2024 American Control Conference (ACC)
July 8-12, 2024. Toronto, Canada

Investigating the Effectiveness of Reinforcement Learning in Closed-Loop Systems with Time Delays

Moh Kamalul Wafi, Milad Siami, Mario Sznaier

Abstract— **Data-driven controllers have gained prominence in diverse control applications, attributed to their inherent flexibility and adaptability to complex system dynamics. However, managing time delays in closed-loop systems remains a significant challenge in their deployment. These delays can arise from various sources, such as computational latency, actuator reaction time, and communication delays. Unaddressed, these time lags can induce system instability and degrade performance. This paper rigorously analyzes the impact of time delays on data-driven controllers and introduces methodologies to mitigate their adverse effects. Specifically, we explore the integration of the Smith predictor with Deep Reinforcement Learning (SP-DRL) to formulate a control law capable of effectively managing both time delays and system uncertainties, while ensuring robust performance. We demonstrate that this DRL-based framework, initially trained in stable environments, generalizes well to unstable systems. Our investigation delineates the scenarios conducive to the successful application of this approach and identifies factors influencing its effectiveness. To substantiate our findings, we present a case study involving a first-order delayed linear system with nonlinear actuation modules. Numerical simulations are employed to compare the robustness of SP-DRL scheme against the DRL standalone and the classical controls, such as PID and Linear Quadratic Regulator (LQR), in the presence of delays.**

I. Introduction

The study of learning-based control has made significant strides in recent years, particularly with the advent of deep-learning breakthroughs and the exploration of semi-supervised reinforcement learning (sSRL). While the discrete Deep Q-Network (DQN) featuring infinite observation spaces and finite actions has achieved success, continuous actions are more desirable in closed-loop control domains [1]. Continuous deterministic-actor $\pi_\theta(x)$ approaches have shown promising performance in deep reinforcement learning (DRL) [2], [3], challenging classical methods under certain constraints. However, the performance of modern DRL in practical systems with dead-time and distributed (rather than lumped) time delays remains uncertain. Moreover, it is of interest to investigate how DRL trained in a stable environment adapts to dynamic changes in non-minimum phase unstable systems.

This material is based upon work supported in by grants ONR N00014-21-1-2431, NSF 2121121, NSF 2208182, the U.S. Department of Homeland Security under Grant Award Number 22STESE00001-03-02, and by the Army Research Laboratory under Cooperative Agreement Number W911NF-22-2-0001. The views and conclusions contained in this document are solely those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security, the Army Research Office, or the U.S. Government.

The authors are with the Department of Electrical & Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mails: {wafi.m, m.siami, m.sznaier}@northeastern.edu.

The concept of addressing time-delay using the Smith Predictor (SP) control dates back to 1950s. This technique was developed to mitigate the impact of delay on closed-loop control designs by employing a predictive model to anticipate the system's delayed output and compensate for it in the control signal [4]. This approach becomes particularly challenging and intriguing when systems involve integrative uncertainty exacerbated by unknown distributed delays. By incorporating modified techniques explored in [5], [6], the control loop can be designed using traditional methods like Proportional-Integral-Derivative (PID) controllers and LQR without explicitly accounting for the delay. Consequently, SP allows for a more straightforward control design and improved performance. Incorporating DRL into the control loop can further enhance the system's performance by adapting the control signal based on its current state.

Assuming a system featuring states, actions, and rewards, DRL can be formulated as a Markov decision process (MDP), utilizing single or combined actor-critic methods, depending on the system. However, this typically assumes that the instantaneous state is updated following the action taken or vice versa [7]. To address cases with delays, [8] proposed Markov control model by augmenting the states and the preceding actions while [9] alleviated the lumped delay problem with stochastic bounded delays, considering no actions if the delays exceed the bounds. Suggestions from previous studies [10], [11] indicate the need to compensate for enlarged states, which may lead to increased computation, memory usage, and convergence time. However, [8]–[11] focus on limited discrete actions, leaving the enigmatic problems of handling time delays in infinite continuous state and action domains unresolved, even without delays [12].

In this paper, we present first the impact of delay to stability limits and the relaxation of Smith control to cope with the delays, considering the plant and the delay mismatch. We then deliver a comprehensive analysis of two control method scenarios, one with and one without the inclusion of SP, in industrial plant with nonlinear actuation. Our investigation compares classical controls with modern DRL approach, focusing on the application of DRL-trained model in both stable and unstable systems featuring a single delay parameter. Furthermore, we explore the effectiveness of the combined SP in mitigating the effects of delays by studying the Lyapunov stability, the small gain theorem, and the gap metric, where values deviate from the benchmark stable system. Finally, we analyze the performance according to the feedback of the systems, assuming the perfect matching, the plant and the delay mismatch.

979-8-3503-8265-5/$31.00 ©2024 AACC

Authorized licensed use limited to: Northeastern University. Downloaded on September 22,2024 at 19:40:53 UTC from IEEE Xplore. Restrictions apply.

II. Preliminaries and Problem Formulations

In this section, we present a linear time-invariant (LTI) system with a single lumped delay $\zeta > 0$ in order to encapsulate various real-world practical and industrial scenarios. The delay occurs within the loop at the input with initial condition $x_0(t) = \vartheta(t), t \in [-\zeta, 0]$ in which $\vartheta : [-\zeta, 0] \to \mathbb{R}$ or simply $\vartheta \in \mathcal{C}([-\zeta, 0], \mathbb{R})$. The system is described as:

$$\dot{x}(t) = Ax(t) + Bu(t-\zeta), \quad y(t) = Cx(t) + Du(t), \quad (1)$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $y \in \mathbb{R}^p$ define the states, control signals, and the outputs, while $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, and $D \in \mathbb{R}^{p\times m}$ denote constant real matrices. When a time delay is introduced, the stability of the delayed closed-loop system cannot be guaranteed even if the non-delayed system is stable. Let us consider the transfer function $P(s)$ and a single lumped delay defined by the transfer function $e^{-\zeta s}$ from (1), the closed-loop characteristic equation, given a control $C(s)$, results in $\psi(s)$, which then can be decoupled into $\psi(s) = \psi_d(s) + \psi_n(s)e^{-\zeta s}$. Thus, we have:

$$P(s) = G_p(s)e^{-\zeta s}, \quad (2)$$

and to examine stability of such systems, the approximations are frequently employed to replace the pure delay ζ, e.g.: the $z-$domain $e^{-\zeta s} = z^{-nT}$, the Padé approximation, i.e., the first-order $e^{-\zeta s} = (2-\zeta s)/(2+\zeta s)$, or the Taylor series $e^{-\zeta s} \approx 1 - \zeta s + f_n(\zeta s)$, where $f_n(\zeta s)$ is the function of the $n-$term series. However, the Taylor series approach results in infinite poles. Assuming a tiny delay ζ where $\zeta s \ll 1$, by the first two terms of the series, $e^{-\zeta s} = 1 - \zeta s$, the system $P(s)$ in (2) yields in,

$$P(s) = \frac{ke^{-\zeta s}}{\tau s + 1 + ke^{-\zeta s}} \equiv \frac{ke^{-\zeta s}}{(\tau - k\zeta)s + (k+1)} \quad (3)$$

reaching the boundary of $-1 < k < \tau/\zeta$ for Routh-Hurwitz stability. Additionally, using the three terms approximation of the series and the Padé approximation, the system $P(s)$ can be constructed as follows,

$$P(s) = \begin{cases} \dfrac{2ke^{-\zeta s}}{k\zeta^2 s^2 + 2(\tau - k\zeta)s + 2(k+1)}, & \text{Taylor} \\ \dfrac{k(2-\zeta s)}{\tau\zeta s^2 + (2\tau + \zeta - k\zeta)s + 2(k+1)}, & \text{Padé} \end{cases}$$

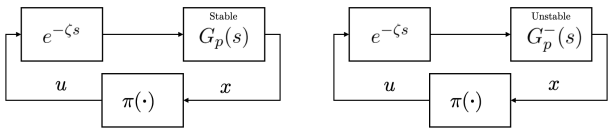
with $0 < k < \tau/\zeta$ and $-1 < k < 1 + 2\tau/\zeta$ for the stability criteria, in turn.

Remark 1. *Due to approximations, the stability limits diverge, and according to the Nyquist criterion, the stability of $P(s)$ comprises $-1 < k < \sqrt{1+\Omega^2}$, while the value of Ω depends on the behavior of ζ and τ.*

Furthermore, we intend to design the control input $u(t)$ with the DRL in order to minimize the cost function:

$$\min_{u\in\mathcal{A}, \forall t\in[t_0, t_0+T]} \int_{t_0}^{t_0+T} c(\mathbf{x}, u, t)\, dt, \quad \text{subject to (1)} \quad (4)$$

where $\mathcal{A}$ denotes the space of actions and c is a piecewise bounded continuous function in $(\mathbf{x}, u)$. Fig. 1 illustrates the feedback control design, in which the goal is to learn the feedback policy $u = \pi_\theta(\mathbf{x})$ where θ shows the trained weights of actor in DRL. For a time instant $t_i, \forall i \in [t_0, t_{\max}]$, an observation $\mathbf{x} \in \mathcal{S}$ is obtained, the deterministic control action $u \in \mathcal{A}$ is applied, and the associated cost $c(\mathbf{x}, u, t)$ is computed. Those three result in a set of observation-input-cost tuples $\mathbb{T} = \{\mathbf{x}_i, u_i, c_i\}, \forall i$. The DRL algorithm $\pi(\mathbf{x})$ is trained on the stable system $G_p(s)$. Subsequently, the trained DRL is analyzed in environments beyond its training conditions, specifically for non-minimum phase systems with time delay ζ and unstable systems $G_p^-(s)$. This is accomplished by converting a few stable modes of $G_p(s)$ into unstable ones, by altering the sign of their real parts.



(a) Training Environment **(b)** Implementation

Fig. 1: (a) The DRL algorithm is trained on the stable system $G_p(s)$ using the state $\pi(x)$. (b) The pre-trained DRL algorithm is evaluated on the unstable system $G_p^-(s)$.

III. The Smith Predictor Revisited

To compensate the delay ζ, the Smith predictor algorithm is proposed, as depicted in Fig. 2a. The terms of ref $\in \mathbb{R}^p$ and $u \in \mathbb{R}^m$ denote the reference and the control input while $\{y, y_n, y_t\} \in \mathbb{R}^p$ define the true output, the nominal-model output and the predicted output in turn with disturbance d. The idea is to construct the model $P_n(s) \coloneqq G_n(s) \times T_n(s)$ to separate the information of the plant and its delay term. The algorithm then subtracts the prediction y_t, considering the estimated delay T_n, from the actual output y obtained from the noisy process $P(s) \coloneqq G_p(s) \times T_p(s)$. The transfer function of the system in Fig. 2b with disturbance-free $D(s) = 0$ is formulated as $Y_r(s) \coloneqq Y(s)/R(s)$, effectively removing the delay-term. Additionally, the output-disturbance perfect-matching system is given as $Y_d(s) \coloneqq Y(s)/D(s)$,

$$Y_r(s) = \frac{CP}{\Phi_n + CP}, \qquad Y_d(s) = \frac{P\Phi_n}{1 + CG_n}, \quad (5)$$

where $\Phi_n = 1 + CG_n(1 - T_n)$ and $Y_r(s)$ will reduce to $CP/(1 + CG_n)$ if $P_n \coloneqq P$.

Lemma 1. (Smith-control stability)**.** *Given a Smith-control in a closed-loop feedback system in Fig. 2b as $C_\Phi = C/\Phi_n$, the instability condition yields in $CG_n(1-T_n) = -1$ where $T_n \coloneqq e^{-\zeta_n s} = (1+CG_n)/CG_n =: \Pi_n$ such that $\exists \Pi_n \neq 1$. Therefore $\lim_{t\to\infty} C_\Phi \to 0$.*

Nonetheless, the inner-loop stability of the Smith-control is independent to the overall closed-loop system. Now, if the predicted system (G_n, T_n) fails to match the true system $(G_p = G, T_p = T)$, then there exists deviations of the system ΔG and the pure delay $\Delta\zeta_p$ as follows,

$$G_n \coloneqq G + \Delta G, \quad T_n \coloneqq e^{-s(\zeta_p + \Delta\zeta_p)} = T\Delta T, \quad (6)$$

where using (5), Fig. 2b is now arranged as in Fig. 2c and $G_nT_n = GT\Delta T + \Delta GT\Delta T$ in which ΔG is denoted as

Authorized licensed use limited to: Northeastern University. Downloaded on September 22,2024 at 19:40:53 UTC from IEEE Xplore. Restrictions apply.

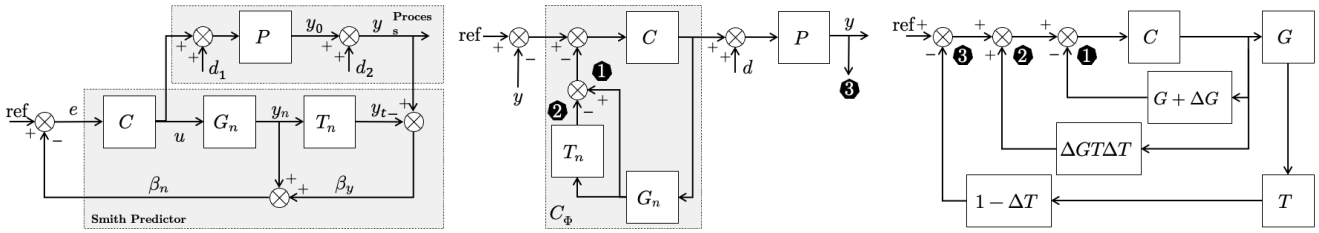


Fig. 2: (a) General Smith predictor with control $C(s)$ and the predicted system G_nT_n of the plant P; (b) Smith predictor reconstruction of (a) as C_Φ; and (c) Smith predictor reconstruction illustrating effects of model and plant mismatches.

$\Delta G \coloneqq C_d(sI - A_d)^{-1}B_d + D_d$. The failures of capturing the true dynamics such as delay ζ, high frequency gain, and time constant τ cause the followings: first, supposed the dynamics of the predicted model G_n is different from that of G_p, $\Delta G \neq 0$, while the time delay is on par $T_n = T_p, \Delta T = 1$, the lack of predicting the gain G_n leads to instability due to a positive feedback of ΔGT; second, with the perfect matching dynamic $\Delta G = 0$ and lack of predicting the delay $\Delta T \neq 1$, the inner feedback control brings $(1 - \Delta T)$, leading to instability. These mismatches should be in certain boundary as a function of the characteristic equation $\psi(s)$ so that the system is stable.

Supposed the plant be $G_p = 1/(\alpha s + 1)$, the predicted model be $G_n = 1/(\beta s + 1)$, and the controller be $C = \gamma(0.5s + 1)$ where $\gamma \geq \beta \geq \alpha \geq 1$ along with $\zeta_n = \zeta_p$. Using Padé approximation for $T_n = T_p$ in (5) with the characteristic polynomial $\psi_g(s) \coloneqq \sum_{i-1}^{3} f_i s^{i-1}$ and the parameters (α, β, γ), the system is stable. Furthermore, if the dynamic is similar $G_n = G_p \coloneqq 1/(\alpha s + 1)$ with control $C = \gamma(0.5s + 1)$, the stability depends on the relationship between T_n and T_p in denominator of $(T_p - T_n)$, in which for $\zeta_n \ll \zeta_p$ the system is unstable. To end this relaxation, we bring the positive real lemma [13] to characterize the transfer function which is used together with the passivity.

Proposition 1. (Kalman-Yakubovich-Popov)**.** *Let* (1) *be* (2) *where $\zeta = 0$ and G_p is minimum phase. G_p is strictly positive real iff $\exists Q \coloneqq Q^\top > 0$, Γ, W, and $\epsilon > 0$, such that*

$$QA + A^\top Q = -\Gamma^\top \Gamma - \epsilon Q$$

where $QB = C^\top - \Gamma^\top \xi$ and $\xi^\top \xi = D + D^\top$. For a positive real of G_p, a constant ϵ equals to zero, $\epsilon \coloneqq 0$.

IV. Integrated Smith-Predictor and Deep Reinforcement Learning (SP-DRL)

In this section, we introduce a linear time-invariant (LTI) system described in (2), with a feedback deep deterministic policy gradient (DDPG) control $u = \pi(\cdot)$ using off-policy data and Bellman equation. Here, the state $x \in \mathbb{R}^n$ is also considered as the output $x \coloneqq y$. This DDPG learns the $Q-$function $Q_\eta(\mathbf{x}, u)$ as critic and uses that to learn the policy as actor. The actor-critic DDPG control conducts four feed-forward neural networks (FFNN) with parameters η, θ and their target networks η^t, θ^t using replay buffer $\mathbb{D}_i(\mathbf{x}, u, r, \mathbf{x}') \coloneqq \mathbb{D}_i(\cdot)$ as shown in Fig. 3. Note that, there are three times applied, where each episode $e_p \coloneqq 1 \to t_c$, there are $j \coloneqq 1 \to t_b$ time steps consisting of $i \coloneqq 1 \to t_a$ iterations. Indeed, the method interleaves learning using the two target networks, in which they initially copy the weights of actor-critic networks, $\eta^t \leftarrow \eta$ and $\theta^t \leftarrow \theta$. It then samples a batch $\mathbb{B}_j$ from the buffer $\mathbb{D}_j$ and updates the policy, the $Q-$function, and their target networks with gradient descent $\mathbb{L}$ and gradient ascent Π as,

$$\begin{aligned} \mathbb{L} &= \frac{1}{\mathbb{B}} \sum_{(\cdot)\in\mathbb{B}} \underbrace{[r + \gamma_t \max Q_{\eta^t}(\mathbf{x}', \pi_{\theta^t}(\mathbf{x}'))}_{\mathbb{Y}} - Q_\eta(\mathbf{x}, u)]^2 \\ \Pi &= \frac{1}{\mathbb{B}} \sum_{\mathbf{x}\in\mathbb{B}} Q_\eta(\mathbf{x}, \pi_\theta(\mathbf{x})) \end{aligned} \tag{7}$$

such that,

$$\begin{aligned} \eta_{j+1} &\leftarrow \eta_j + \alpha_\eta \nabla_\eta \mathbb{L}, \quad \eta^t_{j+1} \leftarrow \rho\eta^t_j + (1-\rho)\eta_{j+1} \\ \theta_{j+1} &\leftarrow \theta_j + \alpha_\theta \nabla_\theta \Pi, \quad \theta^t_{j+1} \leftarrow \rho\theta^t_j + (1-\rho)\theta_{j+1}. \end{aligned} \tag{8}$$

where γ_t denotes the discount factor, $\rho = \{0, 1\}$ is the hyper parameter, whereas α_θ and α_η are the learning rates of actor and critic in turn. Moreover, r defines the reward of the form, $r = 0.1$ for $|e(t)| \leq 0.1$ and $r = 1/|e(t)|$ otherwise, where $e(t)$ is the error between noisy state and the reference.

The actor takes the observations of the state $x(t)$, the error $e(t)$ and the integral of the error $e_I(t)$ with ℓ_a-layer as,

$$\mathbf{x}(t) = \begin{bmatrix} x^\top(t) & e^\top(t) & e_I^\top(t) \end{bmatrix}^\top \in \mathcal{S}$$

while the critic relies on two inputs, the observations $\mathbf{x}(t)$ and control action $u(t)$, from the batch $\mathbb{B}_j$ under ℓ_c-layer. In critic, the observation comprises $\ell_{co}-$layer and the action constitutes $\ell_{ca}-$layer before being added with $\ell_{cs}-$layer. The value of ℓ_c-layer is defined as $\ell_c \coloneqq \max(\ell_{ca}, \ell_{co}) + \ell_{cs}$. The FFNN actor with ℓ_a-layer is denoted as,

$$\phi_0(t) = \mathbf{x}(t) \tag{9a}$$

$$\phi_k(t) = \Lambda_k \left(W_k \phi_{k-1}(t) + b_k\right), \quad \forall k = 1, \ldots, \ell_a \tag{9b}$$

$$u(t) = W_{\ell_a+1}\phi_{\ell_a}(t) + b_{\ell_a+1} \coloneqq v_{\ell_a+1} \tag{9c}$$

where $\phi_k \in \mathbb{R}^{z_k}$ denote the output activations from the $k-$th layer with $z_0 = n$. The weight matrix $W_k \in \mathbb{R}^{z_k \times z_{k-1}}$ and the bias $b_k \in \mathbb{R}^{z_k}$ handles the linear operations described as $v_k \in \mathbb{R}^{z_k}$ containing z_k neurons, running the calculation of

$$v_k(t) \coloneqq \left[v_1(t), \ldots, v_{z_k}(t)\right]^\top = W_k \phi_{k-1}(t) + b_k. \tag{10}$$

Authorized licensed use limited to: Northeastern University. Downloaded on September 22,2024 at 19:40:53 UTC from IEEE Xplore. Restrictions apply.

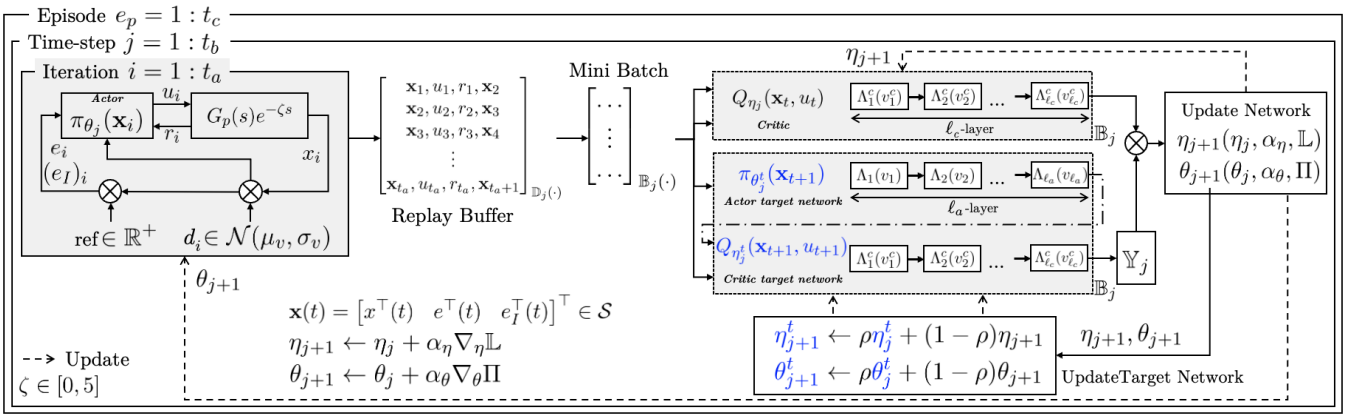


Fig. 3: Implementation of Deep Deterministic Policy Gradient (DDPG).

The non-linear operations of activation functions $\Lambda_k : \mathbb{R}^{z_k} \to \mathbb{R}^{z_k}$ is a row matrix and Λ_k is represented as the element-wise $\Lambda_k(v_k) \coloneqq [\lambda(v_1), \cdots, \lambda(v_{z_k})]^\top$ where λ is the chosen activation function. The size of the last linear operation have the same that of control signal u, $v_{\ell_a+1} \in \mathbb{R}^m$. Therefore, the actor collects the values of the linear $v_q \coloneqq [v_1, \cdots, v_{\ell_a}] \in \mathbb{R}^{n_q}$, $\Lambda_q(v_q)$, and the non-linear $\phi_q \coloneqq [\phi_1, \cdots, \phi_{\ell_a}] \in \mathbb{R}^{n_q}$ where $n_q \coloneqq \sum_{\ell_a} z_k$, resulting $u(t)$ to the plant $P(s)$.

With the same procedures, the critic FFNN with ℓ_c-layer from the batch $\mathbb{B}_j$ is also denoted as,

$$\varphi_0(t) = \begin{bmatrix} u(t) & \mathbf{x}(t) \end{bmatrix}^\top \tag{11a}$$

$$\begin{bmatrix} \varphi_{k^1}^1(t) \\ \varphi_{k^1}^2(t) \end{bmatrix} = \begin{bmatrix} \Lambda_{k^1}^1 \\ \Lambda_{k^1}^2 \end{bmatrix} \left(\begin{bmatrix} W_{k^1}^1 & 0 \\ 0 & W_{k^1}^2 \end{bmatrix} \varphi_{k^1-1}(t) + \begin{bmatrix} b_{k^1}^1 \\ b_{k^1}^2 \end{bmatrix} \right)$$

where $k^1 = 1, \ldots, \max(\ell_{ca}, \ell_{co})$. Supposed $\ell_{ca} > \ell_{co}$, then the calculation from $\ell_{co}+1$ to ℓ_{ca} works only for $u(t)$. The addition of the two inputs follows till ℓ_{cs}, therefore

$$\varphi_{k^2}^3(t) = \varphi_{\ell_{ca}}^1(t) + \varphi_{\ell_{co}}^2(t) \tag{12a}$$

$$\varphi_{k^3}^3(t) = \Lambda_{k^3}^3 \left(W_{k^3}^3 \varphi_{k^3-1}(t) + b_{k^3}^3 \right) \tag{12b}$$

$$\varphi_Q(t) = W_{\ell_{cs}+1}^3 \varphi_{\ell_{cs}}(t) + b_{\ell_{cs}+1}^3 \coloneqq v_{\ell_{cs}+1}^c \tag{12c}$$

where $k^2 \coloneqq \max(\ell_{ca}, \ell_{co}) + 1$ and $k^3 = k^2, \ldots, \ell_{cs}$. For simplicity, in Fig. 3, we name $\Lambda_k^c = \Lambda_{k^1}^1 = \Lambda_{k^1}^2 = \Lambda_{k^3}^3$ and it acts as element-wise with associated v_k^c, where subscript k works for the respected k^1, k^2, and k^3, resulting $\Lambda_k^c(v_k^c) \coloneqq \left[\lambda^c(v_1^c), \cdots, \lambda^c(v_{z_k}^c)\right]^\top$. Indeed, the neural networks of the target networks for actor and critic follow the associated processes, having the difference to the inputs with $(\mathbf{x}', u')$ as in (7). If the optimal values (x^*, u^*) satisfy (2), then the state x^* could be propagated via FFNN to reach the equilibrium values v_k^*, w_k^* for the inputs/outputs of every activation function, resulting $(v_q, w_q) = (v^*, w^*)$ [14]. This paper shows a healthy stable-system training in (2) given to DRL and see how it performs beyond the environment

The stability conditions are discussed for perfect matching ($\Delta G = 0$, $\Delta T = 1$), plant mismatch ($\Delta G \neq 0$), and delay mismatch ($\Delta T \neq 1$). Regarding the perfect matching, we establish the connection between the passivity of plant G and Lyapunov stability, drawing on [13], [15], [16].

Lemma 2. (Perfect matching). *If $G_p = G_n$, $\zeta_p = \zeta_n$, then the feedback dynamic is $G_n \coloneqq G$. Given the storage function $V(x) = \frac{1}{2}x^\top Q x$ the stability is written as,*

$$\begin{aligned} \pi_\theta^\top y - \frac{\partial V}{\partial x}\dot{x} &= \pi_\theta^\top (Cx + D\pi_\theta) - x^\top Q(Ax + B\pi_\theta) \\ &= \pi_\theta^\top Cx + \frac{1}{2}\pi_\theta^\top (D + D^\top)\pi_\theta \\ &\quad - \frac{1}{2}x^\top (QA + A^\top Q)x - x^\top QB\pi_\theta \\ &= \pi_\theta^\top (B^\top Q + \xi^\top \Gamma)x + \frac{1}{2}\pi_\theta^\top \xi^\top \xi \pi_\theta \\ &\quad + \frac{1}{2}x^\top \Gamma^\top \Gamma x + \frac{1}{2}\epsilon x^\top Q x - x^\top QB\pi_\theta \\ &= \frac{1}{2}(\Gamma x + \xi\pi_\theta)^\top (\Gamma x + \xi\pi_\theta) + \frac{1}{2}\epsilon x^\top Q x \geq \frac{1}{2}\epsilon x^\top Q x. \end{aligned} \tag{13}$$

When the system $\dot{x} = f(x, \pi_\theta) =: G$ is strictly passive, then the origin $\dot{x} = f(x, 0)$ is stable and $\pi_\theta^\top y \geq \dot{V} + \mu(x)$ where $(\partial V/\partial x)Ax \leq -\mu(x)$. Let $\bar{\phi}(t; x)$ be the solution of $\dot{\bar{x}} = f(\bar{x}, 0)$, $\bar{x}_0 = x$, then there exists $\Delta V \coloneqq V(\bar{\phi}(\tau; x)) - V(x)$ such that $V(x) > 0$ and $\dot{V} \leq -\mu(x)$, therefore

$$\Delta V \leq -\int_0^\tau \mu(\bar{\phi}(t; x))\, dt =: -\mu_\phi(\tau, x), \forall \tau \in [0, \bar{\gamma}] \tag{14}$$

where $V(\bar{\phi}(\tau; x)) \geq 0$ and $V(x) \geq \mu_\phi(\tau, x)$. For $V(z) = 0$, then $\mu_\phi(\tau, z) = 0, \forall \tau \in [0, \bar{\gamma}]$ and it follows $\mu(\bar{\phi}(t; z)) \equiv 0 \to \bar{\phi}(t; z) \equiv 0 \to z = 0$.

Corollary 3. (Plant mismatch). *If $G_p \neq G_n$ with uncertainty $\Delta(s)$ and $\zeta_p = \zeta_n$, the transfer function of the system $Y(s)$ from reference r to the output y result in,*

$$Y = \frac{\Phi^0(1 + \Delta)}{1 + \Phi^0 \Delta e^{-\zeta s}} e^{-\zeta s}, \quad \Phi^0 = \frac{CG}{1 + CG} \tag{15}$$

and based on the small gain theorem, the system is stable if $|\Phi^0(j\omega)\Delta(j\omega)| < 1, \forall \omega \geq 0$.

Moreover, the plant mismatch could also be seen using $\nu-$gap metric. Given two plants of $P_1(s), P_2(s)$ with left normalized coprime factorizations $P_i = N_i D_i^{-1}, i = 1, 2,$

Authorized licensed use limited to: Northeastern University. Downloaded on September 22,2024 at 19:40:53 UTC from IEEE Xplore. Restrictions apply.

the corresponding $\nu-$gap is defined as

$$\delta_\nu(P_1,P_2)=\begin{cases}\Psi_p^{(1,2)}, & \text{if}\det\left[\Phi(j\omega)\right]\neq 0,\forall\omega\\ & \text{wno}(\det\left[\Phi(s)\right])=0\\ 1, & \text{otherwise}\end{cases}\tag{16}$$

where $\Psi_p^{(1,2)}\coloneqq\|\Psi(P_1,P_2)\|_\infty$, $\Phi(s)\coloneqq N_2^\sim N_1+D_2^\sim D_1$, $\Psi(P_1,P_2)\coloneqq -\tilde{N}_2D_1+\tilde{D}_2N_1$, $\tilde{P}(s)=P^\top(-s)$, and wno denotes winding number. The significance of the gap metric resides in the fact that it can be used to establish whether a controller C that stabilizes P_1 will also stabilize P_2. Specifically, given a controller C that stabilizes P_1 define

$$b_{P_1,C}=\left\|\begin{bmatrix}I\\C\end{bmatrix}(I+P_1C)^{-1}\begin{bmatrix}I & P_1\end{bmatrix}\right\|_\infty^{-1}.\tag{17}$$

Then, as shown in Theorem 17.8 in [17], if $\delta_\nu(P_1,P_2)<b_{P_1,C}$, then the controller C also stabilizes P_2.

Lemma 4. (Delay mismatch)**.** *If $G_p=G_n$, $\zeta_p\neq\zeta_n\coloneqq T\Delta T$ and $V(x)=x^\top Px$, then the feedback dynamic is $G+GT(1-\Delta T)$. Given a storage function $V(x)=x^\top QPx$, with $Q>0$ and a small constant $\epsilon>0$, then $V(x)\geq\epsilon\|x\|^2$.*

Proof: The delay mismatch $\dot{x}=Ax+B\pi_\theta(t-\zeta)$ is asymptotically stable if $\exists\alpha>0$ and a symmetrical matrix Q such that $\Xi<0$. Let us pick $\dot{V}$ along trajectory of $\dot{x}$, then

$$\dot{V}(x(t))=2x(t)^\top Q\left[Ax(t)+B\pi_\theta(t-\zeta)\right]\tag{18}$$

satisfying $V(x(t+\varsigma))<\bar{q}V(x(t)),\forall-\zeta\leq\varsigma\leq 0$ for $\bar{q}>1$ where for arbitrary $\alpha>0$,

$$\begin{aligned}\dot{V}(x(t))&\leq 2x(t)^\top Q\left[Ax(t)+B\pi_\theta(t-\zeta)\right]\\&\quad+\alpha\left[\bar{q}x^\top(t)Q(x(t))-\pi_\theta^\top(t-\zeta)P\pi_\theta(t-\zeta)\right]\\&=\Theta_g^\top\underbrace{\begin{bmatrix}QA+A^\top Q+\alpha\bar{q}Q & QB\\ B^\top Q & -\alpha Q\end{bmatrix}}_{\Xi}\Theta_g\end{aligned}\tag{19}$$

where $\Theta_g=\begin{bmatrix}x^\top(t) & \pi_\theta(t-\zeta)\end{bmatrix}^\top$. This yields $\Xi<0$ such that $V(x)\geq\epsilon\|x\|^2$.

V. Numerical Examples and Insights

In this section, we examine the impact of delays on the robustness of various control strategies, including classical PID, LQR, and the learning-based DDPG reinforcement learning (RL). Furthermore, we investigate the affect of the Smith predictor in capturing delays and analyze the changes in the feedback as a result. In this analysis, we employ a stable system $P(s)\coloneqq G_p(s)e^{-\zeta s}=ke^{-\zeta s}/(1+\tau s)$ DRL agent in two distinct scenarios: stable and unstable systems with $k=3.8163$, $\tau=156.46$, and $\zeta=2.5$ as described in [18]. The primary distinction between the stable and unstable systems is by altering the sign of real parts in stable system into unstable. However, since $|\tau/\zeta|\gg 0.5$, determining the control gains for the PID case becomes straightforward. Moreover, we also consider the non-linearity of the actuator output signal $\varphi_t\coloneqq\varphi(t)$, arising from valve stiction, where the initial friction within the valve exceeds the dynamic friction, denoted as $\varphi_t=N_v(\varphi_{t-1},u_t)$. With the inherent property that $f_s\geq f_d$ where f_s and f_d represent the static and dynamic friction, such that

$$\varphi_t=\begin{cases}u_t-f_d, & \text{if } u_t-\varphi_{t-1}>f_s\\ u_t+f_d, & \text{if } u_t-\varphi_{t-1}<-f_s\\ \varphi_{t-1}, & \text{if } |u_t-\varphi_{t-1}|\leq f_s\end{cases}\tag{20}$$

comprising a multi-mode discontinuous model. Here we then consider the increment multiple delays ζ_i, where $\zeta_i=[5,10,25,50,100,150,300]$. For the small delay ζ_s, the Padé approximation approach closely agrees, but for the higher delay ζ_h, the behavior might be divergent. In this case, performance can be improved, at the cost of more expensive computations, by considering higher-order approximations.

Delay ζ	Stable	Unstable	Delay ζ	Stable	Unstable
5	0.1120	0.4904	100	0.8133	0.9116
10	0.2069	0.4904	150	0.8913	0.9699
25	0.4181	0.5251	300	0.9648	0.9999
50	0.6254	0.7345			

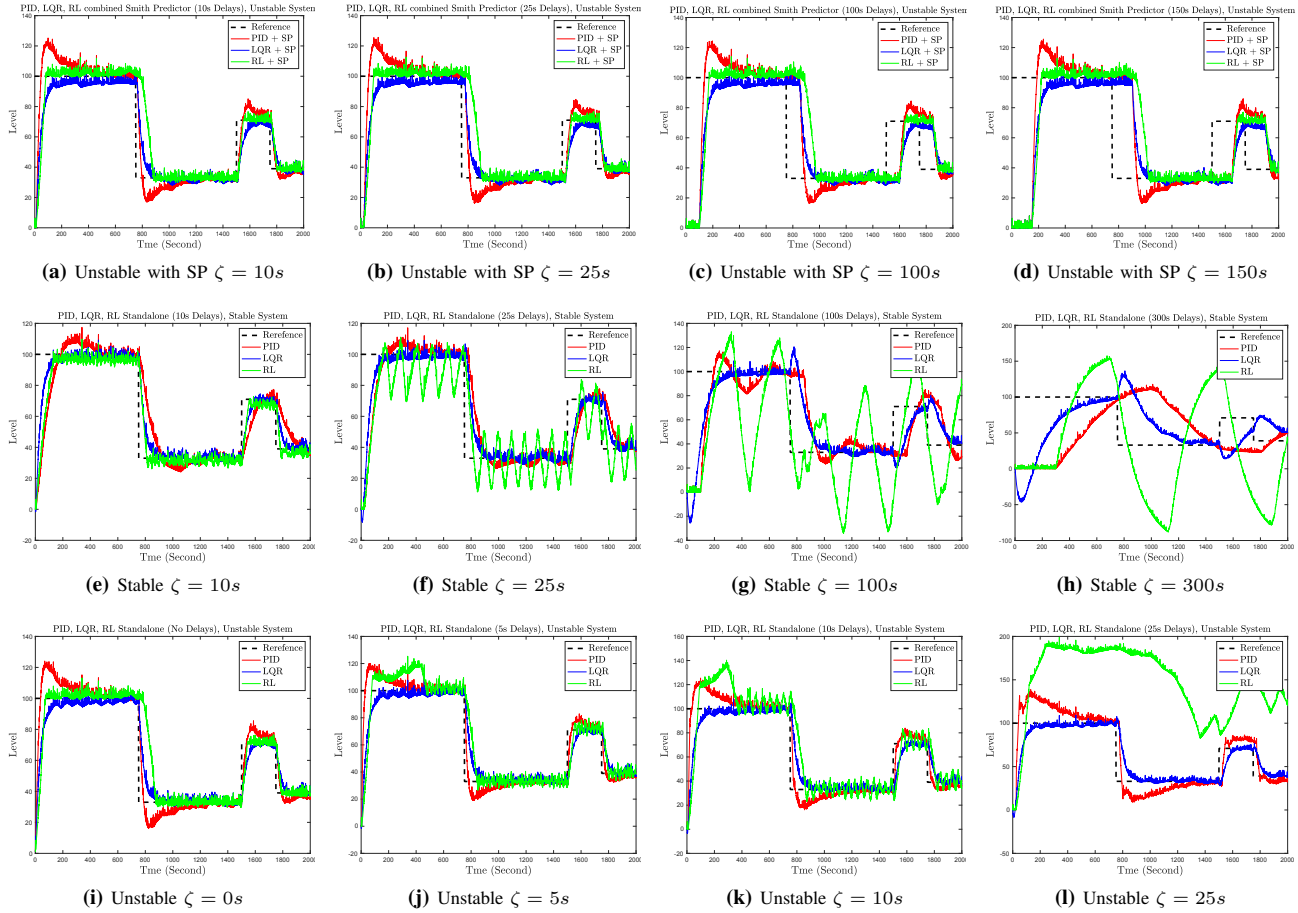
TABLE I: The $\nu-$gap metrics for various ζ_i from the benchmark $G_p(s)$.

Using the well-behaved system $G_p(s)$ as a benchmark, we present the $\nu-$gap metrics between $G_p(s)$ and the non-minimum phase systems with time delays ζ_i as well as their unstable counterparts in Table I. For the non-minimum stable systems, it is evident that as ζ_i increases, the gap grows, and this effect is exacerbated for the unstable systems where $0\leq\delta_v\leq\delta_g\leq 1$. The term ν_ζ represents the $\nu-$gap metrics between the stable and unstable systems for the same delay ζ_i, yielding identical values of 0.4909. The H^∞ optimal cost, which serves as a threshold for the existence of a controller that stabilizes the original plant $P(s)$, amounts to $b_p=0.7917$. Here, $\delta_v(P,P_1(\zeta_i))<b_p$ and b_p is derived from the left-normalized coprime factorization. Fig. 4a-4d demonstrate the effectiveness of capturing delays even in unstable systems, and the performances with those delays ζ_i are precisely the same as those without delay ($\zeta=0$). However, in Fig. 4e-4h, when considering the inability to handle delays without the Smith predictor (SP), the well-trained DRL with $G_p(s)$ can manage delays of about $\zeta_i<20s$, while the considered classical controllers guarantee stability for delays with $|\tau/\zeta|\gg 0.5$ and begin to exhibit unbounded behavior if $|\tau/\zeta|\approx 0.5$ and $\zeta=300s$. The $\nu-$gap metrics increase as the system becomes unstable, as shown in Figs. 4i-4l, causing the boundary of the trained DRL without SP to decrease and resulting in worse performance for the same delay $\zeta=10s$ compared to the stable system.

VI. Conclusions

In this study, we have investigated between the deep reinforcement learning (DRL) operating beyond its training environment and classical control methods in non-minimum phase systems. Our findings demonstrate that combining DRL with a Smith predictor (SP-DRL) can significantly improve performance, even in unstable systems. When the DRL is combined with a Smith predictor, it outperforms

Authorized licensed use limited to: Northeastern University. Downloaded on September 22,2024 at 19:40:53 UTC from IEEE Xplore. Restrictions apply.



(a) Unstable with SP $\zeta = 10s$ **(b)** Unstable with SP $\zeta = 25s$ **(c)** Unstable with SP $\zeta = 100s$ **(d)** Unstable with SP $\zeta = 150s$

(e) Stable $\zeta = 10s$ **(f)** Stable $\zeta = 25s$ **(g)** Stable $\zeta = 100s$ **(h)** Stable $\zeta = 300s$

(i) Unstable $\zeta = 0s$ **(j)** Unstable $\zeta = 5s$ **(k)** Unstable $\zeta = 10s$ **(l)** Unstable $\zeta = 25s$

Fig. 4: This figure presents eight subplots, each comparing three distinct controllers for tracking a desired reference. The controllers are applied to two first-order linear systems with time delays: one stable and one unstable, with varying time delays. The specific time delays are indicated in each subplot.

classical controllers. However, DRL alone, in the absence of a compensatory mechanism such as the Smith predictor, fails to capture the time delay, leading to unbounded states as the time delay becomes larger. Notably, the well-trained learning-based control can manage delays ranging from $10s$ to $20s$, with the performance gap decreasing as the system dynamics become more unstable. Additionally, we considered the analytical $\nu-$gap metrics derived from the benchmark transfer function $P(s)$ for various time delay values and unstable systems to provide further insights into the comparative performance of the control methods.

REFERENCES

[1] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015.

[2] M. K. Wafi and M. Siami, *A Comparative Analysis of Reinforcement Learning and Adaptive Control Techniques for Linear Uncertain Systems*, pp. 25–32.

[3] M. K. Wafi, R. Hajian, B. Shafai, and M. Siami, "Advancing fault-tolerant learning-oriented control for unmanned aerial systems," in *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2023, pp. 1688–1693.

[4] A. Bahill, "A simple adaptive smith-predictor for controlling time-delay systems: A tutorial," *IEEE Control Systems Magazine*, vol. 3, no. 2, pp. 16–22, 1983.

[5] M. Matausek and A. Micic, "A modified smith predictor for controlling a process with an integrator and long dead-time," *IEEE Transactions on Automatic Control*, vol. 41, no. 8, pp. 1199–1203, 1996.

[6] S. Sang and P. Nie, "Modified smith predictor based on h_2 and predictive pi control strategy," *Mathematical Problems in Engineering*, vol. 2021, p. 7228637, Aug 2021.

[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018.

[8] E. Altman and P. Nain, "Closed-loop control with delayed information," *SIGMETRICS Perform. Eval. Rev.*, vol. 20, no. 1, p. 193–204, jun 1992. [Online]. Available: https://doi.org/10.1145/149439.133106

[9] K. Katsikopoulos and S. Engelbrecht, "Markov decision processes with delays and asynchronous cost collection," *IEEE Transactions on Automatic Control*, vol. 48, no. 4, pp. 568–574, 2003.

[10] M. Agarwal and V. Aggarwal, "Blind decision making: Reinforcement learning with delayed observations," *Pattern Recognition Letters*, vol. 150, pp. 176–182, 2021.

[11] T. J. Walsh, A. Nouri, L. Li, and M. L. Littman, "Learning and planning in environments with delayed feedback," *Autonomous Agents and Multi-Agent Systems*, vol. 18, no. 1, p. 83, Jul 2008.

[12] M. Sznaier, A. Olshevsky, and E. D. Sontag, "The role of systems theory in control oriented learning," in *25th International Symposium on Mathematical Theory of Networks and Systems*, 2022.

[13] H. K. Khalil, *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002.

[14] H. Yin, P. Seiler, and M. Arcak, "Stability analysis using quadratic constraints for systems with neural network controllers," *IEEE Transactions on Automatic Control*, vol. 67, no. 4, pp. 1980–1987, 2022.

[15] K. Gu, V. L. Kharitonov, and C. Jie, *Stability of Time-Delay Systems*, 1st ed., ser. Control Engineering. Birkhäuser Boston, MA, 2012.

[16] Q.-C. Zhong, *Robust Control of Time-delay Systems*, 1st ed. Springer London, 2006.

[17] K. Zhou and J. C. Doyle, *Essentials of Robust Control*. Prentice-Hall, 1998.

[18] R. Siraskar, "Reinforcement learning for control of valves," *Machine Learning with Applications*, vol. 4, p. 100030, Jun 2021.

Authorized licensed use limited to: Northeastern University. Downloaded on September 22,2024 at 19:40:53 UTC from IEEE Xplore. Restrictions apply.