IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 March 2024, accepted 25 March 2024, date of publication 1 April 2024, date of current version 5 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3382993

RESEARCH ARTICLE

Optimal PD Control Using Conditional GAN and Bayesian Inference

IVAN HERNANDEZ[1], WEN YU[1], (Senior Member, IEEE), AND XIAOOU LI[2], (Senior Member, IEEE)

[1]Departamento de Control Automático, Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV-IPN), Mexico City 07360, Mexico

[2]Departamento de Computación, Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV-IPN), Mexico City 07360, Mexico

Corresponding author: Wen Yu (yuw@ctrl.cinvestav.mx)

This work was supported in part by the research projects under Grant CONACyT-A1-S-8216 and Grant CONACyT CF-2023-I-2614.

ABSTRACT PD control is a widely used model-free method; however, it often falls short of guaranteeing optimal performance. Optimal model-based control, such as the Linear Quadratic Regulator (LQR), can indeed achieve the desired control performance, but only for known linear systems. In this paper, we present a novel approach for designing optimal PD control for unknown mechanical systems. We utilize a conditional Generative Adversarial Network (GAN) and a Long Short-Term Memory (LSTM) neural network to approximate an optimal PD control. We employ Bayesian inference to generate PD control that can be applied at different operating points. This design mechanism ensures both stability and optimal performance. Finally, we apply this control methodology to lower limb prostheses, and the results demonstrate that the optimal PD control, using GAN and Bayesian inference, outperforms other classical controllers.

INDEX TERMS Optimal PID, Bayesian inference, generative adversarial network, deep learning.

I. INTRODUCTION

The design of controllers has traditionally focused on improving the performance of classical controllers, such as linear PD and PID controllers, which are widely used in industry due to their well-known advantages. However, recent research has explored non-linear structures and the integration of advanced techniques like artificial intelligence to improve the performance of controllers in the face of external disturbances and parameter changes.

Some examples of such research include [1], which mapped a PD controller onto a non-linear structure based on fuzzy logic, achieving minimization of trajectory following error. Reference [2] proposed the identification of an online dynamic system to improve the performance of a PD controller and demonstrated robustness to external disturbances and parametric changes. Reference [3] showed improvements in trajectory tracking under parametric uncertainty by adding a compensator designed with neural networks to a PD controller. Similarly, [4] added a disturbance cancellation term to a PD controller, defining three different cancellation functions.

In particular, several investigations have explored optimal PD upgrades based on Linear Quadratic Regulator (LQR). For example, [5] presented a model-free design of a stochastic LQR controller for linear systems subjected to Gaussian noise, from the perspective of primal-dual optimization, which provides insight into understanding common Reinforcement Learning (RL) algorithms. Reference [6] proposed an online iterative learning LQR with adaptive iterative learning control to control the trajectory tracking of a leg exoskeleton for rehabilitation. Reference [7] presented an LQR controller based on a PD optimization controller to improve the dynamic performance of an Automatic Voltage Regulation (AVR) system. A biogeography-based optimization (BBO) was used to tune controller gains, and the Mean Absolute Percentage Error (MAPE) cost function was used to ensure effective performance. Similarly, [8] introduced a hybrid control methodology through the combination of a traditional PID controller and an LQR optimal controller.

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaojie Su.

The gain parameters of the classic PID controller were determined using the elements of the LQR feedback gain matrix. The performance of the LQR controller was improved using the genetic algorithm optimization method, which was adopted to obtain optimal values for the gain of the LQR controller parameters. The PD-LQR controller has been improved to work with non-linear systems, and some improvements have been made to adapt to parameter changes, such as [6]. However, there are currently no methods that generate controllers adaptable to structural changes of systems, especially those with more complex architectures.

Recently, the integration of artificial intelligence and control theory has seen significant development, allowing for the direct design of intelligent controllers. For example, [9] applied neural networks to model uncertain nonlinear functions and designed a controller by combining backstepping and adaptive control. Similarly, [10] adjusted controller parameters using the convergence of synaptic weights of neural networks based on radial basis functions and adaptive control with a reference model. Multilayer perceptron-based neural networks and recurrent neural networks have been used for modeling and identification of dynamic systems by [11] and [12], and high-order recurrent neural networks have been applied in adaptive control by [13], as well as in reinforcement learning, where their analogy with optimal control has been observed and exploited by [14] and [15].

The original GAN design was developed for generating images from a latent space of noise [16]. To generate an image with specific characteristics, conditional GANs were introduced by [17], where the discriminator recognizes conditional information as a necessary feature in incoming images. Some of the most significant works on GAN, primarily focused on image applications, include [18], where antagonistic generative deep convolutional networks (DCGAN) were proposed to improve the performance of convolutional neural networks (CNNs) in image representation; [19] presents techniques to address the instability problems of GAN networks; in [20], C-GAN was applied to generate images conditioned to other reference images [21]. C-GAN resolved the issue of encountering a condition that is not directly related to an output by applying a low pass filter on the gains produced [17]. In [22], LSTM-CGAN was used to generate song lyrics.

In this paper, we aim to explore and analyze the use of GAN in controller design. We first develop a reference framework for implementing GAN in nonlinear system control. As highlighted in works by [22] and [23], defining useful truth spaces is necessary when dealing with problems beyond conventional GAN applications. In our case, these truth spaces are useful in controlling nonlinear systems. We expand the latent space by adding the input (control signal) and output (tracked trajectory) of the nonlinear system. We exploit the capability of GANs to learn and produce adaptable control schemes for parameter changes through the particularity of C-GAN networks. Additionally, we use GANs to generate controllers capable of adapting to more complex systems than those with which they were trained, provided that there is some structural compatibility. To test this capability, we train the C-GAN network with a truth space provided by information concerning a simple pendulum. Later, we use the trained network to control a lower extremity prosthesis testing robot.

The primary contribution of this paper is the development of a novel controller that can generate control signals for complex systems, such as the n-link rigid robot, after being trained with a simple nonlinear system like the pendulum. This controller's adaptability to various systems is achieved by utilizing the GAN's conditional capacity, which is based on the system's stimulus-response. The paper's contributions can be further detailed as follows:

1) The proposed PD control ensures optimal performance through GAN and neural network training.
2) The proposed mechanism offers a new approach to model-free optimal control.
3) An optimization theory-based model-free stochastic LQR for nonlinear systems is derived through zero-sum games.

II. STABLE OPTIMAL PD CONTROL

The dynamics of a serial n-link rigid robot manipulator can be described by the following equation:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F = \tau \tag{1}$$

where $q \in \Re^n$ denotes the links positions, $\dot{q} \in \Re^n$ denotes the links velocity, $M(q) \in \Re^{n\times n}$ is the inertia matrix, $C(q,\dot{q}) \in \Re^{n\times n}$ is the centripetal and Coriolis matrix, $G(q) \in \Re^n$ is the gravity vector, $F \in R^n$ is the frictional term and the other disturbances. $\tau \in \Re^n$ is the input control vector.

A classical industrial PD control law for the robot is given by

$$\tau_{PD} = -K_p(q - q^d) - K_d(\dot{q} - \dot{q}^d) \tag{2}$$

where K_p and K_d are positive definite constant matrices that correspond to proportional and derivative coefficients, respectively. q^d is the desired joint position, while $\dot{q}^d \in \Re^n$ represents the desired joint velocity. In regulation case, the desired position is constant, i.e., $\dot{q}^d = 0$.

The following proof shows that the position regulation error of the PD control law (2) is bounded within a ball with radius $\bar{d}$. We use the Lyapunov function candidate as

$$V_{PD} = \frac{1}{2}\dot{q}^T M\dot{q} + \frac{1}{2}\tilde{q}^T K_p\tilde{q}, \quad \tilde{q} = q - q^d \tag{3}$$

The n-link rigid robot (1) has the following property [24],

$$\dot{q}^T\left[\dot{M}(q) - 2C(q,\dot{q})\right]\dot{q} = 0 \tag{4}$$

We use the following matrix inequality

$$X^T Y + \left(X^T Y\right)^T \leq X^T\Lambda^{-1}X + Y^T\Lambda Y \tag{5}$$

where $X, Y, \Lambda \in \Re^{n\times k}$ are any matrices, Λ is any positive definite matrix, to $\dot{q}^T (G+F)$, then

$$\dot{q}^T (G+F) \le \dot{q}^T K_1 \dot{q} + (G+F)^T K_1^{-1} (G+F) \quad (6)$$

where $K_1 > 0$

Using (4) and (6), the derivative of (3) is

$$\dot{V}_{PD} = -\dot{q}^T K_d \dot{q} + \dot{q}^T (G+F) \le -\dot{q}^T (K_d - K_1)\dot{q} + \bar{d} \quad (7)$$

where $(G+F)^T K_1^{-1} (G+F) \le \bar{d}$, $\bar{d}$ can be regarded as upper bound of $G+F$. If we choose $K_d > K_1$, the regulation error $\tilde{q}$ is bounded (stable), and $\|\dot{q}\|_{(K_d-K_1)}$ converges to $\bar{d}$.

So, for any bounded disturbance F, the PD control (2) with $K_p > 0$ and $K_d > K_1$ ensures stability in the closed-loop system. However, the performance may vary across different operating points and in the presence of different disturbances. The goal of the paper is to design an stable and optimal controller, which will satisfy both stability conditions, $K_p > 0$ and $K_d > K_1$, and an optimal index, such as LQR,

$$\min_u J = \min_u \int_0^\infty (x^T Q x + u^T R u) dt \quad (8)$$

where the state $x = \left[q^T, \dot{q}^T\right]^T$, the control $u = \tau$ in (1), R and Q are given positive definite constant matrices.

In order to obtain the optimal control for the robot (1), we first rewrite (1) in the following nonlinear system,

$$\dot{x} = f(x) + g(x)u \quad (9)$$

where $x = \left[q^T, \dot{q}^T\right]^T$, $f(x) = \left[\dot{q}^T, -M^{-1}(C\dot{q} + G + F)\right]^T$, $g(x)u = \left[0, 0, M^{-1}(q)\tau\right]^T$. (9) can be linearized at an operating point $x_1 = (q_1, \dot{q}_1)$ as

$$\dot{x} = A_1 x + B_1 u \quad (10)$$

where

$$A_1 = \frac{\partial f(x)}{\partial x}|_{x=x_1}, \quad B_1 = \frac{\partial g(x)u}{\partial u}|_{x=x_1}$$

The optimal control is

$$u_1 = -K_1 x, \quad K_1 = R^{-1} B_1^T P$$
$$A_1^T P + P A_1 + Q - P B_1 R^{-1} B_1^T P = 0 \quad (11)$$

However, this paper deals with unknown quantities such as $M(q)$, $C(q,\dot{q})$, $g(q)$, and $F(\dot{q})$ in the robot model (1). Consequently, $f(x)$, $g(x)$, A, and B also remain unknown for any operating point. To address this, we employ A_1 and B_1 from (10) for designing a model-based optimal PD controller. Subsequently, we utilize the optimal controller (11) to develop a model-free optimal PD control using deep learning methods.

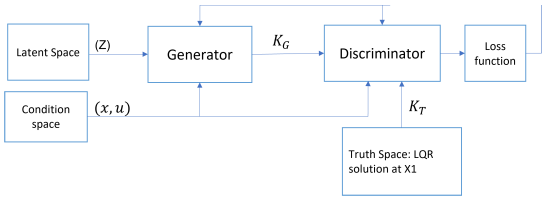


FIGURE 1. C-GAN based optimal PD control.

III. OPTIMAL PD CONTROL USING CONDITIONAL GENERATIVE ADVERSARIAL NETWORK

The optimal PD control utilizing LQR requires the linearized system (10) at a specific operating point x_1. When the robot model (10) is partially unknown, or the operating points are altered, the LQR controller becomes unavailable, necessitating the design of a new optimal controller. To tackle this issue, we propose a gain generation scheme based on a conditional generative adversarial network (C-GAN) that provides a data-driven optimal PD control, independent of the robot model or operating points. In this approach, we employ a generative adversarial network (GAN) to realize the optimal PD control.

The fundamental idea is to use a model to learn the probability distribution of the optimal controller at the operating point x_1. We use GAN to generate $p(u_i)$. We use a two-player game [25], wherein one player is a generator that creates samples aiming to have the same distribution as the training data, while the other player is a discriminator that evaluates the samples to determine their authenticity. The generator is trained to deceive the discriminator.

To generate the controller gains using input (u) and output (x) data, a conditional GAN (C-GAN) is utilized. The C-GAN comprises six essential blocks: latent space (noise), conditioning signal space, truth space (mode-based LQR), generator, discriminator, and loss function. The structure of the C-GAN optimal PD control is illustrated in Figure 1.

The basic idea of a GAN is a two-player game [25]. One player is a generator that creates samples intended to have the same distribution as the training data, while the other player is a discriminator that examines the samples to determine if they are real or fake. The generator is trained to deceive the discriminator. To generate the controller gains using input and output data, a conditional GAN (C-GAN) is used. The C-GAN includes 6 essential blocks: latent space (noise), conditioning signal space, truth space (mode-based LQR), generator, discriminator, and loss function.

1) Latent space. The latent space is noise space, it is

$$Z = \{z_i\}, \quad z_i \sim \mathcal{N}(0,1) \quad (12)$$

where z_i are components, z_i are normally distributed random numbers with normalized amplitude under $Z = \frac{Z}{max\{|Z|\}}$.

2) Conditioning signal space. The C-GAN utilizes information from the dynamic system in two ways. Firstly, when the parameters of the dynamic system can be identified, C-GAN can directly use the parameter vector as a conditioning vector. Secondly, when the parameters are not

available, C-GAN can utilize input-output data as conditional information. Both of these approaches can be considered as processes of nonlinear system modeling. In this paper, we assume that the parameters are unknown, and therefore, we employ the control signal u applied to the system as an input signal, and the state of the system x as the output to establish the conditional signals.

3) Truth space: The truth gains are the solution of optimal PD control based on (A_1, B_1) and the Riccati equation. The truth probabilistic space is defined as (K_T, E_T, P_T), where K_T is the sample space of truth gains, E_T is the event variable at the operating point x_1, and P_T is the probability function given by

$$P_T : E_T \to [0, 1], \quad P_T(E_T) = \int p_r d\tau$$

where p_T is the probability distribution of the truth gains, $E_T \sim p_T$

4) Generator: The generated probabilistic space is defined as (K_G, E_G, P_G), where K_G is the sample space of generated gains, E_G is the event that stabilizes the nonlinear system in the neighborhoods of the operating point x_1, and P_G is the probability function given by

$$P_G : E_G \to [0, 1], \quad P_G(E_G) = \int p_G d\tau$$

where p_G is the probability distribution of the generated gains, $E_G \sim p_G$. The generator is an application that associates an input of the latent space Z as in (12), with a gain of the space K_G, conditioned by the control u and the state x,

$$G : Z \times U \times X \to K_G \tag{13}$$

5) Discriminator: The discriminator is responsible for distinguishing between the input vectors $E_T \in K_T$ and $E_G \in K_G$. It computes the probability $p(a \in K)$ of correctly identifying whether a comes from the true distribution K_T or the generator's distribution K_G. It also takes as input the conditioning signals for both input and output, which complement the generated gain vector. The discriminator produces "valid" gains if its distribution matches that of the true distribution. This mapping can be expressed as follows:

$$D : K \times U \times X \to P \tag{14}$$

where P is a probability. The discriminator is defined as a function that takes a gain $k \in K$ as input and outputs the probability that $k \in K_T$. This function can be parameterized with θ^D as:

$$p(k \in K_T) = D(k, u, x; \theta^D) \tag{15}$$

or

$$p(k \in K_G) = 1 - D(k, u, x; \theta^D) \tag{16}$$

6) Two-Player Game: The two-player game is represented by equations (26) and (15), which are both differentiable with respect to their inputs and parameters. Each player has a cost function that depends on the parameters of both players. The discriminator aims to minimize $J^D(\theta^D, \theta^G)$ by optimizing over θ^D alone [26]. Conversely, the generator seeks to minimize $J^G(\theta^D, \theta^G)$ by adjusting its own parameters θ^G only. We define $\theta^D \in \Theta^D$ and $\theta^G \in \Theta^G$ as the discriminator and generator strategies, respectively. The strategy spaces are denoted by Θ^D and Θ^G.

We define the probability distribution function of the generated space p_G as a function parameterized by the parameters θ^G, $p_G(K_T, \theta^G)$. The training goal is to estimate θ^G, which can be achieved by maximizing the likelihood between the spaces K_T and K_G:

$$\theta^{G*} = \arg\max_{\theta^G} \mathbb{E}_{K_T \sim p_T} \log p_G(K_T; \theta^G), \tag{17}$$

which can be considered as a minimization of the divergence KL

$$\theta^{G*} = \arg\min_{\theta^G} D_{KL}(p_T(K_T) || p_G(K; \theta^G)). \tag{18}$$

where D_{KL} is the Kullback-Leibler divergence (KL distance), defined by

$$KL\left(p\left(a \mid x\right) \| p\left(b \mid x\right)\right) = \sum_{i}^{n} p_i\left(b \mid x\right) \log\left(\frac{p_i\left(b \mid x\right)}{p_i\left(a \mid x\right)}\right) \tag{19}$$

Then the generator produces K_G with the same probability distribution of p_T

$$p_T(K) = p_G(K, \theta^G) \tag{20}$$

The player cost functions are

$$J^D, J^G : \theta^D \times \theta^G \to \mathcal{R} \tag{21}$$

Then we have a local Nash equilibrium (θ^D, θ^G) if

$$\frac{\partial J^D}{\partial \theta^D} = 0, \quad \frac{\partial J^G}{\partial \theta^G} = 0 \tag{22}$$

and

$$\frac{\partial^2 J^D}{\partial \theta^{D2}} \geq 0, \quad \frac{\partial^2 J^G}{\partial \theta^{G2}} \geq 0 \tag{23}$$

Since the discriminator function can be interpreted as a binary classifier to distinguish between true and false, it is beneficial to use the cross-entropy function for binary classification as the cost function for the discriminator, as suggested by [25]. The cross-entropy function can be defined as follows:

$$J^D = E_{k_T \sim p_T(k_T)}[logD(k_T, u, y; \theta^D)] + E_{z \sim p_z(z)}[log(1 - D(G(z, u, y; \theta^G)), u, y)]$$

Then,

$$J^D = E_{k_T \sim p_T(k_T)}[log(p(k_T \in K_T)] + E_{z \sim p_z(z)}[log(p(k_G \in K_G)]$$

Considering the game as zero-sum,

$$J^G + J^D = 0 \tag{24}$$

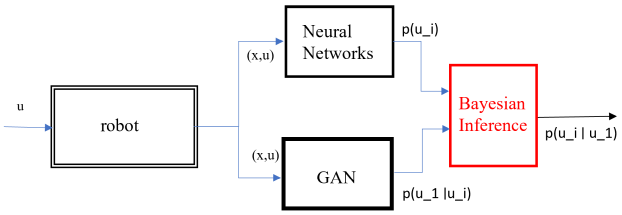


FIGURE 2. Optimal PD control using neural networks.

The objective function of the GAN is

$$\min_G \max_D V(D, G), \quad V(D, G) = J^G = -J^D \tag{25}$$

Once the GAN has been trained, G becomes a mapping from the latent space to the generated gain space, conditioned by the response of a dynamic system. Specifically,

$$K_G = G(Z, u, x, \theta^G) \tag{26}$$

where θ^G is the parameter vector.

IV. OPTIMAL PD CONTROL USING NEURAL NETWORKS

The above GAN method provides us with the probability distribution $p(u_1)$ of the optimal controller. However, we require the probability distribution under different operating points x or various unknown uncertainties, denoted as $p(u_i \mid u_1)$. The use of these signals as conditioning factors for the GAN. By employing the signals (u, x), the generator can identify the system online and produce a gain corresponding to an LQR controller, without requiring the linearization of the network or the identified model, as in the case of instant linearization [27].

To achieve this, we employ Bayesian inference to generate the probability distribution of the optimal controller $p(u_i \mid u_1)$ at different operating points x_i

$$\begin{aligned} p(u_i \mid u_1) &= p(x \mid u_i)\, p(u_i) = \frac{p(u_1 \mid u_i)\, p(u_1)}{\sum p(x)} \\ &\propto p(u_1 \mid u_i)\, p(u_1) \end{aligned} \tag{27}$$

where u_i is the optimal controller at the operating point x_i, or at the operation condition (with unknown disturbances). $p(u_i \mid u_1)$ is the probability property (posterior distributions) of u_i under the the probability distribution u_1. $p(u_1)$ is from the GAN model (prior distribution), $p(u_1 \mid u_i)$ is the likelihood, which will be modeled by deep neural networks in the case of u_1. The structure of the GAN based method is shown in Figure 2.

We use the following neural network model to generate the likelihood $p(u_1 \mid u_i)$. The inputs to the generator of GAN (u, x), are also the input to the neural networks. LSTM is a suitable method for identifying nonlinear systems when the data is sequential [28]. The assumption that (u, x) are sufficient for identifying the system. Because

$$p(a \mid b) = \frac{p(b, a)}{p(b)}, \quad E[b|a] = \sum_x bp(b|a) \tag{28}$$

the neural network modeling in fact is to minimize the likelihood distribution error as,

$$L = \prod_{i=1}^{n} p\left(\theta^i, x^i\right) = \prod_{i=1}^{n} p\left(\theta^i | x^i\right) p\left(x^i\right) \tag{29}$$

The objective of calculating $p(x \mid u_i)$ for the neural network is to update the weights value. In order to maximize the likelihood, we use the logarithm cost function,

$$E = -\ln L = -\sum_{i=1}^{n}\sum_{k=1}^{N} \ln p\left(\theta_k^i | x^i\right) - \sum_{i=1}^{n} \ln p(x^i) \tag{30}$$

where N is the training data number.

1) Given that the control u and the system response x are available, the structure of the GAN for identification can be implemented as follows:

$$\hat{x} = \sigma(W_O(k)[x(k-1), u(k)]) \tanh(x(k)) \tag{31}$$

where

$$x(k) = \sigma(W_F(k)[x(k-1), u(k)])x(k-1) \tag{32}$$

where W_F and W_O are the weights of the LSTM. More details of the (31) and (32) equations are found in [28].

2) To compensate for the mapping between the identified system and the gain K_G, multilayer perceptrons are used as

$$\hat{K}_G = F(W\Psi) \tag{33}$$

where W is the weight matrix and $\Psi = [u, x]^T$ is the input vector.

The calculation process is shown in Algorithm 2.

Algorithm 1 Optimal PD Control Using Neural Networks

1: Sampled $p(u_1)$ from the GAN model
2: Calculated posterior $p(u_i|u_1|)$
3: Calculate the $\hat{x}(k)$ from neural model
4: Use $\hat{x}(k)$ as the observations to construct $p(u_1||u_i)$
5: Use $p(u_1||u_i)$ to actualize the prior $p(u_i)$
6: Calculate the new posterior $p(u_i|u_1|)$
7: Analyze the KL distance like $KL = -ln[p(u_i|p(u_1)/p(u_1)]$ using the actual distribution on the data $p(u_i)$

Given that the stochastic gains $p(u_i)$ from GAN, it is expected that the optimal gains will align with the characteristics of the identified system $p(x \mid u_i)$. Moreover, we have investigated the potential of achieving excellent tracking performance for a more complex system than the one used to train the GAN. This approach aims to develop an adaptable controller capable of handling variations not only in system parameters but also in system complexity, leveraging the nature of GANs. Importantly, the generator produces gains based on the same distribution as the truth space formed by LQR-conditioned controllers.

V. ANALYSIS OF CONVERGENCE AND STABILITY

Analyzing convergence and stability is a fundamental aspect when designing and implementing control systems. In the context of the proposed LSTM-GAN optimal PD control, it is crucial to ensure that the system remains stable and converges towards the desired output. The convergence of the generator and discriminator networks plays a pivotal role in achieving optimal gains to stabilize the system and ensure its high performance. Therefore, a comprehensive analysis of the LSTM-GAN's convergence and stability is essential to assess the effectiveness and reliability of this control approach.

Several methods can be employed to scrutinize the convergence and stability of the LSTM-GAN, including examining training curves, evaluating loss functions, and monitoring the generator and discriminator networks' outputs. Additionally, techniques such as Lyapunov stability analysis can be used to assess the stability of the closed-loop system. Conducting a thorough analysis of convergence and stability offers valuable insights into the performance and dependability of the LSTM-GAN optimal PD control, thus facilitating its practical implementation.

Nonlinear dynamic systems can be represented by several piecewise linear dynamic systems through the piecewise affine modeling technique. This approach involves partitioning the state space of the nonlinear system into multiple regions and approximating the nonlinear dynamics with linear models within each region. Each region corresponds to a specific operating points under which the system dynamics can be considered linear.

We define the linearized system at the time slot $\tau_i = [t_0^i, t_f^i)$ as

$$f_i = A_i x + B_i u + \zeta_i \tag{34}$$

where $A_i = \frac{\partial f(x,u)}{\partial x}|_{x=x(t_0^i)}$, $B_i = \frac{\partial f(x,u)}{\partial u}|_{x=x(t_0^i)}$, and ζ_i is linear approximation error. We define the Heaviside unitary function as $rect(\tau) = H(t - t_0) - H(t - t_f)$. With these definitions,

$$f_i = f(x,u,t)rect(\tau_i) = \begin{cases} 0, & t < t_0^i \\ f(x,u,t) & t_0^i < t < t_f^i \\ 0 & t > t_f^i \end{cases} \tag{35}$$

So only in the time slot τ_i, f_i is equal to f. In a finite time interval $(0, T)$, there are finite time slot τ_i, $i = 1 \cdots N$,

$$\dot{x} = f(x,u,t) = \sum_{i=1}^{N} f(x,u,t)rect(\tau_i) = \sum_{i=1}^{N} (A_i x + B_i u) + \zeta \tag{36}$$

where $\zeta = \sum_{i=1}^{N} \zeta_i$. The accuracy ζ of the approximation depends on the choice of partitioning and the quality of linearization at each region.

Lemma 1: For two random variables $k_T \in K_T$ and $k_G \in K_G$, their distribution functions are $\phi_{p_T} \in p_T$ and $\phi_{p_G} \in p_G$. If their distributions are similar as

$$||\phi_{p_T} - \phi_{p_G}|| < \delta \tag{37}$$

where $||\phi_{p_T} - \phi_{p_G}|| = \sqrt{\left(\sum_i^2 (\phi_{i,1} - \phi_{i,2})^2\right)}$ is the Frechet Inception Distance (FDI), $\delta > 0$, and high-order distance is bounded as

$$\xi + \delta < \epsilon$$

where δ is the bound of FDI, ξ is the distance norm of higher-order moments, ϵ the distance norm bound of all moments, then k_G and k_T satisfy

$$||k_G|| \leq ||\epsilon|| + ||k_T|| \tag{38}$$

Proof: If two distributions p_T and p_G are equal, then

$$||\phi_{p_T} - \phi_{p_G}|| = \lim_{n\to\infty} \sqrt{\left(\sum_{i=1}^{n} (\phi_{i,p_T} - \phi_{i,p_G})^2\right)} = 0 \tag{39}$$

If the distributions are not equal, but similar, the Frechet Inception Distance (FDI) is

$$||\phi_{p_T} - \phi_{p_G}|| = \sqrt{\left(\sum_{i=1}^{2} (\phi_{i,p_T} - \phi_{i,p_G})^2\right)} + \lim_{n\to\infty} \sqrt{\left(\sum_{i=3}^{n} (\phi_{i,p_T} - \phi_{i,p_G})^2\right)} = 0 \tag{40}$$

Combine (37) and (40),

$$||\phi_{p_T} - \phi_{p_G}|| = \sqrt{\left(\sum_{i=1}^{2} (\phi_{i,p_T} - \phi_{i,p_G})^2\right)} + \xi < \delta + \xi < \epsilon$$

This means $\forall a \in K_T \exists \bar{a} \in K_G$, then $||p_T(a) - p_G(\bar{a})|| \leq \epsilon$,or $||k_T - k_G|| \leq \epsilon$. Using the triangle inequality

$$||k_T - k_G|| \leq ||k_T|| + ||k_G|| \leq \epsilon \tag{41}$$

(41) is (38). ■

Theorem 1: If the distribution of LQR-PD ϕ_{p_T}, and the distributions GAN-PD ϕ_{p_G}, are similar as (37), then the GAN-PD control

$$u = -k_G x \tag{42}$$

can stabilize the linearization model

$$\dot{x} = A_t x + B_t u \tag{43}$$

asymptotically.

Proof: For the linear system (43), we define the Lyapunov function as

$$V = x^T P x, \quad P = P^T > 0 \tag{44}$$

Applying C-GAN control (42) to the linearization model (10) or (43),

$$\dot{V} = (A_t x - B_t k_G x)^T P x + x^T P (A_t x - B_t k_G x) \tag{45}$$

From Lemma 1, k_G satisfies (41),

$$||k_G|| = ||\epsilon|| + ||k_T||$$

where $\epsilon = k_G - k_T$, is the error between the LQR PD control and C-GAN PD control. The LQR PD is

$$u = -k_T x, \quad k_T = R^{-1}B^T P \tag{46}$$

where P satisfies the Riccati equation,

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{47}$$

$Q \geq 0$ and $R > 0$ are the matrices in the cost function

$$J = \int_0^\infty x^T Qx + u^T Rudt$$

Because $\epsilon >, k_T > 0$,

$$k_G = \epsilon + k_T$$

Using (46) and (47), (45) becomes

$$\begin{aligned}\dot{V} &= x^T(A_t^T P - \epsilon^T B_t^T P - (R^{-1}B_t^T P)^T B_t^T P \\ &\quad + PA_t - PB_t\epsilon - PB_t R^{-1}B^T P)x \\ &= x^T(-\epsilon^T B_t^T P - k_T^T B_t^T P - PB_t\epsilon - Q)x \\ &\leq x^T(\epsilon^T\epsilon + PBB^T P - k_T^T B_t^T P - Q)x \\ &\leq -x^T Wx\end{aligned}$$

Using LaSalle lemma, x is asymptotically stable ■

Theorem 2: If the distribution of LQR-PD ϕ_{p_T}, and the distributions GAN-PD ϕ_{p_G}, are similar as (37), then the GAN-PD control

$$u = -k_G x \tag{48}$$

can stabilize the nonlinear system (1) or (9) and have the optimal performance (8).

Proof: We definite the Lyapunov function as

$$V = x^T Px, \quad P = P^T > 0 \tag{49}$$

From Lemma 1 the nonlinear system can be expressed as the following piecewise linearization systems

$$\dot{x} = \lim_{n\to\infty}\sum_{t=1}^{n}(A_t x + B_t u) + \zeta \tag{50}$$

We calculate the derivative of (49) along with (50),

$$\dot{V} = \lim_{n\to\infty}\left\{\begin{array}{c}\left[\sum_{t=1}^{n}(A_t x + B_t u) + \zeta\right]^T Px \\ +x^T P\left[\sum_{t=1}^{n}(A_t x + B_t u) + \zeta\right]\end{array}\right\}$$

Using the GAN-PD control (48),

$$\dot{V} = \lim_{n\to\infty}\left\{\begin{array}{c}(\sum_{t=1}^{n}(A_t x - B_t k_G x) + \zeta)^T Px \\ +x^T P(\sum_{t=1}^{n}(A_t x - B_t k_G x) + \zeta)\end{array}\right\} \tag{51}$$

From Lemma 2, k_G satisfies

$$\begin{aligned}\epsilon &= k_G - k_T \\ ||k_G|| &\leq ||\epsilon|| + ||k_T|\end{aligned}$$

Using Theorem 1, any $\Lambda = \Lambda^T > 0$, and Riccati equation

$$A^T P + PA - PBR^{-1}B^T P + P\Lambda P + Q = 0 \tag{52}$$

(51) becomes

$$\begin{aligned}\dot{V} &= \lim_{n\to\infty}\left\{\begin{array}{c}\sum_{t=1}^{n}x^T(A_t - B_t\epsilon - B_t k_T)^T Px \\ +\sum_{t=1}^{n}x^T P(A_t - B_t\epsilon - B_t k_T)x \\ +x^T P\zeta + \zeta^T Px\end{array}\right\} \\ &\leq \lim_{n\to\infty}\sum_{t=1}^{n}\left\{\begin{array}{c}x^T((A_t - B_t\epsilon - B_t k_T)^T P + \zeta^T\Lambda^{-1}\zeta \\ +P(A_t - B_t\epsilon - B_t k_T) + P\Lambda P)x\end{array}\right\} \\ &\leq \lim_{n\to\infty}\sum_{t=1}^{n}(x^T Qx + \zeta^T\Lambda^{-1}\zeta) \\ &\leq -\lim_{n\to\infty}\sum_{t=1}^{n}x^T Qx + \bar{\zeta}\end{aligned}$$

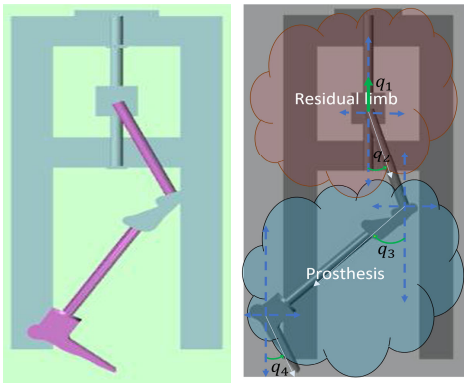
where $\bar{\zeta} = \zeta^T\Lambda^{-1}\zeta$ is the upper bound of LSTM approximation error. So

$$\lim_{t-\infty}\left(\lim_{n\to\infty}\sum_{t=1}^{n}\|x\|_Q^2\right) = \bar{\zeta}$$

All sates are bounded and convergence the upper bound of the approximation error. ■

VI. APPLICATION TO LOWER LIMB PROSTHESES

he lower limb prosthesis we have designed is an active control prosthesis, as described in [29]. It can be viewed as a 4-degree-of-freedom (4)-DOF) robot, as shown in Figure 3. The robot consists of one prismatic joint and three rotational joints, which correspond to the vertical movement of the hip and the rotations in the sagittal plane of the femur, knee, and ankle. The transformer prosthesis is part of the femur, while the lower part of the limb includes the knee, tibia, ankle, and foot, and the rest is the residual limb.



(a) Prostheses (b) 4-DOF robot

FIGURE 3. Active lower limb prostheses.

The active lower limb prosthesis can be represented by (1) and linearized using (9). For the first link, the matrices A and B are given by

$$\begin{aligned}A &= \frac{\partial f}{\partial x}\Big|_{x=x_0} = \begin{bmatrix}0 & 1 \\ \frac{9.8}{l}\cos q_1 & -\frac{1}{m}\end{bmatrix} \\ B &= \frac{\partial g}{\partial x}\Big|_{x=x_0} = \begin{bmatrix}0 \\ \frac{10}{ml^2}\end{bmatrix}\end{aligned} \tag{53}$$

because for the first DOF, it is a pendulum as

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\frac{g}{l}\sin x_2 - \frac{k}{m}x_2 + \frac{1}{ml^2}u \tag{54}$$

Here, $g = 9.8$ and $k = 1$. We use the pendulum model (54) to linearize each link of the prosthesis, as each joint does not move in a large workspace, and the pendulums can be assumed to have variable parameters. For example, changes in the center of mass can be considered as variations in the length of the pendulum.

The PD control law is given by

$$\tau = k_p e + k_d \dot{e}, \quad e = q_d - q \tag{55}$$

where e is the tracking error.

The GAN is trained using only data derived from the pendulum, and therefore, the generation of the truth space is accomplished through the application of an LQR control to the pendulum. This is because the pendulum serves as a simplified model for a more complex system like lower limb prosthetics. The objective is to verify whether the generator, trained with a simple pendulum, can adapt to a more complex system while maintaining a strong relationship in behavior. Each joint of the robot is viewed as a set of pendulums with variable parameters, where the movement of a lower link can be interpreted as a displacement of the center of mass of a pendulum, which can be further understood as a variable-length pendulum.

The truth space consists of the K_T values that produce the control by solving the Riccati equation and optimizing (8) for the following expression:

$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \tag{56}$$

with

$$Q_1 = 2877.8(ml)^2 + 22.765ml - 1.2698 \tag{57}$$
$$Q_2 = Q_1/10 \tag{58}$$

and $R = .1$. (57) was obtained through interpolation in such a way as to ensure a tracking error less than 0.05.

We train the system with different parameters: $l \in [0.2, 2]$, $m \in [0.4, 4]$, $x_0 \in [1, 6]$. There are 700 samples.

Input and output signals were generated by varying the excitation signal from 1 to 6 for each system, and the control signals u received by the system were measured. These control signals were considered as the input signal for training the GAN. The trajectory tracked by the system, denoted by y, was saved as the output signal for the training data. To obtain this output signal, the system was first stabilized with the PD obtained by LQR, using the parameters described above.

The Generator in Figure 1 has three inputs: the control signal u of the system, the tracked trajectory y, and a one-dimensional Gaussian noise vector z. It has two outputs, K_p and K_d. We used LSTM neural networks and two multilayer perceptrons (MLP) to learn them. The approximation structure consisted of an LSTM with 3 layers, each with 10 nodes, and an MLP with 2 layers, each with 2 nodes. The inputs to the discriminator were the gains K_p and K_d, as well as the input-output of the system (u, y). The loss function for the discriminator was defined as follows:

$$L_D = -E\left[log(D(k_T, u, y))\right] - E\left[log(1 - D(G(z, u, y), u, y)\right] \tag{59}$$

while the loss for the generator is defined as

$$L_G = -E\left[log(D(G(z, u, y), u, y)\right] \tag{60}$$

To improve the variability of the generator, a random change of associations between the truth space and the conditioning signals is considered. The probability of a random flip is set to $p_f = 0.2$. The optimization algorithm used in Figure 1 is Adam. The GAN-PD control is given by:

$$u = K_g(z, u, y)e \tag{61}$$

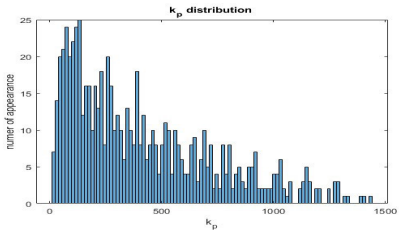
The histograms depicted in Figure 4 show the truth space and generated space. The Frechet Inception Distance (FDI) values for K_p and K_d are 31 and 56, respectively. These values are high, indicating that the variances are significant. Therefore, there is still room for improvement of the GAN to obtain a more appropriate generation.

Figure 5 depicts the scores obtained by the generator and discriminator throughout the training process. It can be observed that an optimal point has been reached in the two-player game, where the generator can no longer enhance the gains it generates and the discriminator can no longer improve its ability to differentiate between the generated K and the true K. Furthermore, it is evident that the generator score remains approximately constant at 0.4, while the discriminator score hovers around 0.5. This is attributed to the random parity disruptions between the truth space and the conditioning signals that were applied to achieve greater variability.

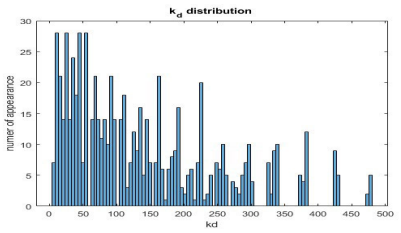
To validate the good performance of the proposed GAN-PD, we switched from a plastic to a steel prosthesis. Plastic prostheses are low-cost and are sufficient for functional implementation in various applications. Figure 6 shows the tracking results of the hip, thigh, knee, and ankle. We can see that GAN-PD is capable of satisfactorily controlling the prosthesis with plastic elements, maintaining small errors comparable to PD.

For the steel prosthesis, the densities of the materials were changed from $920kg/m^3$ to $7850kg/m^3$. Figure 7 shows the tracking results of the steel prosthesis with the same LSTM-GAN PD control. Clearly, classical PD control fails to follow the desired trajectory since this controller is a linear control that can only control the system for which it was designed and possibly one with slight variations. However, GAN-PD can stabilize the prosthesis after modifying the density of the materials without making any modifications to the controller.

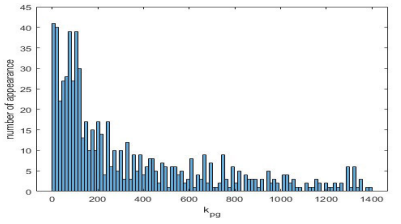
Finally, we compare our GAN-PD with several other controllers, including classical PD (PD), classical linear



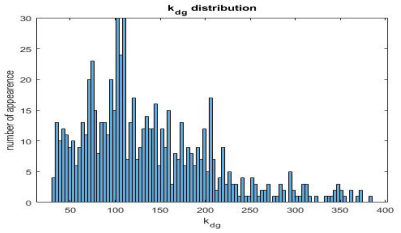
(a) Histogram of truth space k_p



(b) Histogram of truth space k_d



(c) Histogram of generated k_p



(d) Histogram of generated k_d

FIGURE 4. Histogram of gains.

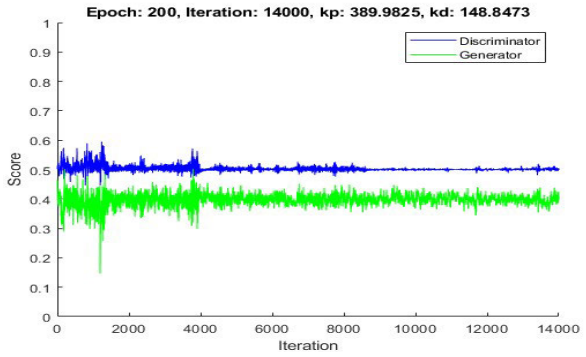
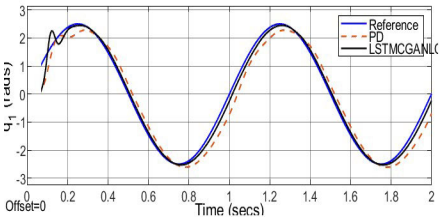
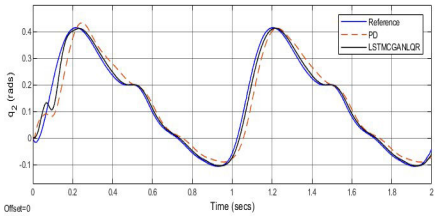


FIGURE 5. Scores of LSTM-GAN training.

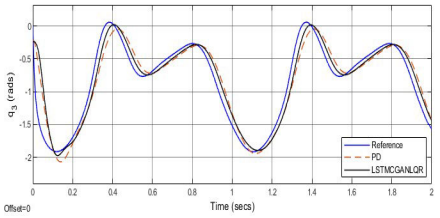
control (LQR), PD control with LSTM (PD-LSTM), and LSTM-based LQR control (L2). The results are summarized in Table 1. We can see that the PD, LQR, and PD-LSTM



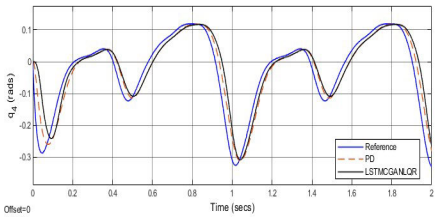
(a) Hip



(b) Thigh



(c) Knee



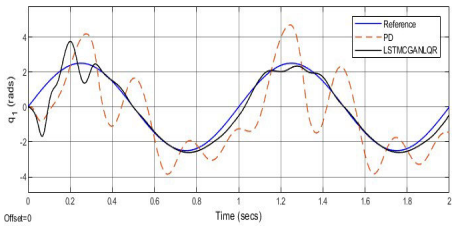
(d) Ankle

FIGURE 6. Plastic prosthesis.

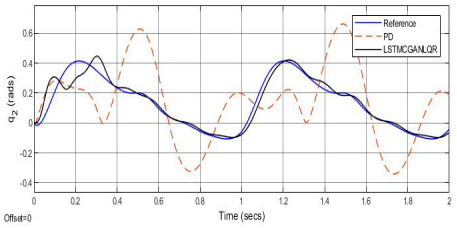
TABLE 1. Comparison of varying mass pendulum.

	m=0.5	m=2.5	m=6.5	
controller	e(t=10)	e(t=30)	e(t=70)	rmse
PD	0.0525	0.2261	0.4439	0.289209
LQR	0.0492	0.2138	0.4263	0.276805
PD+LSTM	0.0118	0.1929	0.4211	0.267504
NARMA-L2	0.0031	0.0223	0.0634	0.038844
LSTMCGAN LQR	0.0038	0.0178	0.04	0.025372

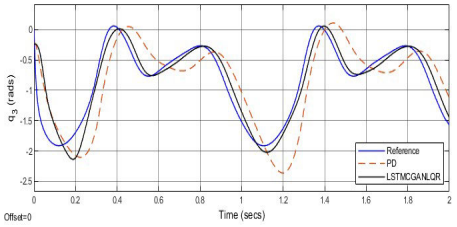
controllers have similar performance, with a large error when the mass changes. Among these three, the PD-LSTM shows better tracking when the mass corresponds to what was considered to design the controllers, but it fails to track the reference for different masses. On the other hand, L2 and GAN-PD show similar performances, but L2 exhibits oscillations at the instants of mass change, while GAN-PD is more stable and maintains a lower tracking error. Furthermore, it is observed that the error remains bounded according to the predefined performance in the truth space, i.e., an error



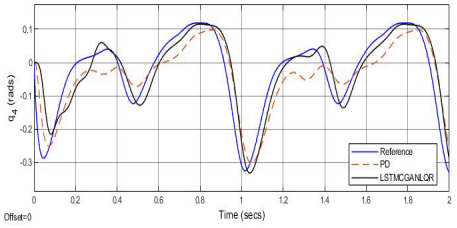
(a) Hip



(b) Thigh



(c) Knee



(d) Ankle

FIGURE 7. Steel prosthesis.

less than 0.05, as shown in Table 1. It is also observed that GAN-PD produces the smallest errors in the face of mass variations, while NARMA-L2 presents an error of less than 0.05 when the mass is $0.5kg$ and $2.5kg$, but not for the mass of $6.5kg$, where it presents a greater error of 0.06. These results demonstrate that the performance of GAN-PD is generally better than the controllers that were implemented for comparison. Figure 8 illustrates the comparison of the performance.

It is important to note that, in our implementation, even though the truth space is small and based solely on a particular nonlinear system, the pendulum, the generated controls are capable of adapting to a robot with four degrees of freedom, which not only includes rotational joints, more affine to the pendulum, but also prismatic joints. A great connection between the dynamics of both systems is possible, and the conditioning signals as the input and output of the system are adequate.

By comparing our work with previous studies on image-based GANs, we note that having a truth space with a wide

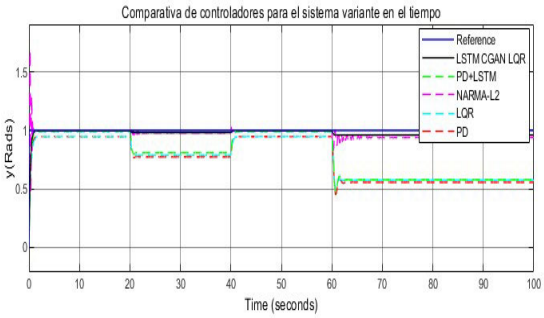


FIGURE 8. Comparison of the controllers.

variety of systems would allow for a control system that is adaptable to any system. In general, it is necessary to have a base structure that generates all the space of systems, making it possible to control any kind of system. This is one of the most important characteristics of GANs, the ability to generate, rather than just map like a standard neural network, and we exploit this property here. Alternatively, it is possible to define a characteristic that works as a conditioning signal that identifies any system. We believe that including inverse convolutional filters that receive the output and the input would generate that conditioning signal that identifies the system, since the response of the system is given as the convolution between the system and the input. The conditioning scheme can be extended so that the system is conditioned, for example, by the reference signal and the error, which could even generate control signals directly without the need for the base structure of the PD.

VII. CONCLUSION

In this paper, we propose a mechanism for designing optimal control using GAN and LSTM. Our proposed controller has numerous advantages over other controllers such as PID control, neural control, and other robust controllers. We explore the advantages and applicability of C-GAN, which allows conditioning signals to adapt to time-varying systems. We believe that this can be further extended by considering a database with a greater number of controllers and different types of controllers. It is possible to establish the weights of the LQR as conditions or choose between various controllers. With an additional intelligent system, we can obtain a scheme that chooses the most suitable control under certain desired conditions.

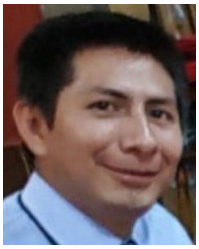
Currently, we are working on expanding the controller database, redefining the GAN architecture, and improving the conditioning signals. These improvements will result in a generator that can adapt to a wide range of systems and achieve better performance.

REFERENCES

[1] R. E. Meligy, A. H. M. Bassiuny, E. M. Bakr, and A. A. Tantawy, "Systematic design and implementation of decentralized fuzzy-PD controller for robot arm," *Global Perspect. Artif. Intell.*, vol. 1, no. 1, pp. 8–14, Jan. 2013.

[2] M. Ticherfatine and Q. Zhu, "Fast ferry smoothing motion via intelligent PD controller," *J. Mar. Sci. Appl.*, vol. 17, no. 2, pp. 273–279, Jun. 2018.

[3] R. Lopez, M. A. Llama, and R. García-Hernández, "Controlador PD con compensación neuro-adaptable aplicado a la dinámica de un RMR omnidireccional," in *Proc. Congreso Nacional Control Automático*, 2019, pp. 1–6.
[4] M. Haddar, R. Chaari, S. C. Baslamisli, F. Chaari, and M. Haddar, "Intelligent PD controller design for active suspension system based on robust model-free control strategy," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 233, no. 14, pp. 4863–4880, Jul. 2019.
[5] M. Li, J. Qin, W. X. Zheng, Y. Wang, and Y. Kang, "Model-free design of stochastic LQR controller from a primal-dual optimization perspective," *Automatica*, vol. 140, Jun. 2022, Art. no. 110253. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000510982200098X
[6] N. Ajjanaromvat and M. Parnichkun, "Trajectory tracking using online learning LQR with adaptive learning control of a leg-exoskeleton for disorder gait rehabilitation," *Mechatronics*, vol. 51, pp. 85–96, May 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741581830045X
[7] E. H. Jadoua, N. M. Diwood, and Y. F. Azeez, "The LQR based on optimized tuning PD controller for AVR system," *Int. J. Appl. Sci.*, vol. 5, no. 1, p. 8, Sep. 2022.
[8] I. K. Mohammed and A. I. Abdulla, "Elevation, pitch and travel axis stabilization of 3DOF helicopter with hybrid control system by GA-LQR based PID controller," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 2, p. 1868, Apr. 2020.
[9] H. Wang, P. X. Liu, J. Bao, X.-J. Xie, and S. Li, "Adaptive neural output-feedback decentralized control for large-scale nonlinear systems with stochastic disturbances," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 3, pp. 972–983, Mar. 2020.
[10] S. Slama, A. Errachdi, and M. Benrejeb, "Model reference adaptive control for MIMO nonlinear systems using RBF neural networks," in *Proc. Int. Conf. Adv. Syst. Electric Technol. (IC_ASET)*, Mar. 2018, pp. 346–351.
[11] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou, "High-order neural network structures for identification of dynamical systems," *IEEE Trans. Neural Netw.*, vol. 6, no. 2, pp. 422–431, Mar. 1995.
[12] A. Y. Alanis, E. N. Sanchez, A. G. Loukianov, and E. A. Hernandez, "Discrete-time recurrent high order neural networks for nonlinear identification," *J. Franklin Inst.*, vol. 347, no. 7, pp. 1253–1265, Sep. 2010.
[13] G. A. Rovithakis and M. A. Christodoulou, *Adaptive Control With Recurrent High-Order Neural Networks: Theory and Industrial Applications*. Berlin, Germany: Springer, 2012.
[14] L. Buğoniu, T. de Bruin, D. Tolić, J. Kober, and I. Palunko, "Reinforcement learning for control: Performance, stability, and deep approximators," *Annu. Rev. Control*, vol. 46, pp. 8–28, Jan. 2018.
[15] A. Perrusquía and W. Yu, "Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview," *Neurocomputing*, vol. 438, pp. 145–154, May 2021.
[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.
[17] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
[18] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.
[19] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.
[20] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.
[21] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*.
[22] Y. Yu, A. Srivastava, and S. Canales, "Conditional LSTM-GAN for melody generation from lyrics," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 17, no. 1, pp. 1–20, Feb. 2021.
[23] X. Ding, Y. Wang, Z. Xu, W. J. Welch, and Z. J. Wang, "CcGAN: Continuous conditional generative adversarial networks for image generation," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–30.
[24] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Contro*. Hoboken, NJ, USA: Wiley, 1989.
[25] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," 2017, *arXiv:1701.00160*.
[26] L. J. Ratliff, S. A. Burden, and S. S. Sastry, "Characterization and computation of local Nash equilibria in continuous games," in *Proc. 51st Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Allerton, IL, USA, Oct. 2013, pp. 917–924.
[27] M. Norgaard, O. Ravn, N. Poulsen, and L. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems a Practitioner's Handbook* (Advanced Textbooks in Control and Signal Processing), 2000.
[28] J. Gonzalez and W. Yu, "Non-linear system modeling using LSTM neural networks," *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 485–489, 2018.
[29] I. Hernández and W. Yu, "Control of active lower limb prosthesis using human-in-the-loop scheme," *Cogent Eng.*, vol. 9, no. 1, Dec. 2022, Art. no. 2067026.



IVAN HERNANDEZ received the B.S. degree in electrical engineering from Metropolitan Autonomous University (UAM), Mexico, in 2014, and the M.S. degree in electrical engineering and the Ph.D. degree in automatic control from the Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV-IPN), Mexico, in 2019 and 2023, respectively. Since 2005, he has taught at various institutions. His main research interests include biomechatronics, robotics, artificial intelligence, and nonlinear control. He received the medal for university merit from UAM, in 2014.

WEN YU (Senior Member, IEEE) received the B.S. degree in automation from Tsinghua University, Beijing, China, in 1990, and the M.S. and Ph.D. degrees in automatic control from Northeastern University, Shenyang, China, in 1992 and 1995, respectively. From 1995 to 1996, he was a Lecturer with the Department of Automatic Control, Northeastern University. Since 1996, he has been with the Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV-IPN), Mexico, where he is currently a Professor with the Departamento de Control Automático. From 2002 to 2003, he held research positions with the Instituto Mexicano del Petróleo. He was a Senior Visiting Research Fellow with Queen's University Belfast, Belfast, U.K., from 2006 to 2007, and a Visiting Associate Professor with the University of California, Santa Cruz, from 2009 to 2010. Since 2006, he has been a Visiting Professor with Northeastern University (China). He has more than 500 publications, including more than 200 journal articles and eight monographic books. He is a member of Mexican Academy of Sciences. He was the General Chair of the IEEE Flagship Annual Meeting SSCI 2023. He serves as an Associate Editor of IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *Neurocomputing*, and *Journal of Intelligent and Fuzzy Systems*.

XIAOOU LI (Senior Member, IEEE) received the B.S. degree in applied mathematics and the Ph.D. degree in automatic control from Northeastern University, Shenyang, China, in 1991 and 1995, respectively. Since 2000, she has been a Professor with the Department of Computer Science, Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV-IPN), Mexico. In 2006, 2010, and 2018, she was a Senior Research Fellow with Queen's University Belfast; the University of California, Santa Cruz; and Shenzhen University, respectively. She has published more than 100 papers in international journals, books, and conferences in these areas. She is a regular member of Mexican Academy of Science and a member of the National Researcher System of Mexico. In addition, she is an active organizer of IEEE events and conferences, such as RAS, SMC, CIS, and WIE societies. She served as the General Chair for ICNSC'2016 and CASE'2022 and the Program Chair for SSCI'2023. She was an Editor of IEEE Press and *Engineering Applications of Artificial Intelligence* and an Associate Editor of IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, IEEE/CAA JOURNAL OF AUTOMATICA SINICA, and *IEEE SMC Magazine*.