Reinforcement Learning based Adaptive PID Controller for Non-minimum Phase Unstable Systems with delay

Sourabh Yadav
Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, India
sourabhy24@iitk.ac.in

Ketan P. Detroja
Electrical Engineering
Indian Institute of Technology Hyderabad
Telangana, India
ketan@ee.iith.ac.in

***Abstract*—Control of unstable systems is a pretty challenging task, since even a slight mismatch in controller gains can cause the system to go out of bounds. The control task becomes even more difficult if an unstable system has a zero in the right half-plane (RHP) of the complex s-plane. In this paper, a reinforcement learning (RL) based controller tuning method for non-minimum phase unstable systems is proposed. An adaptive PID controller is designed using RL and a modified Smith predictor structure is used to accommodate any delay in the system. The Deep Deterministic Policy Gradient (DDPG) algorithm has been employed along with a long short term memory (LSTM) layer in the actor as well as the critic network to design an RL-based adaptive PID controller. The proposed framework with the modified Smith predictor and RL based PID controller provide improved control performance. The performance has been evaluated against a state-of-the-art PID controller based on Integral Absolute Error (IAE) and Integral Square Error (ISE). Various case studies highlight the applicability of the proposed framework to a wide range of unstable and non-minimum phase systems.**

***Index Terms*—Reinforcement Learning, PID Controller, Adaptive Control, Unstable systems, Process Control, LSTM(Long-Short Term Memory)**

I. Introduction

The following types of systems are considered difficult to control for the reasons discussed here.

1) Unstable systems - The systems with RHP poles cause the system to become unstable, as RHP poles will contribute an exponentially increasing term to the system output, making the output of the system to increase without bounds. Due to unbounded output, unstable systems are difficult to control.
2) Systems with dead-time - The systems with transportation delay are also difficult to control as they limit the controller bandwidth and affect the closed-loop stability of the system.
3) Systems with RHP zeroes - the systems with an RHP zero, also known as non-minimum phase (NMP) systems, which introduces an inverse response in the system, are also difficult to control because RHP zero introduces phase lag into the system which, if not taken into account appropriately can cause the system states to become unstable.
4) Integrating process - designing a controller for an integrating process poses different kinds of difficulty as the controller's integrating action combines with the integrator function of the process which leads to excessive oscillations in the closed-loop system response.

In this manuscript, a controller design method is proposed for a class of systems which is a combination of the above four types of systems. A general Unstable NMP system with one zero and two poles can be represented as follows:

$$G = \frac{K(s - z_r)\prod(s + z_i)}{(s - p_r)\prod(s + p_i)} e^{-\theta s} \tag{1}$$

where, K is the process gain, z_r represents the RHP zero and p_r represents the RHP pole. There can be more than one zero and pole in the RHP but in this work unstable systems with only one RHZ and two RHP have been discussed. p_i and z_i are LHP poles and LHP zeroes, respectively. θ is the dead time of the plant model. The design procedure for a stable system is comparatively easy as the systems are BIBO (Bounded Input Bounded Output) stable and they can accept a wide range of controller gains without going unstable. Unstable systems that too non-minimum phase systems require careful consideration while designing a controller as they are very susceptible to control input changes and can go out of bounds and behave abruptly if proper care is not taken while designing the controller. There are several methods available for designing a PID controller for unstable systems. Few of them involve IMC (Internal model control) along with H_2 minimization [1] and [2], utilizing Taylor series expansion and then solving a set of linear equations [3] and frequency domain design method [4] for designing classical PID-Controller for Unstable systems with delay. Machine learning-based, in particular, reinforcement learning (RL) based methods [5] for designing controllers for Unstable systems without delay have been reported in the literature.

Traditional controller design methods (excluding learning-based approaches) typically rely on an explicit plant trans-

fer function, with controller parameters derived directly from the plant model. This tight coupling can lead to degraded performance when there is a plant-model mismatch. Reinforcement learning (RL) methods, in contrast, learn control policies by interacting with the environment—either real or simulated—without requiring an explicit analytical model. This decoupling has the potential to improve robustness to modeling errors. However, while RL provides flexibility and adaptability while giving improved performance in the presence of dead-time variations, it does not yet handle plant-model mismatch effectively and exhibits unstable responses under discrepancies in gain and time constant of plant. It is important to note here that this limitation arises due to the use of Smith predictor, which is required to compensate for the dead-time.
The proposed method does not require a plant model for designing controllers for delay-free systems and requires plant model information only for the implementation of Smith-predictor for systems with delay. Application of RL algorithms directly on systems with delay is a tough task, owing to the fact that the delay in systems introduces the same amount of delay in the outcome of an action taken by an agent during training. This causes a consequential decrease in the learning of the algorithm. This shortcoming in the RL training is the motivation behind using the Smith Predictor in the proposed work. Typically, the performance of any control scheme is measured based on its IAE (Integral Absolute Error) and/or ISE (Integral Squared Error) values.

$$IAE = \int |e|\, dt \tag{2}$$

$$ISE = \int e^2\, dt \tag{3}$$

The main contributions of this manuscript are as follows:

- Use of reinforcement learning in addition to Smith-Predictor to design an adaptive PID controller for NMP-Unstable systems with delay.
- LSTM-based actor-critic network along with early training stopping criteria which significantly reduces the training and convergence time.
- Design of a control objective relevant reward function that improves the transient as well as the steady-state response, simultaneously.

II. Preliminaries

A. Reinforcement Learning

Reinforcement learning (RL) is a subset of machine learning method that was introduced during 1950-60s by Richard Bellman. It is a feedback-based machine learning algorithm where an agent learns to behave in an environment by taking certain actions and then looking at the rewards produced by those actions. This process of taking action and checking rewards runs in a loop and hence the term feedback-based machine learning algorithm. The main benefit of employing reinforcement learning is that it can find solutions to complex problems quite comfortably which otherwise will require a lot of effort and control theory knowledge.

B. RL-based Adaptive Controller

A PID controller, the most common form of controller used in industries, contains three components: a proportional term K_p, an integral term K_i, and a derivative term K_d.
The K_p term takes into account the current error ($e(t)$), K_i term accounts for error summed up to a point ($\int e(t)$) while K_d term handles the error occurring in future by looking at the rate of change of error ($\Delta e(t)$). By combining these 3 components the control input (u) is obtained as:

$$u(t) = K_p e(t) + K_i \int e(t) + K_d \Delta e(t) \tag{4}$$

Adaptive controllers are not new and there are adaptive controller design methods available, such as model reference adaptive control (MRAC) [6] and fuzzy-logic based PID controllers [7]. An Adaptive PID controller applies different values of controller gains at different time instances contrary to a classical PID controller, which applies fixed values of gains. Due to added flexibility in design of an adaptive controller, it is expected that adaptive controllers should give better control and performance as compared to a traditional PID Controller. The proposed method utilizes the deep deterministic policy gradient (DDPG) algorithm to find the controller gains and then these gains are applied with the help of a trained DDPG agent.

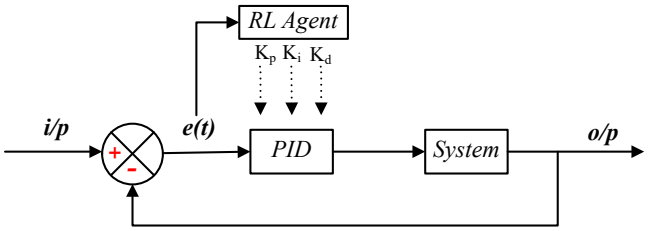

Fig. 1. RL-based Adaptive Controller

C. DDPG Algorithm

DDPG is known to be a model-free off-policy algorithm that simultaneously learns a policy and a Q-function. It utilizes the Bellman equation and off-policy data to learn the Q-function and then uses that Q-function to learn a policy. It is a deterministic algorithm meaning it gives a deterministic policy instead of a probability distribution of actions [8]. DDPG performs pretty well in continuous state and action spaces [9]. DDPG uses two deep neural network (DNN) networks, i.e., an actor and a critic network. The actor-network is used to approximate the policy of the agent, whereas the critic network approximates the value function to calculate future rewards.

D. LSTM Network

In addition to DNN, an LSTM(Long-Short Term Memory) layer is introduced in the actor and critic network. LSTM is a modification to the existing RNNs (Recurrent Neural Network) that can find long-term dependencies between sequential data that also solves the problem of gradient explosion and vanishing gradient associated with a typical RNN. LSTM is known to

provide better convergence performance for other applications such as speech recognition [10].

E. Smith-Predictor

The procedure of designing an RL-Adaptive controller on systems with delay is challenging because the dead time of the system introduces a delay in the system's response which inturn introduces the same delay in the observation of the states of the system. This causes a mismatch between state action pairs, which in turn deteriorates the learning of the agent. Fig. 2

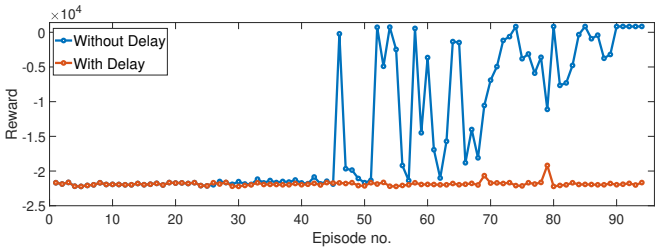

Fig. 2. training rewards for process $G_p = \frac{100(1-0.2s)}{(100s-1)(s-1)}e^{-0.2s}$

clearly depicts the effect of including dead time while training a controller for the system. The training without dead time converges in just 94 episodes, whereas the same system with delay struggles to learn anything during training. To address this problem, a Smith-Predictor is employed. The Smith predictor was proposed by $O.J.M.\ Smith$ in 1957 and falls into the category of a predictive controller, which is employed to reduce the effect of dead-time in any process systems. The main advantage of the Smith-Predictor is that it removes the delay from the characteristic equation of the system. Therefore one only has to design the controller for the delay-free part of the system. Carlos Mejía et al. have suggested a Modified form of the Smith Predictor [11], which gives robust performance even with variable delay.

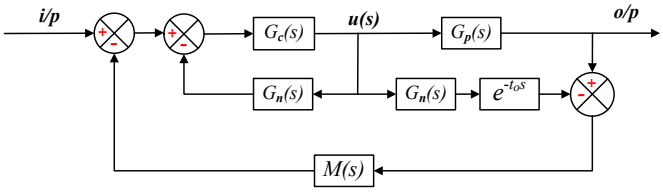

Fig. 3. Modified Smith Predictor

In Fig. 3, $G_c(s)$ is the proposed RL-based Adaptive controller, $G_p(s)$ is the plant with delay, $G_n(s)$ is estimated plant model without delay, t_o is the estimated delay of the plant $G_p(s)$ while $M(s)$ is a low pass filter designed as discussed in [2].

III. Proposed Controller Design Method

The training of the proposed controller through RL would require an environment consisting of the plant model, observable states, and a reward function that the algorithm will try to maximize. The observable states used are error (e_n) and change in error (Δe_n).

Observable state vector

$$x = \begin{bmatrix} e_n \\ \Delta e_n \end{bmatrix}$$

where, $e_n = y^{ref} - y_n$, y^{ref} is a reference signal, while y_n is the actual output at time instance n. The Δe_n is defined as $\Delta e_n = e_n - e_{n-1}$.

The reward function r, for the proposed RL based controller is chosen as $r = r_1 + r_2 + r_3$, where,

$$\begin{aligned} r_1 &= 8 - t_n|e_n| - e_n^2; \\ r_2 &= 1; && |e_n| - |e_{n-1}| < 0 \\ &= 0; && Otherwise \\ r_3 &= -10; && \begin{cases} y \le y_{min} \\ y \ge y_{max} \end{cases} \\ &= 0; && Otherwise \end{aligned}$$

The term $t_n|e_n|$ in r_1 will make errors at later time instances more penalizing than errors at the initial stages hence minimizing this term makes the system settle quickly. However, this may make the initial response sluggish. While minimizing the term e_n^2 prioritizes larger error more than smaller error hence minimizing this term leads to faster responses but with considerable oscillations for a longer duration of time. In this way, both terms complement each other's strengths and overcome each other's weaknesses. The r_2 term is used to provide a discrete reward, whenever the error at the current step is less than the error at the previous step. As suggested in [12], a positive constant term is added in r_1 to provide better sample efficiency and faster convergence by promoting conservative exploitation and balance between exploration-exploitation. The constant was chosen empirically, as discussed in [12].

Here, r_1 and r_2 shape the performance of the controller, whereas a saturation block has been used to confine the system's output such that $y_{min} \le y \le y_{max}$ using a penalty associated with r_3. The reward r_3, prevents the system from going unstable during training. The Hyper-parameters used for the algorithm are given in Table I The actor-network and

TABLE I
HYPER-PARAMETERS

Hyper-parameter	Value
actor learn-rate (α)	$1e$-5
critic learn-rate (β)	$1e$-4
Discount factor (γ)	0.99
Experience buffer length	$1e6$
Exploration noise (Variance)	0.3
Exploration noise decay	$1e$-5
Sample time	1
Max episode length	100
Mini batch size	64
Optimizer	Adam

critic-network architecture used are given in Fig. 4 and Fig. 5, respectively. Each fully connected layer (FC_i) has 150 neurons, and the LSTM layer has 20 neurons, add represents the addition layer whereas Relu represents the ReLU(Rectified Linear Unit) activation function. Along with the mentioned settings, the proposed method employs early-stopping criteria where the training is stopped if there is no significant improvement in the rewards for up to 10 episodes.

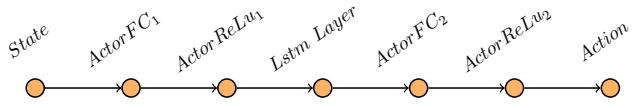

Fig. 4. Actor Network for Adaptive PID

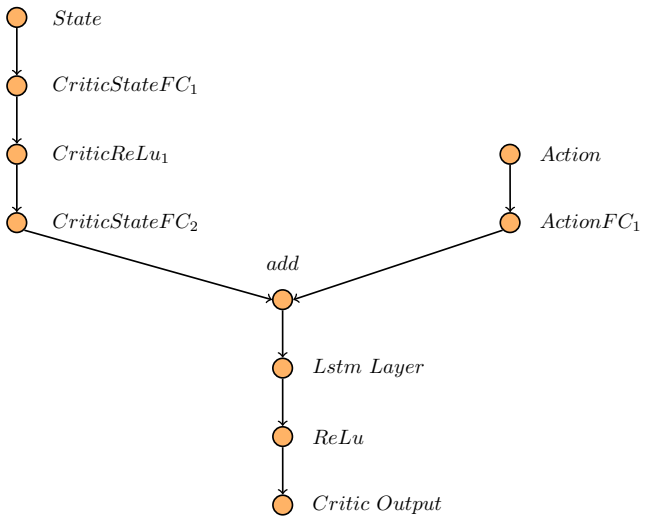

Fig. 5. Critic Network for Adaptive PID

IV. Simulation Examples

For evaluating the efficiency of the proposed RL-based Adaptive controller three second-order NMP Unstable Systems ($G_p(s)$) from [3] are selected and performance is compared based on IAE and ISE values. The first two systems are integrating processes, while the third system is the boiler steam-drum process.

A. Example-1

For the first simulation, the process given as $G_{p1} = \frac{100(1-0.2s)}{(100s-1)(s-1)}e^{-0.2s}$ is chosen. which is an approximation of an integrating process $\frac{e^{-0.2s}}{s(s-1)}$. The filter $M(s)$ for Smith-Predictor is selected as $M(s) = \frac{1.3s+1}{21.9s^2+5.0833s+1}$ as discussed in [2]. In Fig. 6 there is a significant reduction in overshoot along

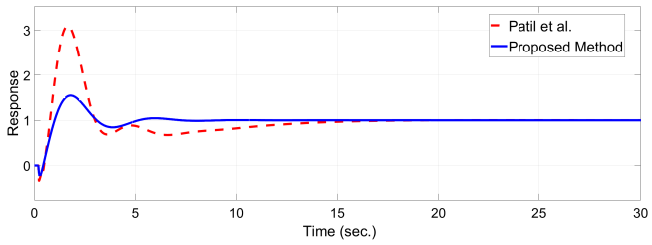

Fig. 6. Comparison of the controller performance for G_{p1}

with the system's settling time while the control effort required is reduced significantly when compared against the existing controller proposed by [3] (Fig. 7). While Fig. 8 depicts the controller gains applied via a trained agent at different instances while classical PID gains are K_p = 0.4451, K_i = 0.0853 and K_d = 1.9272 [3]. Table II draws the comparison of IAE & ISE

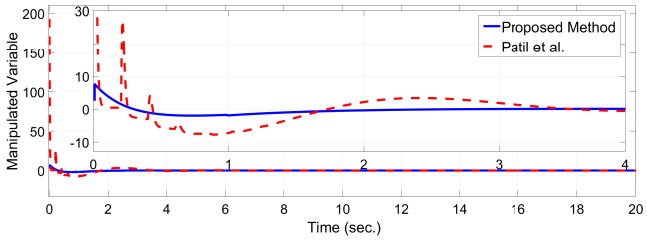

Fig. 7. Manipulated Variable (u)

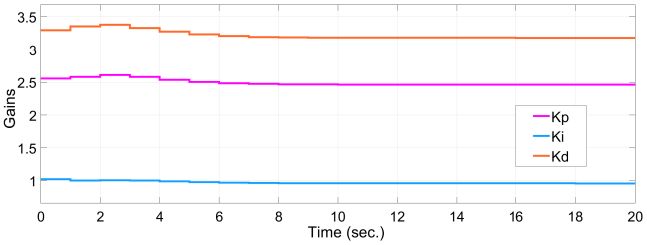

Fig. 8. Adaptive Controller gains

obtained with both the controllers. There is a significant improvement in both transient response and settling time through the proposed RL-based controller. Also, the proposed controller gives superior performance when implemented on the original integrating plant model as indicated in Fig. 9

TABLE II
Performance Comparison against classical PID

	Proposed Method	Patil et al.	Ghousiya et al.
IAE	1.72	5.6099	7.4258
ISE	1.014	5.517	7.0560

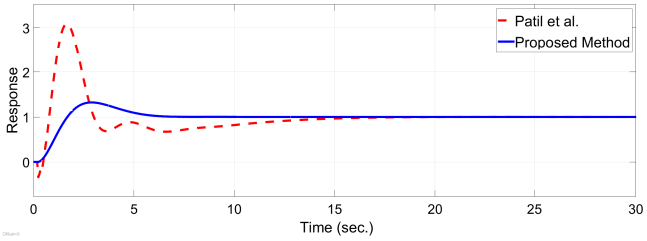

Fig. 9. Comparison of the controller performance on the original integrating system of *Example-1*

B. Example-2

For this simulation, the process transfer function $G_{p2} = \frac{51.83(1-0.4699s)}{(116.09s^2+98.8391s-1)}e^{-0.81s}$ is selected. While filter $M(s)$ is taken as $M(s) = \frac{1.984s+1}{8.3863s^2+8.0915s+1}$. The step response of the system along with the control effort required to control the system is shown below, in Fig. 10 and Fig. 11, respectively. In Fig. 10 even though there is an increase in the inverse response of the system the response is much quicker and the settling time is also less when compared to the classical PID controller. It is important to note that this improved performance is achieved

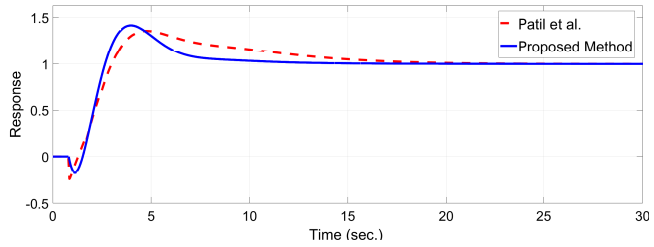

Fig. 10. Step response of process

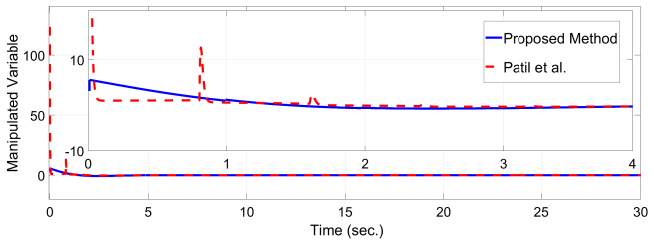

Fig. 11. Manipulated Variable (u)

with significantly less control effort (See Fig. 11). Fig. 12 shows the controller gains of the algorithm whereas gains used by [3] are $K_p = 0.9947$, $K_i = 0.1197$ and $K_d = 1.2314$. A performance measure of the proposed controller along with a comparison between both methods is shown in Table III. For this system,

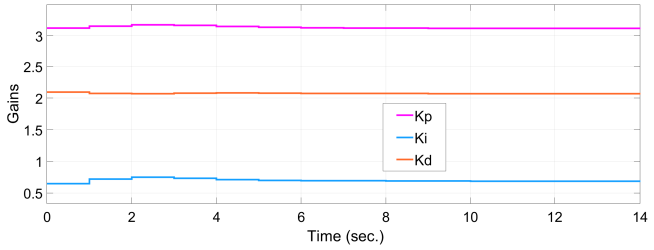

Fig. 12. Adaptive Controller gains

TABLE III
PERFORMANCE COMPARISON AGAINST CLASSICAL PID

	Proposed Method	Patil et al.	Ghousiya et al.
IAE	3.232	4.624	5.3601
ISE	2.162	2.5606	2.8307

a significant improvement in settling time through the proposed controller can be seen over the classical PID controller. Also, a decent reduction in controller effort is observed with the proposed RL-based controller.

C. Example-3

Here the process transfer function of the industrial boiler steam drum given as $G_{p3} = \frac{54.7(1-0.418s)}{(106.09s^2+98.94s-1)}e^{-0.1s}$ is selected [13]. Here the derivative filter for the proposed controller is chosen as $\frac{100s}{s+100}$. The filter $M(s)$ is taken to be $M(s) = \frac{0.85s+1}{2.5018s^2+3.4186s+1}$. Figure. 13 shows the training performance for 250 and 310 episodes, respectively. It is visible that after 250 episodes there are several reward collapses along with no significant increase in the rewards. Hence, with the proposed early stopping criteria there is no adverse impact on the performance. In Fig. 14 a slight increase in inverse

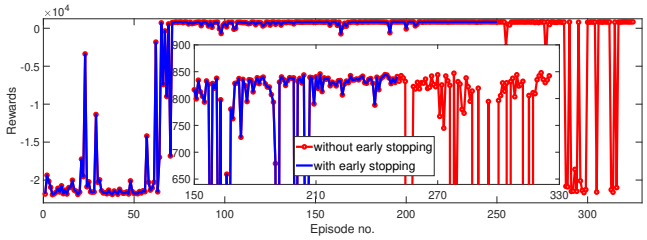

Fig. 13. Rewards for example-3 with and without early stopping

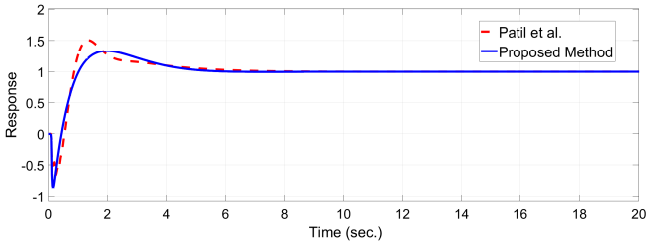

Fig. 14. Step response of process

response can be observed along with a reduction in settling time and at the same time the control effort Fig. 15 required is comparatively less than that of a traditional PID. The Proposed

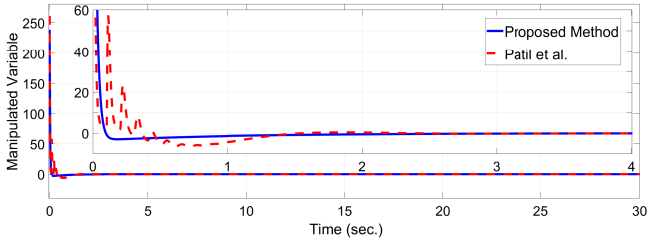

Fig. 15. Manipulated Variable (u)

Controller gains are shown in Fig. 16 while classical PID controller gains given in [3] are $K_p = 3.2306$, $K_i = 0.9215$ and $K_d = 2.5783$. The performance on basis of IAE & ISE values

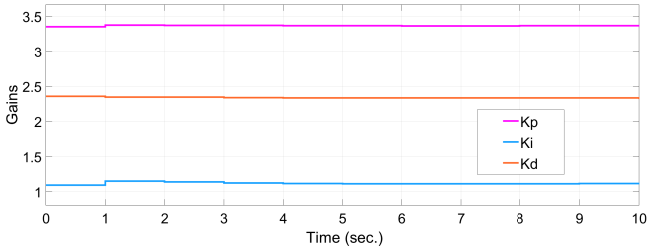

Fig. 16. Adaptive Controller gains

have been compared in TableIV. Here the proposed controller gives a similar performance to state of the art PID controller at a significantly less control effort.

TABLE IV
PERFORMANCE COMPARISON AGAINST CLASSICAL PID

	Proposed Method	Patil et al.	Ghousiya et al.
IAE	1.609	1.8646	1.9998
ISE	1.19	1.13	1.1954

V. ROBUSTNESS ANALYSIS

One of the major concerns regarding learning-based control strategies is whether or not they will give robust performance and how will they behave in the presence of external disturbances, and/or plant-model mismatch. In this section, the performance of the proposed control strategy has been discussed in the presence of external disturbance and in the presence of delay uncertainty. One of the main benefits of using the modified Smith predictor discussed in [11] is that it performs very well even when there is a variation in the dead time of the plant model since in practice the dead time keeps changing depending on the operating conditions and load on the plant. Table V demonstrates the performance of the proposed RL-based controller on varying dead time in contrast to the classical PID controllers. Again, the proposed

TABLE V
PERFORMANCE COMPARISON WITH DELAY UNCERTAINTY

		+40%		−40%	
Process	**Method**	**IAE**	**ISE**	**IAE**	**ISE**
$\frac{100(1-0.2s)e^{-0.2s}}{(100s-1)(s-1)}$	**Proposed**	**1.85**	**1.11**	**1.65**	**0.92**
	Patil et al.	12.05	14.83	5.16	3.76
	Ghousiya et al.	4.18	2.63	3.98	2.67
$\frac{51.83(1-0.4699s)e^{-0.81s}}{(116.09s^2+98.8391s-1)}$	**Proposed**	**3.91**	**2.62**	**2.63**	**1.73**
	Patil et al.	5.27	3.34	3.94	1.99
	Ghousiya et al.	5.95	4.34	6.00	4.11
$\frac{54.7(1-0.418s)e^{-0.1s}}{(106.09s^2+98.94s-1)}$	**Proposed**	**1.71**	**1.51**	**1.58**	1.41
	Patil et al.	1.95	1.55	1.65	**1.21**
	Ghousiya et al.	2.89	2.08	2.92	2.05

controller gives superior performance on the first two examples and comparatively similar performance on the third system. Fig. 17 shows the response to the introduction of an external disturbance $\frac{1}{2s+1}$ into plant output at 5 s. on G_{p1}.

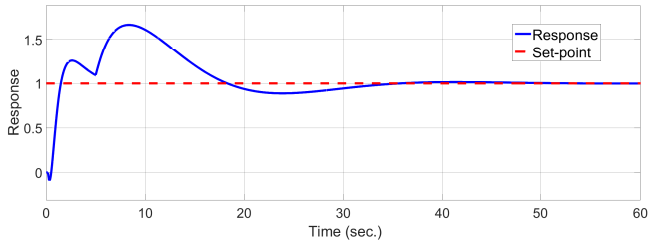

Fig. 17. Response in the presence of output disturbance on G_{p1}

The proposed controller can mitigate the effects of output disturbance on the plant efficiently.

VI. CONCLUSION AND FUTURE WORKS

An RL-based adaptive PID controller is proposed incorporating an actor-critic network based on LSTM for NMP unstable second-order systems. The controller obtained via the DDPG algorithm with an LSTM-based actor-critic network along with early training stopping criteria provides an improvement in performance over existing classical PID controllers with a significant improvement in transient response. The proposed adaptive controller gives superior performance even when subjected to external disturbances and variations in dead time. The proposed method successfully deals with dead-time uncertainty, proving system resilience. However, it is not effective in handling plant-model mismatches in other parameters. In future, a systematic study will be conducted to improve performance when there is some plant-model mismatch through proper training.

REFERENCES

[1] A. Anusha and A. S. Rao, "Design and analysis of imc based pid controller for unstable systems for enhanced closed loop performance," *IFAC Proceedings Volumes*, vol. 45, no. 3, pp. 41–46, 2012.

[2] K. Ghousiya Begum, A. Seshagiri Rao, and T. Radhakrishnan, "Enhanced imc based pid controller design for non-minimum phase (nmp) integrating processes with time delays," *ISA Transactions*, vol. 68, pp. 223–234, 2017.

[3] P. Patil, S. S. Anchan, and C. S. Rao, "Improved pid controller design for an unstable second order plus time delay non-minimum phase systems," *Results in Control and Optimization*, vol. 7, p. 100117, 2022.

[4] Z. Shafiei and A. Shenton, "Frequency-domain design of pid controllers for stable and unstable systems with time delay," *Automatica*, vol. 33, no. 12, pp. 2223–2232, 1997.

[5] T. Shuprajhaa, S. K. Sujit, and K. Srinivasan, "Reinforcement learning based adaptive pid controller design for control of linear/nonlinear unstable processes," *Applied Soft Computing*, vol. 128, p. 109450, 2022.

[6] H. Gai, X. Li, F. Jiao, X. Cheng, X. Yang, and G. Zheng, "Application of a new model reference adaptive control based on pid control in cnc machine tools," *Machines*, vol. 9, 2021.

[7] D. Somwanshi, M. Bundele, G. Kumar, and G. Parashar, "Comparison of fuzzy-pid and pid controller for speed control of dc motor using labview," *Procedia Computer Science*, vol. 152, pp. 252–260, 2019.

[8] T. Tiong, I. Saad, K. T. K. Teo, and H. b. Lago, "Deep reinforcement learning with robust deep deterministic policy gradient," *2020 2nd International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*, pp. 1–5, 2020.

[9] T. Lilicrap, J. Hunt, A. Pritzel, N. Hess, T. Erez, D. Silver, Y. Tassa, and D. Wiestra, "Continuous control with deep reinforcement learning," in *International Conference on Representation Learning (ICRL)*, 2016.

[10] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128*, 2014.

[11] O. C. Carlos Mejía, Danilo Chavez, "A modified smith predictor for processes with variable delay," *2019 IEEE 4th Colombian Conference on Automatic Control (CCAC)*, vol. 13, p. 6, 2019.

[12] H. Sun, L. Han, R. Yang, X. Ma, J. Guo, and B. Zhou, "Optimistic curiosity exploration and conservative exploitation with linear reward shaping," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022.

[13] M. Lee and M. Shamsuzzoha, "Pid controller design for integrating processes with time delay," *Korean Journal of Chemical Engineering*, vol. 25, pp. 223–234, 2008.