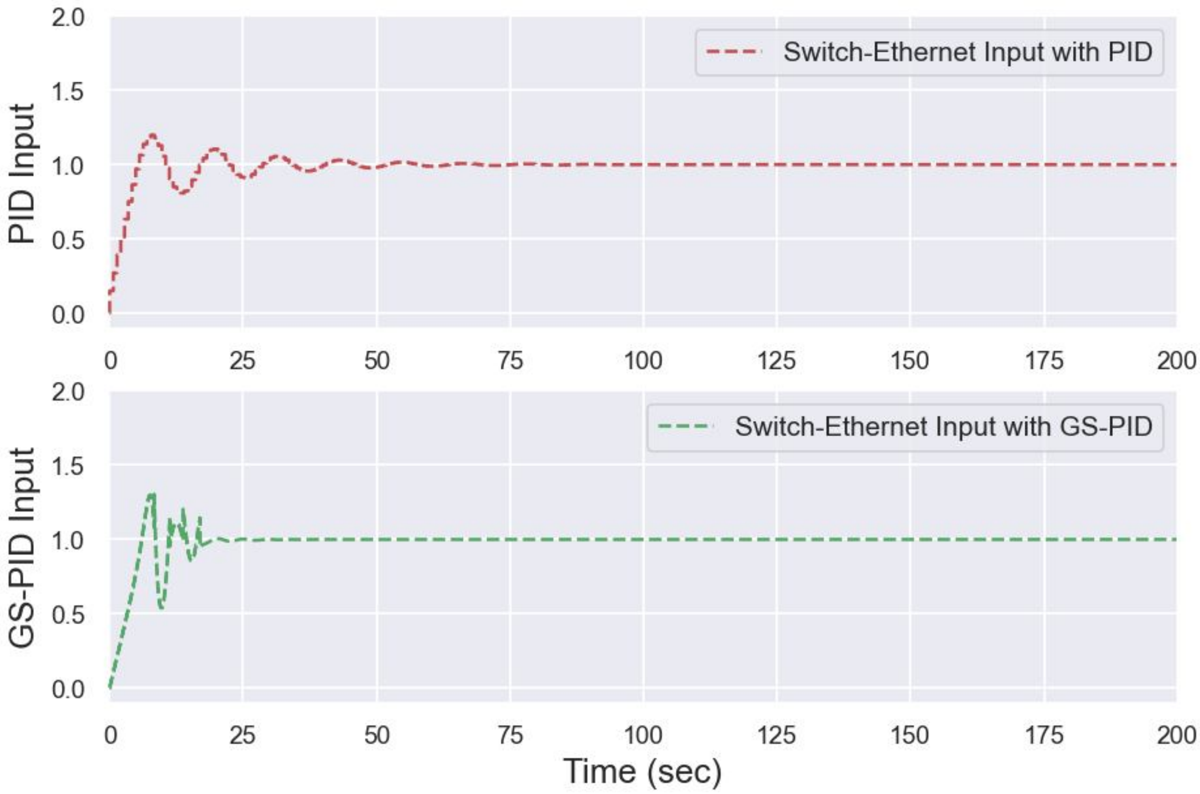


Fig. 14. Simulation of NCS control Input using Ethernet with respect to time

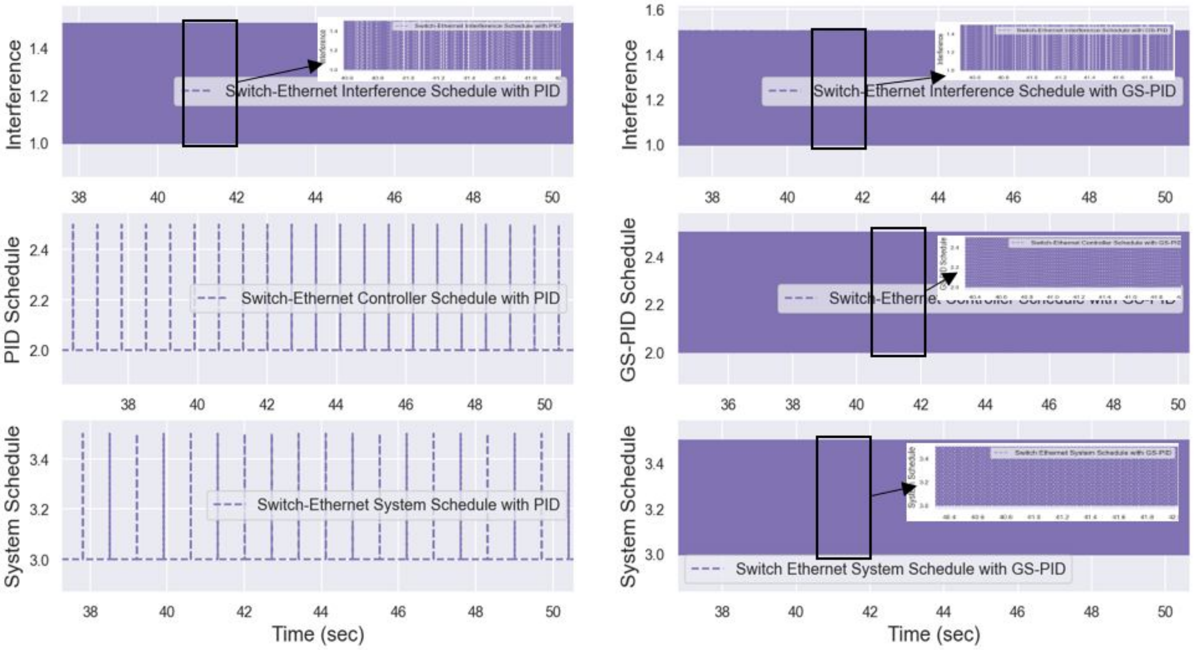


Fig. 15. The Switched Ethernet communication Network Schedule

3.4 Comparison

In a networked controlled system with a PID controller and a GS-PID controller with TrueTime in MATLAB, Table 3 compares the in-induced time delays of the three networks: Ethernet, CAN, and Switched Ethernet. The PID controller has high values for peak time, settling time, and overshoot for the three network channels. While the GS-PID controller significantly improves peak time and settling time, it also significantly reduces the overshoot percentage.

Table 3. Comparison of three different types of networks with GS-PID and PID controllers

Network	Controller	Peak Time	Settling Time	Overshoot
Ethernet	PID	10 seconds	113 seconds	70%
	GS-PID	8 seconds	56 seconds	45%
CAN	PID	11 seconds	80 seconds	60%
	GS-PID	8 seconds	30 seconds	45%
Switched Ethernet	PID	10 seconds	100 seconds	65%
	GS-PID	8 seconds	25 seconds	43%

4. CONCLUSION

In this research work, an NCS model for a nonlinear system is developed using the TrueTime toolbox in MATLAB. The communication network setup considers three nodes: the interference node, the Controller node, and the System node. The nonlinear model is controlled by a gain-scheduling PID controller and compared to the conventional PID controller in three different network communication channels: Ethernet, CAN, and Switched Ethernet. As shown in Table 3, GS-PID outperforms the PID controller in settling time and overshoot by 57 seconds and 25%, respectively, in the Ethernet network. The GS-PID controller outperforms the PID controller in the CAN network by 50 seconds of settling time and 15% overshoot. Finally, the GS-PID controller outperformed the PID controller for Switched Ethernet networks with a settling time of 75 seconds and an overshoot of 22%. The Switched Ethernet GS PID controller out of these networks has a smaller overshoot, while the CAN network's GS PID controller has a shorter settling time.

REFERENCES

1. Gautam M K, Pati A, Mishra S K, Appasani B, Kabalci E, Bizon N, Thounthong P. A comprehensive review of the evolution of networked control system technology and its future potentials. Sustainability (Switzerland). 2021;13(5):1-39. https://doi.org/10.3390/su13052962.
2. Huang S, Zhou C J, Yang S H, Qin Y Q. Cyber-physical system security for networked industrial processes. Int. J. Autom. Comput. 12, 567–578 (2015). https://doi.org/10.1007/s11633-015-0923-9
3. Qi Q, Zhang H, Ji Z. Further results on stabilization for NCSs with packet losses and transmission delay: UDP case. J Franklin Inst. 2019;356(8):4601-4621. https://doi.org/10.1016/j.jfranklin.2019.04.022
4. Bushnell L, Ye H. Networked Control Systems: Architecture and Stability Issues. In: Encyclopedia of Systems and Control. Springer London; 2015:835-842. https://doi.org/10.1007/978-1-4471-5058-9-151
5. Park P, Ergen S C, Fischione C, Lu C, Johansson K H. Wireless Network Design for Control Systems: A Survey. IEEE Communications Surveys and Tutorials. 2018;20(2):978-1013. https://doi.org/10.1109/COMST.2017.2780114
6. Zhao Y B, Sun X M, Zhang J, Shi P. Networked Control Systems: The Communication Basics and Control Methodologies. Math Probl Eng. 2015;2015:1-9. https://doi.org/10.1155/2015/639793
7. Gupta R A, Mo-Yuen C. Networked Control System: Overview and Research Trends. IEEE Transactions on Industrial Electronics. 2010;57(7):2527-2535. https://doi.org/10.1109/TIE.2009.2035462
8.] Dallolio A, Øveraas H, Johansen T A. Gain-scheduled steering control for a wave-propelled unmanned surface vehicle. Ocean Engineering. 2022;257:111618. https://doi.org/10.1016/j.oceaneng.2022.111618
9. Bett C J. Gain-Scheduled Controllers. In: The Electrical Engineering Handbook. Elsevier; 2005:1107-1114. https://doi.org/10.1016/B978-012170960-0/50086-4
10. Gao J, Budman H M. Design of robust gain-scheduled PI controllers for nonlinear processes. J Process Control. 2005;15(7):807-817. https://doi.org/10.1016/j.jprocont.2005.02.003
11. Vesel´y V, Ilka A. Gain-scheduled PID controller design. J Process Control. 2013;23(8):1141-1148. https://doi.org/10.1016/j.jprocont.2013.07.002
12. Park B, Nah J, Choi J Y, Yoon I J, Park P. Robust Wireless Sensor and Actuator Networks for Networked Control Systems. Sensors. 2019;19(7):1535. https://doi.org/10.3390/s19071535

13. Hamdani C N, Adi W W, Anggriawan D O. Comparative Study on Fuzzy Supervision and Gain Scheduling for Nonlinear Level Control System. In: Atlantis Press; 2021:493-497. https://doi.org/10.2991/aer.k.210810.085
14. Eke-Nnodu C O, Udeze C C, Chukwudozie C, Ndefo M I, Ugwu H I. Performance Evaluation of an Optimized PID Controller for Networked Control System Using ANT Colony Optimization. Iconic Research And Engineering Journals. 2021;5(5):183-191.
15. Salman R S, Abdulsadda A T. Adaptive Gain Scheduling Based controller for Networking Control Systems. Al-Furat Journal of Innovations in Electronics and Computer Engineering. 2020;1(2):1. https://doi.org/10.46649/110420-01
16. Popovi´c N, Naumovi´c M. Networked and Cloud Control Systems - Modern Challenges in Control Engineering. IJEEC - INTERNATIONAL JOURNAL OF ELECTRICAL ENGINEERING AND COMPUTING. 2019;2(2). doi:10.7251/IJEEC1802091P
17. Salim L, Awad O A. Gain Scheduling PID Controller For Networked Control System with Random Delay. Vol 4.; 2015. www.erpublications.com
18. Li H, Sun Z, Chow M Y, Sun F. Gain-Scheduling-Based State Feedback Integral Control for Networked Control Systems. IEEE Transactions on Industrial Electronics. 2011;58(6):2465-2472. https://doi.org/10.1109/TIE.2010.2066533
19. Singh M, Kundu A. A scheduling algorithm for networked control systems. IFAC-PapersOnLine. 2022;55(13):37-42. https://doi.org/10.1016/j.ifacol.2022.07.232
20. Kalaivani CT, Kalaiarasi N. Earliest deadline first scheduling technique for different networks in network control system. Neural Comput Appl. 2019;31(S1):223-232. https://doi.org/s00521-018-3729-1
21. Chaparro L F, Akan A. Discrete-Time Signals and Systems. In: Signals and Systems Using MATLAB. Elsevier; 2019:487-557. doi:10.1016/B978-0-12-814204-2.00020-X
22. Cervin A, Henriksson D, Ohlin M. TRUETIME 2.0 – Reference Manual. Available from: http://archive.control.lth.se/media/Research/Tools/TrueTime/report_2016-02-10.pdf
23. Ghosh A B, Lal Priya P S. Adaptive Discrete Sliding Mode Controller for Networked Control Systems. In: 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET). IEEE; 2018:1-5.
24. Juanole G, Mouney G, Sauter D, Aubrun C, Calmettes C. Decision making improvement for diagnosis in Networked Control Systems based on dynamic message scheduling. In: 18th Mediterranean Conference on Control and Automation, MED'10. IEEE; 2010:280-285. doi:10.1109/MED.2010.5547681
25. Phyo A N, Min N Z, Mo M W M, Myo T H. Simulation Of Wireless Networked Control System Using Truetime and Matlab. International Journal Of Scientific \& Technology Research. 2016;5(06). Available from: https://www.ijstr.org/final-print/june2016/Simulation-Of-Wireless-Networked-Control-System-Using-Truetime-And-Matlab.pdf
26. Shah D, Mehta A. Discrete-time sliding mode controller subject to real-time fractional delays and packet losses for networked control system. Int J Control Autom Syst. 2017;15(6):2690-2703. doi:10.1007/s12555-016-0761-8
27. Magalhães S C, Fontella G, Borges R F O, Vega M P, Calçada L A, Scheid C M. Construction of software using gain-schedule/auto-adaptive control strategy for automation in drilling fluid production plants. J Pet Explor Prod Technol. 2020;10(2):501-514. doi:10.1007/s13202-019-00746-x
28. Bagyaveereswaran V, Arulmozhivarman P. Gain Scheduling of a Robust Setpoint Tracking Disturbance Rejection and Aggressiveness Controller for a Nonlinear Process. Processes. 2019;7(7):415. doi:10.3390/pr7070415
29. Tipsuwan Y, Chow MY. Gain Scheduler Middleware: A Methodology to Enable Existing Controllers for Networked Control and Teleoperation—Part I: Networked Control. IEEE Transactions on Industrial Electronics. 2004;51(6):1218-1227. doi:10.1109/TIE.2004.837866