A Comparative Analysis of Reinforcement Learning and Adaptive Control Techniques for Linear Uncertain Systems

Moh. Kamalul Wafi* Milad Siami*

Abstract

In this paper, we consider uncertain linear systems with input quantizers over communication channels subject to packet loss, and we assume dynamic switching from an unstable state matrix to a more unstable one during the operation. We then investigate the effectiveness of two learning-based control strategies for stabilizing this class of dynamical systems: the Adaptive Quantized Control (AQC) and the Deep Reinforcement Learning (DRL). The adaptive setup assumes acknowledgment messages on packet losses are received by the adaptive controller, while the state matrix is unknown and the input matrix is known. On the other hand, the DRL operates without acknowledgment messages and relies on the knowledge of both the state and input matrices. Results show that DRL outperforms adaptive techniques in damping amplitudes and improving convergence speed. However, when faced with both packet loss and model uncertainty, the mathematical guarantees provided by AQC can better handle stability and uncertainty across a wider range of model parameters.

1 Introduction

Modern systems' complexity has presented several challenges for classical and modern control techniques. However, there has been a recent trend toward utilizing learning-based control methods, particularly in high-dimensional spaces [1]. Over the past decade, deep learning has significantly impacted both the theoretical and practical aspects of machine learning by enabling its application to non-Euclidean spaces, such as graphs with interdependencies. This revolution has expanded the scope and capabilities of machine learning beyond traditional Euclidean spaces [2]. The integration of deep learning and reinforcement learning, known as deep reinforcement learning (DRL), has been the subject of extensive control-related studies [3, 4, 5]. The ideas behind DRL are similar to those of deep Q-network (DQN) as seen in [6]. The DQN is limited to discrete and finite actions whereas DRL, on the other hand, allows for continuous and infinite control actions, as described in [7]. In this work, we analyze the behavior of an unstable system under two different types of control: the deep deterministic policy gradient (DDPG) reinforcement learning, which is learning-oriented and the quantized control which is adaptive-oriented.

Additionally, the term "quantization" in this context refers to the limitation of communication within a specific bandwidth in networked systems. The concept of using quantization for stabilization of linear systems with finite control signals and measurements was introduced in the reference [8]. The state information is quantized in a coarse manner, with the level of precision becoming finer as it approaches the origin in a logarithmic manner. This can also be alternatively explained using the more widely accepted sector-bounded quantizer [9]. Beyond that, dealing with the uncertain system, the adaptive control with input quantizer is studied by [10] which is also implemented into systems with packet loss η [11]. This means there exists a probability $\bar{p}$ of control signal not being sent to the plant, with some variations of the non-linear uncertain system [12]. Note that with stabilization guaranteed [8, 9, 10, 11], it is interesting to see how the learning approach performs under some limitations [13].

The DDPG reinforcement learning is a combined deterministic-actor $a = \mu_\theta(s)$ instead of the stochastic $\pi_\theta(a|s) = \mathbb{P}[a|s;\theta]$ and the Q-value critic $Q(s,a)$ which both applies the feed-forward neural network (FFNN) [7, 14]. The stability is guaranteed as presented in [15] over uncertain systems with sector-bounded for the non-linear activation function. However, we design the dynamic changes in the simulation to see how robust the trained DRL control performs under specific trained system A considering the dynamic change beyond the environment A_w. The two dynamics encompass systems ranging from modestly unstable to highly unstable, as defined later.

*Department of Electrical & Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mails: {wafi.m, m.siami}@northeastern.edu).

This research was supported in part by grants ONR N00014-21-1-2431, NSF 2121121, and the Army Research Laboratory and accomplished under Cooperative Agreement Number W911NF-22-2-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government.

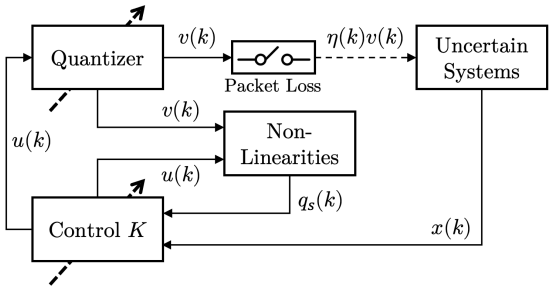


Figure 1: Adaptive quantized control (K, v) method with time-varying quantizer $q(k, u(k))$. The solid arrows between blocks mean the connected information whereas the dashed arrow show there exists a possibility not being connected. The dashed arrows behind blocks deduces the time-varying.

The paper begins by providing an overview of adaptive quantized control with full acknowledgment messages, as depicted in Fig.1. In Section 2, we consider packet loss, represented by $\eta(k) \coloneqq 0, 1$, and in Section 3, we present actor-critic DDPG reinforcement learning as a comparison. Sections 4 and 5 provide numerical examples and a performance analysis of the two controllers, both with and without quantization, ultimately leading to the conclusion.

2 Adaptive Quantized Control

In this section, we introduce the adaptive quantized control method presented in [10], which was further developed for harsh systems with packet loss in [11]. We consider the quantized control signal $v(k)$ to be successfully transmitted to the system according to a constant parameter $\eta(k)$ to decide whether or not the information was received by the system Fig.1. Thus, the discrete-time linear uncertain plant $\mathcal{G}$ over noisy channel is modeled as,

$$x(k+1) = Ax(k) + \eta(k)Bv(k) \tag{2.1}$$

with $x(0) = x_0, k \in \mathbb{N}_0$ where $x \in \mathbb{R}^n$ is the state vector, $v \in \mathbb{R}^m$ denotes the quantized control signal, whereas $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ represent the matrices of unknown system and known input, respectively. The term $\eta(k), \forall k$ represents binary independent random variables, where a value of $\eta(k) \coloneqq 1$ indicates that the control signal was completely received, while a value of $\eta(k) \coloneqq 0$ indicates that information was lost. This transmission condition is affected by some probability $\bar{p}$ such that $\mathcal{P}\{\eta(k) = 0\} \leq \bar{p}, \forall k$ where $0 \leq \bar{p} < 1$. The quantized control is designed as function of control $u(k)$,

$$v(k) = q(k, u(k)) \tag{2.2}$$

where $u(k) = H(k)x(k)$ and $q(k, u_i)$ describes the time-varying logarithmic quantized function written as

$$q_i \triangleq \begin{cases} \varphi_i(k,j), & \text{if } u_i \in (\varphi_i(k,j+1), \varphi_i(k,j)] \\ -\varphi_i(k,j), & \text{if } u_i \in [-\varphi_i(k,j), -\varphi_i(k,j+1)) \\ 0, & \text{if } u_i = 0 \end{cases}$$
$$j \in \mathbb{I}, i = 1, \ldots, m \tag{2.3}$$

with $\varphi_i(k, j) = a_i(k)\rho_i^j(k), i = 1, \ldots, m$ and $a_i(k) > 0$, $0 < \rho_i(k) < 1$ while $q_i(\cdot, \cdot)$ and u_i define the $i-$th term of $q(\cdot, \cdot)$ and u in turn. If the term j is preferred to be negative, the value of $\varphi_i(k, -j)$ alters to $\varphi_i(k, -j) \coloneqq a_i(k)/\rho_i^j(k), i = 1, \ldots, m$. Moreover, $\rho_i(\cdot)$ declares the coarseness of the quantizer $q_i(\cdot, \cdot)$ for certain u_i. Note that (2.3) could be transformed as a time-varying sector-bounded memoryless input nonlinearities $\mathcal{Q}$ given by,

$$\begin{aligned} \mathcal{Q} \triangleq \{ & q : \mathbb{N}_0 \times \mathbb{R}^m \to \mathbb{R}^m : q(\cdot, 0) = 0, \\ & \left[q(k,u) - M_1(k)u\right]^\top \left[q(k,u) - M_2(k)u\right] \leq 0, \\ & u \in \mathbb{R}^m, k \in \mathbb{N}_0 \} \end{aligned} \tag{2.4}$$

where $M_1 \in \mathbb{R}^m$ and $M_2 \in \mathbb{R}^m$ imply the diagonal matrix formed as $M_1 \triangleq \text{diag}[M_{1_1}, \ldots, M_{1_m}] > 0$ and $M_2 \triangleq \text{diag}[M_{2_1}, \ldots, M_{2_m}] > 0$ satisfying positive definiteness $M_2 - M_1 > 0$ as portrayed in Fig.2 with $\rho_i = M_{1_i}/M_{2_i}, \forall i \in \mathbb{R}^m$. For the scalar perspective, the sector-bounded $\mathcal{Q}$ (2.4) is translated into,

$$M_{1_i}(k)u_i^2 \leq q_i(k, u_i)u_i \leq M_{2_i}(k)u_i^2, \tag{2.5}$$

where $u_i \in \mathbb{R}, k \in \mathbb{N}_0, \forall i \in \mathbb{R}^m$. Recalling

$$\rho_i(\cdot) = \frac{M_{1_i}(\cdot)}{M_{2_i}(\cdot)} = \frac{1 - \beta\delta_i(\cdot)}{1 + \beta\delta_i(\cdot)}, \quad i = 1, \ldots, m \tag{2.6}$$

in which $\delta(\cdot) \triangleq \frac{1}{\beta}\left[M_2(\cdot) + M_1(\cdot)\right]^{-1}\left[M_2(\cdot) - M_1(\cdot)\right]$ for $\beta \neq 0$, the coarseness $\rho_i(\cdot, \cdot)$ is now set as a function of $\delta_i(\cdot)$ while due to the change of the time k, the quantizer $q(k, \cdot)$ is according to the formula of

$$\begin{aligned} \Delta(k) &\triangleq \text{diag}[\delta_1(k), \ldots, \delta_m(k)] \\ &= \frac{1}{\beta}\left[M_2(\cdot) + M_1(\cdot)\right]^{-1}\left[M_2(\cdot) - M_1(\cdot)\right], \end{aligned} \tag{2.7}$$

In this work, we aim to compare the performance of adaptive quantized control with that of actor-critic reinforcement learning. Both control methods utilize quantization to reduce communication bandwidth. The detailed concept and stability analysis of the adaptive quantized control can be found in [10]. We next decompose the quantizer $q(\cdot, \cdot)$ into its linear and nonlinear components as follows:

$$q(k, u) = \frac{1}{2}\left[M_1(k) + M_2(k)\right]u + q_s(k, u) \tag{2.8}$$

where the non-linear modified $q_s : \mathbb{N}_0 \times \mathbb{R}^m \to \mathbb{R}^m$ is a set of $\mathcal{Q}_s$ expressed as,

$$\mathcal{Q}_s \triangleq \{q_s : \mathbb{N}_0 \times \mathbb{R}^m \to \mathbb{R}^m : q_s(\cdot, 0) = 0,$$
$$q_s^\top(k,u) q_s(k,u) - \frac{1}{4} u^\top \left[M_2(k) - M_1(k)\right]^2 u \leq 0,$$
$$u \in \mathbb{R}^m, k \in \mathbb{N}_0\}. \tag{2.9}$$

For the stochastic system, we adopt the definition and proof of Lyapunov stability in probability from [16].

DEFINITION 2.1. Let us consider a stochastic discrete-time system as follows,

$$x_{k+1} = f(x_k, y_{k+1}), \quad k \in \mathbb{N}_0 \tag{2.10}$$

where $x_k \in \mathbb{R}^n$ and $\{y_k : k \in \mathbb{N}\}$ shows a $\mathbb{R}^d-$valued stochastic process on a probability space $(\Omega, \mathbb{F}, \mathbb{P})$. The notation of Ω comprises the sample space, $\mathbb{F}$ makes up a set of events, and $\mathbb{P} : \mathbb{F} \to [0,1]$ constitutes a function matching the probabilities to events. The measurable function y_k maps Ω into state-space $\Omega_0 \in \mathbb{R}^d, \forall \omega \in \Omega$ and for $\mathbb{F}_k = \sigma(y_1, \ldots, y_k), \forall k \geq 1$, $\mathbb{F}_0 = \{\emptyset, \Omega\}$, $\{\mathbb{F}_k\}, \forall k$ is an increasing sequence of $\sigma-$field. The origin of (2.10) is classified to be:

1. stable in probability if $\lim_{x_0 \to 0} \mathbb{P}[\sup_{k \in \mathbb{N}} \|x_k\| > \epsilon] = 0$ for any $\epsilon > 0$;
2. asymptotically stable in probability if the stable holds and $\lim_{x_0 \to 0} \mathbb{P}[\lim_{k \to \infty} \|x_k\| = 0] = 1$;
3. exponentially stable in probability if for $\gamma > 1$ independent of ω, $\lim_{x_0 \to 0} \mathbb{P}[\lim_{k \to \infty} \|\gamma^k x_k\| = 0] = 1$

For a set $\mathcal{Q} \subseteq \mathbb{R}^n$ the origin of (2.10) is classified to be:

1. locally (globally) a.s in $\mathcal{Q}$ if starting from $x_0 \in \mathcal{Q}$ ($x_0 \in \mathbb{R}^n$), all the sample paths x_k stay in $\mathcal{Q}$ ($\mathbb{R}^n$) $\forall k \geq 0$ and converge to the origin almost surely;
2. locally (globally) exponentially stable in $\mathcal{Q}$ if it is locally (globally) a.s and the convergence moves exponentially fast

The asymptotic convergence of (2.10) and its stability inspired by [16, 17] is presented as follows,

LEMMA 2.1. *For the stochastic system being defined in (2.10), let x_k be a Markov chain and let $V : \mathbb{R}^n \to \mathbb{R}$ be a Lyapunov positive definite function. For some $\lambda > 0$, given a set $\mathcal{Q}_\lambda \coloneqq \{x_k : 0 \leq V(x_k) < \lambda\}$ provided that*

$$\mathbb{E}[V(x_{k+1})] - V(x_k) = -\vartheta(x_k) \leq 0, \forall k \tag{2.11}$$

where $x_k \in \mathcal{Q}_\lambda$ and $\vartheta(\cdot)$ is continuous, therefore the two conditions might apply:

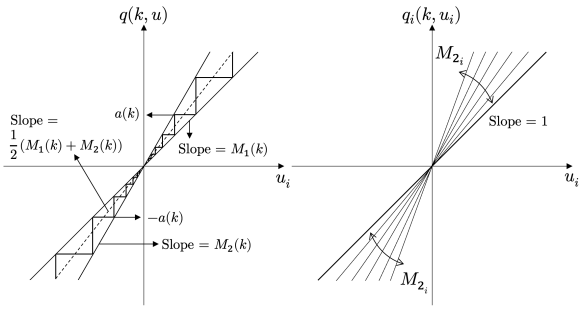


Figure 2: Left: logarithmic quantized q for $m = 1$, Right: the instance for $[M_2(k) \in \{1 + \hat{a}\mu_i^j : j \in \mathbb{I}\}, M_1(k) \equiv 1]$.

(i) if $x_0 \in \mathcal{Q}_\lambda$, the paths remain in $\mathcal{Q}_\lambda$ with at least probability $1 - V(x_0)/\lambda$, $V(x_k)$ converges to some limit and $\lim_{k \to \infty} \vartheta(x_k) = 0$ with probability 1

(ii) recalling the case (i), for every $\gamma > 0, \exists \delta > 0$, such that $\vartheta(x_k) \geq \delta$ for $|x_k| > \gamma$ and $\vartheta(0) = 0$, therefore the origin of (2.10) is globally a.s

We continue the overview of the adaptive quantized control, which is divided into several conditions [10, 11] and we here consider the acknowledgement messages to the gain control K of whether or not the packet loss $\eta(k)$ occurs using the time-varying variable $q_s(k,u)$. Finally, we are asserting the adaptive control with parameters written as follows, in order, as steps along with the associated key theorem from [11] on how to design the adaptive quantized control (AQC),

1. Let $R \in \mathbb{R}^{n \times n} > 0$ and let $\gamma \in (0,1)$
2. Solve the Riccati equation with $P \geq I_n$,
$$P = \tilde{A}^\top P \tilde{A} + R - \tilde{A}^\top P B \left(B^\top P B\right)^{-1} B^\top P \tilde{A}$$
3. Let $A_s \triangleq \tilde{A} + B K_g''$ where $\tilde{A} \triangleq A + B K_g'$ and $K_g'' \triangleq -(B^\top P B)^{-1} B^\top P \tilde{A}$. The matrix A_s is then Hurwitz provided that (A, B) is stabilizable. Note that $\tilde{A}$ is assumed to be known and unstable
4. Let $Q \in \mathbb{R}^{m \times m}$ and $\varepsilon > 0$, the $Q \in (0, 2I_m)$ and
$$\frac{1}{\varepsilon}(2I_m - Q) - 2B^\top P B \geq 0,$$
are guaranteed. Thanks to P, Q, ε for always exist

THEOREM 2.1. *Recalling the discrete linear uncertain system $\mathcal{G}$ in (2.1) where A is an unknown system where $\sigma(A) < \bar{\sigma}_A$, the rank of B denoted as $\rho(B) = m$, and the pair of (A, B) is stabilizable. Given that the controller knows whether or not the packet loss happened and assuming the upper bound of the probability $\bar{p}$ satisfies*

$$\frac{\bar{p}}{1 - \bar{p}} \left(\lambda_{\max}(P) \bar{\sigma}_A^2 I_n - P\right) < \gamma R \tag{2.12}$$

therefore, according to steps 1-4, the adaptive control with acknowledgement message

$$u(k) = 2\left[M_1(k) + M_2(k)\right]^{-1} K(k)x(k), \tag{2.13}$$

where $K(k) \in \mathbb{R}^{m\times n}$*, the quantized* $M_1(k)$ *and* $M_2(k)$ *satisfying the following equation*

$$R - 2K^\top(k)\Delta(k)B^\top PB\Delta(k)K(k) \geq \gamma R > 0 \tag{2.14}$$

for all $k \in \mathbb{N}_0$ *with (2.2) and the updated gain*

$$\begin{aligned} K(k+1) = K(k) - \frac{\eta(k)}{1 + x^\top(k)Px(k)} QB^\dagger \left[x(k+1)\right. \\ \left. - A_s x(k) - Bq_s(k, u(k))\right] x^\top(k) \end{aligned} \tag{2.15}$$

guarantees the Lyapunov stability such that the solution $(x(k), K(k)) \equiv (0, K_g)$ *where* $K_g \triangleq -(B^\top PB)^{-1}B^\top PA$ *and* $K_g \coloneqq K_g' + K_g''$ *given by (2.1), (2.2), (2.13), and (2.15) converges as* $\lim_{k\to\infty} x(k) = 0, \forall x_0 \in \mathbb{R}^n$

Finally, equation (2.14) should hold at all times, implying that the sector bounds of $M_1(k)$ and $M_2(k)$ must also be time-varying. One simple method to ensure compliance with equation (2.14) is to fix $M_1(k)$ and make $M_2(k)$ time-varying. This can be achieved by defining $M_1(k)$ as a constant equal to I_m and $M_2(k)$ as $1 + \hat{a}\mu_i^j : j \in \mathbb{I}$, where $\hat{a} > 0$ and $\mu_i > 0$ for all i. Since $M_2(k) - M_1(k)$ is positive definite then $M_2(k) > I_m$ and supposed $M_2(k)$ is bounded with a parameter M_ϕ as the maximum such that $M_1(k) < M_2(j), j \in \mathbb{I} \leq M_\phi$. Then the issue is to find the suitable j as the best parameter of $M_2(k)$ at given step k to be delivered to the controller. The design of the AQC algorithm is outlined in Algorithm 1. To evaluate its performance against the AQC, the reinforcement learning design will be presented next.

Algorithm 1 Adaptive Quantized Control

Require: $A, B, x_0, k(\text{on}), K_0, R \coloneqq I_2 \to P \geq I_2, \bar{p}$
$Q \leftarrow (0, 2I_m)$ ▷ $(a, b) \coloneqq a < Q < b$
$K_g \leftarrow -(B^\top PB)^{-1}B^\top PA$
$A_s \leftarrow A + BK_g$
for $k = 1 : t(\max)$ **do**
  $p_c \leftarrow \mathcal{N}(\mu, \sigma)$ ▷ *Gaussian between* 0 *and* 1
  if $p_c > \bar{p}$ **then**
    $\eta(k) \leftarrow 1$, *otherwise* $\eta(k) \leftarrow 0$
  end if
  if $k \geq t_d$ **then** ▷ $t_d \coloneqq$ *dynamic change*
    $(A, B) \leftarrow (A_w, B_w)$
  end if
  $\gamma \leftarrow (0, 1)$, $\delta \leftarrow$ *Eq.* 2.13
  if $\delta \geq 1$ **then** ▷ $0 < \delta < 1$
    $M_2(k) \leftarrow M_\phi$, $\rho \leftarrow \frac{1}{\phi}$ ▷ $M_2(k) \coloneqq M_\phi$ *is max*
  else
    for $j_i, \forall i = 1 : n$ **do** ▷ $M_1 < M_2(j_i) < M_\phi$
      if $\delta > \frac{1}{\beta}(M_2 + M_1)^{-1}(M_2 - M_1)$ **then**
        $j \leftarrow j_i$ and **stop**
      end if
    end for
    $u(k) \leftarrow$ *Eq.* 2.13, $v(k) \leftarrow$ *Eq.* 2.3
  end if
  $x(k+1) = Ax(k) + \eta(k)Bv(k)$
  $q_s(k) \leftarrow$ *Eq.* 2.8, $K(k+1) \leftarrow$ *Eq* 2.15
end for

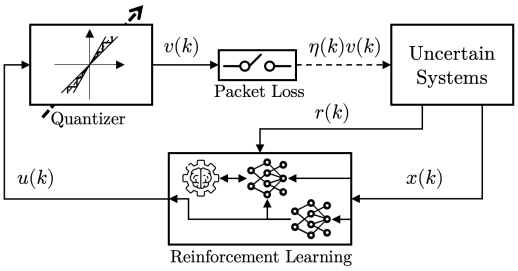


Figure 3: Actor-critic DDPG Reinforcement learning with time-varying quantizer $q(k, u(k))$.

3 DDPG Reinforcement Learning

We consider deep-deterministic policy gradient (DDPG) developed by [7, 14] which uses the deterministic direct mapping states to action $u = \mu_\theta(x)$ instead of the probabilistic ones $\pi_\theta(u|x) = \mathbb{P}[u|x;\theta]$. The two terms x and u represent the spaces of the states $x \subseteq \mathcal{S}$ and the actions $u \subseteq \mathcal{A}$. The problem is built as the Markov decision process (MDP) comprising the spaces of $(\mathcal{S}, \mathcal{A})$ and the distributions of an initial state with density $p_1(x_1)$ and the stationary transition $p(x_{t+1}|x_t, u_t) \coloneqq p(x_{t+1}|x_1, u_1, \ldots, x_t, u_t)$ for arbitrary trajectory $(x_k, u_k), \forall k \in (\mathcal{S}, \mathcal{A})$. Furthermore, the agent applies the off-policy information and the Bellman equation to learn the Q-value function which is then determined to learn the policy μ_θ with the optimum one μ_θ^*. The control mechanism is designed to maximize the expected cost function $J(\mu_\theta) = \mathbb{E}[r_t^\gamma|\mu_\theta]$,

$$J(\mu_\theta) = \int_{\mathcal{S}} \rho^\mu(x) \int_{\mathcal{A}} \mu_\theta(x, u) r(x, u)\, du\, dx \tag{3.16}$$

where the return r_t^γ denotes the sum of the discounted future rewards from time instant t onwards defined as,

$$r_t^\gamma = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots = \sum_{k=t}^{\infty} \gamma^{k-t} r(x_k, u_k) \tag{3.17}$$

with the discount factor $\gamma \in (0, 1)$. If the policy μ_θ and the associated cost function $J(\mu_\theta)$ are taken such that it maximizes the function, $\forall x \in \mathbb{R}, \forall t \in [0, T]$, then $J(\mu_\theta) = J(\mu_\theta^*)$ and $\mu_\theta = \mu_\theta^*$.

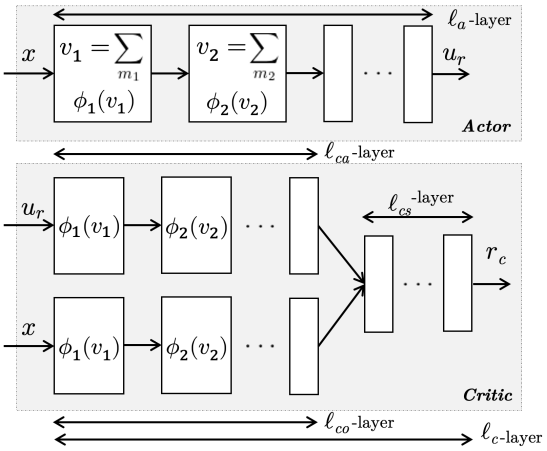


Figure 4: Actor-critic DDPG reinforcement learning design.

This DDPG is closely related to the DQN with the following explanations. The value function of the state-action $Q^\mu(x,u)$ is expressed as the expected return given by (x,u,μ_θ) and the discounted state distribution $\rho^\mu(x')$. If the optimal action function $Q^*(x,u)$ is obtained, then in arbitrary given state, the optimal action $u^*(x)$ is declared as $u^*(x) = \arg\max_u Q^*(x,u)$. Due to the fact the action is continuous, the value function $Q^*(x,u)$ is assumed as differentiable over action argument and it leads to construct the gradient-based rule for $\mu_\theta(x)$. This allows to approach the expensive computation optimization as $\max_u Q(x,u) \approx Q(x,\mu_\theta(x))$. For the discrete perspective, the Bellman equation showing the optimal value function $Q^*(x,u)$ is written as follows,

$$Q^*(x,u) = \mathbb{E}_{x'\sim\mathbb{P}}\left[r(x,u) + \gamma \max_{u'} Q^*(x',u')\right] \tag{3.18}$$

where $x' \sim \mathbb{P}$ says that the next state x' is sampled by the system given a certain distribution $\mathbb{P}(\cdot|x,u)$. Now, as the action is deterministic and continuous, the Q-value function is constructed with neural network ϕ such that $Q^\phi(x,u)$ collects a set Φ of transition (x,u,r,x',d) where d points whether or not the next state x' is the terminal. Finally, the mean squared Bellman error (MSBE) describing the error of $Q_\phi(x,u)$ into Bellman's,

$$\begin{aligned} \xi_\Phi &= r + \gamma(1-d)\max_{u'} Q^*(x',u') \\ L(\phi,\Phi) &= \mathbb{E}_{x,u,r,x',d\sim\Phi}\left[\left(Q_\phi(x,u) - \xi_\Phi\right)^2\right] \end{aligned} \tag{3.19}$$

and the reward is presented as a function of error $e \coloneqq x$ with some arbitrary small constant ψ to negate the zero division and a parameter threshold T_h if the system performs beyond the permitted values, therefore

$$r(k) = \begin{cases} -100, & x < -T_h, x > T_h \\ \dfrac{1}{e+\psi}, & \text{otherwise} \end{cases} \tag{3.20}$$

Now, recall a linear uncertain system in (2.1) with a feedback deep-reinforcement learning (DRL) control as in Fig.3 with quantizer (2.2). The state $x(k) \in \mathbb{R}^{n_p}$ is also the output $x(k) \coloneqq y(k)$ and $u(k) \in \mathbb{R}^{n_u}$ makes up the input. The actor-critic DRL control performs two sequential feed-forward neural network (FFNN). The *actor*-DRL takes the observation x using single direction with ℓ_a-layer while the *critic*-DRL relies on two inputs, the observation x and control action u under ℓ_c-layer. This control u comes from the *actor*-DRL, where 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*-DRL with ℓ_a-layer is denoted as,

$$\phi_0(k) = x(k) \tag{3.21a}$$

$$\phi_i(k) = \Delta_i \underbrace{(W_i\phi_{i-1}(k) + b_i)}_{v_i(k)}, \forall i = 1 \to \ell_a \tag{3.21b}$$

$$u^{(n_u)}(k) = W_{\ell_a+1}\phi_{\ell_a}(k) + b_{\ell_a+1} \coloneqq v^{(n_u)}_{\ell_a+1} \tag{3.21c}$$

where the weight matrix $W_i \in \mathbb{R}^{m_i\times m_{i-1}}, \forall i = 1,\dots,\ell_a$ and the bias $b_i \in \mathbb{R}^{m_i}$ handles the linear operations for $i-$th layer resulting $v_i \in \mathbb{R}^{n_i}$ containing m_i-neurons with $m_0 = n_p$ so that for $j-$th neuron, $\forall j = 1,\dots m_i$ runs the calculation of (3.22),

$$v_i(k) \coloneqq v_i^{(j)}(k) = \sum_{t=1}^{m_{i-1}} W_i^{(t)}\phi_{i-1}^{(t)}(k) + b_i^{(t)}, \tag{3.22}$$

for all $j = 1,\dots m_i$ The non-linear operations of activation functions $\Delta_i \in \mathbb{R}^{n_i}$ in $i-$th layer drive the v_i with n_i is a row matrix with length of the scalar multiplication of $n_i \coloneqq m_i \times m_{i-1}$ with outputs of $\phi_i \in \mathbb{R}^{n_i}$. The activation function Δ_i is represented as the element-wise $\Delta_i(v_i) \coloneqq [\lambda(v_1),\cdots,\lambda(v_{n_i})]$ where $\lambda(v) \coloneqq \tanh(v)$ or ReLU $\lambda(v) \coloneqq \max(0,v)$. The size of the last linear operation should have the same that of control signal u, $v_{\ell_a+1} \in \mathbb{R}^{n_u}$. Therefore, the actor collects the values of the linear $v_q \coloneqq [v_1,\cdots,v_{\ell_a}] \in \mathbb{R}^{n_q}$ and the non-linear $\phi_q \coloneqq [\phi_1,\cdots,\phi_{\ell_a}] \in \mathbb{R}^{n_q}$ in which $n_q \coloneqq \sum_{\ell_a} n_i$ along with $\Delta_q(v_q)$ resulting the control signal $u(k)$ to the plant and as the input of *critic*-RL. The input-output relationship of actor is shown as,

$$\begin{bmatrix} u(k) \\ v_q(k) \end{bmatrix} = N_{\ell_a}\begin{bmatrix} x(k) \\ \phi_q(k) \\ 1 \end{bmatrix}, \quad \phi_q = \Delta(v_q(k)) \tag{3.23}$$

and the matrix N consists of the weights $W_q \in \mathbb{R}^{n_q+1}$ and the biases $b_q \in \mathbb{R}^{n_q+1}$ corresponding to the inputs

Algorithm 2 Reinforcement Learning with Quantizer

Require: $A, B, x_0, k(\text{on}), R \coloneqq I_2 \to P \geq I_2$, FFNN $\psi, \tau_a, \tau_c, \gamma, t_s, \mathcal{E}, \mathcal{T}$

for $k = 1 : t(\max)$ **do**
  observe state $x(k)$ and *take an action* $u_r(k)$
  observe a reward $r(k)$
  if $k \geq t_d$ **then** ▷ $t_d \coloneqq$ *Dynamic change*
    $(A, B) \leftarrow (A_w, B_w)$
  end if
  $\gamma \leftarrow (0, 1)$ ▷ $(a, b) \coloneqq a < \gamma < b$
  $\delta \leftarrow$ *Eq.* 2.13 *with* $K \coloneqq u_r x^{-1}$
  $v(k) \leftarrow$ *do step 10 - 20 of Algorithm 1, changing* Kx *with the control signal from RL* $\to u_r(k)$
  $x(k+1) = Ax(k) + \eta(k)Bv(k)$
end for

$[w, \phi_q, 1]$ and outputs $[u, v_q]$

$$N \coloneqq \left[\begin{array}{c|cccc|c} 0 & 0 & 0 & \cdots & W_{\ell_a+1} & b_{\ell_a+1} \\ \hline W_1 & 0 & \cdots & 0 & 0 & b_1 \\ 0 & W_2 & \cdots & 0 & 0 & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & W_{\ell_a} & 0 & b_{\ell_a} \end{array}\right] \tag{3.24}$$

This results in decomposing the FFNN from the nonlinear activation function and stability of these forms are studied in [18]. With the same procedures, the FFNN *critic*-RL with ℓ_c-layer is also denoted as,

$$\begin{aligned} \varphi_0(t) &= \left[\begin{array}{cc} u(t) & x(t) \end{array}\right]^\top \\ \varphi_i(t) &= \left[\begin{array}{c} \Delta_i\left(W_i\varphi_{i-1}(t) + b_i\right), \forall i \in \ell_{ca} \\ \Delta_i\left(W_i\varphi_{i-1}(t) + b_i\right), \forall i \in \ell_{co} \end{array}\right]^\top \\ \varphi_j(t) &= \varphi_{\ell_{ca}}(t) + \varphi_{\ell_{co}}(t), \quad j \coloneqq \max(\ell_{ca}, \ell_{co}) \\ \varphi_j(t) &= \Delta_j\left(W_i\varphi_{j-1}(t) + b_j\right), \quad \forall j = 1, \ldots, \ell_{cs} \\ r_c(t) &= W_{\ell_{cs}+1}\varphi_{\ell_{cs}}(t) + b_{\ell_{cs}+1} \coloneqq v_{\ell_{cs}+1} \in \mathbb{R}^{r_c} \end{aligned} \tag{3.25}$$

where the construction is straight-forward exactly the same as that of *actor*-DRL (3.21-3.24) and from here, to differ the control signal of DRL from AQC, we use the term $u_r(k) \coloneqq u(k)$. If the optimal values (x^*, u_r^*) satisfy (2.1), then the state x^* could be propagated via FFNN to reach the equilibrium values v_i^*, w_i^* for the inputs/outputs of every activation function, resulting $(v_q, w_q) = (v^*, w^*)$. Thus, (x^*, u_r^*, v^*, w^*) is an equilibrium point of (2.1), (3.21) and (3.25) if the following dynamic holds, $x(k+1) = Ax^*(k) + Bq(k, u_r^*(k))$. This paper shows a training given to DRL and see how it performs beyond the environment. While the AQC works with unknown A around the gain K_g such that $A_s \coloneqq A + BK_g$ is known, this DRL performs under a known linear system A, which is then examined beyond the trained system, with A_w and packet loss.

4 Numerical Results and Findings

In this section, we provide numerical illustrations to show the effectiveness of the suggested control methods. There are four scenarios to analize: 1) the adaptive quantized control (AQC); 2) the AQC such that $\exists \bar{p}_i, \forall i = \{1, 2, 3\} : \bar{p}_1 = 0.15$, $\bar{p}_2 = 0.30$ and $\bar{p}_3 > 0.5$; 3) the trained DRL with A system and input quantizer; 4) example 3 holds with the packet loss $\exists \bar{p}_i, \forall i$. Keep in mind those four scenarios will be also considered the dynamical switching from A to A_w according to time parameter $t_d = \{10s, 15s, 20s, 25s, 30s\}$ and the control starts from $k(\text{on}) = 1\text{s}$. Let us consider the linear uncertain system and the noisy channel given by

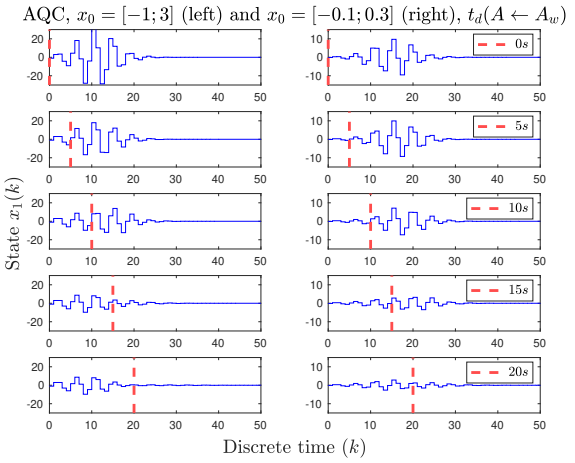
$$\begin{aligned} z(k+2) + \beta_\zeta z(k+1) + \alpha_\zeta z(k) &= \eta(k) b_\zeta v(k), \\ z(0) = z_0, \quad z(1) &= z_1, \quad k \in \mathbb{N}_0, \end{aligned} \tag{4.26}$$

where $\beta_\zeta, \alpha_\zeta \in \mathbb{R}, \forall \zeta = 1, 2$ are the unknown constants, $b_\zeta \in \mathbb{R}, \forall \zeta$ comprise the known constants, $z(k) \in \mathbb{R}$, and $v(k) \in \mathbb{R}$ results in quantizer control $u(k)$. The random process $\eta(k), k \in \mathbb{N}_0$ decides whether or not the signal is transmitted based on $\mathcal{P}\{\eta(k) = 0\} \leq \bar{p}_i, \forall k$. The states would be $x_1(k) = z(k)$ and $x_2(k) = z(k+1)$ where the modest unstable system A and the worse A_w are written as follows

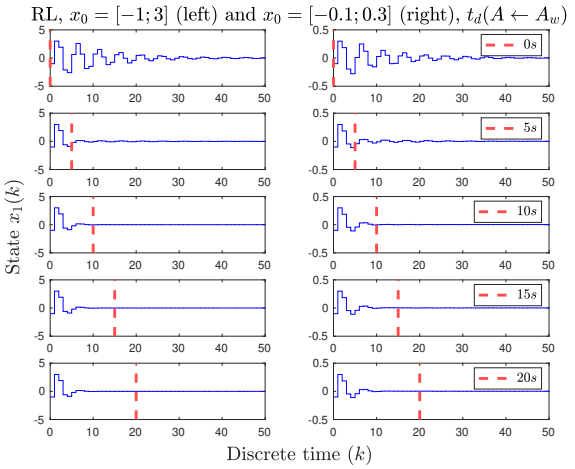
$$A = \left[\begin{array}{cc} 0 & 1 \\ -\alpha_1 & -\beta_1 \end{array}\right], \quad A_w = \left[\begin{array}{cc} 0 & 1 \\ -\alpha_2 & -\beta_2 \end{array}\right]$$

with $x = [x_1, x_2]^\top$, $x(0) = x_0$, $B = [0, b_\zeta]^\top, \forall \zeta$. The R is chosen as $I_m, m \coloneqq 2$ so as to solve Riccati equation in step 2 as $P = \text{diag}[1, 2] > I_m$. The constants $\alpha_1 = 1.5$, $\alpha_2 = 2$, $\beta_1 = \beta_2 = -0.5$ and $b_\zeta = 0.4, \forall \zeta$. Here, we are considering only one state $x_1(k)$ so that the sector-bounds $M_1(k) \equiv 1$ and $M_2(k) \in \{1 + 3 \times (1.3)^j, j \in \mathbb{I}\}$. This j in $M_2(j)$ makes the $M_2(k)$ time-varying such that $M_1 \coloneqq 1 < M_2(j) < M_\phi \coloneqq 10$. The DRL uses the same quantizer yet applying the control signal u_r from the trained DRL in A unstable system. The observation spaces $e(k) = x(k) = y(k)$ are infinite while the action spaces are bounded $|\mathcal{A}| < \mathcal{A}_\phi \coloneqq 100$. The learning rates of actor τ_a and critic τ_c are defined as 10^{-4} and 10^{-3} in turn. The FFNN of actor uses the tanh and softmax activation funtion with 25 neurons while the FFNN of critic combines the ReLU activation functions for both paths (state x, action u_r from actor) with 50 and 25 neurons. The discount factor γ, the sample time t_s, maximum episode $\mathcal{E}$ and the terminal training $\mathcal{T}$ are setup with $\gamma = 0.9$, $t_s = 1s$, $\mathcal{E} = 1000$, and $\mathcal{T} = 750$.

The results are exhibited in Fig.5 and Fig.6. The simulations present various dynamic switches (red-dashed), from unstable A to more unstable system A_w, based on time parameter t_d from three different initial conditions x_0. As for the AQC, it can be seen in Fig.5,



(a) Adaptive Quantized Control, $\eta(k) = 1, \forall k$



(b) Reinforcement Learning, $\eta(k) = 1, \forall k$

Figure 5: Performances of AQC and DRL with two different initial condition x_0 and various dynamic change time t_d.

if the switches happen earlier, it impacts the system performance worse than the later changes. This is due to the adaptive gain K has already closely converged to the optimal values for the later changes. Whilst to the AQC with packet loss $\eta(k)$ with some probabilities $\bar{p}_i$, it applies the acknowledgement messages to the controller, meaning that the control K receives the information whether or not the packet loss happens via time-varying q_s in (2.15). It is because $v(k)$ and the quantized input $u(k)$ via time-varying $M_2(k)$ are always updated at any time instant k regardless any loss $\eta(k) := 0$. In [11], it is mentioned, for control with acknowledgement messages, there is always stability guaranteed for $\bar{p}$ satisfying the upper-bound in (2.12) whereas this range reduces to $0 \leq \bar{p} \leq 0.5$ if the control K does not have the information, q_s not being sent to K. Interestingly, being trained in A unstable system with quantizer as depicted in Fig.2, the DRL control can stabilize the system with dynamic changes A_w less than 10s. This is obvious because, as the MDP, the DRL already knows the optimal values while the impulsive at first is due to different initial conditions compared to the training value with $x_0 = [0, 0]^\top$. However, when dealing with packet loss, the mathematical guaranteed AQC outperforms the DRL which should act beyond the action spaces. Keep in mind that the AQC uses the known A_s and unknown A while the DRL is feedback with the known A, otherwise it could not stabilize the system, i.e., training in A_s and experimenting in A.

$\alpha_2 =$	1	2	3	4	5	...	18	19	20
RL	✓	✓	×	×	×	...	×	×	×
AQC	✓	✓	✓	✓	✓	...	✓	×	×
AC	✓	✓	✓	✓	✓	...	✓	✓	✓

Table 1: Stability comparison of three control methods across different dynamic switches $A \to A_w$ at time t_d

***Remark*:** Table 1 presents a summary of the stability performance for various switching scenarios of $A \to A_w$ at time t_d, as α_2 is varied from 1 to 20. It is noted that the DRL control demonstrates effective performance only within the narrow range of the trained environment, which was trained with $\alpha_2 = 1$. In contrast, the AQC control, being a special case of adaptive control (AC), exhibits stability over a wider range of α_2 values, whereas the AC control demonstrates stability for all reported α_2 values in the table.

5 Conclusion

The comparative framework of the AQC and the DDPG DRL has been constructed along with the numerical examples. The dynamic changes and the packet loss signals to approach the real conditions have been chosen to see the robustness of the DRL beyond the training environments. The results indicate the trade-off and limitations of the learning-oriented control working beyond the training environments. Finally, future research will focus on ensuring the stability of DRL in the presence of packet loss.

References

[1] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3826–3839, 2020.

[2] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[3] X. Wang, S. Wang, X. Liang, D. Zhao, J. Huang, X. Xu, B. Dai, and Q. Miao, "Deep reinforcement

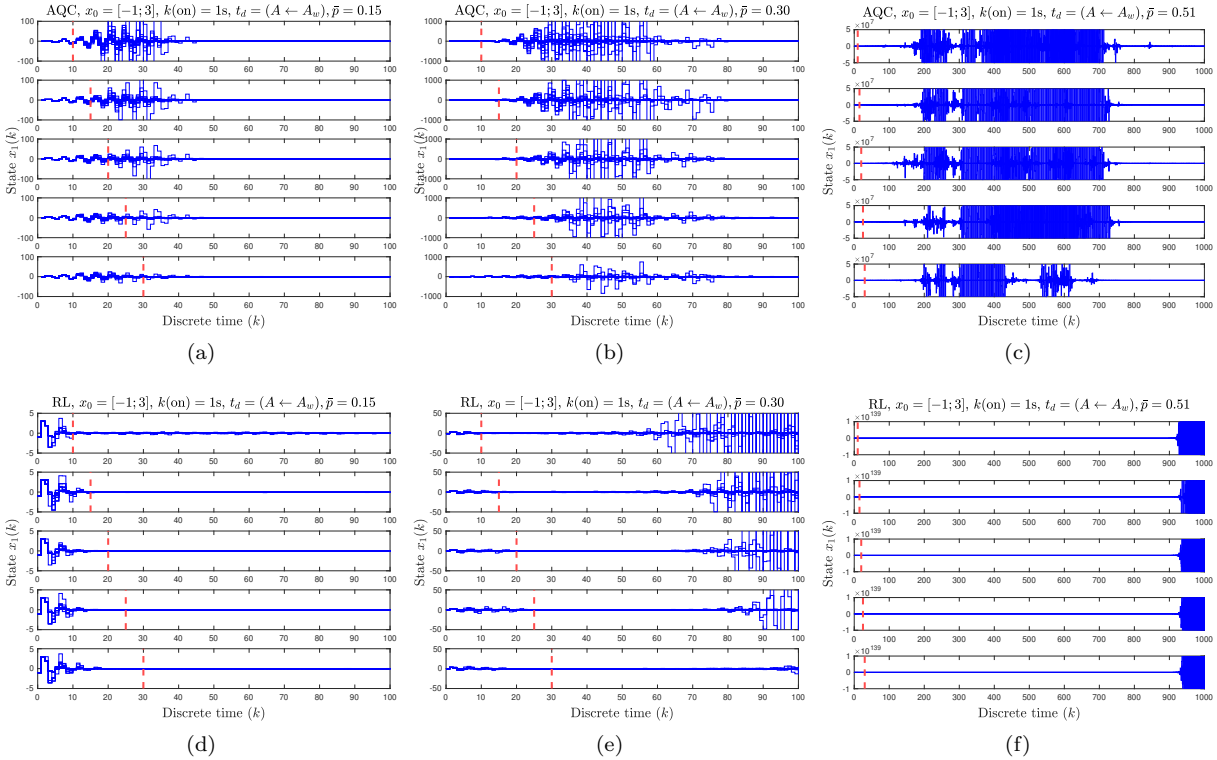


Figure 6: Comparison of AQC and DRL under varying levels of packet loss $\eta(k)$.

learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[4] X. Chen and A. Ray, "Deep reinforcement learning control of a boiling water reactor," *IEEE Transactions on Nuclear Science*, vol. 69, no. 8, pp. 1820–1832, 2022.

[5] H. Li, Q. Zhang, and D. Zhao, "Deep reinforcement learning-based automatic exploration for navigation in unknown environment," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 2064–2076, 2020.

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb 2015.

[7] 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.

[8] N. Elia and S. Mitter, "Stabilization of linear systems with limited information," *IEEE Transactions on Automatic Control*, vol. 46, no. 9, pp. 1384–1400, 2001.

[9] M. Fu and L. Xie, "The sector bound approach to quantized feedback control," *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1698–1711, 2005.

[10] T. Hayakawa, H. Ishii, and K. Tsumura, "Adaptive quantized control for linear uncertain discrete-time systems," *Automatica*, vol. 45, no. 3, pp. 692–700, 2009.

[11] M. Siami, T. Hayakawa, H. Ishii, and K. Tsumura, "Adaptive quantized control for linear uncertain systems over channels subject to packet loss," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 4655–4660, 2010.

[12] T. Hayakawa, H. Ishii, and K. Tsumura, "Adaptive quantized control for nonlinear uncertain systems," *Systems & Control Letters*, vol. 58, no. 9, pp. 625–632, 2009.

[13] 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.

[14] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on Machine Learning* (E. P. Xing and T. Jebara, eds.), vol. 32 of *Proceedings of Machine Learning Research*, (Bejing, China), pp. 387–395, PMLR, 22–24 Jun 2014.

[15] 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.

[16] Y. Qin, M. Cao, and B. D. O. Anderson, "Lyapunov criterion for stochastic systems and its applications in distributed computation," *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 546–560, 2020.

[17] H. J. Kushner, "A partial history of the early development of continuous-time nonlinear stochastic systems theory," *Automatica*, vol. 50, no. 2, pp. 303–334, 2014.

[18] 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.