Asymptotically Stable Controller Design via Inverse Optimal Control

Shuhuai Tan, Kexi Yan, Jie Zhang, Yongting Chen, Qinglai Wei, and Fei-Yue Wang

***Abstract*—The design of stabilizing controllers for general nonlinear systems remains a challenging task due to their inherent complexities and nonconvexities. In this paper, we consider the problem of designing an asymptotically stable controller of a nonlinear dynamic system. We begin by framing the problem as an inverse optimal control problem, aiming to design a pair of cost functions that ensure asymptotic stability for the nonlinear model predictive control closed-loop system. By leveraging the relaxed dynamic programming inequality, a machine learning based algorithm is proposed to learn the cost functions. Finally, we demonstrate the effectiveness of the proposed method through illustrative examples.**

I. INTRODUCTION

Throughout the history of control science and engineering, system stability has always been a central research focus. Traditional analytical frameworks represented by Lyapunov stability theory have provided rigorous mathematical foundations for the long-term reliability and convergence of system dynamic behaviors. The Lyapunov function is a crucial tool within this framework [1]. It serves to analyze and prove the stability of dynamic systems, particularly in nonlinear and time-varying systems. The Lyapunov method offers an analysis approach independent of specific system solutions. However, fundamental challenges persist. Determining the existence of a Lyapunov function for a given system and constructing such functions remain open problems, with no universal solution method established to date. Recently, in Ref. [2], Meta's artificial intelligence (AI) team proposes a new machine learning based method using sequence-to-sequence Transformers [3] to discover global Lyapunov functions. However, this method requires known Lyapunov functions as training data and can only discover global Lyapunov functions. Some other methods also use machine learning based methods to discover Lyapunov functions [4, 5].

Starting from the 1950s, optimal control problems received widespread attention and gradually matured. Solving infinite-horizon optimal control problems provides a way to construct Lyapunov functions, as well as a method to design asymptotically stable controllers [6]. Optimal control problems typically aim to optimize a specific performance metric, solving for control inputs through the mathematical theoretical foundations of the Pontryagin's minimum principle [7] and dynamic programming (DP) [8–10].

With the continuous increase in the complexity of dynamic systems, traditional optimal control methods face new challenges. On the one hand, high-dimensional state spaces lead to an exponential increase in computational complexity, which is known as the "curse of dimensionality". On the other hand, dynamic systems often exhibit complex characteristics such that accurate mathematical modeling presents inherent difficulties. A series of data-driven intelligent methods are proposed to handle these challenges. Among these, reinforcement learning (RL) [11] and adaptive dynamic programming (ADP) [12] are methods that can approximately solve optimal control problems with unknown system dynamics in a forward-in-time manner, thereby avoiding the "curse of dimensionality" problem to some extent. However, these methods typically require a well-designed reward function to guide the learning process, which is often difficult to obtain in practice.

Inverse optimal control (IOC) [13], also known as inverse reinforcement learning (IRL) [14], which can be traced back to Ref. [15], is a control approach used when the reward function or cost function of the system is unknown or partially unknown. Contrary to forward optimal control, IOC methods aim to infer the unknown reward function of the dynamic system in a Markov decision process from expert demonstrations of the optimal policy. IOC currently has applied research in multiple fields, including autonomous driving [16, 17], human-robot collaboration [18], multi-agent system control [19], and anomaly detection [20].

Optimal control requires the value function to satisfy the dynamic programming equality (optimality), whereas asymptotic stability control only requires the value function to satisfy the relaxed dynamic programming (RDP) inequality

Manuscript received: 10 January 2025; revised: 3 April 2025; accepted: 6 May 2025. *(Corresponding author: Jie Zhang.)*

Citation: S. Tan, K. Yan, J. Zhang, Y. Chen, Q. Wei, and F.-Y. Wang, Asymptotically stable controller design via inverse optimal control, *Int. J. Intell. Control Syst.*, 2025, 30(2), 182–188.

Shuhuai Tan, Kexi Yan, Jie Zhang, and Qinglai Wei are with State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: tanshuhuai2025@ia.ac.cn; yankexi2022@ia.ac.cn; jie.zhang@ia.ac.cn; qinglai.wei@ia.ac.cn).

Yongting Chen is with Tandon School of Engineering, New York University, New York, NY 11201, USA (e-mail: yc5739@nyu.edu).

Fei-Yue Wang is with State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: feiyue.wang@ia.ac.cn).

Qinglai Wei and Fei-Yue Wang are also with Institute of Systems Engineering, Macau University of Science and Technology, Macao 999078, China.

Digital Object Identifier 10.62678/IJICS202506.10185

(sub-optimality), which is generally a more lenient condition [21–23]. This fact inspires us to use the IOC method to design cost functions that satisfy the relaxed dynamic programming inequality, which then ensures the asymptotic stability of the closed-loop system.

In this paper, we propose a method to design an asymptotically stable controller via inverse optimal control. Our method first uses the idea of inverse optimal control to infer the cost function of the system, then leverages the cost function to attain an asymptotically stable controller. The contributions of the method can be listed as follows:

(1) As far as we know, this is the first time that inverse optimal control is used to design an asymptotically stable controller.

(2) Compared to typical IOC methods and Meta AI's sequence-to-sequence Transformers method, our method does not require optimal trajectories or expert demonstrations.

Notation Let $\mathbb{N}_0$ be the set of non-negative integers, that is $\mathbb{N}_0 = \{0,1,2,\ldots\}$. Let $\mathbb{R}^n$ be the n-dimensional Euclidean space, $\mathbb{R}_0^+ = [0,+\infty)$ be the set of non-negative real numbers.

II. Inverse Asymptotic Stability Problem

A. DP and RDP

Consider the discrete-time deterministic system

$$x_{k+1} = f(x_k, u_k) \tag{1}$$

where $k \in \mathbb{N}_0$, $f(\cdot,\cdot): \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ is the system dynamics, which is possibly nonlinear, $x_k \in \mathbb{R}^n$ is the state variable, and $u_k \in \mathbb{R}^m$ is the control input.

Dynamic programming is a classic tool for solving optimal control problems. Let $\mathbb{X} \subseteq \mathbb{R}^n$ and $\mathbb{U} \subseteq \mathbb{R}^m$ be the state space and control input space, respectively. The DP principle states that the optimal value function $V(n, x_0)$ satisfies the Bellman equation

$$V(n,x_0) = \min_{\mu}\{\lambda(x_0,\mu(n,x_0)) + V(n+1, f(x_0,\mu(n,x_0)))\} \tag{2}$$

where $n \in \mathbb{N}_0$, $\mu: \mathbb{N}_0 \times \mathbb{X} \mapsto \mathbb{U}$ is a feedback control law, $\lambda: \mathbb{X} \times \mathbb{U} \mapsto \mathbb{R}_0^+$ is a stage cost function, and x_0 is the initial state. The feedback control law obtained from the Bellman equation by solving the infinite-horizon optimal control problem also stabilizes the closed-loop system. Thus, solving infinite-horizon optimal control problems using dynamic programming provides a way to construct Lyapunov functions and design asymptotically stable controllers [6], if λ is given.

Compared with DP, the relaxed dynamic programming is considered a more lenient condition than the Bellman equation

$$V(n,x_0) \geqslant \alpha\lambda(x_0,\mu(n,x_0)) + V(n+1, f(x_0,\mu(n,x_0))) \tag{3}$$

where $n \in \mathbb{N}_0$, $\mu: \mathbb{N}_0 \times \mathbb{X} \mapsto \mathbb{U}$ is a feedback control law, $V: \mathbb{N}_0 \times \mathbb{X} \mapsto \mathbb{R}_0^+$ is the corresponding value function of μ, and $\lambda: \mathbb{X} \times \mathbb{U} \mapsto \mathbb{R}_0^+$ is a stage cost. Let $\mathbb{X}_0$ be a forward invariant set of the closed-loop system and desired equilibrium $x_{\text{ref}} \in \mathbb{X}_0 \subseteq \mathbb{X}$. If the RDP inequality in Formula (3) is satisfied for some $\alpha \in (0,1]$ and all x_0 in $\mathbb{X}_0$ and all $n \in \mathbb{N}_0$, the closed-loop system is asymptotically stable on $\mathbb{X}_0$ [6].

B. Inverse RDP Approach

The DP principle enables the synthesis of asymptotically stable controllers by solving infinite-horizon optimal control problems, but a well-designed cost function is required. Traditional inverse optimal control methods provide a way to infer the cost function, yet they typically rely on access to optimal trajectories or expert demonstrations. In this work, we propose a novel approach to construct asymptotically stable controllers from scratch, leveraging the RDP in conjunction with inverse optimal control to infer suitable cost functions, without requiring predefined cost or expert-generated trajectories.

By inverting RDP, we derive sufficient conditions on the cost functions that guarantee the asymptotic stability of a given closed-loop system. Let the equilibrium point x_{ref} belong to a target set $\mathbb{X}_0 \subseteq \mathbb{X}$, and denote the corresponding N-step feasible set of $\mathbb{X}_0$ as $\mathbb{X}_N$. Suppose that we find a stage cost function $\lambda: \mathbb{X} \times \mathbb{U} \mapsto \mathbb{R}_0^+$ and a terminal cost function $F: \mathbb{X} \mapsto \mathbb{R}_0^+$ such that, for all $x \in \mathbb{X}_N$, the following RDP inequality is satisfied

$$V_N(n+1, f(x,\mu_N(n,x))) + \lambda(x,\mu_N(n,x)) - V_N(n,x) \leqslant 0 \tag{4}$$

and for all $x \in \mathbb{X}_0$, the terminal Lyapunov inequality holds

$$F(f(x,\mu_N(n,x))) + \lambda(x,\mu_N(n,x)) - F(x) \leqslant 0 \tag{5}$$

where the value function $V_N(n,x)$ and feedback control law $\mu_N(n,x)$ are obtained by solving the associated finite-horizon optimal control problem with cost functions λ and F, then the closed-loop system is asymptotically stable on $\mathbb{X}_N$.

Given any candidate cost functions, finite-horizon optimal control problems are solved to generate a collection of demonstration trajectories. The extent to which these trajectories violate Formulas (4) and (5) provides a quantitative basis for evaluating and refining the cost functions. We denote the parameterized cost functions as $\lambda_{\theta_1}(x,u;\theta_1)$ and $F_{\theta_2}(x;\theta_2)$, where θ_1 and θ_2 are the parameter vectors of the functions. The design of these functions can then be posed as the following optimization problem

$$\min_{\theta_1,\theta_2} L(\lambda_{\theta_1}, F_{\theta_2}) \tag{6}$$

where $L(\lambda_{\theta_1}, F_{\theta_2})$ is a stability metric that measures the extent of violation of the RDP, i.e., $L(\lambda_{\theta_1}, F_{\theta_2}) = 0$ if and only if Formulas (4) and (5) hold for all the demonstration trajectories, otherwise we have $L(\lambda_{\theta_1}, F_{\theta_2}) > 0$.

To solve the optimization problem of Eq. (6), we use nonlinear model predictive control (NMPC) to solve a receding horizon optimal control problem to obtain the NMPC value function $V_N(n,x)$, NMPC feedback control law $\mu_N(n,x)$, and state trajectories as demonstrations, and then the stability metric $L(\lambda_{\theta_1}, F_{\theta_2})$ is computed based on the demonstrations. Then, we use gradient descent to update the cost function parameters θ_1 and θ_2 according to the stability metric $L(\lambda_{\theta_1}, F_{\theta_2})$.

III. Illustrative Example

To facilitate a deeper understanding of the proposed

method, this section first provides a linear quadratic system as an illustrative example to demonstrate the key differences between our algorithm and the traditional optimal control methods. Then, a CartPole nonlinear system is used to validate the effectiveness of the algorithm.

A. Linear Quadratic System

Give a linear quadratic system

$$x_{k+1} = x_k + u_k \tag{7}$$

$$J_N = F x_N^2 + \sum_{k=0}^{N-1} (Q x_k^2 + R u_k^2) \tag{8}$$

where $x, u \in \mathbb{R}$, $F, Q, R \in [0, +\infty)$, and $x_{\text{ref}} = 0$ is the equilibrium point. The objective is to design the cost functions such that the optimal control stabilizes the system.

Two cases are considered in this example. The first case assumes that the stage cost parameters Q and R are known, and we need to infer the terminal cost parameter F. The second case only assumes that the stage cost R is known, and we need to infer both Q and F.

NMPC with terminal cost F is often referred to as quasi-infinite horizon NMPC. The reason is that when terminal cost function $F(\cdot)$ is an approximation of the infinite-horizon value function, the finite horizon dynamic programming principle is an approximation of the infinite-horizon dynamic programming principle. Consequently, the NMPC actually solves an approximate infinite-horizon optimal control problem at each time step [6].

For this system, if we consider the infinite-horizon optimal control, the objective function becomes

$$J_\infty = \sum_{k=0}^{\infty} Q x_k^2 + R u_k^2 \tag{9}$$

Solving the Bellman equation allows us to directly compute the optimal value function $V_\infty(x) = \bar{K} x^2$. The corresponding infinite-horizon optimal control law stabilizes the system. In fact, this is also what reinforcement learning is focused on: to approximately solve the Bellman equation, estimate the optimal value function, and thereby obtain the optimal control law, which is also an asymptotically stable control law.

A value function that fulfills the Bellman equation also satisfies the RDP inequality (with $\alpha = 1$) and achieves equality. Considering the relationship between F and $\bar{K}$, for fixed Q and R and a small NMPC horizon N, the stability metric $L(\lambda_{\theta_1}, F_{\theta_2})$ is expected to be close to 0 when the terminal cost parameter F is close to $\bar{K}$. However, the Bellman equation is only a special case of the RDP inequality. RDP inequality being satisfied does not strictly require F to approximate $\bar{K}$, and this highlights the difference between our algorithm and general optimal control methods like ADP or RL.

One of the objectives of the experiment is to validate the above discussion. The stability metric only includes RDP inequality in Formula (4), as it is easy to prove that the resulting terminal cost F must be a Lyapunov function for linear quadratic problems. We use the penalty function $P(z) = \max(0, (z - \epsilon))^2$ to transform the RDP inequality into the stability metric, where ϵ is a small positive number.

a. Inferring F with Known Q and R

In this case, running cost parameters are fixed. NMPC prediction horizon is set to $N = 1$. The initial state x_0 is sampled from $[-1000, 1000]$ uniformly, and $M = 128$ samples are drawn for each epoch.

Given $Q = 1$ and $R = 1$, the infinite horizon optimal value function can be computed by solving the Bellman equation, and the resulting optimal value function is $V_\infty(x) = 1.61803x^2$. Choose $F = 1.5$ as the initial value of the terminal cost parameter. The loss curve is shown in Fig. 1(a) and the change of the terminal cost parameter F is shown in Fig. 1(b). It can be observed that the loss converges to a small value, and the terminal cost parameter F converges to 1.6180.

This convergence is consistent with the expectation that the terminal cost F approaches the optimal value function $\bar{K}$, which indicates that after updating the terminal cost parameter F using the proposed algorithm, the resulting NMPC value function $V_1(n, x)$ can be regarded as an approximation of the infinite-horizon value function $V_\infty(x)$. So, NMPC value function $V_1(n, x)$ satisfies the Bellman equation globally, thus the RDP inequality also holds, which implies that the closed-loop system is asymptotically stable globally.

We also run the experiments with different Q values. For each Q, $\bar{K}$ is computed by solving the Bellman equation, and the initial value of F is set to a value that is less than $\bar{K}$. The results are shown in Table 1, where the second column is the initial value of F, the third column is the converged value of

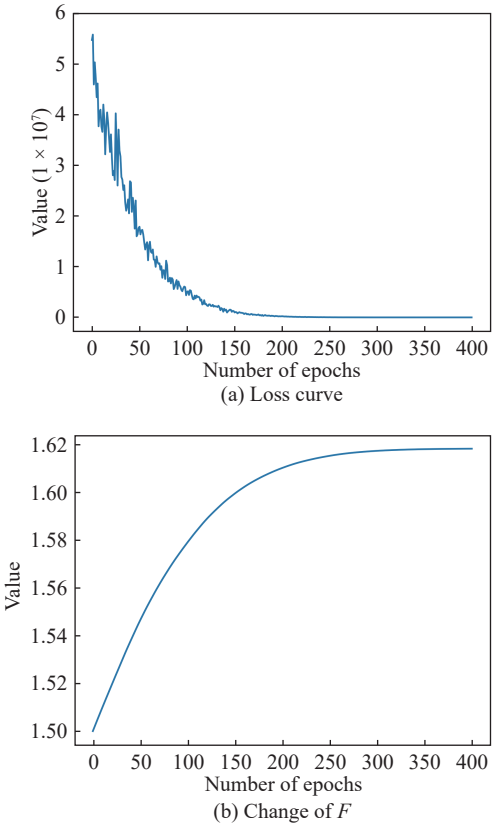


Figure 1 Control results of the linear quadratic system with known Q and R.

F after iteratively updating, and the fourth column is the expected value of $\bar{K}$. The results in Table 1 show that the terminal cost parameter F does converge to $\bar{K}$ as expected.

Table 1 Results under different Q values.

Q	Initial F	Converged F	Expected $\bar{K}$
0.5	0.9	1.0000	1.000,000,00
0.8	1.2	1.3798	1.379,795,91
1.0	1.4	1.6180	1.618,033,99
1.2	1.6	1.8490	1.848,999,65
1.6	2.0	2.2967	2.296,662,98
2.0	2.4	2.7320	2.732,050,81

The following part focuses on other values of F that also make the RDP inequality hold. Considering the objective function of $J_1 = F(x+u)^2 + Qx^2 + Ru^2$, taking the partial derivative with respect to u, and setting it to zero, we can obtain the optimal control law $u^* = -Fx/(F+1)$. Substituting u^* into the objective function, the optimal value function can be obtained, and then substituting both u^* and the optimal value function into the RDP inequality, we have

$$V_1(n+1, f(x,u^*)) + \lambda(x,u^*) - V_1(n,x) \leqslant 0 \tag{10}$$

Eliminating x in Formula (10) leads to $F^3 - 2F - 1 \geqslant 0$, which has the solution of $F \geqslant 1.6180$ (the negative roots are discarded). This indicates that as long as the terminal cost F is greater than or equal to 1.6180, the RDP inequality holds, and the closed-loop system is asymptotically stable globally. When $F \geqslant 1.6180$, the stability metric is always zero, thus the terminal cost parameter F will not be updated by the algorithm. We verified this by running a series of experiments with different initial values of $F \geqslant 1.6180$.

b. Inferring Q and F with Known R

In this case, we fix $R = 1$ and update both Q and F. The NMPC prediction horizon is set to $N = 1$. Specify the initial value of F to be 1.5 and the initial value of Q to be 2.5. The loss curve is shown in Fig. 2(a) and the change of the cost parameters Q and F is shown in Fig. 2(b). The value of Q converges to 1.5510, and the value of F converges to 2.2426. Fixing $Q = 1.5510$ and $R = 1$, the infinite horizon optimal value function can be computed by solving the Bellman equation, and the result is $V_\infty(x) = 2.2426x^2$. Solving the RDP inequality in Formula (10) analytically with $Q = 1.5510$ results in $F \geqslant 2.24260608$. These results are again consistent with the expectation.

B. Nonlinear CartPole System

Consider a nonlinear CartPole system with the following model parameters: cart mass $M = 1$ kg, pendulum mass $m = 0.1$ kg, pendulum length $L = 0.5$ m, and gravitational acceleration $g = 9.8\ \mathrm{m/s^2}$. The system state is denoted as $[x, \dot{x}, \cos\theta, \sin\theta, \dot{\theta}] \in \mathbb{R}^5$, with each dimension representing displacement, velocity, cosine of the pendulum angle, sine of the pendulum angle, and pendulum angular velocity, respectively. The control input is $u \in \mathbb{R}$. When the CartPole system stabilizes at the upright position, the system state

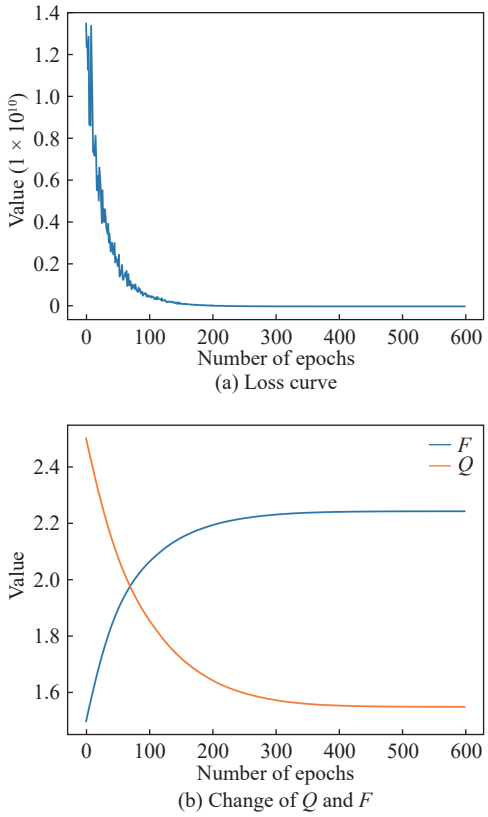


Figure 2 Control results of the linear quadratic system with known R.

becomes $x_{\mathrm{ref}} = [0,0,1,0,0]$, which is an equilibrium point. We set this point as the desired equilibrium for the NMPC. The optimal control objective function is given by

$$J = (x_N - x_{\mathrm{ref}})^{\mathrm{T}} F (x_N - x_{\mathrm{ref}}) + \sum_{k=0}^{N-1} (x_k - x_{\mathrm{ref}})^{\mathrm{T}} Q (x_k - x_{\mathrm{ref}}) + u_k^{\mathrm{T}} R u_k \tag{11}$$

where F and Q are parameter matrices to be learned, and R is a known parameter.

In the experiment, R is fixed to 0.01. The parameter matrix Q_{θ_1} for the running cost is initialized as an identity matrix ($Q_0 = I$) and remains diagonal during subsequent updates (only the diagonal elements are treated as parameters to be learned). The parameter matrix F_{θ_2} for the terminal cost is randomly initialized using a normal distribution. To ensure the positive definiteness of the parameter matrices, the practical running cost is $Q_{\theta_1}^{\mathrm{T}} Q_{\theta_1}$ and the terminal cost is $F_{\theta_2}^{\mathrm{T}} F_{\theta_2}$. The initial parameter for the terminal cost function is

$$F_0 = \begin{bmatrix} 5.8778 & 3.1375 & 4.8410 & -3.5438 & 4.3548 \\ 3.1375 & 4.1936 & 1.7407 & -4.2103 & 1.4381 \\ 4.8410 & 1.7407 & 5.4636 & -1.3833 & 5.8233 \\ -3.5438 & -4.2103 & -1.3833 & 5.7621 & -2.1388 \\ 4.3548 & 1.4381 & 5.8233 & -2.1388 & 9.1122 \end{bmatrix}.$$

The performance metric in the experiment consists of the RDP inequality term, the terminal Lyapunov inequality term, and the parameter regularization term. We use a quadratic function $P(z) = z^2$ as the penalty function to convert the

inequalities into the stability metric. To prevent Q and F from being too small, we regularize the summation of the parameters to a fixed value using the following regularization term

$$\beta(K - \sum \text{abs}(\theta_i))^2 \tag{12}$$

where θ_i are diagonal elements of Q and all elements of F, K is a constant, and β is a hyperparameter. We set $K = 5$ and $K = 25$ for Q and F, respectively. To balance the influence of these three terms on parameter learning, we multiply the terminal Lyapunov inequality term by 100 and let $\beta = 0.001$.

The initial angle of the pendulum is sampled from $[-\pi/6, \pi/6]$ uniformly. The angle range of the target state is set to $[-\pi/60, \pi/60]$, and only when the angle is within this range, the terminal Lyapunov inequality term is activated.

The NMPC prediction horizon is set to $N = 32$ and the batch size is set to $M = 64$. The CartPole environment and the optimal control solver are from Ref. [24]. To better handle objective functions in Bolza form, some modifications are made to the solver.

The training loss curve of the total stability metric and each term is shown in Fig. 3. It is observed that except for the Lyapunov inequality loss, the other terms quickly converged. However, it can be noted that the Lyapunov inequality loss has consistently remained very close to zero. Therefore, it is believed that the terminal cost function has consistently acted as a Lyapunov function on the target set throughout the update process.

Remark 1 During certain epochs, none of the states along the trajectory fall within the target set, causing the Lyapunov inequality term to be 0. Note that unlike during training, the Lyapunov loss curve is not scaled by a factor of 100 when plotting.

To verify the effectiveness of the learned cost functions, we use the initial Q_0 and F_0 as well as the trained Q_{100} and F_{100} as the loss functions for the NMPC algorithm, respectively. We then randomly sample $M = 32$ initial states from the initial state range $[-\pi/6, \pi/6]$ and run the NMPC algorithm with a rolling optimization horizon of $N = 32$, simulating for 100 time steps. The control results obtained using the initial Q_0 and F_0 are shown in Fig. 4(a), while those obtained using the trained Q_{100} and F_{100} are shown in Fig. 4(b).

In Fig. 4, each small square represents the result of one simulation. When the pendulum is positioned in the upper part of the square and remains completely stable in the vertical position, we consider the result stable. It can be observed that the initial cost function only stabilizes the CartPole system at the upright position in a few cases, while in the vast majority of cases, the pendulum remains suspended at the lowest point. In contrast, the trained cost function successfully stabilizes the CartPole system at the upright position in all cases. This further verifies that our algorithm designed an NMPC asymptotically stable control law for the system.

In fact, after training for 30 epochs, the cost function already stabilized the CartPole system at the upright position. The training time of 30 epochs is approximately 1 h on an Intel Core i7-14500HX CPU, which can be a performance indicator of the efficiency of the algorithm.

IV. Conclusion

This work addresses the problem of designing

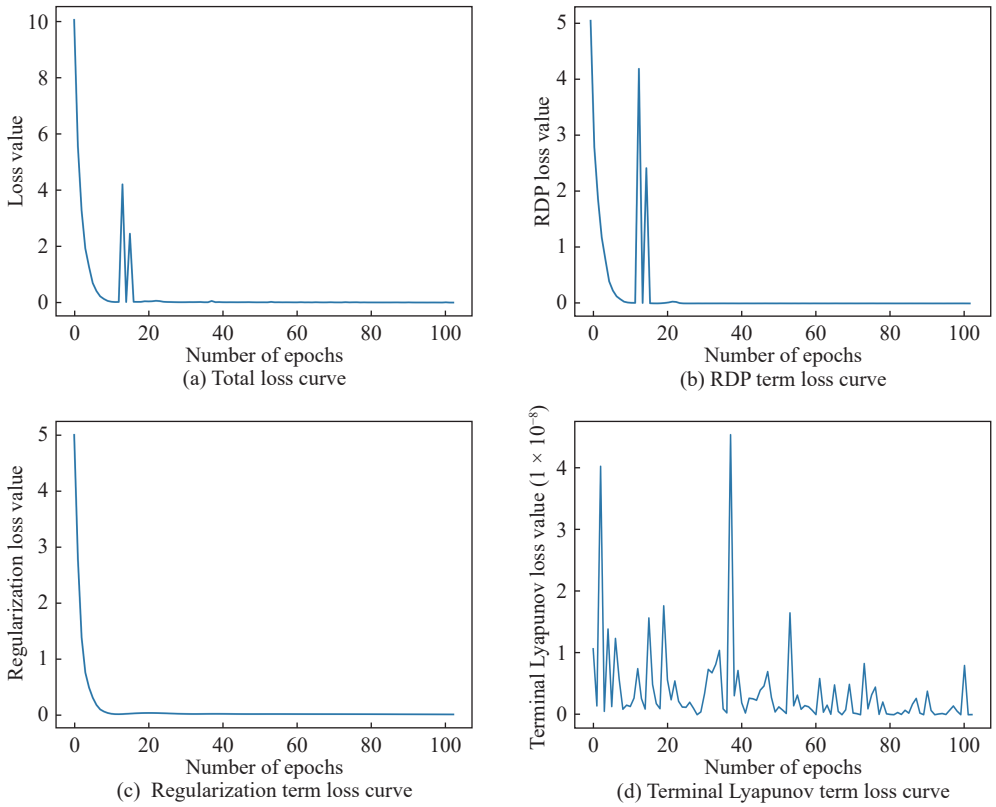
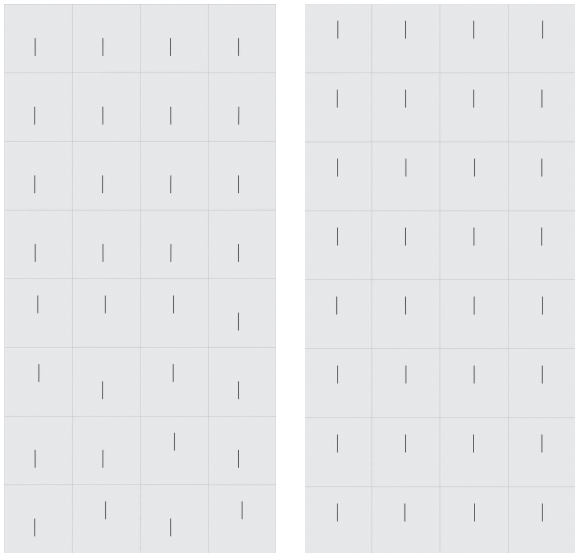


Figure 3 Control results of the CartPole system.



(a) Initial cost function (b) Trained cost function

Figure 4 Control results of the CartPole system using initial and trained cost functions.

asymptotically stable control laws for a system. By employing inverse optimal control and leveraging the RDP inequality, we developed a method to learn cost functions. The resulting NMPC feedback control law, based on these learned cost functions, guarantees asymptotic stability. In the future, we will verify its effectiveness in more realistic scenarios, such as robotic arm control and power system stability. Our future research will also focus on further exploiting the theoretical underpinnings of this method and enhancing its generality. Specifically, integrating a World Model can extend the algorithm to handle systems with unknown dynamics.

REFERENCES

[1] A. M. Lyapunov, The general problem of the stability of motion, PhD dissertation, Kharkov University, Ukraine, 1892.

[2] A. Alfarano, F. Charton, and A. Hayat, Global Lyapunov functions: A long-standing open problem in mathematics, with symbolic transformers, in *Proc. 38th Annual Conference on Neural Information Processing Systems*, Vancouver, Canada, 2024, 93643–93670.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, in *Proc. 31st International Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017, 6000–6010.

[4] D. Grande, A. Peruffo, E. Anderlini, and G. Salavasidis, Augmented neural Lyapunov control, *IEEE Access*, 2023, 11, 67979–67986.

[5] J. Liu, Y. Meng, M. Fitzsimmons, and R. Zhou, TOOL LyZNet: A lightweight Python tool for learning and verifying neural Lyapunov functions and regions of attraction, in *Proc. 27th ACM International Conference on Hybrid Systems: Computation and Control*, Hong Kong, China, 2024, 25.

[6] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms*, 2nd ed. Cham, Germany: Springer, 2017.

[7] L. S. Pontryagin, *Mathematical Theory of Optimal Processes*. London, UK: Routledge, 2018.

[8] R. Bellman, On the theory of dynamic programming, *Proc. Natl. Acad. Sci. USA*, 1952, 38(8), 716–719.

[9] R. Bellman, The theory of dynamic programming, *Bull. Am. Math. Soc.*, 1954, 60(6), 503–515.

[10] R. Bellman, *Dynamic Programming*. Mineola, NY, USA: Dover Publications, 2003.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[12] D. Liu, S. Xue, B. Zhao, B. Luo, and Q. Wei, Adaptive dynamic programming for control: A survey and recent advances, *IEEE Trans. Syst., Man, Cybern.: Syst.*, 2021, 51(1), 142–160.

[13] M. Krstic and Z.-H. Li, Inverse optimal design of input-to-state stabilizing nonlinear controllers, *IEEE Trans. Automat. Contr.*, 1998, 43(3), 336–350.

[14] A. Y. Ng and S. J. Russell, Algorithms for inverse reinforcement learning. in *Proc. 7th International Conference on Machine Learning*, Stanford, CA, USA, 2000, 663–670.

[15] R. E. Kalman, When is a linear control system optimal? *J. Basic Eng.* 1964, 86(1), 51–60.

[16] M. Kuderer, S. Gulati, and W. Burgard, Learning driving styles for autonomous vehicles from demonstration, in *Proc. 2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, 2015, 2641–2646.

[17] Y. Zhang, Z. Ai, J. Chen, T. You, C. Du, and L. Deng, Energy-saving optimization and control of autonomous electric vehicles with considering multiconstraints, *IEEE Trans. Cybern.*, 2022, 52(10), 10869–10881.

[18] J. Mainprice, R. Hayne, and D. Berenson, Goal set inverse optimal control and iterative replanning for predicting human reaching motions in shared workspaces, *IEEE Trans. Robot.*, 2016, 32(4), 897–908.

[19] T. Feng, J. Zhang, Y. Tong, and H. Zhang, Consensusability and global optimality of discrete-time linear multiagent systems, *IEEE Trans. Cybern.*, 2022, 52(8), 8227–8238.

[20] B. Lian, Y. Kartal, F. L. Lewis, D. G. Mikulski, G. R. Hudas, Y. Wan, and A. Davoudi, Anomaly detection and correction of optimizing autonomous systems with inverse reinforcement learning, *IEEE Trans. Cybern.*, 2023, 53(7), 4555–4566.

[21] A. Rantzer, On approximate dynamic programming in switching systems, in *Proc. 44th IEEE Conference on Decision and Control*, Seville, Spain, 2005, 1391–1396.

[22] B. Lincoln and A. Rantzer, Relaxing dynamic programming, *IEEE Trans. Automat. Contr.*, 2006, 51(8), 1249–1260.

[23] A. Rantzer, Relaxed dynamic programming in switching systems, *IEE Proc.-Control Theory Appl.*, 2006, 153(5), 567–574.

[24] B. Amos, I. D. J. Rodriguez, J. Sacks, B. Boots, and J. Z. Kolter, Differentiable MPC for end-to-end planning and control, in *Proc. 32nd International Conference on Neural Information Processing Systems*, Montréal, Canada, 2018, 8299–8310.

Shuhuai Tan received the BS degree from School of Advanced Manufacturing, Nanchang University, Nanchang, China, in 2025. He is currently pursuing the MS degree at State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China, and at School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. His research interests include inverse optimal control and reinforcement learning.

Kexi Yan received the BS degree in applied mathematics from Tsinghua University, Beijing, China, in 2022. He is currently pursuing the PhD degree at State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China, and at School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. His research interests include multi-agent reinforcement learning, mechanism design, and optimal control.

Jie Zhang received the BSc degree in information and computing science from Tsinghua University, Beijing, China, in 2005, and the PhD degree in technology of computer application from University of Chinese Academy of Sciences, Beijing, in 2015. He is currently an associate professor at State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include parallel control, mechanism design, optimal control, and multiagent reinforcement learning.

Yongting Chen is currently pursuing the BS degree in mechanical engineering at Tandon School of Engineering, New York University, USA. His research interests include complex system measurement and advanced control science.

Qinglai Wei received the BS degree in automation and the PhD degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2002 and 2009, respectively. From 2009 to 2011, he was a postdoctoral fellow at State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is currently a professor and the associate director at State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, China. He is the secretary of IEEE Computational Intelligence Society Beijing Chapter since 2015. He was the guest editor for several international journals. He was a recipient of the IEEE/CAA Journal of Automatica Sinica Best Paper Award, the IEEE System, Man, and Cybernetics Society Andrew P. Sage Best Transactions Paper Award, the IEEE Transactions on Neural Networks and Learning Systems Outstanding Paper Award, the Outstanding Paper Award of Acta Automatica Sinica, the IEEE 6th Data Driven Control and Learning Systems Conference Best Paper Award, and the Zhang Siying Outstanding Paper Award of Chinese Control and Decision Conference. He was a recipient of the Shuang-Chuang Talents in Jiangsu Province, China, the Young Researcher Award of Asia Pacific Neural Network Society, and the Young Scientst Award and Yang Jiachi Tech Award of Chinese Association of Automation. He is a board of governor member of the International Neural Network Society and a council member of Chinese Association of Automation. He has authored 4 books and published over 80 international journal papers. His research interests include adaptive dynamic programming, neural networks based control, optimal control, nonlinear systems, and their industrial applications.

Fei-Yue Wang received the PhD degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. He joined University of Arizona, Tucson, AZ, USA, in 1990, and became a professor and director at Robotics and Automation Laboratory and Program in Advanced Research for Complex Systems. In 1999, he founded Intelligent Control and Systems Engineering Center, Institute of Automation, Chinese Academy of Sciences, Beijing, China, under the support of the Outstanding Chinese Talents Program from the State Planning Council, and in 2002, he was appointed as the director of Key Laboratory of Complex Systems and Intelligence Science, CAS, China, and vice president of Institute of Automation, CAS, China, in 2006. In 2011, he became the state specially appointed expert and founding director at State Key Laboratory of Management and Control for Complex Systems, China. He has been the chief judge of Intelligent Vehicles Future Challenge since 2009 and director of China Intelligent Vehicles Proving Center, Changshu, China, since 2015. He is currently the director at Intel's International Collaborative Research Institute on Parallel Driving with CAS and Tsinghua University, Beijing, China. His research interests include methods and applications for parallel intelligence, social computing, and knowledge automation. He is a fellow of International Council on Systems Engineering (INCOSE), International Federation of Automatic Control (IFAC), American Society of Mechanical Engineers (ASME), and American Association for the Advancement of Science (AAAS). In 2007, he was the recipient of the National Prize in Natural Sciences of China, numerous best papers awards from IEEE Transactions, and became an Outstanding Scientist of Association for Computing Machinery (ACM) for his work in intelligent control and social computing. He was the recipient of the IEEE Intelligent Transportation Systems (ITS) Outstanding Application and Research Awards in 2009, 2011, and 2015, respectively, the IEEE Systems, Man, and Cybernetics (SMC) Norbert Wiener Award in 2014, and became the IFAC Pavel J. Nowacki Distinguished Lecturer in 2021. Since 1997, he has been the general or program chair of more than 30 IEEE, Institute for Operations Research and the Management Sciences (INFORMS), IFAC, ACM, and ASME conferences. He was the president of the IEEE ITS Society from 2005 to 2007, IEEE Council of Radio Frequency Identification (RFID) from 2019 to 2021, Chinese Association for Science and Technology, USA, in 2005, American Zhu Kezhen Education Foundation from 2007 to 2008, vice president of the ACM China Council from 2010 to 2011, vice president and the secretary general of Chinese Association of Automation from 2008 to 2018, and vice president of IEEE Systems, Man, and Cybernetics Society from 2019 to 2021. He was the founding editor-in-chief of *The International Journal of Intelligent Control and Systems* from 1995 to 2000, *IEEE ITS Magazine* from 2006 to 2007, *IEEE/CAA Journal of Automatica Sinica* from 2014 to 2017, *Journal of Command and Control* from 2015 to 2021, and *Chinese Journal of Intelligent Science and Technology* from 2019 to 2021. He was the EiC of the *IEEE Intelligent Systems* from 2009 to 2012, *IEEE Transactions on Intelligent Transportation Systems* from 2009 to 2016, and *IEEE Transactions on Computational Social Systems* from 2017 to 2020. He is currently the president of CAA's Supervision Council and new EiC of the *IEEE Transactions on Intelligent Vehicles*.