Robust stability analysis and real-time speed tracking control of stepper motors via matlab/OPC-PLC integration with smith predictor

Nguyen Quang Binh[1], Hoang Van Quyet[1], Le Van Tuan[1*]

[1] Control, Automation in Production and Improvement of Technology Institute (CAPITI), 89B Ly Nam De, Hoan Kiem, Hanoi, Vietnam

*Email: visaosang89@gmail.com

Abstract

This paper presents a robust real-time speed tracking control approach for a two-phase stepper motor based on MATLAB/Simulink–OPC–PLC integration with Smith predictor delay compensation. In OPC-based industrial systems, distributed communication and computation processes introduce significant time delays that may degrade closed-loop stability and tracking performance. An equivalent-delay modeling framework is established to represent MATLAB computation, OPC communication, PLC execution, and sensor feedback delays as a unified pure delay. The experimentally measured total delay is approximately 0.40s. A stability-oriented analysis of the Smith predictor is conducted using small-gain conditions and phase-margin-based delay tolerance criteria. The nominal delay-free system exhibits a gain margin of 9.6dB and a phase margin of 52.3°, with a complementary sensitivity peak $\|T(j\omega)\|_\infty = 1.38$*, ensuring robust stability under 10–15% model uncertainty and practical delay variations. Semi-real-time experiments were performed under three feedback configurations and repeated five times for consistency verification. Results show that the Smith predictor significantly improves tracking performance compared to conventional PID control. The model-based feedback configuration achieves the best performance, with a settling time of 1.5s, zero overshoot, and a steady-state error of 1.83%. The proposed approach provides a practical and computationally efficient solution for delay compensation in PLC-based industrial control systems.*

Keywords: *Robust stability, Smith predictor, distributed time delay, OPC–PLC integration, real-time industrial control.*

1. INTRODUCTION

Programmable Logic Controllers (PLCs) are widely used in industrial process control due to their robustness, flexibility, and suitability for harsh operating environments. PLC-based systems offer reliable solutions for controlling a wide range of electromechanical devices, including electric drives, conveyors, and robotic subsystems. However, most industrial PLCs are typically equipped with basic control structures such as on–off control or conventional PID controllers, which may be insufficient for achieving high-performance control in systems affected by communication and computation delays.

To overcome these limitations, MATLAB/Simulink is often employed as a high-level design and analysis platform, enabling the implementation of advanced control algorithms

https://doi.org/10.65153/s2pkyh47

while PLCs handle low-level execution. The integration of MATLAB/Simulink with PLCs is commonly realized through OPC (Object Linking and Embedding for Process Control) servers, which provide a standardized and flexible communication framework. KEPServerEX, in particular, supports OPC UA and is widely adopted for secure and scalable data exchange between industrial devices and software applications.

Several studies have explored OPC-based control architectures in different application domains. In [1], OPC servers and image processing techniques were used for fruit classification on conveyor systems. The authors of [2] demonstrated PLC–KEPServerEX–MATLAB integration for speed control of DC motors, while [3] applied OPC-based adaptive fuzzy PID control to temperature regulation problems. Educational-oriented OPC–PLC control platforms were reported in [4], and PID model prediction for PLC implementation was investigated in [5]. These studies confirm the feasibility and practicality of OPC-based control systems for industrial and educational purposes.

Nevertheless, a common limitation of the above works is that time delays inherent in OPC–PLC-based systems are not explicitly modeled or rigorously analyzed. In practice, delays arise from multiple sources, including data transmission through OPC servers, computation time in PLCs and MATLAB/Simulink, sensor signal processing, and actuator response. The combined effect of these delays forms a distributed delay structure, which can significantly degrade control performance and may even destabilize the closed-loop system if not properly addressed.

The Smith predictor is a classical and effective approach for compensating pure time delays in control systems. Its application has been reported in various tracking and industrial control problems, including systems with communication and image-processing delays [10]. However, most existing implementations apply Smith compensation in a heuristic manner, without providing formal guarantees on stability and robustness when model mismatch or delay uncertainty is present. This limitation becomes particularly critical in OPC–PLC environments, where delay characteristics may vary depending on network load, sampling periods, and computational burden.

Motivated by these observations, this paper focuses on robust real-time speed tracking control of stepper motors in OPC–PLC-based systems. Unlike servo motors, stepper motors are widely used in industrial automation due to their simplicity, cost-effectiveness, and precise positioning capability, but they are also sensitive to delay-induced oscillations and performance degradation. Therefore, ensuring stability and robustness in delayed control loops is of practical importance. The main contributions of this paper can be summarized as follows:

1. An equivalent-delay modeling framework is established for MATLAB/Simulink–OPC–PLC integrated control systems, explicitly capturing distributed communication, computation, execution, and sensing delays and representing them as a unified pure-delay element suitable for predictor design.

2. A stability-oriented analysis of the Smith predictor is developed for the considered industrial architecture. Small-gain robustness conditions and phase-margin-based delay tolerance criteria are derived and interpreted in the context of OPC-based PLC control loops.

3. A comparative semi-real-time experimental validation is conducted under three different feedback configurations. The results reveal that model-based feedback within the Smith structure significantly enhances robustness and tracking performance in PLC implementations, providing practical insight for industrial deployment and engineering education laboratories.

The remainder of this paper is organized as follows. Section 2 describes the structure of the OPC–PLC-based embedded control system and introduces the Smith predictor. Section 3 presents the stability and robustness analysis of the Smith delay compensation scheme. Section 4 discusses the identification of the system model and evaluates the control performance through semi-real-time experiments. Finally, Section 5 concludes the paper and outlines directions for future research.

2. STRUCTURE OF EMBEDDED CONTROL SYSTEM USING OPC-PLC

2.1. Embedded system control structure using OPC-PLC

The system works based on OPC-PLC with the structure of a 4-layer SCADA system [11], shown in Figure 1. The monitoring layer acts as a central control station. Parameters and control signals from algorithms are displayed, and data storage is managed by SCADA software. Here, Simulink is used to design the interface, establish OPC connections, and output control signals to PLC. The core of the OPC Server layer is formed based on the OPC Server platform, serving to extract data on the OPC bus. This study uses OPC Server KEPServer Ex. The connection between PC and OPC Server is performed through the process of linking and embedding OPC objects. At the embedded control layer, the control program based on algorithms is designed and executed on PLC. The connection between PLC and OPC Server is via Modbus RTU industrial communication. The field layer is a microstep controller that receives control pulses from PLC to control the stepper motors. The rotary encoder sensor feeds back the measured speed to the PLC.

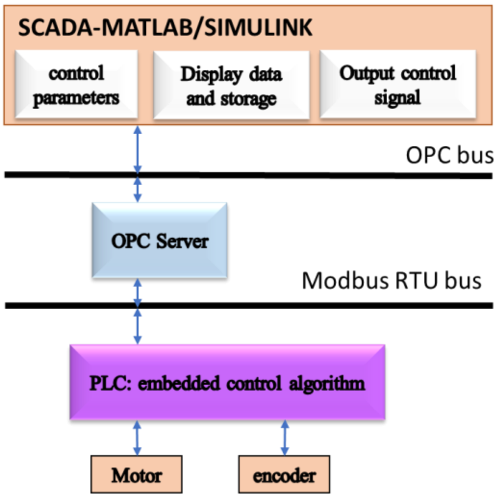

Figure 1. 4-layer embedded diagram configuration of the control system.

The control structure in Figure 1 shows the presence of delays in several components, including communication delays for reading/writing data via OPC server, calculation delays of the controller (PLC, driver), communication delays between the sensor (encoder) and the controller, delays between the controller and the actuator (motor), and calculation delays on Simulink. The paper considers the combined delay in the form of pure delay, which can be determined experimentally.

2.2. Smith Predictor

The Smith predictor proposed by O.J.M. Smith works by using a predictive model of the control object to estimate and predict the future of the system. Instead of reacting directly to the output, the system uses information from the predictive model to adjust the output. This helps to minimize the effect of delay and improve the overall performance of the control system. The Smith predictor uses the predictive model of the delayed object to restructure the control loop, thereby eliminating the effect of pure delay.

Consider the control object with a transfer function $G(s)$, where K is the gain, T is the time constant, and τ is the delay.

$$G(s) = \frac{K}{Ts+1}e^{-\tau s} \quad (1)$$

The Smith delay compensation structure is of the form (2), and the Smith predictor is represented in Figure 2 [6].

$$Y(s) = \frac{\hat{G}(s)}{1 + \hat{G}(s)C(s)} R(s) \tag{2}$$

Where, $C(s)$ is the controller, $R(s)$ is the desired signal, $Y(s)$ is the output, $\hat{G}(s)$ is the predictive model. When the model $\hat{G}(s)$ is asymptote $G(s)$, the system will almost eliminate the effect of the delay element.

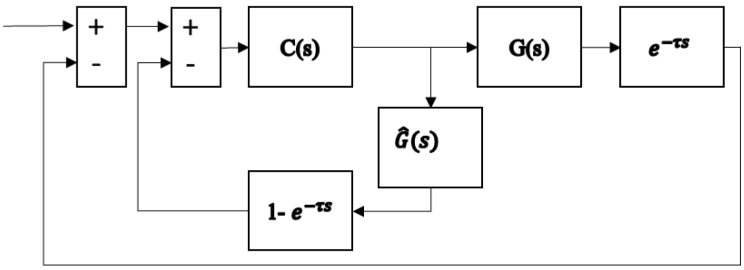

Figure 2. Structural diagram depicting the Smith predictor.

It is observed that the Smith controller contains an object model in the feedback circuit, so it is necessary to find the transfer function $\hat{G}(s)$ (without delay) of the system relatively accurately. Delay makes the system less stable, so the analysis of the stability of the system with delay is a very important task in control systems.

3. STABILITY ANALYSIS OF SMITH DELAY COMPENSATION SYSTEM

3.1. Assumption

A1: Consider the controller $C(s)$ as a linear controller (e.g., P, PI, PID) designed so that the pair $[C, \hat{G}]$ is stable (i.e., the nominal closed-loop system with the model $\hat{G}$ is stable).

A2: $T_0(s)$ is complementary sensitivity of the nominal system (no delay in Smith's internal circuit).

$$T_0(s) = \frac{C(s)\hat{G}(s)}{1 + C(s)\hat{G}(s)}$$

A3: Every $\| * \|_\infty$ standard is an infinite frequency standard: $\|F\|_\infty = |F(j\omega)|$.

3.2. Mathematical basis of stability analysis

Theorem 1 (On equivalent stability when the model is correct)*: If the model is completely correct, meaning* $G(s) = \hat{G}(s)e^{-\tau s}$*, then the actual closed-loop system using the Smith predictor has the same characteristic polynomial as the system* $[C(s), \hat{G}(s)]$ *without delay, specifically, the poles are the solutions of the characteristic equation* $1 + C(s)\hat{G}(s) = 0$. *The*

https://doi.org/10.65153/s2pkyh47

internal stability of the Smith closed-loop system is equivalent to the stability of the pair $[C, \hat{G}]$ *without delay.*

Proof. According to assumption A1, the control signal $U(s)$ generated by controller $C(s)$ (P controller) from the error $E(s)$:

$$U(s) = C(s)E(s) \tag{3}$$

We have an actual object:

$$Y(s) = G(s)U(s) = \hat{G}(s)e^{-\tau s}U(s) \tag{4}$$

The model in the predictor generates two quantities: Model output without delay (5) and model output with delay (6).

$$\tilde{Y}(s) = \hat{G}(s)U(s) \tag{5}$$

$$\hat{Y}(s) = \hat{G}(s)e^{-\tau s}U(s) \tag{6}$$

The Smith predictor determines the current estimate (predicted current output):

$$Y_{est}(s) = \tilde{Y}(s) + [Y(s) - \hat{Y}(s)] \tag{7}$$

When the model is correct:

$$Y(s) = \hat{G}(s)e^{-\tau s}U(s) = \hat{Y}(s) \tag{8}$$

Therefore:

$$Y(s) - \hat{Y}(s) = 0; Y_{est}(s) = \tilde{Y}(s) = \hat{G}(s)U(s) \tag{9}$$

The error given to the controller is:

$$E(s) = R(s) - Y_{est}(s) = R(s) - \hat{G}(s)U(s) \tag{10}$$

So:

$$U(s) = C(s)\left(R(s) - \hat{G}(s)U(s)\right) \Rightarrow [1 + C(s)\hat{G}(s)]U(s) = C(s)R(s) \tag{11}$$

From there:

$$\frac{Y(s)}{R(s)} = \frac{\hat{G}(s)e^{-\tau s}C(s)}{1 + C(s)\hat{G}(s)} \tag{12}$$

The poles of the system are the solutions of the equation $1 + C(s)\hat{G}(s) = 0$ (the numerator only adds the factor $e^{-\tau s}$ without changing the characteristic polynomial.

Theorem 1 has been proven.

□

Theorem 2 (On the stability under factored errors, small-gain condition): *Suppose the model error* $G(s) = \hat{G}(s)\big(1 + W_m(s)\Delta(s)\big)e^{-\tau s}$ *with acceptable uncertainty* $\Delta(s)$ *and* $\|\Delta\|_\infty \le 1$. *If* $\|W_m T_0\|_\infty < 1$, *then the Smith closed-loop system is stable against all* Δ *with* $\|\Delta\|_\infty \le 1$.

Proof. From the general Smith expression (similar to Theorem 1), but with a factorization error, we have:

$$Y(s) = \hat{G}(s)\big(1 + W_m(s)\Delta(s)\big)e^{-\tau s}U(s) \tag{13}$$

By the predictor formula:

$$Y_{est} = \hat{G}U + (Y - \hat{G}e^{-\tau s}U) = \hat{G}U + \hat{G}e^{-\tau s}W_m\Delta U \tag{14}$$

We have the error:

$$E = R - Y_{est} = R - \hat{G}U - \hat{G}e^{-\tau s}W_m\Delta U \tag{15}$$

Therefore:

$$U = CE = C(R - \hat{G}U - \hat{G}e^{-\tau s}W_m\Delta U) \tag{16}$$

$$(1 + C\hat{G})U + C\hat{G}e^{-\tau s}W_m\Delta U = CR \tag{17}$$

Multiply both sides of (17) by $\big(1 + C\hat{G}\big)^{-1}$ (It is valid because of the assumption A1):

$$U + \frac{C\hat{G}}{1 + C\hat{G}}e^{-\tau s}W_m\Delta U = \frac{C}{1 + C\hat{G}}R \tag{18}$$

According to assumption A2, we have the expression:

$$(I + T_0 e^{-\tau s}W_m\Delta)U = U_0; \quad U_0 := \frac{C}{1 + C\hat{G}}R \tag{19}$$

Thus, the condition for U (and the whole system) to be well-defined and stable for all Δ is that the matrix or linear combination $I + T_0 e^{-\tau s}W_m\Delta$ is invertible for all Δ in the norm set. By the small-gain theorem for linear feedback interfaces, a sufficient condition for all Δ with $\|\Delta\|_\infty \le 1$ not to lose invertibility is:

$$\|T_0 e^{-\tau s}W_m\|_\infty < 1 \tag{20}$$

Since $e^{-\tau s}$ has amplitude 1 on the imaginary axis $\big(|e^{-j\omega\tau}| = 1\big)$. This is equivalent to:

$$\|T_0 W_m\|_\infty < 1 \tag{21}$$

The above condition is sufficient for stability for the given factor error.

Theorem 2 has been proven.

Remark 1: The above condition is a standard small-gain condition widely used in robust control. It can be directly checked by frequency plotting (we can draw $|W_m T_0|$ and check $|W_m T_0| < 1$ for all ω), and is directly related to criteria such as peak $|T_0|$ around the bandwidth (if the peak is small, the error tolerance is larger).

Theorem 3 (On robustness to delay error - phase/magnitude condition): *Suppose the real object has a delay different from the delay $\Delta\tau$ used in the predictor, i.e., $G(s) = \hat{G}(s)e^{-(\tau+\Delta\tau)s}$. Let $\delta(j\omega) = e^{-j\omega\Delta\tau} - 1$. If $|\delta(j\omega)|\,) < 1$, then the Smith closed-loop system is stable to delay error $\Delta\tau$. Since $|\delta(j\omega)| = \left|e^{-j\omega\Delta\tau} - 1\right| = 2sin\left(\frac{\omega\Delta\tau}{2}\right) \leq \{2, \omega|\Delta\tau|\}$, a simple and conservative condition is that $|T_0(j\omega)| \, * \{2, \omega|\Delta\tau|\} < 1$.*

In addition, a practical approximation at the cutoff bandwidth ω_c gives us the interval condition: $\omega_c|\Delta\tau| < \varphi_m$ where φ_m is the phase margin of the pair $[C, \hat{G}]$.

Proof. We arrange:

$$G(s) = \hat{G}(s)e^{-\tau s}e^{-\Delta\tau s} = \hat{G}(s)e^{-\tau s}\left(1 + (e^{-\Delta\tau s} - 1)\right) \tag{22}$$

Set: $\delta(s) = e^{-\Delta\tau s} - 1$

Returning to Smith's equation (similar to that in Theorem 2), we get, after similar arrangements:

$$[I + T_0 e^{-\tau s}\delta(s)]U = U_0 \tag{23}$$

Since $e^{-\tau s}$ has units of amplitude, the small-gain condition for $I + T_0 e^{-\tau s}\delta$ to be invertible for all frequencies is

$$\|T_0\delta\|_\infty < 1 \tag{24}$$

It means:

$$|T_0(j\omega)| \, * \delta(j\omega) < 1 \tag{25}$$

Using trigonometric formula:

$$\left|e^{-j\omega\Delta\tau} - 1\right| = 2\left|sin\left(\frac{\omega\Delta\tau}{2}\right)\right| \tag{26}$$

and the inequality $|sin\, sin\, x\,| \leq min\,\{1, |x|\}$ we have $|\delta(j\omega)| \leq min\{2, \omega|\Delta\tau|\}$, which leads to the above conservation condition.

Regarding the phase margin relationship: The additional delay $\Delta\tau$ causes an additional phase lag $\omega\Delta\tau$. In order not to exceed the existing phase margin φ_m at the cutoff frequency ω_c (when $|\tau(j\omega_c| = 1)$, needs $\omega_c\Delta\tau < \varphi_m$. This is a practical approximation condition when $|T_0|$ reaches level 1 at ω_c.

Theorem 3 has been proven.

□

Remark 2: Theorem 1 explains that if the identification model is accurate, the Smith predictor eliminates the delay from the characteristic polynomial, thus making it easy to control and stable according to the design for$[C, \hat{G}]$. This is the basis for explaining why Case 2 in the simulation gives good results. Theorem 2 gives the robust condition (small-gain). To keep the system stable when the model is not completely accurate (factor error), it is necessary to reduce the peak $|T_0|$ around the bandwidth or reduce the relative error W_m. Empirically, if the identification is good (small W_m) and the design of C is reasonable (no large peak in T_0), the Smith predictor is still stable. Theorem 3 gives the condition with the delay error. If the delay error is small, satisfying the inequality with $min\{2, \omega|\Delta\tau|\}$ the system remains stable, the practical approximation is $\omega_c\Delta\tau < \varphi_m$.

3.3. Identification of the object model

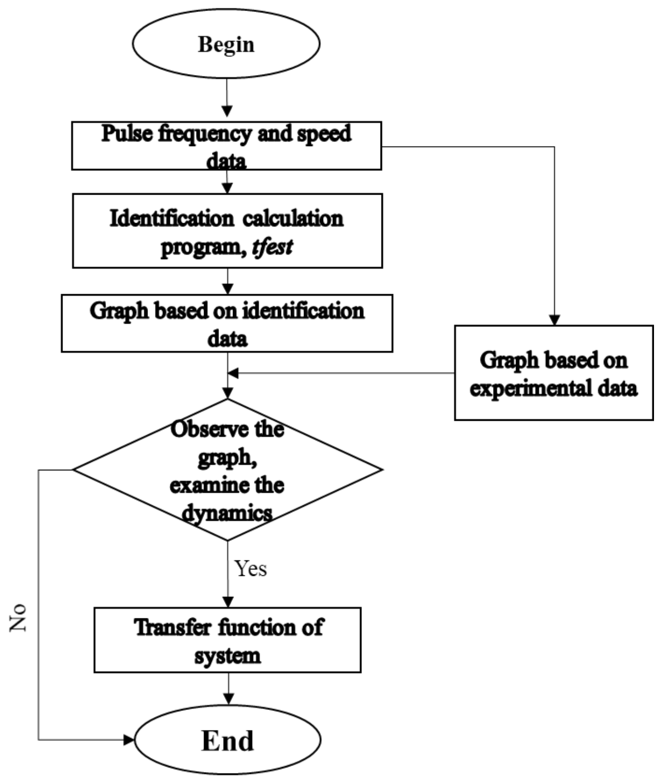

Figure 3. Flowchart of the system transfer function identification algorithm.

Next, to design the Smith delay compensator, the object model $\hat{G}(s)$ is identified using MATLAB Identification Toolbox.

In the continuous time domain, the transfer function $G(s)$ with a delay component describing the system's dynamics is a polynomial function of the following form:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + 1} e^{-\tau s}, n \geq m \qquad (27)$$

To design a highly efficient controller, feedback system modeling is very important, especially with the Smith predictor. Studies [7-9] have proposed various system identification methods such as frequency analysis, spectrum analysis, recursive least squares method, maximum reliability method, state variable filter-based method, or existing toolbox-based method (CONTSID-Garnier Toolbox, 2008; CAPTAIN-Young, 2009). This study makes full use of MATLAB's identification toolbox with the tfest function. The model identification algorithm is implemented as shown in Figure 3. The system is activated with the input being the control pulse frequency, and the output being the motor speed. Figure 4 depicts the distributed delay structure in the MATLAB/Simulink–OPC Server–PLC-based real-time control system. The overall delay is modeled as an equivalent pure delay resulting from computation, communication, execution, and sensing processes.

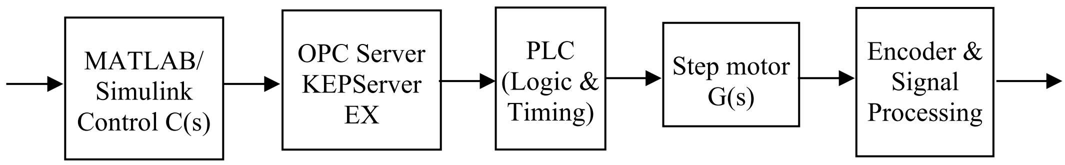

Figure 4. Distributed delay structure in the MATLAB/Simulink–OPC Server–PLC-based real-time control system.

To quantify the distributed delay in the MATLAB/Simulink–OPC–PLC architecture, each delay component was experimentally measured using timestamp-based logging and cycle-time monitoring within the PLC and Simulink environments, concretely:

τ_1: MATLAB/Simulink computation delay (0.05s); τ_2: OPC read/write communication delay (0.15s); τ_3: PLC execution and actuator interface delay (0.12s); τ_4: Sensor measurement and feedback delay (0.8s). The sum of the equivalent delay: $\tau = \tau 1 + \tau 2 + \tau 3 + \tau 4 = 0.4s$

The identified transfer function has the following form:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{0.7255}{s^2 + 15.19s + 77.4} e^{-0.4s} \qquad (28)$$

4. SIMULATION RESULTS

4.1. Semi-natural simulation model

The authors use MATLAB's OPC Toolbox library as an OPC Client to connect to the OPC Server (KEPServer Ex). This library consists of two main blocks, OPC Read and OPC Write, to read/write data to the Server. The sampling frequency for transmitting/receiving data between the MATLAB/Simulink Client and the OPC Server is 0.1s.

The OPC sampling period of 0.1s reflects a trade-off between communication load and control performance. Experimental results indicate that the identified total delay (approximately 0.4 s) remains within the stability margin derived in Section 3.3. Reducing the OPC sampling time further improves tracking performance but increases network and PLC computational load, which may limit scalability in large-scale industrial systems.

Table 1 lists the tagged objects for linking and embedding OLE. For the experiment, the authors use PLC FX3U, Omron CWZ6C-1000 encoder, 2-phase stepper motor with DM422C driver.

Table 1. Tag definitions in KEPServerEx.

Variables	Modbus address	Data type	Describe
X0	X000	Boolean	Phase A encoder
X1	X001	Boolean	Phase B encoder
X5	X005	Boolean	Bit allows speed measurement
X7	X007	Boolean	Bit reset
D0	D0000	DWord	Save the pulse value
D10	D0010	DWord	Save the pulse frequency value

4.2. Evaluate the quality of the system

Before implementing the Smith predictor, the nominal closed-loop system (without delay) was analyzed in the frequency domain to evaluate robustness margins. Using MATLAB frequency response analysis, the following results were obtained: gain margin (GM): 9.6 dB, phase margin (PM): 52.3°, and crossover frequency (ωc): 3.1 rad/s. These margins indicate that the nominal delay-free system possesses sufficient robustness prior to delay compensation.

Using PID App Tuner of MATLAB/Simulink to simulate the design of a PID controller for the identification model. The coefficients $(K_p;\ K_i;\ K_d) = (27{,}65; 243{,}85; 0)$ give the following system quality: Rise time 0.5 seconds; settling time 0.9 seconds and overshoot 1.61%.

Next, we experiment with 3 different cases:

Case 1: Feedback from the encoder in the feedback circuit, using the Smith predictor.

Case 2: Feedback from the identification in the feedback circuit, using the Smith predictor.

Case 3: Feedback from the encoder in the feedback circuit and no Smith predictor.

Figures 5, 6 and Table 2 show that in real-time control of the speed tracking system, to achieve a speed of 40 (rpm) when the feedback signal is the recognition model with the Smith predictor, it will give a better response. If we consider the raw signal from the encoder as a stable boundary state, there is almost no overshoot, the transition time is 1.5 seconds, and it starts to fluctuate around the set value with a steady-state error of 1.83% (0.7332 rpm); the error integral is 0.37 rpm².s (after 1.5 seconds). Then the stable pulse frequency reaches 4267Hz.

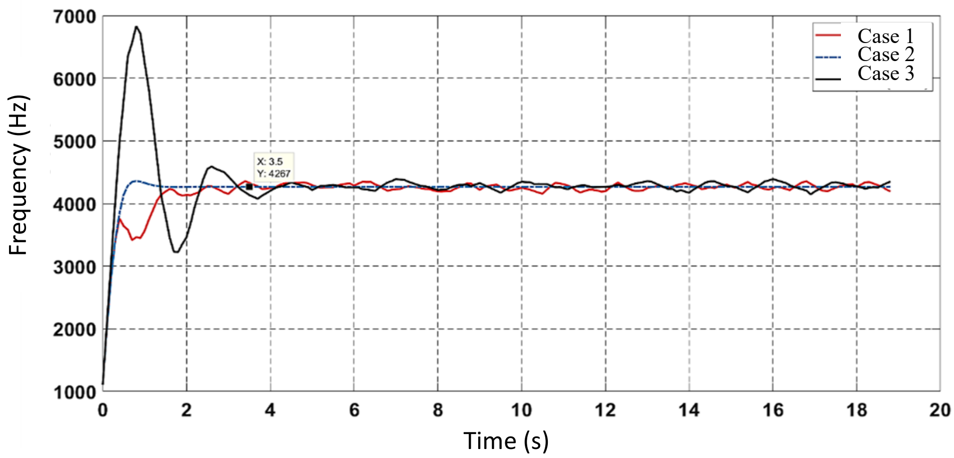

Figure 5. Control signal of the system.

The complementary sensitivity function $T(s)$ of the nominal system was computed. The peak value was found to be: $\|T(j\omega)\|_\infty = 1.38$. Considering that the identified model uncertainty level is approximately 10–15% in gain and time constant variation, the small-gain condition: $\|T(j\omega)\Delta(j\omega)\|_\infty < 1$, is satisfied since: $1.38 \times 0.15 = 0.207 < 1$. From Theorem 3, delay robustness depends on the available phase margin. The maximum tolerable additional delay can be approximated as: $\Delta\tau_{max} \approx \frac{\phi_m}{\omega_c}$

https://doi.org/10.65153/s2pkyh47

Substituting: $\Delta\tau_{max} \approx \frac{52.3^{\circ}\times\frac{\pi}{180}}{3.1} \approx 0.29\ s$

Since the experimentally observed delay variation was within $\pm 0.08\ s$, the system operates well inside the derived delay stability bound. This confirms that the closed-loop system remains robustly stable under the considered model uncertainty.

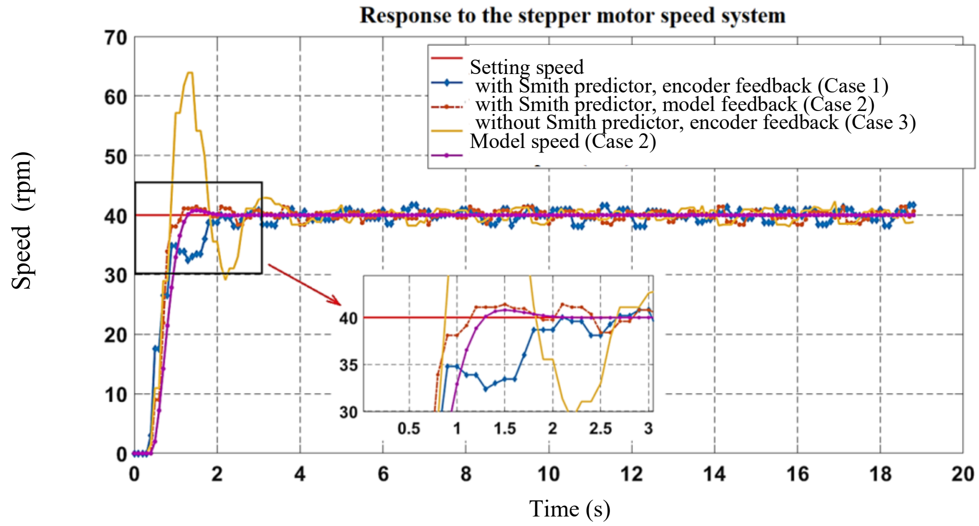

Figure 6. System response at setting speed V=40 rpm.

Table 2. System performance in comparison.

Scenario	Settling time (s)	Overshoot (%)	Steady-state error (%)	ISE (rpm².s) (after 1.5s)	RMSE (rpm) (after 1.5s)
Case 1	2.1	0	2.47	1068.2	2.40
Case 2	1.5	0	1.83	927.9	0.22
Case 3	4.6	59.75	3.07	1316.4	0.85

Each experimental case was repeated five times under identical operating conditions. This confirms that the reported results are consistent and reproducible within the considered PLC-based control environment. Table 2 shows that, in real-time control of the speed tracking system, to achieve a speed of 40 (rpm) when the feedback signal is the recognition model with Smith predictor (Case 2), it will give the best response compared to Case 1 (encoder + Smith) and Case 3 (no Smith). Concretely: Settling time is reduced to 1.5s compared to 4.6s (Case 3); overshoot is 0% compared to 59.75% (Case 3); the lowest settling error is 1.83%.

In addition, we also tested the robustness for the case of the recognition model having a deviation (K decreased by 10%, T increased by 15%). Through simulation, the tracking quality remains stable, but the settling error increases by ~1%. This shows that the method is relatively sustainable against model errors.

Thus, when the system is recognized relatively accurately, and the feedback signal in the feedback circuit from the model using the Smith predictor gives the best speed tracking quality of the stepper motor, as well as a stable control signal.

Compared with advanced delay compensation strategies such as Model Predictive Control (MPC), which require online optimization and higher computational resources, the Smith predictor provides a computationally efficient and easily implementable solution. This makes it particularly suitable for PLC-based industrial systems where memory and processing capability are limited.

5. CONCLUSION

This paper has presented a robust real-time speed tracking control approach for a two-phase stepper motor based on MATLAB/Simulink–OPC–PLC integration with Smith predictor delay compensation. An equivalent-delay modeling framework was developed to represent distributed communication and computation delays as a unified pure delay. Stability analysis showed that when the model is accurate, the Smith predictor eliminates the delay from the characteristic equation. Furthermore, robustness against model uncertainty and delay mismatch was verified using small-gain conditions and phase-margin-based delay tolerance analysis. Frequency-domain verification confirmed a gain margin of 9.6dB and a phase margin of 52.3°, with a complementary sensitivity peak $\|T\|_\infty = 1.38$, ensuring robust stability under 10–15% model uncertainty. The experimentally measured total delay of 0.40s remains within the derived delay tolerance bound. Semi-real-time experimental results demonstrate that the Smith predictor significantly improves speed tracking performance compared to conventional PID control. The configuration using model-based feedback within the Smith structure achieved the best performance, with a settling time of 1.5s, zero overshoot, and 1.83% steady-state error. The proposed approach provides a practical bridge between classical delay compensation theory and industrial PLC-based control implementation. It is particularly suitable for small-scale industrial automation systems and engineering education laboratories. Future work will focus on long-term validation under load disturbances and extension toward adaptive delay estimation strategies.

REFERENCES

[1]. Onur Ayan, D. Zeynep Demirez, H. Kagan Kiziloz, Gizem Inci, Seckin Isleyen, Semih Ergin. The Detection of Spoiled Fruits on a Conveyor Belt Using Image Processing Techniques and OPC Server Software, International Journal of Computational and Experimental Science and Engineering (IJCESEN) Vol. 4-No.1 (2018) pp. 11-15.

[2]. K.N Bagal, C.B.Kadu, P.S.Vikhe. PLC-Based Real-Time Process Control using SCADA and MATLAB. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Doi: 10.1109/ICCUBEA.2018.8697491.

[3]. Omur Aydogmus. A web-based educational tool using programmable logic controller-connected MATLAB-OPC server, International Journal of Electrical Engineering Education, Doi: 10.1177/0020720915571491.

[4]. Qingbao Huang, Qianzhong She, Xiaofeng Lin, Adaptive Fuzzy PID Temperature Control System Based on OPC and Modbus/TCP Protocol, 2010 2nd International Asia Conference on Infonnatics in Control, Automation and Robotics, Doi: 10.1109/CAR.2010.5456559.

[5]. Erwani Merry Sartika, T. Rudi Sarjono, Diki Dwi Saputra. Prediction of PID control model on PLC, TELKOMNIKA, Vol. 17, No.1, February 2019, pp.529-536. Doi: 10.12928/TELKOMNIKA.v17i1.11589.

[6]. K.Warwick, D.Rees. Industrial digital control systems (1988). Peter Peregrinus Ltd, London, United Kingdom.

[7]. Lennart Ljung. Experiments with Identification of Continuous Time Models, Proceedings of the 15th IFAC Symposium on System Identification, Saint-Malo, France, July 6-8, 2009.

[8]. Lennart Ljung. On the estimation of transfer functions. 1985 International Federation of Automatic Control, Vol. 21, No. 6, pp. 677-696, 1985.

[9]. Kyeong Ha Lee, Seung Guk Baek, Jong Yoon Choi, Jae Jun Park, Hyouk Ryeol Choi, Hyungpil Moon, and Ja Choon Koo. System Identification of Hydraulic System Using RLS Algorithm for Feedforward Position Control.

[10]. Vũ Quốc Huy, Nguyễn Văn Đức, Chu Đức Chình, Nguyễn Thị Lê Na. “Predictor control PI-Smith in the tracking system with time delay in image processing and communication”, Journal of Military Science and Technology, FEE-2021, pp. 37-43, 2021.

[11]. Vũ Quốc Huy, Nguyễn Thị Lê Na, "A flexible SCADA architecture for auto-inspection of the electric motorcycles based on OPC server and PLC", Journal of Military Science and Technology, No. 75, pp. 65-73, 2021.