Proceedings of CONF-MLA 2025 Symposium: Intelligent Systems and Automation: AI Models, IoT, and Robotic Algorithms
DOI: 10.54254/2755-2721/2025.LD28120

Balance Control Algorithms for Robotic Arm

Chengyuan Gao

High School Affiliated to Nanjing Normal University, Jiangning Campus, Nanjing, China
gaochengyuan111@gmail.com

Abstract. Balance control in robotic arms refers to the ability to maintain both dynamic and static equilibrium during task execution, adapting to varying external and internal conditions, such as disturbances, payload changes, and environmental uncertainties. This capability is crucial for ensuring practical functionality in robotic implementations. This paper provides an in-depth analysis of the fundamental principles, key strengths, and inherent limitations of various balance control algorithms. The paper introduces classical feedback control techniques, including PID, LQR, and LQG, as well as robust and adaptive control methods. Additionally, the paper explores impedance and admittance approaches for interaction, along with Model Predictive Control (MPC) and learning-based strategies, such as reinforcement learning. The specific strengths of each algorithm are discussed, emphasising their suitability for diverse applications. The paper also compares these methods in terms of computational efficiency, implementation complexity, and performance under real-world constraints, revealing necessary trade-offs among them. Furthermore, the paper summarises the open challenges that hinder widespread adoption of these control techniques, indicating critical areas for future research and development efforts.

Keywords: Balance control, Robotic arm, Impedance control, Model predictive control, Robust control.

1. Introduction

Balance control in robotic arms is the ability to maintain dynamic and static equilibrium during tasks under variable conditions. It involves compensating for gravity, countering payload-shift disturbances, ensuring interaction compliance, and stabilising mobile manipulators. The balance control of the robot arm is a case of robot balance control. For whole robots, the centre of gravity must be calculated in real time and kept within the support base by adjusting motion states. In robotic arms, as the load is mainly at the end, the impact of end-load changes on the arm angle needs consideration. Historically, robotic arms used joint-level feedback controllers, and some standard control methods are applicable. This paper reviews existing balance control strategies, then introduces standard balance control algorithms in a robot arm. The survey aims to compare algorithms for selection and innovation. Finally, applications and open challenges are described.

2. The principle and steps

The general steps for implementing balance control for a robotic arm involve the following stages.

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.LD28120

Model the robotic arm. Create a mathematical model including mass inertia and forces to describe its physical characteristics. Balance control aims to control motor torques for stability and trajectories despite disturbances, payload variations, or inertial force. Select a control strategy. There are control methods such as classical control, adaptive control, robust control, and machine learning. Select one according to practical requirements. Control methods are crucial for achieving robotic arm balance control, which is discussed in Section 3. Deploy a feedback control system. Use sensors to measure the arm's state (e.g., joint positions, velocities, accelerations) and state observers if necessary. Then feed the robot arm's responses back to the controller to realise the feedback control loop.

3. Balance-control algorithms

Balance control methods can be categorised into three kinds. They are classical control methods, modern control methods and learning-based control. The following section will present the specifics in terms of distinctive principles, strengths, and limitations.

3.1. Classical control

3.1.1. PID control

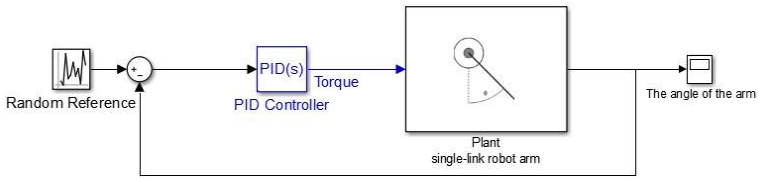

Figure 1. PID control for robot arm

As shown in Figure 1, PID control is a widely used feedback control method in industrial robotic arms. It measures the current error and uses three terms (P, I, D) to correct system deviation in real-time. The proportion term responds quickly to the error, the integral term eliminates steady-state error, and the differential term suppresses overshoot. Its strengths include a simple structure, relatively easy tuning, no need for an accurate model, and suitability for simple systems like single-link robot arms. However, it's not suitable for heavy payloads and interaction, such as in multi-DOF robot arms.

3.1.2. Linear Quadratic Regulator LQR/LQG

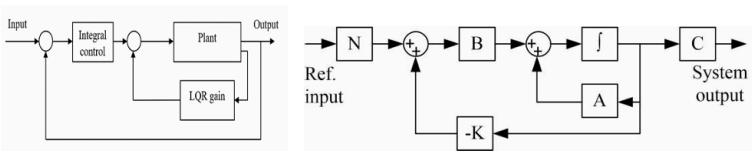

Figure 2. Block diagram of LQR control and the structure of the LQR-controlled system.

LQR is a popular method based on linear models. The technique aims to find the optimal control inputs to minimise a cost function that balances tracking a desired trajectory and minimising control effort. The Linear Quadratic Regulator (LQR) is the controller. The Kalman Filter is used to estimate

DOI: 10.54254/2755-2721/2025.LD28120

the system's states [1]. The whole system is modulated as a linear system with Gaussian noise. The control structure is given in Figure 2.

LQR is based on optimal control decisions for the system depicted in state space. Suppose the state equation of the system is:

$$\dot{x}(t) = AX(t) + BU(t) \tag{1}$$

$$Y(t) = CX(t) + DU(t) \tag{2}$$

The block diagram of LQR control is shown in Figure 3. The difference between LQR and LQG is that LQR control requires the measurement of all state variables, while in the linear quadratic Gaussian LQG controller, this measurement of all states is not necessary [2].

LQR is an optimal control method for linearised and lightweight robots. It provides efficient disturbance rejection and an optimal trade-off between stability and energy. Although it strictly relies on the linear model, it is unsuitable for real-world applications.

3.2. Robust and adaptive control

Robust and Adaptive Control. Robust control techniques, such as H-infinity, are developed to handle uncertainties in system dynamics. They aim to maintain performance despite these disturbances and uncertainties. They are employed in robotic arm manipulation with load variation and external force influence.

3.2.1. Robust control

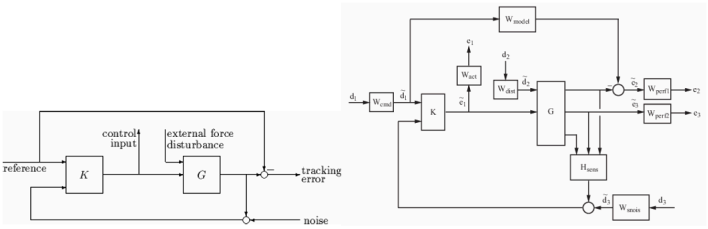

Figure 3. Typical Closed-Loop Performance and H∞ control Block Diagram.

The H∞ control principle involves designing controllers to minimise the worst-case system gain and enhance system robustness. Consider a tracking problem with disturbance rejection, measurement noise, and control input signal limitations (Fig. 3). Let K be the controller to design and G the system to control. The design objective is to design K so that tracking errors and the control input signal stay low under disturbances and noise. Thus, the closed-loop gain T is preferred:

$$\begin{bmatrix} \frac{tracking\ error}{control\ input} \end{bmatrix} = T \begin{bmatrix} reference \\ external\ force \\ noise \end{bmatrix} \tag{3}$$

Essentially, better performance means a small T. As the system is a MIMO system, if the performance objective is a matrix norm, it should be a weighted norm $\| WLTWR \|$. An acceptable performance can be characterised by the MIMO $\| \cdot \|_\infty$ (H∞) norm [3].

DOI: 10.54254/2755-2721/2025.LD28120

As the conditions of uncertainty and noise are fully considered, the method works robustly. However, in practical applications, the mathematical solution is challenging to obtain. It is complex in implementation.

3.2.2. Adaptive control

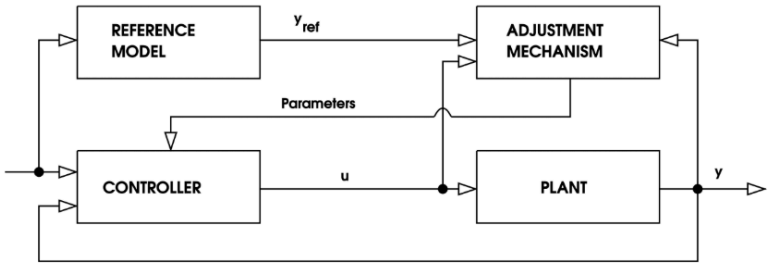

Figure 4. A block diagram of an adaptive control system [4].

An adaptive control system automatically compensates for variations in system dynamics by adjusting the characteristics of the controller to ensure optimal system performance. It takes into account the degradation of plant performance over time. The key elements involve system dynamics identification, decision-making, and modification. It operates by measuring to estimate the process dynamics and then modifying the controller characteristics accordingly. A typical design approach is to estimate parameters online. It recursively estimates the parameters from input–output data and updates the control settings at each sampling instant, as depicted in Figure 4. The adaptive control algorithm can be applied without precise modelling or in the presence of parameter variations; however, it exhibits a slow convergence rate and necessitates reliable state estimation.

3.3. Impedance and admittance control

These algorithms are often used for physical interactions between a robot and its environment. These control methods address how a robot reacts to forces and movements during contact. Impedance and admittance frameworks enable robotic arms to regulate motion-force relationships, facilitating compliant balance in human-robot interaction, as seen in the relationship between force F and movement v (F = kv and v = k×F). When the system operates, stability must be ensured to improve safety in interaction tasks.

3.3.1. Impedance control

Impedance control in robotics regulates the interaction force between a robot and its environment by modulating the robot's "impedance" (resistance to motion or deformation). It enables a robot to act like a spring and a damper system. This is especially useful when robots interact with humans or unstructured environments, such as in human-robot collaboration or surgical robotics[5].

Consider a simplified robot with one degree of freedom (DOF). The robot is defined as a mass at position Xm, and it is to be moved to a desired position Xd, which can be easily achieved by using a classical Proportional-Derivative (PD) controller with the following mathematical equation:

$$Fcontroller = k(Xd - Xm) - d\dot{X} \tag{4}$$

DOI: 10.54254/2755-2721/2025.LD28120

In the equation, the control action of the PD controller can be physically interpreted as a spring-damper combination where the proportional term represents the spring and the derivative term represents the damper. This is a simple impedance controller that provides feedback to the operator during movement (Figure 5).

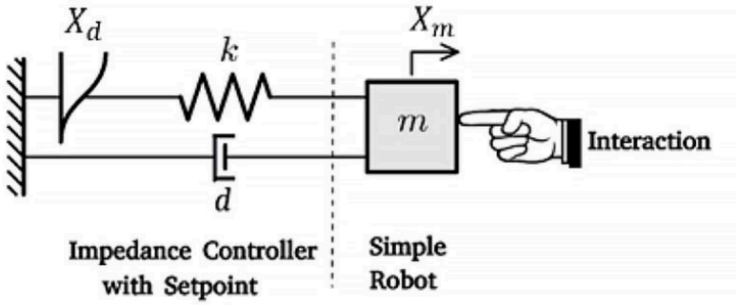

Figure 5. A simple one-DOF impedance-controlled system

3.3.2. Admittance control

Admittance control is the inverse of impedance control(Figure 6). It focuses on calculating motion responding to applied forces. It determines how the robot responds to external forces and moves accordingly. It allows the robot to be compliant and follow the applied forces, rather than resisting them. Admittance control is proper when a robot is interacting with a soft or yielding environment, such as a human hand. Impedance control can be implemented using an inner force loop and an outer position loop, or vice versa.

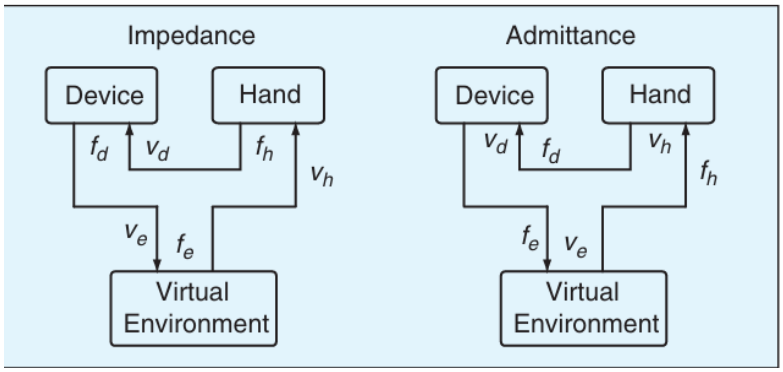

Figure 6. Impedance and Admittance control

3.4. Model Predictive Control (MPC)

MPC was initially utilised at Shell Oil and later appeared in vehicle path planning and control. When controlling a robot arm, it can offer predictive torque commands while taking into account system constraints, payload changes, and joint limits [6]. MPC stems from optimal control. At every time point k, an optimal control issue is resolved. Subsequently, all forecasted states and inputs within a prediction horizon are optimised, and only the first input is implemented in the system. This process is reiterated with the present system state as the starting state for predictions. A benefit of MPC is that it takes state and input constraints into account through an iterative solution.

DOI: 10.54254/2755-2721/2025.LD28120

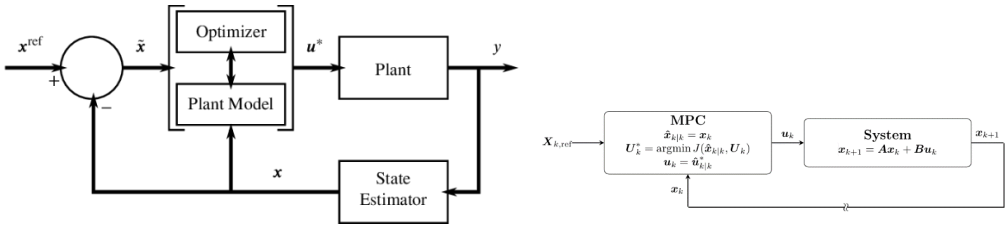

Figure 7. MPC control loop [7].

Compared with the LQR control block diagram (Fig.7), MPC is similar to LQR. When the cost function is quadratic, the plant is linear and unconstrained, and the horizon tends to infinity, MPC is equivalent to LQG control (or LQR control without an estimator). In practice, even with a finite horizon, MPC often inherits useful features of traditional optimal control, such as handling MIMO plants, dealing with time delays, and being robust against modelling errors. Specific terminal constraints can guarantee nominal stability. Other key features include explicitly handling constraints and using future reference and disturbance information when possible. MPC can handle multi-constraint scenarios (torque, joint limits, stability margins). However, like LQR, it has disadvantages in terms of computational demands and real-time feasibility.

3.5. Learning-based methods

Although robotics has made significant progress in mechanical design, perception, and control for object handling, robotic manipulation still lacks human dexterity in complex tasks. Data-driven learning methods, such as reinforcement learning (RL) and imitation learning (IL), offer adaptive strategies for balance in uncertain and complex manipulation. These learning-based methods collect human action data from sensors and train the AR model to generate a MIMO controller. The classification of learning methods is shown in Figure 8. RL, as a subfield of machine learning, is a promising algorithm for robotic arm manipulation[8-10].

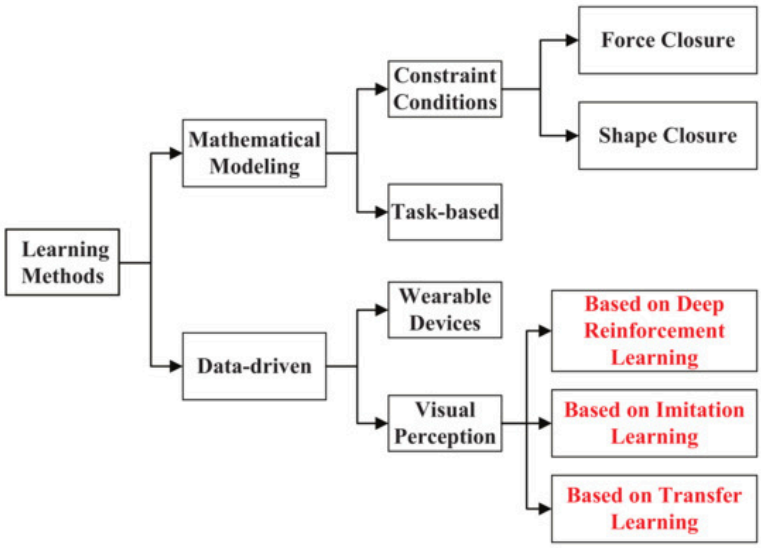

Figure 8. Taxonomy of learning methods

It explores how an agent ought to behave in an environment to maximise its long-term cumulative expected reward. It attempts various methods through trial-and-error for rewards. The

DOI: 10.54254/2755-2721/2025.LD28120

interaction between the agent and the environment is depicted in Figure 9. The agent performs actions and receives feedback in the form of bonuses or penalties from the environment. Subsequently, it utilises this feedback to modify behaviour and enhance performance over time.

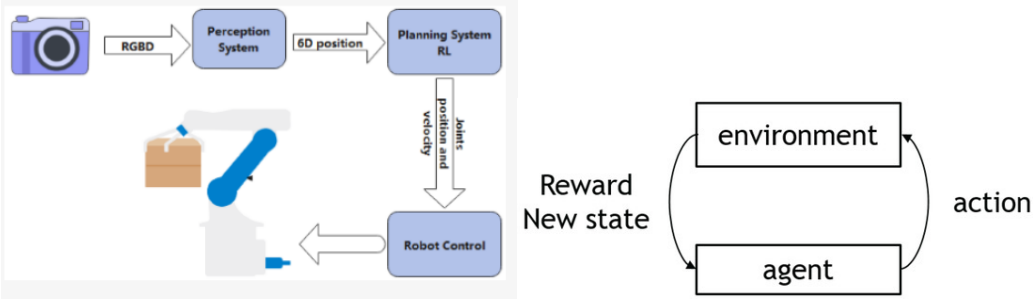

Figure 9. Classic robotic manipulation based on RL and a block diagram of a typical RL

At a time step, an autonomous agent observes the state, interacts with the environment through action, and updates its state to the next state. Upon reaching a new state, it receives a correlated reward. It aims to discover an optimal policy (the best action in any state). Unlike supervised and unsupervised learning, RL is sequential, dealing with state-action pairs one at a time. RL assesses actions according to the states they lead to, is goal-oriented, and focuses on learning action sequences to reach a goal or optimise an objective function.

4. Discussion

Table 1. Characteristics of the control algorithms

Algorithm Family	Strengths	Limitations	Typical Application
PID/LQR	Simple, efficient	Local validity	Industrial manipulators
Robust/Adaptive	Handles uncertainties	May require excitation, conservative	Payload-varying arms
Impedance/Admittance	Safe compliance, interaction	Trade-off stiffness vs. stability	Collaborative robots
MPC	Anticipatory, constraint handling	Computationally heavy	Precision, vibration suppression
Learning	Adaptivity, complex tasks	Data-intensive, safety	Human–robot collaboration

The paper summarises the algorithm characteristics and lists them in Table 1. Traditional linear control system-based algorithms are efficient. Modern control methods can handle complex constraints and uncertain conditions. Most are data-driven, with control parameters trained offline. Advanced strategies like MPC improve system responsiveness and energy efficiency. RL-based methods enhance adaptability but require more simulations. There's a trade-off between computing costs and precision.

Despite progress in robot arm balance control, challenges remain. First, real-time adaptation is challenging because current methods struggle to adapt to varying dynamic conditions in real-time while maintaining stability. Second, it's urgent to boost robustness against unmoulded dynamics and external disturbances without sacrificing performance. Finally, for robots to serve humans, the algorithm must ensure the safety and stability of human-computer interaction.

DOI: 10.54254/2755-2721/2025.LD28120

5. Conclusion

This paper shows the diverse range of balance control methods for robotic arms. Classical control methods offer simplicity but often fall short in dynamic environments. Adaptive and robust controls provide significant improvements but come with complexity and design challenges. Meanwhile, machine learning approaches present promising solutions. Still, face issues related to data requirements and safety. Addressing the difficulties outlined could lead to advancements in robot arm balance control, making robotic systems safer, more efficient, and more capable in real-world applications.

References

[1] Asghar, A., Iqbal, M., Khaliq, A., Rehman, S. U., Iqbal, J. (2022) Performance comparison of structured H∞ based looptune and LQR for a 4 DOF robotic manipulator. PLoS ONE, 17(4), e0266728.
[2] Putra, S., Sadiana, S., Siregar, M., Bachri, K. O. (2024) Analytical design and implementation of inverted pendulum control system using linear quadratic regulator. International Journal of Control and Automation, 12(3), 211–219.
[3] Erkol, H. O. (2018) Linear quadratic regulator design for position control of an inverted pendulum by grey wolf optimizer. International Journal of Advanced Computer Science and Applications, 9(4), 35–42.
[4] González, L., Aguilar, L. (2000) H∞ robust control design for an arm manipulator. Journal of Intelligent and Robotic Systems, 27(1), 21–30.
[5] Zhu, M., Gong, D., Zhao, Y., Chen, J., Qi, J., Song, S. (2025) Compliant force control for robots: A survey. Mathematics, 13(13), 2204.
[6] Lee, W. K., Lee, L. J. (1989) Computer control of extrusion. Elsevier, Amsterdam.
[7] Camacho, E. F., Bordons, C. (2007) Model predictive control. Springer, London.
[8] Krishnan, K. G. (2022) Using deep reinforcement learning for robot arm control. Journal of Artificial Intelligence and Capsule Networks, 4(3), 160–166.
[9] Han, D., Mulyana, B., Stankovic, V., Cheng, S. (2023) A survey on deep reinforcement learning algorithms for robotic manipulation. Sensors, 23(7), 3762.
[10] Hua, J., Zeng, L., Li, G., Ju, Z. (2021) Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning. Sensors, 21(4), 1278