Robust Fuzzy $Q$-Learning-Based Strictly Negative Imaginary Tracking Controllers for the Uncertain Quadrotor Systems

Vu Phi Tran, Mohamed A. Mabrok, Sreenatha G. Anavatti, Matthew A. Garratt, Senior Member, IEEE, and Ian R. Petersen, Fellow, IEEE

Abstract—Quadrotors are one of the popular unmanned aerial vehicles (UAVs) due to their versatility and simple design. However, the tuning of gains for quadrotor flight controllers can be laborious, and accurately stable control of trajectories can be difficult to maintain under exogenous disturbances and uncertain system parameters. This article introduces a novel robust adaptive control synthesis methodology for a quadrotor robot’s attitude and altitude stabilization. The proposed method is based on the fuzzy reinforcement learning and strictly negative imaginary (SNI) property. The first stage of our control approach is to transform a nonlinear quadrotor system into an equivalent negative-imaginary (NI) linear model by means of the feedback linearization (FL) technique. The second phase is to design a control scheme that adapts online the SNI controller gains via fuzzy $Q$-learning. The performance of the designed controller is compared with that of a fixed-gain SNI controller, a fuzzy-SNI controller, and a conventional PID controller in a series of numerical simulations. Furthermore, the proofs for the stability of the proposed controller and the adaptive laws are provided using the NI theorem.

Index Terms—Adaptive fuzzy, $Q$-learning, quadcopter unmanned aerial vehicle (UAV), reinforcement learning, robust and adaptive control, strictly negative imaginary (SNI) controller, uncertainties.

I. INTRODUCTION

ROBUSTNESS and adaptability have been the driving themes behind the development of modern control techniques. Negative-imaginary (NI) systems’ theory provides one such technique that can provide stability and robustness against uncertainties. This has been beneficially in different applications, including large flexible structures [1], multilink robotic arms [2], and aerial robotics [3]–[5].

A stable linear single-input–single-output (SISO) system with a transfer function $\xi(s)$ is said to have the strictly NI (SNI) property when the following requirement is fulfilled: $\forall \omega > 0, \quad f(\xi(j\omega) - \xi(-j\omega)) > 0$. This indicates that the Nyquist contour of $\xi(s)$ always lies below the real axis for all positive frequencies [6]. A stability analysis using the NI theory only uses the loop gains at zero and infinite frequencies. According to [6] and [7], the feedback interconnection between an NI plant, with transfer function $\xi_1(s)$ and an SNI controller with transfer function $\xi_2(s)$, obtains robust stability if the dc gain at the zero frequency is less than one: $\xi_1(0)\xi_2(0) < 1$.

In general, most physical systems are inherently nonlinear and time varying. Hence, it is very difficult to obtain an accurate mathematical model of the system. Several of the existing techniques, such as the linear quadratic regulator (LQR/LQG) [8], model predictive control (MPC) [9], and linear-distributed MPC [10] require an accurate model. To use the NI systems theory, one can linearize the given nonlinear systems around the desired operating point and then apply the traditional linear NI control methods. Due to the limitations on the accuracy of the mathematical models (nonlinear and the linearized version), there is no guarantee of robustness provided by the linear SNI theory. This article provides a novel solution to handle this problem.

Feedback linearization [11], [12] is a convenient tool to obtain a linear system representation from a nonlinear model. The nonlinearity is effectively eliminated using a feedback controller. The equivalent dynamic system has a particular structure, which is used in this article to design the SNI controller. Although a generalized linear model can be obtained by the use of linear approximations, the system’s main parameters are required to be fixed and well known. Hence, the accuracy of the model can be undermined when unknown disturbances or unmodeled parameters are present.

Since a broad class of unstructured disturbances and uncertainties are often present in the system dynamics, adaptive controllers have gained importance over the years in handling uncertainties and parameter variations. To enhance the robustness of an SNI controller, the parameters of the SNI controller, consisting of the dc gain and time constant, can be adaptive. This adaptation can be carried out online using various adaptive control methods [e.g., fuzzy interference system (FIS) [13], [14] or artificial neural networks (ANNs)] [15], [16] to adapt to...
environmental changes, structural damage, or uncertain parameters in the linear SNI/NI systems, where necessary and sufficient conditions for internal stability are still guaranteed (see [4], [17], [18]). Using adaptive control strategies, one can improve the tracking performance and eliminate the necessity for an accurate dynamic model.

Generally, the adaptive control schemes are categorized into two primary techniques: either 1) passive (PT) or 2) active (AT). Unlike the AT control strategies dramatically increasing the computational time due to the design of the additional fault detection and diagnosis observers, PT approaches can reject the faults’ influences faster, accommodating the system recovered from the fault states quicker. Furthermore, many recent AT approaches have just been designed for the SISO case; whereas, many existing dynamic systems in nature are MIMO [15], [19].

Intelligent PT controllers using fuzzy logic have been designed for various nonlinear control applications [20], [21]; system identification techniques [22], [23]; and active disturbance rejection systems [24]. The design of FISs is based on human intuition and, hence, FIS are called “knowledge-based systems (KBSs).” They are not only convenient to design but are also easy to implement and interpret, particularly from the perspective of lay persons (e.g., average drone operators). These systems can also offer computational simplicity and robustness compared to some conventional mathematically based autotuning procedures (e.g., MIT Rules [25]), whose performance entirely depends on the accuracy of mathematical models of the systems. In addition, fuzzy systems can deal with real-time control applications under multiple disturbances (e.g., sensor noise, wind gusts and uncertain dynamics). Furthermore, an adaptive fuzzy controller can be easily designed with adjustable parameters along with an embedded mechanism for adjusting them (e.g., a general fuzzy neural network [26], or a sliding-mode control for Takagi–Sugeno fuzzy systems [27]) in order to improve the performance of the controller.

However, fuzzy-logic controller design requires expert knowledge, which is often not available. Moreover, controller designers can spend many hours manually tuning complicated control loops. Hence, intelligent embedding approaches have been proposed to address these issues. Some of the offline techniques to automatically tune the parameters of a fuzzy-logic controller include genetic algorithms [28]; ant colony optimization [29]; particle swarm optimization [30], [31]; etc. Due to their computational complexity, these are mainly used in an offline mode.

$Q$-learning is a model-free reinforcement learning algorithm that allows an agent to select an action at a given state by exploring various actions in a state and updating the desired reward with the actual reward for that action. The agent directly interacts with the environment to generate responses, and makes updates to state action pairs recorded in a $Q$-table. Hence, it does not require agents to have a model of the environment and an adaption law (e.g., adaptive fuzzy) as required by dynamic programming (DP) [32]. Moreover, it also requires a small amount of computation, compared to other machine-learning approaches [33], [34]. $Q$-learning has gained importance over the last few years as an effective embedding tool for adaptive controllers. A hybrid Ziegler–Nichols (Z-N) fuzzy $Q$-learning multiagent system (MAS) approach for the online adjustment of the gains of a PID controller is proposed in [35]. Using the online adjustment capability of the PID gains, the overshoot is controlled and, simultaneously, problems regarding the dynamic nature of the process are addressed. The three gains of the PID controller are initially set by the Z-N method and then, through the control process, are adjusted online via the MAS. Fuzzy $Q$-learning has been used in the recently published works in order to supervise a PID controller for a motor speed control application [36]. In addition, fuzzy $Q$-learning in MAS has been successfully applied in energy management applications for direct control of the power exchange amongst the units of a microgrid [37]. Here, the fuzzy $Q$-learning approach is used to tune the parameters of the SNI control technique.

The contributions of this article for advancing current research boundaries include the following.

1) This article presents a novel method to transform the general nonlinear dynamical equations into a MIMO NI linear model.

2) The above transformation is used, through a hybrid approach between an adaptive NI MIMO feedforward control loop and an adaptive SNI MIMO controller, to propose a control strategy for the second-order NI linear dynamics with double poles at the origin and to improve the quadrotor’s altitude and attitude response and accuracy. The proposed control and conversion technique can deal with time-varying and coupling uncertainties, which are one of the main drawbacks of the recently published adaptive SNI SISO control methods [17], [18], [38].

3) An adaptive control scheme, called fuzzy $Q$-learning SNI (Fuzzy-QL-SNI), is employed to design the adaptive MIMO SNI controller. The Fuzzy-QL-SNI controllers are capable of adapting the SNI gains online according to the environmental changes without the need for pretraining or the use of an expert’s system experience.

4) Through extensive simulations of a quadrotor system that include the nominal system, external disturbances, and internal parameter variations, the relative merits of our adaptive control system are investigated with respect to a fixed-gain SNI, fuzzy-SNI, and traditional PID control methods.

The structure of this article is given as follows. Section II presents the main theory of NI systems, the feedback linearization technique, fuzzy-logic systems, and reinforcement learning. Section III describes a dynamic model of a quadrotor and presents the design of the proposed control strategy, while Section IV analyzes the stability of the closed-loop control system. Section V simulates the capabilities and performance of the proposed Fuzzy-QL-SNI control techniques. Finally, Section VI presents conclusions drawn from this work.

For convenience, Table I summarizes a list of major variables used in this article.
the properties of the imaginary part of the frequency response system. For instance, the NI property is defined by considering understanding the NI property by considering the case of the SISO to accommodate systems with free body motion. We can in [6] and [10]. A generalized definition was developed in [7].

II. Preliminaries

A. Negative-Imaginary Systems

In this section, we briefly present the NI systems theory and its formal definition for linear time-invariant (LTI) systems.

NI systems theory was introduced by Lanzon and Petersen in [6] and [10]. A generalized definition was developed in [7] to accommodate systems with free body motion. For instance, the NI property is defined by considering the properties of the imaginary part of the frequency response $P(j\omega)$ and requiring the condition $j(P(j\omega) - P(j\omega)^*) \geq 0$ to be satisfied for all $\omega \in (0, \infty)$. Roughly speaking, the NI systems are stable systems having a phase lag between 0 and $-\pi$ for all $\omega > 0$. That is, their Nyquist plot lies below the real axis when the frequency varies in the open interval $(0, \infty)$. This is similar to positive real systems, where the Nyquist plot is constrained to lie in the right half of the complex plane.

The NI property arises in many practical systems. For example, such systems arise when considering the transfer function from a force actuator to a corresponding colocated position sensor (for instance, a piezoelectric sensor) in a lightly damped structure [6], [39], [40]. Another area where the underlying system dynamics are NI, are nano-positioning systems (see [40]–[42]). Similar issues also arise in application to the area of robotics, where position measurements are also widely used. From the control point of view, these NI systems provide a means of obtaining stable robust controller [6]. Hence, there is enhanced research effort in this area to identify the NI systems or convert given systems to NI systems [5]. This article proposes one such technique to convert the quadrator dynamics to the NI system so that the robust stability benefit of the controller can be used.

Consider an LTI system that has the following state-space representation:

\[ \dot{x}(t) = Ax(t) + Bu(t) \]  \hspace{1cm} (1)
\[ y(t) = Cx(t) + Du(t). \]  \hspace{1cm} (2)

Here, the matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{m \times n}$, and $D \in \mathbb{R}^{m \times m}$. The transfer function matrix corresponding to this system is given by

\[ P(s) = C(sl - A)^{-1}B + D. \]  \hspace{1cm} (3)

For the above LTI system, an NI system is defined as follows.

**Definition 1** [7]: A square transfer function matrix $P(s)$ is NI if the following conditions are satisfied.
1) $P(s)$ has no pole in $\text{Re}[s] > 0$.
2) For all $\omega > 0$ such that $s = j\omega$ is not a pole of $P(s)$

\[ j(P(j\omega) - P(j\omega)^*) \geq 0. \]  \hspace{1cm} (4)

3) If $s = j\omega_0$ with $\omega_0 > 0$ is a pole of $P(s)$, then it is a simple pole and the residue matrix $K = \lim_{s \to j\omega_0}(s - j\omega_0)P(s)$ is Hermitian and positive semidefinite.
4) If $s = 0$ is a pole of $P(s)$, then $\lim_{s \to 0}s^kP(s) = 0$ for all $k \geq 3$ and $\lim_{s \to 0}s^2P(s)$ is Hermitian and positive semidefinite.

Similarly, the SNI system is also defined as follows.

**Definition 2** [10]: A square transfer function matrix $N(s)$ is SNI if the following conditions are satisfied.
1) $N(s)$ has no pole in $\text{Re}[s] \geq 0$.
2) For all $\omega > 0$, $j(N(j\omega) - N(j\omega)^*) > 0$.

The fundamental result of the interconnected NI systems theory can be summarized as follows.

**Lemma 1**: The positive feedback interconnection between an NI transfer function $P(s)$ (with no poles at the origin) and an SNI transfer function $N(s)$ as shown in Fig. 1 is internally stable if and only if the following dc gain condition is satisfied [6]:

\[ \lambda_{\max}(P(0)N(0)) < 1 \]  \hspace{1cm} (5)

where $\lambda_{\max}$ is the maximum eigenvalue.

B. Feedback Linearization for Nonlinear Problems

Feedback linearization (FL) is a widely used approach in linearizing nonlinear dynamical systems [43]. The basic idea of FL is to find a transformation that transforms given nonlinear dynamics into an equivalent linear dynamical system (see Fig. 2).

Consider the following nonlinear system:

\[ \dot{x} = f(x) + g(x)u \]  \hspace{1cm} (6)
\[ y = h(x) \]
where \( x \in \mathbb{R}^n \) represents the state vector, \( u \in \mathbb{R}^p \) represents the input vector, and \( y \in \mathbb{R}^m \) represents output vector. The main idea is to design the control law as follows:

\[
u = k(x) + h(x)v \tag{7}\]

such that the dynamics from the new input \( v \) to the output \( y \) are linear.

### C. Sugeno Fuzzy Inference System

A fuzzy controller or model uses fuzzy rules, which are linguistic if-then statements, consisting of fuzzy sets, fuzzy logic, and fuzzy inference, to express complex processes without the use of complicated models [17], [18]. These rules embed expert’s control/modeling knowledge and represent experience in connecting fuzzy variables using linguistic terms [44]. Fig. 3 describes the overall architecture of fuzzy-logic control. The common structure of a fuzzy rule is

\[
\text{IF } (\xi_1 = \epsilon_1) \text{ and/or } (\xi_2 = \epsilon_2) \cdots \text{ and/or } (\xi_n = \epsilon_n) \Rightarrow \text{THEN } \phi = \mu \tag{8}\]

where \( \epsilon_i \forall i = 1, \ldots, n \) is a fuzzy set of the \( i \)th input. \( n \) is the total number of fuzzy rules. While \( \xi = (\xi_1, \xi_2, \ldots, \xi_n) \) represents the script input vector, \( \phi \) denotes the output variable obtained from the defuzzification process, and \( \mu \) indicates a fuzzy consequent offered by the expert.

The fundamental operators “and/or/not” combine the conditions of the variable inputs that are satisfied. Based on the degree of match between fuzzy input and the rules, which rules need to be implemented and the firing strength \( w_{ri} \) of the rule \( i \) is defined according to the given input area.

Sugeno (TS) fuzzy inference, which is the most commonly used, works with singleton output membership functions that are either constant or a linear function of the input values. By using a weighted average over all rule outputs to compute the script output, the defuzzification process for the Sugeno system is more computationally efficient than that of a Mamdani system. The final output can be obtained using the Wang–Mendel technique [45]:

\[
\phi = \frac{\sum_{i=1}^{n} w_{ri} \phi_i}{\sum_{i=1}^{n} w_{ri}} \tag{9}\]

where \( \phi_i \) is the consequent of rule \( i \).

### D. Q-Learning

Q-learning is a subclass of reinforcement learning, which is one of the three basic machine-learning platforms; along with supervised learning and unsupervised learning. Reinforcement learning represents the learning process in the nature around us. Reinforcement learning relies on a goal-directed problem solving mechanism. Q-learning is a model-free control method, where the control policy generated depends on the interaction or the feedback between the environment and the agent without knowing any information about the system model. Therefore, a Q-reinforcement learning policy is defined as a learning control law for the system to decide on the actions to be performed.

The main components in reinforcement learning in general are the agent states \( S \), the agent allowable actions \( A \), and the rewards \( R \). Here, the agent or the controller observes states \( S_t \in S \) and, therefore, performs an action (control signal) \( A_t \in A \). Hence, the environment (the system under control) updates the new states \( S_{t+1} \in S \) and produces a reward \( R_{t+1} \in R \). The reward is then received by the agent and chooses a new action \( A_{t+1} \in A \). This feedback interaction proceeds until a terminal state \( S_T \in S \) is reached.

The main aim of the agent (the controller) in reinforcement learning is to find or learn the optimal control policy. This control policy is to maximize a discounted accumulated reward function \( J_t \), which may take the following form:

\[
J_t(\tau) = R_{t+1} + \sigma R_{t+2} + \sigma^2 R_{t+3} + \cdots = \sum_{\tau=0}^{\infty} \sigma^\tau R(s_t, a_t). \tag{10}\]

Here, \( \sigma \) is a discounting factor that defines the significance of the future rewards, \( 0 < \sigma < 1 \).

The action-value function (Q-Function), \( Q^\pi(s, a) \), indicates the expected return with initial state \( s \) and initial action \( a \) and all forward actions controlled by policy \( \pi \):

\[
Q^\pi(s, a) = E_{\tau\leftarrow\pi} [J(\tau) | s_t = s, a_t = a]. \tag{11}\]

In the Q-learning algorithm, the update rule is governed by the following equation:

\[
Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \eta \left[ R_{t+1} + \sigma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right] \tag{12}\]

where the learning rate \( \eta \) represents the degree the new information overrides the old one.

Fuzzy Q-learning is a fuzzy extension of the Q-learning algorithm. In creating an FQL model, one first needs to specify the input states and their corresponding fuzzy sets and then build an FIS to integrate with the Q-learning algorithm. In Section III-B, a fuzzy Q-learning algorithm for our quadrotor system is developed.
III. QUADROTOR MODEL AND CONTROL STRATEGY

A. Negative-Imaginary Quadrotor Model Using the Feedback Linearization Technique

The general dynamic model of a quadrotor using the Euler–Lagrangian method has been derived in the articles [46], [47]. To derive the quadrotor’s motions in 3-D space, an inertial frame $E$ whose origin is at the defined home location and a body frame $B$ whose origin is at the quadrotor’s center of gravity are formed as in Fig. 4.

Let $\delta = (x, y, z, \alpha, \theta, \psi) \in \mathbb{R}^6$ denote the generalized coordinates and let $\rho = (x, y, z) \in \mathbb{R}^3$ denote the position vector of the air vehicle with respect to an inertial frame. Moreover, the orientation of the quadrotor is represented by the three Euler angles, namely: 1) the roll angle $\alpha$ around the $x$ axis; 2) the pitch angle $\theta$ around the $y$ axis; and 3) the yaw angle $\psi$ around the $z$ axis, that generate a rotational vector $\Omega = (\alpha, \theta, \psi) \in \mathbb{R}^3$. Based on the Euler angles, coordinates are transferred between inertial and body frames as required.

The overall dynamic model describing the nonlinear quadrotor behavior is summarized as follows [48]:

\begin{align}
\dot{x} &= \frac{(c_s s_\theta c_\alpha + s_\psi s_\alpha) U_1 - C_{dx} \dot{x}^2}{m} \\
\dot{y} &= \frac{(s_\phi s_\theta c_\alpha - c_\psi s_\alpha) U_1 - C_{dy} \dot{y}^2}{m} \\
\dot{z} &= \frac{(c_\phi c_\theta) U_1 - C_{dz} \dot{z}^2 - g}{m} \\
\dot{\alpha} &= \frac{U_2 - C_{ax} \dot{\alpha} - J_\alpha \dot{\theta} - (I_z - I_x) \dot{\theta} \dot{\psi}}{I_x} \\
\dot{\theta} &= \frac{U_3 - C_{ay} \dot{\theta} + J_\theta \dot{\alpha} - (I_z - I_x) \dot{\theta} \dot{\psi}}{I_y} \\
\dot{\psi} &= \frac{U_4 - C_{az} \dot{\psi} - (I_y - I_z) \dot{\alpha} \dot{\theta}}{I_z}
\end{align}

(13)

where

\begin{align}
U_1 &= LK_f (\omega_{p1}^2 + \omega_{p2}^2 + \omega_{p3}^2 + \omega_{p4}^2) \\
U_2 &= LK_f (\omega_{p2}^2 - \omega_{p4}^2) \\
U_3 &= LK_f (\omega_{p3}^2 - \omega_{p1}^2)
\end{align}

(14)

in which $I_i$ stands for the moments of inertia about the respective axes and $m$ represents the mass of the quadrotor. $(C_{dx}, C_{dy}, C_{dz})$ and $(C_{ax}, C_{ay}, C_{az})$ are the translational and rotational drag coefficients. Furthermore, $(c_\alpha, c_\theta, c_\psi)$ and $(s_\alpha, s_\theta, s_\psi)$ represent $(\cos(\alpha), \cos(\theta), \cos(\psi))$ and $(\sin(\alpha), \sin(\theta), \sin(\psi))$, respectively. While $K_m$ is the rotor torque coefficient, $K_f$ is the rotor thrust coefficient. In addition, $J_i$ stands for the inertia of the propeller and $\Omega_i = \sum_i^{i=1} (-1)^{i+1} \omega_i$ is the total angular speed of propellers, wherein $\omega_i$ is the speed of the $i$th rotor. Moreover, the control signal vector $U = [U_1, U_2, U_3, U_4]$ includes the total thrust and the pitch, roll, and yaw moments. Finally, $L$ represents the distance from the center of the quadrotor to the actuator axes.

The proposed state-space model of the quadrotor in (13) cannot be converted into an equivalent linear and controllable system by static state feedback due to under-actuation. It is, however, possible to resort to dynamic state feedback to achieve full-state linearization. After that, starting from the new feedback linearized design, a linear controller can be designed. In this section, the FL approach for the quadrotor unmanned aerial vehicle (UAV) attitude and altitude will be presented.

According to (7), the control input vector $U$ can be chosen to eliminate all nonlinear terms in (13), leading to a linear system. Using dynamic inversion, $U$ can be obtained by

$$U = g(x)^{-1}(-f(x) + v).$$

(15)

Substituting (15) into (6) results in the following linear system:

$$\dot{x} = v.$$

(16)

In our approach, for convenience, drag coefficients are assumed to be 0 since drags are negligible at low speed [49]. The attitude and altitude quadrotor dynamic equations in (13), therefore, can be rewritten as:

\begin{align}
\ddot{\alpha} &= \frac{U_2 - J_\alpha \dot{\theta} - (I_z - I_x) \dot{\theta} \dot{\psi}}{I_x} \\
\ddot{\theta} &= \frac{U_3 + J_\theta \dot{\alpha} - (I_z - I_x) \dot{\theta} \dot{\psi}}{I_y} \\
\ddot{\psi} &= \frac{U_4 - (I_y - I_z) \dot{\alpha} \dot{\theta}}{I_z} \\
\dot{v} &= \frac{(c_\alpha c_\theta) U_1}{m} - g.
\end{align}

(17)

Based on (17), $U_1, U_2, U_3,$ and $U_4$ can be selected as follows:

\begin{align}
U_1 &= \frac{m}{c_\phi c_\theta} (g + v_1) \\
U_2 &= \frac{1}{h_1} (k_1 \dot{\alpha} \dot{\psi} + k_2 \Omega_i \dot{\theta} + v_2) \\
U_3 &= \frac{1}{h_2} (k_3 \dot{\alpha} \dot{\psi} - k_4 \Omega_i \dot{\alpha} + v_3) \\
U_4 &= \frac{1}{h_3} (k_5 \dot{\alpha} \dot{\theta} + v_4)
\end{align}

(18)
where \( k_1 = [(I_x - I_y)/I_x] \), \( k_2 = (J_y/I_y) \), \( k_3 = [(I_x - I_z)/I_y] \), 
\( k_4 = (J_z/I_z) \), \( k_5 = [(I_y - I_x)/I_z] \), 
\( h_1 = (1/I_x) \), \( h_2 = (1/I_y) \), 
and \( h_3 = (1/I_z) \).

By substituting (18) into (17), the following linear system is obtained:
\[
\begin{bmatrix}
\ddot{\psi} \\
\ddot{\phi} \\
\ddot{\theta} \\
\ddot{\xi}
\end{bmatrix} = P \begin{bmatrix}
v_1 \\
v_2 \\
v_3 \\
v_4
\end{bmatrix}.
\]

The Laplace transforms of the above equation gives the following transfer function matrix:
\[
A = \begin{bmatrix}
\frac{1}{\tau_1} & 0 & 0 & 0 \\
0 & \frac{1}{\tau_2} & 0 & 0 \\
\frac{1}{\tau_3} & 0 & \frac{1}{\tau_4} & 0 \\
0 & 0 & 0 & \frac{1}{\tau_5}
\end{bmatrix}
= \begin{bmatrix}
v_1 \\
v_2 \\
v_3 \\
v_4
\end{bmatrix}.
\]

The transfer function matrix \( P \) in (20) has two poles on the imaginary axis (double integrator). Based on the NI definition given in [50], it is clear that the linearized attitude and altitude dynamic model satisfies the NI property.

### B. Fuzzy Q-Learning Adaption Law

In the Q-learning method, the values of \( Q \) for the state-action (SA) pairs in the table are continuously computed and updated at each iteration using a \( Q \) function. However, if the number of SA pairs is large, the computational load will dramatically increase, which might fail in real-time implementation. For scaling up to large or continuous SA spaces, TS fuzzy function approximation can be used in combination with Q-learning to generate the Fuzzy Q-learning control theory [36], [51]. In this new method, the state of the agent \( S \) is the crisp set of the fuzzy inputs \( \zeta \). Hence, the firing strength of each rule determines the degree to which the agent matches a state.

In addition, the rules do not have fixed consequences, meaning that there are no predefined SA pairs. They can be increased through the exploration/exploitation algorithm. Thus, the learning agent has to find the best conclusion for each rule via competing actions (e.g., the action with the best \( Q \)-value)

\[
\text{IF } (\zeta \text{ is } S_i) \text{ THEN } \begin{cases} 
[1, 1] \text{ with } q[i, 1] \\
[1, 2] \text{ with } q[i, 2] \\
\vdots \\
[i, J] \text{ with } q[i, J]
\end{cases}
\]

where \( a[i, j] \) is the \( j \)th possible action in rule \( i \) and \( Q[i, j] \) is a corresponding \( Q \)-value. Furthermore, \( S_i \) is defined by \( \xi_1 \) is \( S_{i, 1} \) and \( \xi_2 \) is \( S_{i, 2} \cdots \) and \( \xi_n \) is \( S_{i,n} \), in which \( S_{i,j}, j = 1, \ldots, n \) are TS fuzzy rules.

Therefore, the global action \( a(\zeta) \) for input vector \( \zeta \) and the selected rule consequences \( a_i \) can be expressed as follows:
\[
a(\zeta) = \sum_{i=1}^{n} w_{ri}(\zeta) a_i = \sum_{i=1}^{n} w_{ri}(\zeta),
\]

On the other hand, the corresponding \( Q \)-value is calculated by the following equation:
\[
Q(\zeta, a) = \frac{\sum_{i=1}^{n} w_{ri}(\zeta) q[i, i^*]}{\sum_{i=1}^{n} w_{ri}(\zeta)}
\]

where \( q[i, i^*] \) is the actual \( Q \)-value of the chosen action \( i^* \) in the fired rule \( i \), using an exploration/exploitation strategy and an action \( ^* \) that has the maximum \( Q \)-value for the fired rule \( i \), such that \( q[i, i^*] = \max q[i, j], j = 1, \ldots, J \).

Based on gradient descent, the rate of change in value \( q \) are calculated as follows:
\[
\Delta q[i, i^*] = \eta \Delta Q \frac{w_{ri}(\zeta)}{\sum_{i=1}^{n} w_{ri}(\zeta)}
\]

where
\[
\eta \Delta Q = R_{i+1} + \sigma \max_a Q(S_{i+1}, a) - Q(S_i, A_i)
\]

Thus, the \( q \) values are updated as
\[
q[i, i^*]_{i+1} = q[i, i^*]_i + \eta \left( R_{i+1} + \sigma \max_a Q(S_{i+1}, a) - Q(S_i, A_i) \right) \frac{w_{ri}(\zeta)}{\sum_{j=1}^{n} w_{rj}(\zeta)}
\]

where \( \eta \) indicates the learning rate that determines how much the model updates the action value depending on the reward prediction error.

The pseudocode of this algorithm is presented as in Table II.

### C. Control Strategy

The control strategy for the quadrotor plant is shown in Fig. 5. The tracking errors arise from the subtraction of the desired attitude and altitude set points and their actual values. The error signals are considered as the input vector of both SNI tracking controllers and the FQL agents. The SNI tracking control system whose gains are tuned offline by minimizing the tracking error function and then adjusted online by the fuzzy Q-learning algorithm has the following general representation as follows:
\[
N(s) = \frac{\gamma}{\tau s + 1} - \beta
\]

where \( \gamma \) is the dc gain, \( \tau \) is the time constant, and \( \beta \) is a feedforward gain.
the one used in [3], [4], and [18], since it contains a negative feedforward term in the SNI controller is due to stability conditions.

The proposed SNI gains are tuned online by the fuzzy Q-learning agent. The input signal \( \zeta \) of the fuzzy Q-learning agent is bounded within a range of \([-2, 2]\). There are five Gaussian-shaped membership functions (e.g., NB, NS, Z, PS, PB) for the state variables, as depicted in Fig. 6.

The output of each agent is a vector with two output variables \( \phi = [\gamma, \tau] \), each representing a gain of the SNI controller. Each output variable involves a group of three singleton fuzzy actions. For the output variable \( \gamma \), its singleton fuzzy sets are \([-20, 0, 20]\) and for the output variable \( \tau \), the group of the fuzzy singletons is \([-0.002, 0, 0.002]\). Each linguistic value, produced at the output variable, denotes the percentage variation of each gain according to its initial value.

As mentioned in Section II-D, the agent communicates with the environment and performs updates to the SA pairs in the \( Q \)-table. Moreover, there are two primary operations: 1) exploring and 2) exploiting to acquire information through its experience. While the exploiting action uses the available data to make a decision via the max value of those actions, the exploring action enables the agent to explore new states that may not be selected during the exploitation process. To accelerate the learning process at the beginning of the experiment, where the fuzzy \( Q \)-learning agent mainly explores, the agent will search for new knowledge during a certain number of iterations; see Fig. 9(c). After that, the agent conducts exploitation for 99% of the time and the remaining 1% of the time is exploration for a given state.

The agent reward \( R \) is selected the same as in [36]

\[
R = \frac{1}{1 + |e_t|} - \frac{1}{1 + |e_{t-1}|}
\]

where \( e_t \) and \( e_{t-1} \) are the tracking errors in the current and previous states. The fraction \( [1/(1+|e_t|)] \) is limited in range of \([0, 1]\). If the tracking error is 0, the value of this fraction becomes 1. The reward \( R \) can take either positive or negative values. If the reward values are positive, the error in the current state is decreased (reward). If the error increases, then reward values are negative (punishment).

The future discounted accumulated reward, therefore, is rewritten as

\[
J_t = \sum_{i=0}^{t} \sigma^i \left( \frac{1}{1+|e_i|} - \frac{1}{1 + |e_{i-1}|} \right)
\]

IV. Stability Analysis

The linear transfer function matrix from the new input vector variable \( v \) to the output vector \( y \) is a double integrator as depicted in (20), which represents a lossless NI system as presented in [7] and [50]. In this case, as indicated in [7], the SNI controller \( N(s) \) must have a negative dc gain. The reason for this is the poles’ location of the linearized system, which implies that the dc gain of the linearized system \( P(s) \) is infinity. The transfer function matrix \( N(0) \) and \( P(0) \) at the zero frequency can be expressed as follows:

\[
\lim_{\epsilon \to 0} P(0) = \begin{bmatrix}
\frac{1}{\epsilon} & 0 & 0 & 0 \\
0 & \frac{1}{\epsilon} & 0 & 0 \\
0 & 0 & \frac{1}{\epsilon} & 0 \\
0 & 0 & 0 & \frac{1}{\epsilon}
\end{bmatrix}
\]

\[
N(0) = \begin{bmatrix}
\gamma_1 - \beta_1 & 0 & 0 & 0 \\
0 & \gamma_2 - \beta_2 & 0 & 0 \\
0 & 0 & \gamma_3 - \beta_3 & 0 \\
0 & 0 & 0 & \gamma_4 - \beta_4
\end{bmatrix}
\]

Therefore, for the stability dc-gain condition in Lemma 1 to be satisfied, the controller has to have a negative dc gain at the zero frequency, and hence

\[
\lambda_{\max}[P(0)N(0)] = \lim_{\epsilon \to 0} \frac{\gamma \epsilon - \beta}{\epsilon} = -\infty < 1 \quad \forall \gamma < \beta.
\]

Equation (32) shows that the stability condition is satisfied for any negative dc gain in the SNI controller. Therefore, the controller proposed in (27) will stabilize the outer loop from the new output to the output \( y \).
V. SIMULATION RESULTS

Since quadrotor systems are highly nonlinear and time-varying, designing a high-performance tracking controller is challenging in the presence of uncertainties and variations in the system dynamics. A hybrid approach, including an adaptive SNI controller based on the FQL theory and fixed-gain NI feedforward gains, is developed to handle extreme (fast and huge variations) pose attitudes and altitude without any prior training. Simulations are conducted to test the capabilities of the proposed adaptive SNI controller to achieve resilient and precise flight control under wind disturbance and pose attitudes recovery under multiple uncertainties.

Fig. 7 shows the entire system implemented in MATLAB/Simulink. The dynamic model of the system (13)–(19), mentioned in Section III-A, is used to test controller design results on a full nonlinear simulation of the quadrotor UAV. The main parameters of the quadrotor UAV used in our experiments can be seen in Table III. Moreover, there are four input signal pairs (desired and actual roll/pitch/yaw and height). The tracking error $e_i$ is treated as the input of the tracking controllers and the adaption laws. The MIMO output of each adaptive strategy using FQL adjusted gains $\gamma$ and $\tau$, respectively. Meanwhile, each SNI tracking controller’s output was connected with the corresponding adaptive NI feedforward control gain $\beta$, which is chosen as $\gamma + 1$, in order to enhance their convergence rate and to guarantee system stability as mentioned in (32). After control moment values are produced, they are given to the input of the linear-system generator, where the nonlinear system dynamics are transformed into a fully linearized system. The roll/pitch/yaw moment and total thrust, produced by the linear-system generator, are fed to the quadrotor plant.

The sample time is set to 0.01 s since the physical VICON Motion Tracker system in our lab published the pose data to the relevant drone at the same frequency. Moreover, Xrotor Pro X9 brushless motors mounted on the quadrotor also normally operates at 0.01 s. During the simulation, various values of the setpoints (roll, pitch, yaw, and altitude) are given to the quadrotor system. The roll angle ($\alpha$) is controlled to follow a sine-wave; whereas, the pitch ($\theta$) and yaw ($\psi$) angles are commanded to track a square wave and a step function, respectively. Moreover, the altitude is also maintained at 1.5 m after the 1-s flight time.

The simulation will compare the robustness and performance of a conventional PID controller, tuned using the Z-N, a hybrid fixed-gain SNI controller, tuned by trial-and-error, and a hybrid fuzzy-SNI with those of the proposed method under two scenarios: 1) without and 2) with added multiple uncertainties. It is worth noting that the comparative controllers’ detailed parameters can be found in our previously published works [17], [18]. This comparison aims to highlight the improvement of the SNI and fuzzy-SNI through the fuzzy $Q$-learning algorithm’s supervision and its adaptive ability (robustness) against internal and external changes. The learning rate and the discount factor are set to $\eta = 0.1$ and $\sigma = 0.7$, respectively. The proposed strategy will work in the exploration phase for 0.7 s at the beginning of the simulation before being switched into the exploitation phase for the remaining flight time. The selections of optimal exploration time, $\eta$, and $\sigma$ parameters will be described in the next Section V-A.

The evaluation metrics for the simulation tests are as follows; the Root Mean Square error (RMSE), the steady-state offset (SO), and the rise time ($tr$). The following equations calculate the RMSE and SO:

$$RMSE = \sqrt{\frac{1}{T} \sum_{j=1}^{T} e_{ij}^2} \quad \forall i \in A \quad (33)$$

where $T$ is the sample size.

$$SO = \max|e_{ij}| \quad \forall j = T - 500, \ldots, T > 0. \quad (34)$$

A. Nominal Dynamics

The performance of the proposed adaptive controller is evaluated under a nominal condition. According to the simulation results observed in Fig. 8, it is obvious that most of the proposed controllers are capable of tracking the desired reference values with satisfactory performance, reasonable rise time and low RMSE values, as summarized in Table IV. In contrast, although the PID controller tracks well the constant setpoints and the sine-wave trajectory, it cannot handle the desired extreme square and step trajectories by experiencing a high overshoot of approximately 25%. Thus, it can be seen that the proposed SNI control methods are more robust and have a better tracking capability than the traditional PID controller.

In addition, the above experiment has been repeated under the different samples of $\eta$, $\sigma$, and the initial exploration period. First, as shown in Fig. 9(b), increasing the constant discount factor results in the slight variations of the RMSE metrics. However, the best improvement is observed as $\sigma = 0.7$. Similarly, learning rate $\eta = 0.1$ produces the smallest RMSE values among the potential candidates; see Fig. 9(a). Finally,
the obtained results in Fig. 9(c) demonstrate that having an exploration phase at the start time noticeably reduces the tracking errors, especially the vertical direction, due to the faster learning time. Furthermore, the smallest RMSE results are obtained if the beginning exploration phase is implemented in the first 0.7s of the simulation. Consequently, a 0.7s exploration phase is set at the beginning of all simulations.

B. Disturbances and Uncertainties

For robustness analysis, the performance of the closed-loop control system will be examined against exogenous disturbances (e.g., wind gusts and air turbulence) and parameter changes. The Dryden wind gust model in MATLAB is applied to replicate a realistic wind gust scenario under realistic flight circumstances. The model employs continuous gusts along three axes based on a spatially varying stochastic process with

| Metrics | Fuzzy-QL-SNI | Fuzzy-SNI | SNI | PID | Units |
|---------|--------------|-----------|-----|-----|-------|
| SO      | 0.007        | 0.09      | 0.083 | 0.077 | [m]   |
| $t_e$   | 1.5          | 1.8       | 1.84 | 1.5  | [s]   |
| RMSE    | 0.007        | 0.09      | 0.083 | 0.077 | [m]   |

| Metrics | Fuzzy-QL-SNI | Fuzzy-SNI | SNI | PID | Units |
|---------|--------------|-----------|-----|-----|-------|
| SO      | 0.25         | 0.37      | 0.38 | 0.39 | [s]   |
| $t_e$   | 8.5          | 8.5       | 7.9  | 2.36 | [s]   |
| RMSE    | 0.32         | 0.33      | 0.38 | 0.72 | [rad] |

| Metrics | Fuzzy-QL-SNI | Fuzzy-SNI | SNI | PID | Units |
|---------|--------------|-----------|-----|-----|-------|
| SO      | 0.017        | 0.055     | 0.057 | 0.46 | [rad] |
| $t_e$   | 0.16         | 0.16      | 0.23 | 1.4  | [s]   |
| RMSE    | 0.012        | 0.058     | 0.071 | 0.32 | [rad] |
a maximum of 5 m/s [52]. In addition to the Dryden gust turbulence, the effects of air stream are also considered in our numerical experiment by the use of the 1-cos air turbulence model in [53]. The combined effects of extreme wind gusts and turbulence applied to the system throughout the simulation time is shown in Fig. 10.

Furthermore, the model uncertainties, consisting of the moment of inertia about each axis and mass, are set in consideration of 15% bias. However, the quadrotor simulation model parameters in the linear-system generator are kept as in Table III. Therefore, the residual nonlinearities and uncertainties acting on the quadrotor must be accommodated by the tracking controllers.

Evidently, the effects of the multiple disturbances alters the performance of all four controllers. Nonetheless, the fuzzy-QL-SNI controller outperforms the fuzzy-SNI, SNI, and PID by stabilizing the system with less drift from the desired pose references, as shown in the plots in Fig. 11. The PID method is unable to achieve the desired tracking performance when tracking the extreme pitch trajectory. Although the tracking performances are better for the yaw and z step trajectories, the flight paths experience the high-overshoot for yaw control and chattering phenomena in the steady-state phase, and chattering phenomena in the steady-state phase, demonstrating a slow recovery from disturbances. Similar results can be seen in Table V, the PID control approach produces large SO figures of 0.48 rad for roll, 0.25 rad for pitch, 0.51 rad for yaw, and 0.11 m for altitude. Moreover, the PID controller has the maximum RMSEs of 0.31 rad for roll, 0.72 rad for pitch, 0.34 rad for yaw, and 0.09 m for altitude while its $t_s$ values for the roll and pitch are high, namely 2.5 and 1.4 s. However, the SNI controllers are able to handle the wind turbulence and the system’s uncertain parameters, much faster and with less error than the PID. This is reflected in both columns of Table V and Fig. 11. Furthermore, the Fuzzy-QL-SNI controller’s performance yields much more desirable values than the other ones. Particularly, the proposed adaptive strategy obtains the lowest RMSE values of

![Fig. 10. Wind gust and turbulence profile added in both directions for the second scenario.](image)

![Fig. 11. Closed-loop control responses of the four compared controllers for the attitude and altitude tracking problem of the quadrotor in the presence of exogenous disturbances and time-varying parameter uncertainties. (a) Roll dynamics. (b) Pitch dynamics. (c) Yaw dynamics. (d) z-axis position.](image)
[0.026 rad, 0.32 rad, 0.25 rad, 0.067 m] for roll, pitch, yaw, and height, which are approximately 78%, 61%, 61%, and 60% lower than those with the PID, SNI and Fuzzy-SNI. The response of the Fuzzy-QL-SNI is noticeably faster with the $t_e$ numbers of [0.35, 0.17, 5.8, 1.5] s, almost twice as fast as that of the other control schemes. Furthermore, there are only slight differences in the evaluation metrics of the two given cases considered for the Fuzzy-QL-SNI. These observations validate the fast adaption capability and the robustness of our closed-loop adaptive control system.

Another strength of the Fuzzy-QL-SNI controller is that it delivers low-frequency and unsaturated output control signals. This is demonstrated in Fig. 12, where the Fuzzy-QL-SNI controller produces smooth moment and thrust control signals.

VI. CONCLUSION

In our study, a novel concept of a hybrid adaptive SNI MIMO controller and a feedback linearization technique has been presented for the accurate attitude and altitude tracking problem of an uncertain nonlinear quadrotor drone under various flight schemes. The mathematical model of a quadrotor’s dynamics was transformed into a corresponding NI linear model by means of nonlinear feedback, performing a desirable cancellation of nonlinearities. Nevertheless, model uncertainties create uncertain nonlinear terms. Therefore, the proposed hybrid control system, consisting of a feedforward NI gain and the adaptive SNI controller using the FQL algorithm, was developed to enhance the robust tracking performance and accommodate uncertainties and variation in plant dynamics. Our simulation results showed that the fuzzy-QL-SNI exhibits better trajectory tracking compared to the adaptive fuzzy-SNI, SNI, and PID control schemes based on the achievement of the lowest RMSE and SO values. In addition to being robust and transparent, the proposed control system also demonstrated better robustness against uncertainties, namely, errors in modeling and exogenous disturbances (wind gusts and air turbulence) and a superior noise rejection capability with respect to the performance of the other methods.

For future work, such controllers will be executed in real-time applications or extended to control networked multirobot system consensus.

REFERENCES

[1] M. A. Mabrok, A. G. Kallapur, I. R. Petersen, and A. Lanzon, “Spectral conditions for negative imaginary systems with applications to nanopositioning,” IEEE/ASME Trans. Mechatronics, vol. 19, no. 3, pp. 895–903, Jun. 2014.
[2] O. Skeik and A. Lanzon, “Robust output consensus of homogeneous multi-agent systems with negative imaginary dynamics,” Automatica, vol. 113, Mar. 2020, Art. no. 108799.
[3] V. P. Tran, M. Garratt, and I. R. Petersen, “Switching time-invariant formation control of a collaborative multi-agent system using negative imaginary systems theory,” Control Eng. Pract., vol. 95, Feb. 2020, Art. no. 104245.
[4] V. P. Tran, F. Santos, M. A. Garratt, and S. G. Anavatti, “Distributed artificial neural networks-based adaptive strictly negative imaginary formation controllers for unmanned aerial vehicles in time-varying environments,” IEEE Trans. Ind. Informat., vol. 17, no. 6, pp. 3910–3919, Jun. 2021.
[5] V. P. Tran, F. Santos, and M. A. Garratt, “Adaptive trajectory tracking for quadrotor systems in unknown wind environments using particle swarm optimization-based strictly negative imaginary controllers,” IEEE Trans. Aerosp. Electron. Syst., vol. 57, no. 3, pp. 1742–1752, Jun. 2021.
[6] I. R. Petersen and A. Lanzon, “Feedback control of negative-imaginary systems,” IEEE Control Syst. Mag., vol. 30, no. 5, pp. 54–72, Oct. 2010.
[7] M. A. Mabrok, A. G. Kallapur, I. R. Petersen, and A. Lanzon, “Generalizing negative imaginary systems theory to include free body dynamics: Control of highly resonant structures with free body motion,” IEEE Trans. Autom. Control, vol. 59, no. 10, pp. 2692–2707, Oct. 2014.
[8] H. Jafari, M. Zareh, J. Roshanian, and A. Nikkhah, “An optimal guidance law applied to quadrotor using LQR method,” Trans. Inst. Soc. Aeronaut. Space Sci., vol. 53, no. 179, pp. 32–39, 2010.
[9] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, “Model predictive control for power converters and drives: Advances and trends,” IEEE Trans. Ind. Electron., vol. 64, no. 2, pp. 935–947, Feb. 2017.
[10] A. Lanzon and I. R. Petersen, “Stability robustness of a feedback interconnection of systems with negative imaginary frequency response,” IEEE Trans. Autom. Control, vol. 53, no. 4, pp. 1042–1046, May 2008.
[11] H. Voos, “Nonlinear control of a quadrotor micro-UAV using feedback-linearization,” in Proc. IEEE Int. Conf. Mechatronics, 2009, pp. 1–6.
[12] M. A. Lotufo, L. Colangelo, C. Perez-Montenegro, C. Novara, and E. Camuto, “Embedded model control for UAV quadrotor via feedback linearization,” IFAC-PapersOnLine, vol. 49, no. 17, pp. 266–271, 2016.
[13] Y. Li and S. Tong, “Adaptive fuzzy output constrained control design for multi-input multioutput stochastic nonstrict-feedback nonlinear systems,” IEEE Trans. Cybern., vol. 47, no. 12, pp. 4086–4095, Dec. 2017.
[14] W. Sun, S.-F. Su, Y. Wu, J. Xia, and V.-T. Nguyen, “Adaptive fuzzy control with high-order barrier Lyapunov functions for high-order uncertain nonlinear systems with full-state constraints,” IEEE Trans. Cybern., vol. 50, no. 8, pp. 3424–3432, Aug. 2020.
[15] Y. Li, K. Sun, and S. Tong, “Observer-based adaptive fuzzy fault-tolerant optimal control for SISO nonlinear systems,” IEEE Trans. Cybern., vol. 49, no. 2, pp. 649–661, Feb. 2019.
[16] W. Bai, Q. Zhou, T. Li, and H. Li, “Adaptive reinforcement learning neural network control for uncertain nonlinear system with input saturation,” IEEE Trans. Cybern., vol. 50, no. 8, pp. 3433–3443, Aug. 2020.
[17] V. P. Tran, F. Santos, M. A. Garratt, and I. R. Petersen, “Adaptive second-order strictly negative imaginary controllers based on the interval type-2 fuzzy self-tuning systems for a hovering quadrotor with uncertainties,” IEEE/ASME Trans. Mechatronics, vol. 25, no. 1, pp. 11–20, Feb. 2020.
[18] V. P. Tran, F. Santos, M. A. Garratt, and I. R. Petersen, “Fuzzy self-tuning of strictly negative-imaginary controllers for trajectory tracking of a quadcopter unmanned aerial vehicle,” IEEE Trans. Ind. Electron., vol. 68, no. 6, pp. 5036–5045, Jun. 2021.
[19] Y. Li, Y. Liu, and S. Tong, “Observer-based neuro-adaptive optimized control of strict-feedback nonlinear systems with state constraints,” IEEE Trans. Neural Netw. Learn. Syst., early access, Jun. 26, 2021, doi: 10.1109/TNNLS.2021.3051030.
[20] J. Qiu, K. Sun, T. Wang, and H. Gao, “Observer-based fuzzy adaptive event-triggered control for pure-feedback nonlinear systems with prescribed performance,” IEEE Trans. Fuzzy Syst., vol. 27, no. 11, pp. 2152–2162, Nov. 2019.
[21] X. Zhang et al., “Compound adaptive fuzzy quantized control for quadrotor and its experimental verification,” IEEE Trans. Cybern., vol. 51, no. 3, pp. 1121–1133, Mar. 2021.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
C.-W. Chang and C.-W. Tao, “A novel approach to implement Takagi-Sugeno fuzzy models,” IEEE Trans. Cybern., vol. 47, no. 9, pp. 2353–2361, Sep. 2017.

C.-M. Lin, T.-L. Le, and T.-T. Huynh, “Self-evolving function-link interval type-2 fuzzy neural network for nonlinear system identification and control,” Neurocomputing, vol. 275, pp. 2239–2250, Jan. 2018.

J. I. Meza and T. Wong, “Adaptive fuzzy-neural-network based on RBFNN control for active power filter,” Int. J. Mach. Learn. Cybern., vol. 10, no. 5, pp. 1139–1150, 2019.

N. Tariba, A. Bouknadel, A. Haddou, N. Ikken, H. E. Omari, and H. E. Omari, “Comparative study of adaptive controller using MIT rules and Lyapunov method for MPPT standalone PV systems,” in Proc. AIP Conf., vol. 1801, no. 1, 2017, p. 040008.

P. M. Kebria, A. Khosravi, S. Nahavandi, D. Wu, and F. Bello, “Adaptive type-2 fuzzy neural-network control for teleoperation systems with delay and uncertainties,” IEEE Trans. Fuzzy Syst., vol. 28, no. 10, pp. 2543–2554, Oct. 2020.

Y. Wang, H. R. Karimi, H. Shen, Z. Fang, and M. Liu, “Notice of violation of IEEE publication principles: Fuzzy-model-based sliding mode control of nonlinear descriptor systems,” IEEE Trans. Cybern., vol. 49, no. 9, pp. 3409–3419, Sep. 2019.

S. Loussafi and A. Abdelkerim, “Convolutional neural network hyper-parameters optimization based on genetic algorithms,” Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 10, pp. 252–266, 2018.

F. D. Yao, F. Vázquez, O. Castillo, C. I. González, G. Martinez, and P. Melin, “Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems,” Appl. Soft Comput., vol. 53, pp. 74–87, Apr. 2017.

F. Aboura, “Tuning PID controller using hybrid genetic algorithm particle swarm optimization method for AVR system,” in Proc. Int. Aegean Conf. Electr. Mach. Power Electron. (ACEMP) Int. Conf. Optim. Electr. Electron. Equip. (OPTIM), 2019, pp. 570–574.

D. D. Chen and Y. C. Lin, “A particle swarm optimization-based multi-level processing parameters optimization method for controlling microstructures of an aged superalloy during isothermal forging,” Metals Mater. Int., vol. 25, no. 5, pp. 1246–1257, 2019.

A. K. Mondal and N. Jamali, “A survey of reinforcement learning techniques: Strategies, recent development, and future directions,” 2020, arXiv:2001.06921.

D. Liu, X. Yang, D. Wang, and Q. Wei, “Reinforcement-learning-based robust controller design for continuous-time uncertain nonlinear systems subject to input constraints,” IEEE Trans. Cybern., vol. 45, no. 7, pp. 1372–1385, Jul. 2015.

H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double Q-learning,” in Proc. 30th AAAI Conf. Artif. Intell., 2016, pp. 2094–2100.

I. Carlucho, M. De Paula, S. A. Villar, and G. G. Acosta, “Incremental Q-learning strategy for adaptive PID control of mobile robots,” Expert Syst. Appl., vol. 80, pp. 183–199, Sep. 2017.

P. Kofinas and A. I. Doumos, “Fuzzy Q-learning agent for online tuning of PID controller for DC motor speed control,” Algorithms, vol. 11, no. 10, p. 148, 2018.

M. Esmaeili, H. Shayeghi, H. M. Nejad, and A. Younesi, “Reinforcement learning based PID controller design for LFC in a microgrid,” Int. J. Comput. Electr. Electron. Eng., vol. 36, no. 4, pp. 1287–1297, 2017.

V. P. Tran, F. Santoso, M. A. Garrant, and S. G. Anavatti, “Neural network-based self-learning of an adaptive strictly negative imaginary tracking controller for a quadrotor transporting a cable-suspended payload with minimum swing,” IEEE Trans. Ind. Electron., vol. 68, no. 10, pp. 10258–10268, Oct. 2021.

J. E. Jensen and T. K. Caughley, “Positive position feedback control for large space structures,” AIAA J., vol. 28, no. 4, pp. 717–724, Apr. 1990.

B. Bhikkaji and S. O. R. Moheimani, “Fast scanning using piezo-electric tube nanopositioners: A negative imaginary approach,” in Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics, AIM, Singapore, Jul. 2009, pp. 274–279.

A. Sebastian and S. M. Salapaka, “Design methodologies for robust nano-positioning,” IEEE Trans. Control Syst. Technol., vol. 13, no. 6, pp. 868–876, Nov. 2005.

I. M. Díaz and E. Pereira, and P. Reynolds, “Integrated resonant control scheme for cancelling human-induced vibrations in light-weight pedestrain structures,” Struct. Control Health Monitor, vol. 19, pp. 55–69, Feb. 2012.

F. Garcia, V. M. Becerra, C. Kambhampati, and K. Warwick, “Introduction to feedback linearisation,” in Strategies for Feedback Linearisation, London, U.K.: Springer, 2003, pp. 27–60.
Sreenatha G. Anavatti received the Ph.D. degree from the Indian Institute of Science, Bengaluru, India, in 1990.

He was with the Indian Institute of Technology Bombay, Mumbai, India, as an Associate Professor. He moved to Australia, in 1998. He is a Senior Lecturer with the School of Engineering and Information Technology, University of New South Wales, Canberra, ACT, Australia. He has supervised a number of Ph.D. scholars over these years. His current research interests include application of artificial intelligence techniques, such as fuzzy and neural systems for UAVs, underwater vehicles, and ground vehicles, identification and control of dynamic systems, navigation and path planning for autonomous vehicles, evolutionary fuzzy systems, and their applications to dynamic systems.

Matthew A. Garratt (Senior Member, IEEE) received the B.E. degree in aeronautical engineering from The University of Sydney, Camperdown, NSW, Australia, in 1990, the graduate Diploma degree in applied computer science from Central Queensland University, Norman Gardens, QLD, Australia, in 1997, and the Ph.D. degree in the field of biologically inspired robotics from The Australian National University, Canberra, ACT, Australia, in 2008.

He is a Professor with the School of Engineering and Information Technology, University of New South Wales, Canberra. His research interests include sensing, guidance, and control for autonomous systems with particular emphasis on biologically inspired and computational intelligence approaches.

Prof. Garratt is currently the Chair of the CIS Task Force on the Ethics and Social Implications of CI.

Ian R. Petersen (Fellow, IEEE) was born in Korumburra, VIC, Australia. He received the Ph.D. degree in electrical engineering from the University of Rochester, Rochester, NY, USA, in 1984.

From 1983 to 1985, he was a Postdoctoral Fellow with Australian National University, Canberra, ACT, Australia. From 1985 to 2016, he was with the University of New South Wales, Canberra, where he was most recently a Scientia Professor and an Australian Research Council Laureate Fellow with the School of Engineering and Information Technology. He has previously been a ARC Executive Director for Mathematics Information and Communications, an Acting Deputy Vice-Chancellor Research for UNSW, and an Australian Federation Fellow. Since 2017, he has been a Professor with the Research School of Engineering, Australian National University. His current research interests include robust control theory, quantum control theory, and stochastic control theory.

Dr. Petersen has served as an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, Systems and Control Letters, Automatica, the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, and the SIAM Journal on Control and Optimization. He is currently an Editor of Automatica. He is a Fellow of IFAC and the Australian Academy of Science.