International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I5P130

Volume 72 Issue 5, 290-298, May 2024


*Original Article*

# Trajectory Tracking Control for Wheeled Mobile Robot System with Uncertain Nonlinear Model based on Integral Reinforcement Learning Algorithm

Doan Van Hoa[1*], Tran Duc Chuyen[1], Lai Khac Lai[2], Le Thi Thu Ha[2]

[1]*Faculty of Electrical Engineering, University of Economics – Technology for Industries, Ha Noi, Viet Nam.*
[2]*Faculty of Electrical Engineering, Thai Nguyen University of Technology, Thai Nguyen City, Viet Nam.*

[*]*Corresponding Author : rvhoa@uneti.edu.vn*

Received: 15 December 2023 Revised: 31 March 2024 Accepted: 01 April 2024 Published: 26 May 2024

***Abstract*** *- A mobile robot is a type of robot that is capable of moving on its own and performing tasks without human intervention. Mobile robots are equipped with sensors and control systems to detect and react to the surrounding environment. Designing a controller for mobile robots so that the working process achieves optimal performance is of interest to many scientists. In this study, the author proposes an Integral Reinforcement Learning (IRL) method combined with a disturbance observer to design a robust adaptive optimal controller to track the trajectory of the WMR system. The optimal controller uses a traditional Actor-Critic structure consisting of two neural networks, Critic NN and Actor NN. External disturbances and wheel slippage of the WMR are estimated by the Disturbance Observer (DO) and compensated for by the disturbance compensation controller. System simulation results on Matlab software show us the effectiveness of the proposed combined method.*

***Keywords*** *- Reinforcement learning, Integral reinforcement learning, Actor-Critic, Wheeled mobile robot, Disturbance observer, Hamilton-Jacobi-Bellman.*

## 1. Introduction

A nonholonomic Wheeled Mobile Robot (WMR) is an inherently unstable system that lacks actuators and is nonlinear. When the WMR moves in an environment that depends heavily on external factors such as friction between the wheels and the road surface, the impact of the wind, the slope of the road surface and the load of the WMR may change. Therefore, the mathematical model of WMR contains many uncertain and difficult-to-control elements. Many classic control methods, such as PID [1], and modern control methods, such as backstepping [2-5], adaptive control [6-8], robust control [9-11], fuzzy control [12-18] and neural networks [19-22] have been applied to WMR. However, these methods are largely based on the mathematical model of WMR. In addition, these methods only consider the problem of orbital tracking for WMR and do not take into account optimal criteria related to tracking quality and control energy.

In modern control theory, two control methods are adaptive control and optimal control to solve two different big problems. Optimal control provides methods to find control laws that help stabilize the system while optimizing a certain objective function. However, to find the optimal control law, old methods require explicit information about the system's model, which hinders the ability to apply the algorithm in practice due to model uncertainty. Meanwhile, the adaptive control method allows the design of a controller with an uncertain model based on adaptive rules for the controller, possibly indirectly through an object recognition mechanism or control mechanism.

Directly adjust controller parameters. However, adaptive control does not consider the factor of optimizing the quality of the control law. Taking advantage of the advantages of optimal control and adaptive control, reinforcement learning techniques are considered a combination method of adaptive optimal control developed by adding optimization factors in the control design adaptive. For example, controller parameters are a variable in the optimization problem, or additional adaptive factors are included in the optimal control design, such as approximating the system parameters used in the control law optimal. Normally, by solving the Hamilton-Jacobi-Bellman (HJB) equation, the optimal control problem will be solved. For a linear system, the HJB equation becomes the Riccati algebraic equation (ARE). If the state matrix (A, B) of the linear system is available, the ARE solution can be found analytically.

On the contrary, if one of the matrices is missing, the analytical method cannot be applied. For nonlinear systems,

the HJB equation is a nonlinear differential equation. Therefore, it is generally impossible to solve these equations even for systems with deterministic models. To overcome the above limitation, many algorithms that approximate the solution of the ARE or HJB equation based on the basic theory of reinforcement learning have been proposed.

The reinforcement learning technique is a branch of machine learning aiming to obtain a policy, which can be understood as an operating process or control law optimal for an agent based on the observed responses from the interaction between agent and environment [23]. A reinforcement learning algorithm generally has two steps; first, each agent evaluates the performance of a current policy through interaction with the environment; this step is called Policy Evaluation.

Next, based on the evaluated results, the actor updates the policy to increase quality, which is equivalent to minimizing the cost function. This step is named Policy Improvement. Recently, researchers have focused on applying reinforcement learning techniques in feedback control of dynamic systems. One of the popular methods of reinforcement learning applied in control is the Policy Iteration algorithm (PI) [24].

Instead of using mathematical methods to solve the HJB equation directly, the PI algorithm starts by evaluating the cost function of an acceptable initialization control law. This is usually obtained by solving the nonlinear Lyapunov equation [25]. This new cost function is used to improve the control law, which is equivalent to minimizing the Hamilton function corresponding to that cost function. This two-step iterative process is carried out until the control law converges to the optimal control law.

With the development of reinforcement learning, many real-time methods have been applied to find the optimal control law online without needing a completely accurate understanding of the system dynamics, and this approach is often called Integral Reinforcement Learning (IRL) [26]. Based on its ability to approximate smooth nonlinear functions, neural networks are often used to implement iterative learning algorithms. The algorithms will be executed online on the Actor-Critic structure, which includes two function approximating neural networks.

The first neural network is called actor, used to approximate the control law, and the second neural network is called Critic, used to approximate the cost function. For continuous linear systems, research [27] introduced two offline PI iteration algorithms, which are mathematically equivalent to the Newton method. These methods eliminate the need to model the internal dynamics of the system by evaluating the cost function corresponding to the control law on a steady-state trajectory or by using measured state variables to construct the Lyapunov equation. Developing Murray's research direction, in [23], Vrabie and his colleagues presented a control design using reinforcement learning to solve the global linear optimal control problem online.

Specifically, the method uses the PI iterative algorithm based on measured kinetic data to solve the Riccati equation iteratively. In the design, the internal dynamic matrix of the system is also eliminated during the design process. However, the external dynamic matrix is still needed, so it is also called the algorithm for partially uncertain systems.

The method for fully model-free systems was developed in [28], with the use of a probe disturbance signal in addition to the input signal during the learning process. For nonlinear systems, in [29] and [30], an online algorithm for partially indeterminate affine nonlinear systems is presented, providing a local solution to the nonlinear HJB equation.

The method for completely uncertain systems is presented in the work [26], which can be considered an extension of the method for linear systems [28]. Although it is only a semi-global stable optimization method (because it does not guarantee complete stability but only in the case of satisfying certain assumptions), it is still a breakthrough in finding a law that regulates optimal control that completely eliminates the need for a system model. Extending the results, the authors presented a global stability method for a class of polynomial systems in [31].

Thus, it can be seen that by applying reinforcement learning and adaptive dynamic programming, not only can the optimization problem be solved online using measurement data, but also without using the full kinematic model and system accuracy. This has great significance in practice when obtaining sufficiently accurate models of systems is very difficult, not to mention that the parameters in the system can change during operation.

Some other studies extend to systems affected by external disturbances, combining adaptive optimal control with robust nonlinear methods such as sliding control to take advantage of the advantages of each method [32]. In this paper, we apply the IRL algorithm combined with the DO set for the uncertain nonlinear WMR system. The control quality was verified through numerical simulation on Matlab software, showing that the WMR tracked the reference trajectory with small errors and the cost function was minimized.

## 2. Geometric Structure of WMR and Modeling

Considering a three-wheeled mobile robot structure, two independent driving wheels and one passive wheel are used as a fulcrum to create a gravity balance, subject to nonholonomic constraints, as shown in Figure 1. *OXY* coordinates are the fixed coordinate system, and *MX'Y'* is the local coordinate system mounted on the robot. The parameters of WMR are presented in Table 1 [33].

**Table 1. WMR parameters**

| Variable name | Meaning | Value |
|---|---|---|
| $m_G$ | Weight of the platform | 10 kg |
| $I_G$ | Inertial moment of the platform | 4 kgm$^2$ |
| $m_W$ | Weight of each wheel | 2 kg |
| $I_W$ | Inertial moment of each wheel rotation axis | 0.1 kgm$^2$ |
| $I_D$ | Inertial moment of each wheel diameter axis | 0.05 kgm$^2$ |
| $b$ | Radius of the wheel shaft | 0.3 m |
| $a$ | Distance between the M and G | 0.2 m |
| $r$ | Radius of the wheel | 0.15 m |

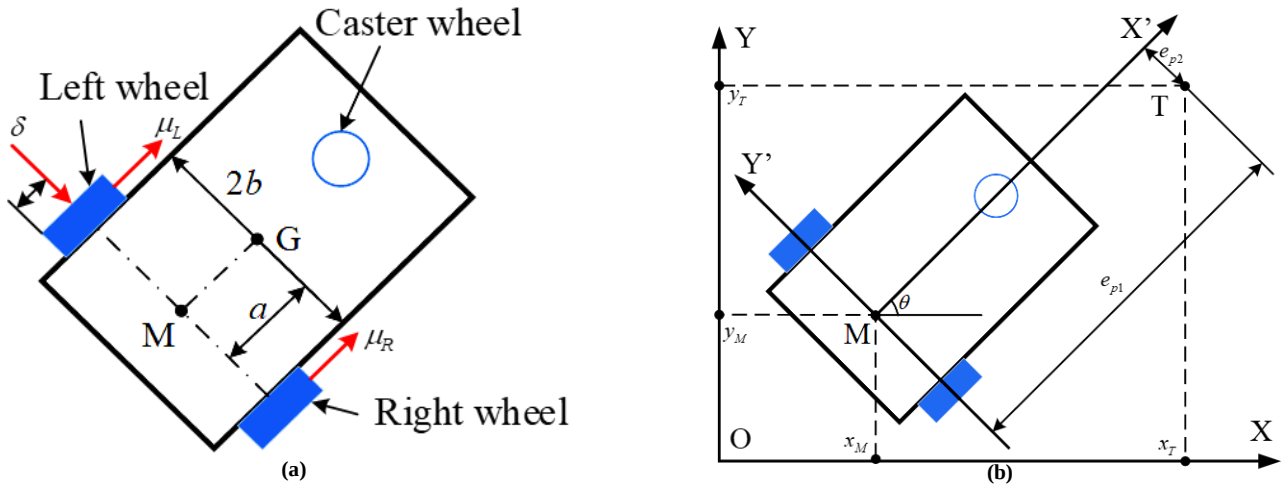


**Fig. 1 Geometric structure of WMR. (a) Model, and (b) Coordinate system**

Suppose the components of the longitudinal and transverse slip of the wheel axis are $\mu_R$, $\mu_L$, $\delta$ respectively; $\beta$ is the linear velocity; $\varpi$ is the angular velocity of the WMR.

According to the document [33], the kinetic equation of WMR is:

$$\begin{cases} \dot{x}_M = \beta cos\theta - \dot{\delta} sin\theta \\ \dot{y}_M = \beta sin\theta + \dot{\delta} cos\theta \\ \dot{\theta} = \varpi \end{cases} \quad (1)$$

In there:

$$\beta = \frac{r(\dot{\varphi}_R + \dot{\varphi}_L)}{2} + \frac{\dot{\mu}_R + \dot{\mu}_L}{2} \quad (2)$$

$$\varpi = \frac{r(\dot{\varphi}_R - \dot{\varphi}_L)}{2b} + \frac{\dot{\mu}_R - \dot{\mu}_L}{2b} \quad (3)$$

According to the document [33], the dynamic equation of WMR is:

$$M\dot{v} + Bv + Q\ddot{\mu} + C\dot{\delta} + G\ddot{\delta} + \tau_d = \tau \quad (4)$$

Where: $\tau_d$ is the input disturbance, $v = [\dot{\varphi}_R \quad \dot{\varphi}_L]^T$, $\mu = [\mu_R \quad \mu_L]^T$.

$$M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, m_{11} = m_{22}, m_{12} = m_{21}$$

$$m_{11} = m_G\left(\frac{r^2}{4} + \frac{a^2r^2}{4b^2}\right) + \frac{r^2}{4b^2}(I_G + 2I_D) + 2m_Wr^2 + I_W$$

$$m_{12} = m_G\left(\frac{r^2}{4} - \frac{a^2r^2}{4b^2}\right) - \frac{r^2}{4b^2}(I_G + 2I_D)$$

$$B = m_G\frac{r^2}{2b}\varpi\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

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

$$Q_{1,2} = m_G\frac{r}{4}\left(1 \pm \frac{a^2}{b^2}\right) \pm \frac{r}{4b}(I_G + 2I_D)$$

$$C = m_G\frac{r}{2}\varpi\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$G = m_G\frac{ar}{2b}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

The position error between point M and target point T is calculated as follows:

$$e_p = \begin{bmatrix} e_{p1} \\ e_{p2} \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix}\begin{bmatrix} x_T - x_M \\ y_T - y_M \end{bmatrix} \quad (5)$$

Differentiating (3) with respect to time, we have [34]:

$$\dot{e}_p = \begin{bmatrix} \dot{e}_{p1} \\ \dot{e}_{p2} \end{bmatrix} = \kappa v + \xi_1 \quad (6)$$

Where:

$$\kappa = \begin{bmatrix} \left(\frac{e_{p2}}{b} - 1\right)\frac{r}{2} & -\left(\frac{e_{p2}}{b} + 1\right)\frac{r}{2} \\ -\frac{e_{p1}r}{2b} & \frac{e_{p1}r}{2b} \end{bmatrix}$$

$$\xi_1 = \begin{bmatrix} \left(\frac{\dot{\mu}_R - \dot{\mu}_L}{2b}\right) e_{p2} - \frac{\dot{\mu}_R + \dot{\mu}_L}{2} \\ -\left(\frac{\dot{\mu}_R - \dot{\mu}_L}{2b}\right) e_{p1} - \dot{\delta} \end{bmatrix} + \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} \dot{x}_T \\ \dot{y}_T \end{bmatrix}$$

Set state variables $\zeta_1 = e_p$; $\zeta_2 = \dot{\zeta}_1 + \lambda\zeta_1$ where $\lambda$ is a positive scalar quantity.

The first and second derivatives of $\zeta_1$ with respect to time, we have:

$$\dot{\zeta}_1 = \dot{e}_p = \kappa v + \xi_1 \quad (7)$$

$$\ddot{\zeta}_1 = \kappa\dot{v} + \dot{\kappa}v + \dot{\xi}_1 \quad (8)$$

From equation (4), multiplying both sides by $M^{-1}$ we get:

$$\dot{v} = -M^{-1}Bv - M^{-1}\left(Q\ddot{\mu} + C\dot{\delta} + G\ddot{\delta} + \tau_d\right) + M^{-1}\tau = -M^{-1}Bv + M^{-1}\tau + \xi_2 \quad (9)$$

Where: $\xi_2 = -M^{-1}\left(Q\ddot{\mu} + C\dot{\delta} + G\ddot{\delta} + \tau_d\right)$.

Substituting (9) into (8), we get:

$$\ddot{\zeta}_1 = -\kappa M^{-1}Bv + \kappa M^{-1}\tau + \kappa\xi_2 + \dot{\kappa}v + \dot{\xi}_1 \quad (10)$$

The first derivative of $\zeta_2$ with respect to time, we have:

$$\dot{\zeta}_2 = \ddot{\zeta}_1 + \lambda\dot{\zeta}_1 \quad (11)$$

Substituting (7) and (10) into (11) we get:

$$\dot{\zeta}_2 = \ddot{\zeta}_1 + \lambda\dot{\zeta}_1 = -\kappa M^{-1}Bv + \kappa M^{-1}\tau + \kappa\xi_2 + \dot{\kappa}v + \dot{\xi}_1 + \lambda\kappa v + \lambda\xi_1$$

$$\dot{\zeta}_2 = \mathrm{E}_1 v + \mathrm{Z}\tau + \xi_3 \quad (12)$$

Where: $\mathrm{E}_1 = -\kappa M^{-1}B$, $\mathrm{Z} = \kappa M^{-1}$, $\xi_3 = \kappa\xi_2 + \dot{\kappa}v + \dot{\xi}_1 + \lambda\kappa v + \lambda\xi_1$.

From equation (7), multiply both sides by $\kappa^{-1}$ to derive:

$$v = \kappa^{-1}\dot{\zeta}_1 - \kappa^{-1}\xi_1 = \kappa^{-1}(\zeta_2 - \lambda\zeta_1) - \kappa^{-1}\xi_1 = \kappa^{-1}\zeta_2 - \kappa^{-1}\lambda\zeta_1 - \kappa^{-1}\xi_1 \quad (13)$$

Substituting (2.12) into (2.11), we get:

$$\dot{\zeta}_2 = \mathrm{E}_1\kappa^{-1}\zeta_2 - \mathrm{E}_1\kappa^{-1}\lambda\zeta_1 - \mathrm{E}_1\kappa^{-1}\xi_1 + \mathrm{Z}\tau + \xi_3 = \mathrm{E}\zeta_2 - \lambda\mathrm{E}\zeta_1 + \mathrm{Z}\tau + \xi \quad (14)$$

where: $\mathrm{E} = \mathrm{E}_1\kappa^{-1}$, $d = \xi_3 - \mathrm{E}_1\kappa^{-1}\xi_1$.

From there, we have the state equation describing the system as follows:

$$\begin{cases} \dot{\zeta}_1 = \zeta_2 - \lambda\zeta_1 \\ \dot{\zeta}_2 = \mathrm{E}\zeta_2 - \lambda\mathrm{E}\zeta_1 + \mathrm{Z}\tau + d \end{cases} \quad (15)$$

Write it down:

$$\dot{\zeta} = \mathcal{F}(\zeta) + \mathcal{G}_u\tau + \mathcal{G}_d d \quad (16)$$

where: $\mathcal{F}(\zeta) = \begin{bmatrix} \zeta_2 - \lambda\zeta_1 \\ \mathrm{E}\zeta_2 - \lambda\mathrm{E}\zeta_1 \end{bmatrix}$; $\mathcal{G}_u = \begin{bmatrix} 0 \\ Z \end{bmatrix}$; $\mathcal{G}_d = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

## 3. Design a Controller for WMR

The controller for the nonlinear system (17) is designed as follows:

$$\tau = u = u_r(\zeta) + u_d(\zeta) \quad (17)$$

In which $u_r(\zeta)$ is the adaptive optimal control component when $d = 0$, $u_d(\zeta)$ is the disturbance compensation component.

### *3.1. Design an Optimal Adaptive Controller*

The adaptive optimal controller is designed based on the IRL algorithm. When $d = 0$, the nonlinear system (4) is written as

$$\dot{\zeta} = \mathcal{F}(\zeta) + \mathcal{G}_u(\zeta)u_r \quad (18)$$

Where: $\zeta \in \mathbb{R}^n$ is the state vector;
$u_r \in \mathbb{R}^m$ is the control signal vector;

and $\mathcal{F}(\zeta) \in \mathbb{R}^n$, $\mathcal{G}_u(x) \in \mathbb{R}^{n\times m}$, $\mathcal{F}(0) = 0$, $\mathcal{F}(\zeta) + \mathcal{G}(\zeta)u$ satisfies the Lipschitz continuity property in the set $\Omega \subseteq \mathbb{R}^n$.
Definition of cost function [34]:

$$V(\zeta, u) = \int_t^\infty r(\zeta, u)d\tau \quad (19)$$

In which $r(\zeta, u) = Q(x) + u^T Ru$. With $Q(\zeta)$ being a positive definite function of $\zeta$, $R$ is a positive definite symmetric matrix.

The goal of the design is to find a control law $u(\zeta)$ that helps stabilize the system (1) and minimize the objective function (2). Before designing the IRL algorithm, we define an acceptable control law.

Definition 1: A control law $u(\zeta) \in \Psi(\zeta)$ is considered a set of acceptable control laws if and only if [36]:

- $u(\zeta)$ stabilizes the nonlinear system (18) in the region $\zeta \in \Omega$.
- The cost function, like Equation (19), corresponds to the control law $u(\zeta)$ being finite.

Definition of the Hamilton function [23]:

$$H\left(\zeta, u, V_\zeta\right) = r\left(\zeta(t), u(t)\right) + \left(\nabla V_\zeta\right)^T\left(\mathcal{F}\left(\zeta(t)\right) + \mathcal{G}\left(\zeta(t)\right)u(t)\right) \quad (20)$$

The optimal cost function $V^*(\zeta)$ satisfies the HJB equation:

$$0 = \min_{\mu\in\Psi(\Omega)} H\left(\zeta, u, \nabla V_\zeta^*\right) \quad (21)$$

Based on the stopping condition $\partial H\left(\zeta, \mu, V_\zeta^*\right)/\partial\mu = 0$ [23]

$$u^*(\zeta) = -\frac{1}{2}R^{-1}\mathcal{G}_u{}^T(\zeta)\nabla V_\zeta^*(\zeta) \quad (22)$$

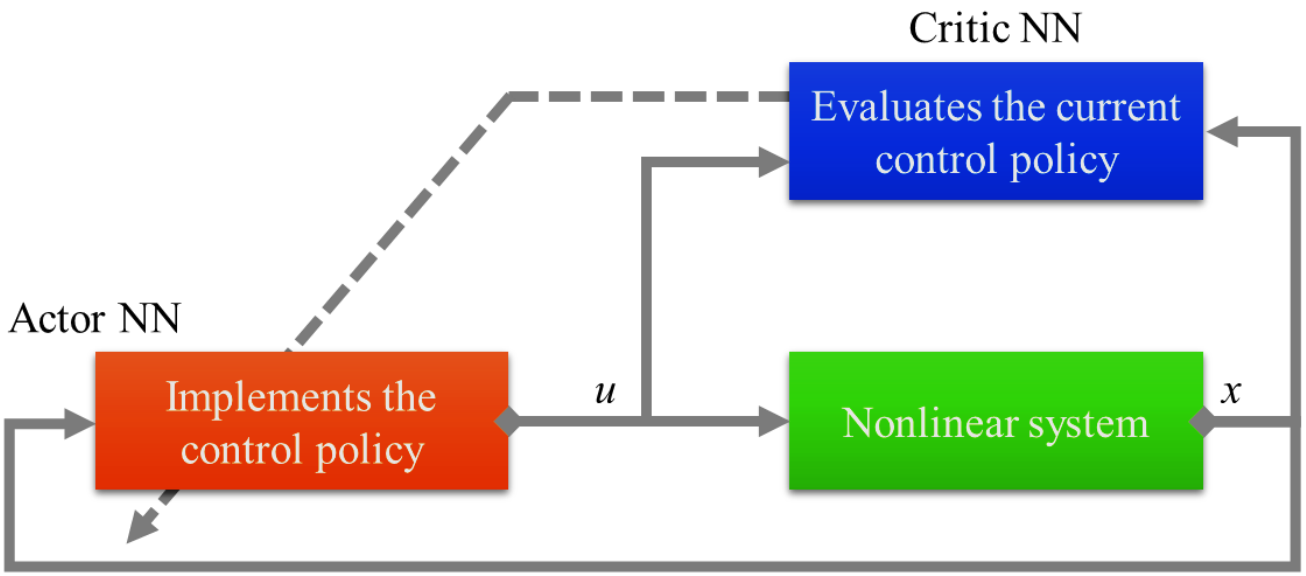


**Fig. 2 Actor-Critic structure**

Substituting (22) into (20), we get the HJB equation. Solving the HJB equation, we find $V_\zeta^*(\zeta)$. However, solving the HJB equation is not simple.

The PI algorithm [23] is an offline method that needs to know $\mathcal{F}(\zeta)$ $\mathcal{G}(\zeta)$ in advance to obtain the solution. Therefore, use the IRL algorithm to overcome the disadvantages of the PI algorithm.

Equation (19) can be transformed into a differential equation as follows:

$$V(\zeta(t)) = \int_t^{t+T} r(\zeta,u)d\tau + \int_{t+T}^{\infty} r(\zeta,u)d\tau = \int_t^{t+T} r(\zeta,u)d\tau + V\big(\zeta(t+T)\big) \quad (23)$$

Next, we use neural networks to approximate the cost function called the Critic neural network to approximate the control law called the Actor neural network. The Actor-Critic structure diagram is shown in Figure 2.

From Equation (23), the Bellman function error can be calculated [36]:

$$\int_t^{t+T} (Q(\zeta) + \mu^T R\mu)d\tau + W_1^T\phi\big(\zeta(t)\big) - W_1^T\phi\big(\zeta(t+T)\big) = \varepsilon_B \quad (24)$$

Equation (24) is written as:

$$\varepsilon_B - p = W_1^T\Delta\phi\big(\zeta(t)\big) \quad (25)$$

where: $p = \int_{t+T}^{t}(Q(\zeta) + \mu^T R\mu)d\tau$, $\Delta\phi\big(\zeta(t)\big) = \phi\big(\zeta(t)\big) - \phi\big(\zeta(t+T)\big)$.

The output of the Critic neural network is:

$$\hat{V}(\zeta) = \widehat{W}_1^T\phi(\zeta) \quad (26)$$

Then, the Bellman function approximation error is [36]:

$$\int_t^{t+T} (Q(\zeta) + \mu^T R\mu)d\tau + \widehat{W}_1^T\phi\big(\zeta(t)\big) - \widehat{W}_1^T\phi\big(\zeta(t+T)\big) = e_1 \quad (27)$$

Similarly (25) we have

$$\widehat{W}_1^T\Delta\phi\big(\zeta(t)\big) = e_1 - p \quad (28)$$

The Critic neural network weight update rule is [36]:

$$\dot{\widehat{W}}_1 = -\alpha_1 \frac{\Delta\phi\big(\zeta(t)\big)^T}{\Big(1 + \Delta\phi\big(\zeta(t)\big)^T\Delta\phi\big(\zeta(t)\big)\Big)^2}\left[\int_{t-T}^{t} (Q(\zeta) + u^T Ru)d\tau + \Delta\phi\big(\zeta(t)\big)^T\widehat{W}_1\right] \quad (29)$$

According to equation (22), we need to know the Critic neural network weight $W_1$ to find the control law. However, this parameter is not determined, so the control law is approximated by the Actor neural network [36]:

$$u_2(\zeta) = -\frac{1}{2}R^{-1}G^T(\zeta)\nabla\phi^T\widehat{W}_2 \quad (30)$$

The rule for updating Actor neural network weights is [36]:

$$\dot{\widehat{W}}_2 = -\alpha_2\Big(F_2\widehat{W}_2 - F_1\Delta\phi\big(\zeta(t)\big)^T\widehat{W}_1\Big) - \frac{1}{4}\alpha_2\bar{D}_1(\zeta)\widehat{W}_2\frac{\Delta\phi\big(\zeta(t)\big)^T}{\Big(1 + \Delta\phi\big(\zeta(t)\big)^T\Delta\phi\big(\zeta(t)\big)\Big)^2}\widehat{W}_1 \quad (31)$$

### *3.2. Nonlinear Disturbance Observer and Disturbance Compensation Controller*

Consider a nonlinear system with disturbance (16). In this section, we use a nonlinear disturbance observer to estimate unknown disturbances according to the document [37,38]:

$$\hat{d} = \eta + \rho(\zeta) \quad (32)$$

$$\dot{\eta} = -h(\zeta)\{g_d(\zeta)[\eta + \rho(\zeta)] + \mathcal{F}(\zeta) + \mathcal{G}_u(\zeta)u_d\} \quad (33)$$

Where: $h(\zeta) = \partial\rho(\zeta)/\partial\zeta$; $\hat{d}$ is the disturbance estimate of $d$

The disturbance compensation control law is designed as follows:

$$u_d(\zeta) = \beta(\zeta)\hat{d} \quad (34)$$

According to model (16), there are:

$$\mathcal{G}_u(\zeta) = Z^{-1}\mathcal{G}_d(\zeta) \quad (35)$$

From there, it can be deduced that the nonlinear disturbance compensation amplification vector is

$$\beta(\zeta) = -Z^{-1} \quad (36)$$

Finally, the disturbance compensation control law is calculated as follows:

$$u_d(\zeta) = -Z^{-1}\hat{d} \quad (37)$$

## 4. Simulation Verification

### *4.1. Control Parameters of the AC2NN Structural IRL Algorithm*

To verify the correctness of the optimal tracking control algorithm based on the AC2NN structural IRL algorithm, researchers performed numerical simulations on Matlab software with WMR parameters as in the document [33].

Choose positive definite matrices $Q = eye(4,4)$; $R = eye(2,2)$. The NN weight initialization values are: $W_1(0) = ones(9,1)$; $W_2(0) = rand(9,1)$; The initial position of WMR is chosen as: $q(0) = [0.75,1,0]^T$.

The random disturbance signal is: $d = [1 + sin(0.2t); 1 + cos(0.2t)]$. The update rate constants are chosen $\alpha_1 = 1.2$; $\alpha_2 = 10$.

Positive adjustment parameters:
$F_1 = F_2 = 10eye(length(W_1))$.

### *4.2. Simulation Results*

To verify the effectiveness of the proposed algorithm, the authors performed system simulations on Matlab software with the WMR scenario following a curved trajectory.

The desired tracking trajectory of the WMR is:

$$\begin{cases} x_{ref} = t \\ y_{ref} = sin(0.5t) + 0.5t + 1 \end{cases} \quad (38)$$

The simulation results shown in Figures 3 to 6 were performed using Matlab software. It can be seen that at first, the two neural networks, Critic NN and Actor NN, are in the learning process, so the results of tracking the reference trajectory of WMR are not good.

However, after this period of time, the weights of the two neural networks converge. The controller designed for WMR approximates and converges to optimal tracking quality. This results in the WMR's trajectory tracking quality increasing and the WMR tracking the reference trajectory, tracking error is almost zero for all variables.

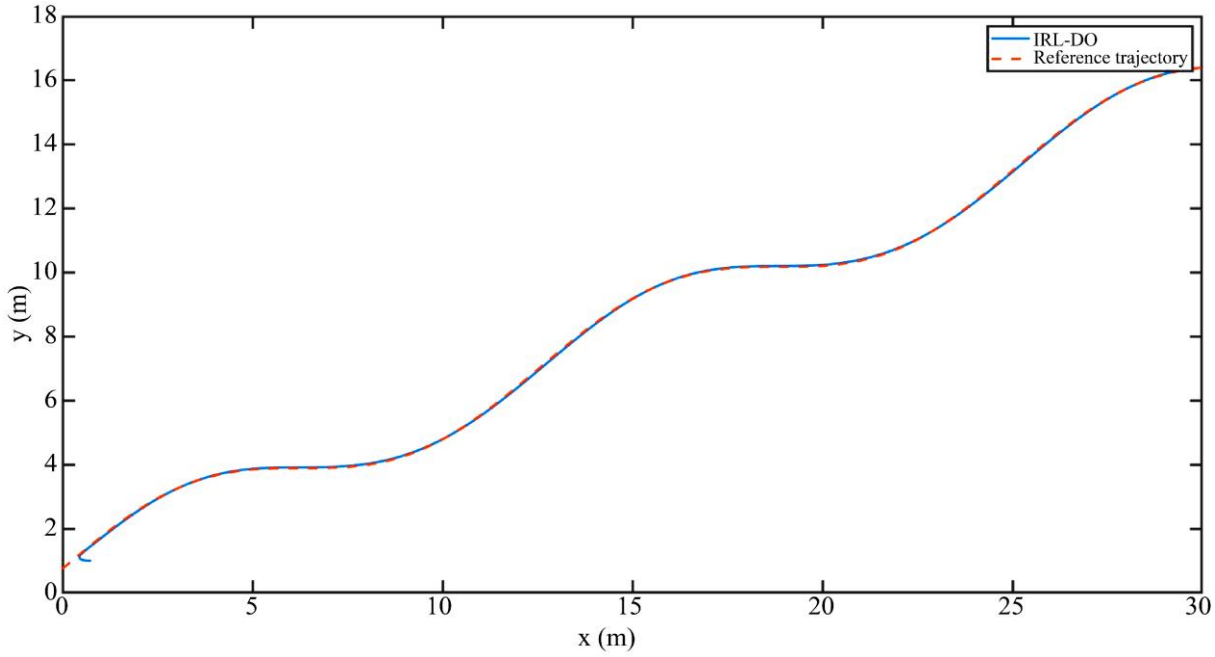


**Fig. 3 Tracking trajectory – curved trajectory using AC2NN structure IRL-DO algorithm**

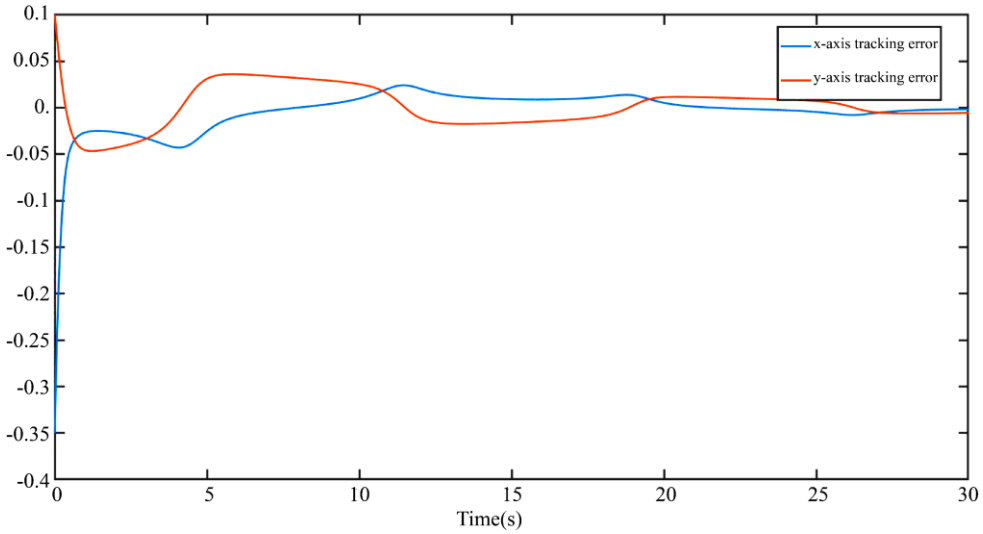


**Fig. 4 Error tracking trajectory along x and y axis - curved trajectory**

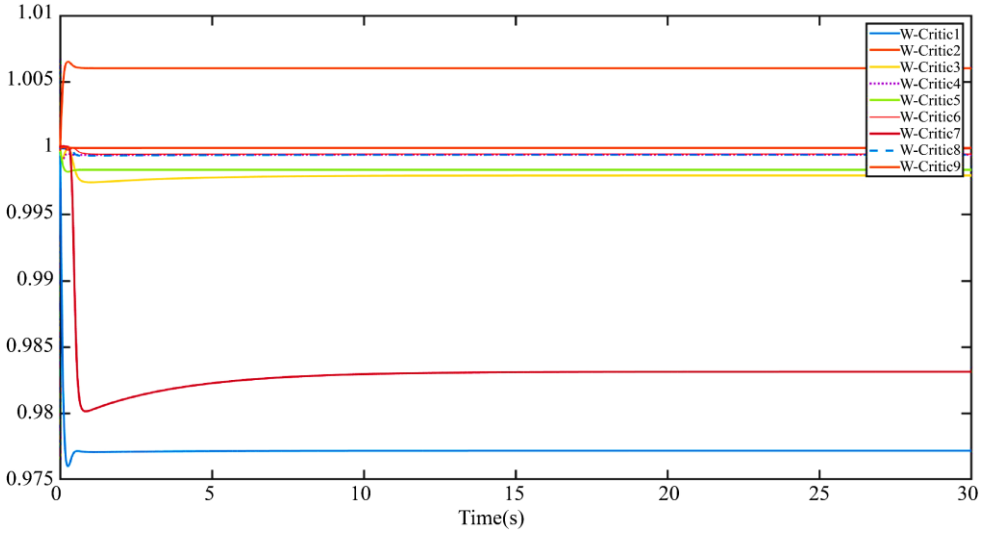


**Fig. 5 Convergence of the critic NN weight matrix during the learning and control process - curved trajectory**

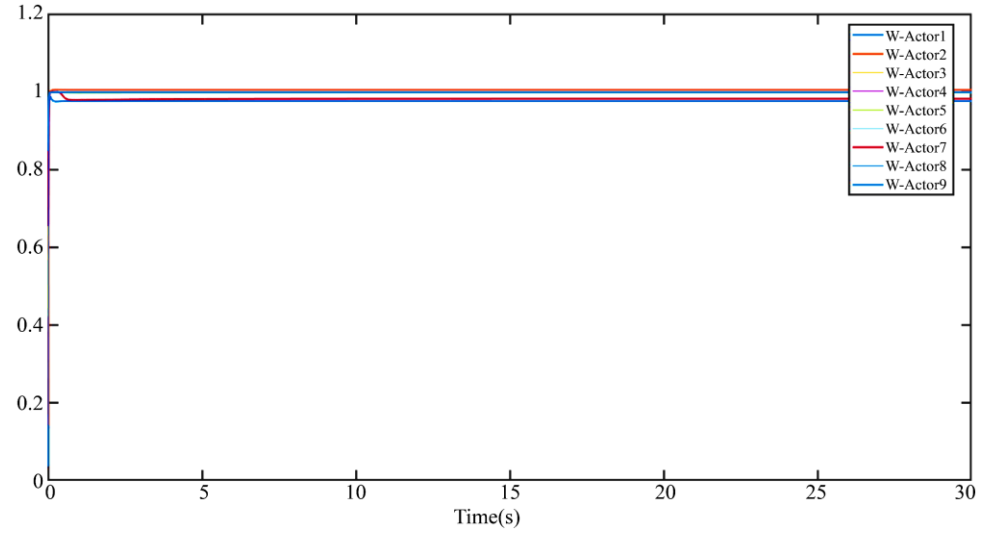


**Fig. 6 Convergence of the actor NN weight matrix during the learning and control process - curved trajectory**

## 5. Conclusion

The IRL adaptive optimal controller has an AC2NN structure, a Critic NN to approximate the cost function and an Actor NN to approximate the optimal control law. This controller is combined with the system's input disturbance component estimator, the disturbance estimation results are of good quality, and the control structure ensures that the system follows the set trajectory. The trajectory tracking error and turbulence estimation error are small. Parameter tuning algorithms have been proposed to learn the optimal control solution online while ensuring system stability.

## Acknowledgments

This study was supported by the University of Economics - Technology for Industries, Ha Noi - Vietnam; http://www.uneti.edu.vn/.

## References

[1] Jie Meng et al., "Two-Wheeled Robot Platform Based on PID Control," *5th International Conference on Information Science and Control Engineering*, Zhengzhou, China, pp. 1011-1014, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[2] R. Fierro, and F.L. Lewis, "Control of a Nonholomic Mobile Robot: Backstepping Kinematics into Dynamics," *Journal of Robotic Systems*, vol. 14, no. 3, pp. 149-163, 1997. [CrossRef] [Google Scholar] [Publisher Link]

[3] Jun Ye, "Tracking Control for Nonholonomic Mobile Robots: Integrating the Analog Neural Network into the Backstepping Technique," *Neurocomputing*, vol. 71, no. 16-18, pp. 3373-3378, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[4] Shubhobrata Rudra, Ranjit Kumar Barai, and Madhubanti Maitra, "Design and Implementation of a Block-Backstepping Based Tracking Control for Nonholonomic Wheeled Mobile Robot," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 14, pp. 3018-3035, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[5] Xing Wu et al., "Backstepping Trajectory Tracking Based on Fuzzy Sliding Mode Control for Differential Mobile Robots," *Journal of Intelligent & Robotic Systems*, vol. 96, pp. 109-121, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] Bong Seok Park et al., "Adaptive Tracking Control of Nonholonomic Mobile Robots Considering Actuator Dynamics: Dynamic Surface Design Approach," *American Control Conference*, St. Louis, MO, USA, pp. 3860-3865, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[7] Omid Mohareri, Rached Dhaouadi, and Ahmad B. Rad, "Indirect Adaptive Tracking Control of a Nonholonomic Mobile Robot via Neural Networks," *Neurocomputing*, vol. 88, pp. 54-66, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[8] Altan Onat, and Metin Ozkan, "Dynamic Adaptive Trajectory Tracking Control of Nonholonomic Mobile Robots Using Multiple Models Approach," *Advanced Robotics*, vol. 29, no. 14, pp. 913-928, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[9] Mohammad Mehdi Fateh, and Aliasghar Arab, "Robust Control of a Wheeled Mobile Robot by Voltage Control Strategy," *Nonlinear Dynamics*, vol. 79, no. 1, pp. 335-348, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[10] P. Navin Chandra, and S.J. Mija, "Robust Controller for Trajectory Tracking of a Mobile Robot," *IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems*, Delhi, India, pp. 1-6, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[11] Spandan Roy et al., "Robust Control of Nonholonomic Wheeled Mobile Robot with Past Information: Theory and Experiment," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 231, no. 3, pp. 178-188, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[12] Chung-Hsun Sun, Yin-Tien Wang, and Cheng-Chung Chang, "Design of T-S Fuzzy Controller for Two-Wheeled Mobile Robot," *Proceedings of International Conference on System Science and Engineering*, Macau, China, pp. 223-228, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[13] Min-Chi Kao et al., "Adaptive Type-2 Fuzzy Tracking Control of Wheeled Mobile Robots," *International Conference on Fuzzy Theory and Its Applications*, Taipei, Taiwan, pp. 1-6, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[14] Qing Xu et al., "Fuzzy PID Based Trajectory Tracking Control of Mobile Robot and its Simulation in Simulink," *International Journal of Control and Automation*, vol. 7, no. 8, pp. 233-244, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[15] Nacer Hacene, and Boubekeur Mendil, "Fuzzy Behavior-Based Control of Three Wheeled Omnidirectional Mobile Robot," *International Journal of Automation and Computing*, vol. 16, pp. 163-185, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[16] Rafael Morales et al., "Robotics and Control Engineering of Wave and Tidal Energy-Recovering Systems," *Mathematical Problems in Engineering*, vol. 2018, pp. 1-2, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[17] Mohamed Abdelwahab et al., "Trajectory Tracking of Wheeled Mobile Robots Using Z-Number Based Fuzzy Logic," *IEEE Access*, vol. 8, pp. 18426-18441, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] Alexandr Štefek et al., "Optimization of Fuzzy Logic Controller Used for a Differential Drive Wheeled Mobile Robot," *Applied Sciences*, vol. 11, no. 13, pp. 1-23, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] R. Fierro, and F.L. Lewis, "Control of a Nonholonomic Mobile Robot Using Neural Networks," *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 589-600, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[20] Zhijun Li et al., "Trajectory-Tracking Control of Mobile Robot Systems Incorporating Neural-Dynamic Optimized Model Predictive Approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 6, pp. 740-749, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[21] Pavol Bozek et al., "Neural Network Control of a Wheeled Mobile Robot Based on Optimal Trajectories," *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, pp. 1-10, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] Ziyu Chen et al., "Adaptive-Neural-Network-Based Trajectory Tracking Control for a Nonholonomic Wheeled Mobile Robot with Velocity Constraints," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 6, pp. 5057-5067, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] D. Vrabie et al., "Adaptive Optimal Control for Continuous-Time Linear Systems Based on Policy Iteration," *Automatica*, vol. 45, no. 2, pp. 477-484, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[24] Richard S. Sutton, and Andrew G. Barto, *Introduction to Reinforcement Learning*, 1998.

[25] Kyriakos G. Vamvoudakis, "*Online Learning Algorithms for Differential Dynamic Games and Optimal Control*," Federated Electronic Theses and Dissertation, pp. 1-218, 2011. [Google Scholar] [Publisher Link]

[26] Yu Jiang, and Zhong-Ping Jiang, *Robust Adaptive Dynamic Programming*, Wiley, pp. 1-216, 2017. [Google Scholar] [Publisher Link]

[27] J.J. Murray et al., "Adaptive Dynamic Programming," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 32, no. 2, pp. 140-153, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[28] Yu Jiang, and Zhong-Ping Jiang, "Computational Adaptive Optimal Control for Continuous-Time Linear Systems with Completely Unknown Dynamics," *Automatica*, vol. 48, no. 10, pp. 2699-2704, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[29] Draguna Vrabie, and Frank Lewis, "Neural Network Approach to Continuous-Time Direct Adaptive Optimal Control for Partially Unknown Nonlinear Systems," *Neural Networks*, vol. 22, no. 3, pp. 237-246, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[30] Kyriakos G. Vamvoudakis, and Frank L. Lewis, "Online Actor-Critic Algorithm to Solve the Continuous-Time Infinite Horizon Optimal Control Problem," *Automatica*, vol. 46, no. 5, pp. 878-888, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[31] Yu Jiang, and Zhong-Ping Jiang, "Global Adaptive Dynamic Programming for Continuous-Time Nonlinear Systems," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 2917-2929, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[32] Quan-Yong Fan, and Guang-Hong Yang, "Adaptive Actor-Critic Design-Based Integral Sliding-Mode Control for Partially Unknown Nonlinear Systems with Input Disturbances," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 1, pp. 165-177, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[33] Hoa Van Doan, and Nga Thi-Thuy Vu, "Adaptive Sliding Mode Control for Uncertain Wheel Mobile Robot," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 4, pp. 3939-3947, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[34] Kyriakos G. Vamvoudakis, and Frank L. Lewis, "Online Actor Critic Algorithm to Solve the Continuous-Time Infinite Horizon Optimal Control Problem," *International Joint Conference on Neural Networks*, Atlanta, GA, USA, pp. 3180-3187, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[35] Murad Abu-Khalaf, and Frank L. Lewis, "Nearly Optimal Control Laws for Nonlinear Systems with Saturating Actuators Using a Neural Network HJB Approach," *Automatica*, vol. 41, no. 5, pp. 779-791, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[36] Kyriakos G. Vamvoudakis, Draguna Vrabie, and Frank L. Lewis, "Online Adaptive Learning of Optimal Control Solutions Using Integral Reinforcement Learning," *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, Paris, France, pp. 250-257, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[37] J. Yang, W.H. Chen, and S. Li, "Non-Linear Disturbance Observer-Based Robust Control for Systems with Mismatched Disturbances/Uncertainties," *IET Control Theory & Applications*, vol. 5, no. 18, pp. 2053-2062, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[38] Keith Dupree et al., "Asymptotic Optimal Control of Uncertain Nonlinear Eulerlagrange Systems," *Automatica*, vol. 47, no. 1, pp. 99-107, 2010. [CrossRef] [Google Scholar] [Publisher Link]