Theory and Practice of Science and Technology
2025, VOL. 6, NO. 4, 191-197
DOI: 10.47297/taposatWSP2633-456934.20250604

State Estimation And Stabilization Control Of A Balanced Robot Based On Extended Kalman Filtering

Boying Zhang, Hujun Luo, Wei Dong

Liaoning University of Science and Technology, Anshan, Liaoning 114051

ABSTRACT
This paper investigates the state estimation and stability control method of balanced robot based on extended Kalman filter (EKF). Due to the nonlinear dynamics of balanced robots, state estimation and stability control have been a hot research topic. This paper firstly introduces the dynamic modeling of the balanced robot, and then elaborates the application of extended Kalman filter in state estimation, which realizes the accurate estimation of the robot state by combining the nonlinear system model and observation data. On this basis, a stabilization control strategy based on state estimation is designed, and the effectiveness of the method is verified by experiments. The results show that the state estimation based on EKF can improve the stability and control accuracy of the balancing robot, which provides theoretical support and technical reference for the practical application of balancing robot.

KEYWORDS
balancing robot; extended Kalman filter; state estimation; stabilizing control; nonlinear system

1 Introduction

At present, research on state estimation and stabilization control for balanced robots has made some progress. Traditional control methods, such as PID control, are widely used in many practical applications due to their simplicity and ease of implementation. However, PID control is difficult to cope with the nonlinear characteristics of balanced robots, and its control effect is often unsatisfactory for complex motion scenes and dynamic environments. In recent years, with the development of control theory, some modern control algorithms, such as fuzzy control, adaptive control and sliding mode control, have been introduced into the control of balancing robots. These algorithms can better deal with the nonlinear characteristics and improve the control accuracy and stability, but at the same time, they also face the problems of high computational complexity and difficult parameter adjustment [1].

In state estimation, Kalman filter and its extended form are widely used in robotic systems. Kalman filter is an optimal estimation method based on linear systems, but its application is limited for nonlinear systems. Extended Kalman filter (EKF) can solve the state estimation problem of nonlinear systems to some extent by linearizing the nonlinear system model.

However, the performance of EKF depends on the accuracy of linearization, and the divergence problem may occur when dealing with strongly nonlinear systems. In addition, nonlinear filtering methods based on particle filtering and other nonlinear filtering methods have been applied in some researches, but their computational complexity is high and their real-time performance is poor [2].

2 The main technological framework

Modeling the dynamics of balanced robots

Dynamics modeling of a balancing robot is the basis for state estimation and stability control. First, the mechanical structure of the robot needs to be described in detail. A balancing robot usually consists of multiple joints and connecting rods, whose degrees of freedom and joint parameters determine the robot's motion capability. For example, a typical two-wheeled balancing robot has two degrees of freedom, which are the rotation of the wheels and the pitching motion of the body.

By analyzing the geometric structure and kinematic relationships of the robot, its kinematic model can be established to describe the position and velocity relationships between the robot components [3].

Next, the dynamic equations of the balanced robot are derived. The dynamics equation describes the changing law of the robot's motion state, taking into account the influence of gravity, friction and other factors on the robot's motion. According to the Newton-Euler equations, the dynamics model of the robot can be established, and the motion state of the robot is expressed as a function of state variables. The state variables usually include the robot's position, velocity, attitude angle and its derivatives. For example, for a two-wheeled balancing robot, its dynamics equation can be expressed as follows.

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \omega \\ \dot{v} = \dfrac{F}{m} - \dfrac{b}{m}v \\ \dot{\omega} = \dfrac{T}{I} - \dfrac{d}{I}\omega \end{cases}$$

where x and y denote the horizontal position of the robot, θ denotes the pitch angle of the robot, v denotes the linear velocity of the robot, ω denotes the angular velocity of the robot, F denotes the driving force, T denotes the driving torque, m denotes the mass of the robot, I denotes the rotational inertia of the robot, and b and d denote the damping coefficients for linear and angular velocities, respectively [4-5].

A state-space model of the balanced robot is developed through mathematical modeling. The state-space model represents the robot's dynamics equations as a function of state variables and control inputs, which facilitates subsequent state estimation and control design. The state-space model can be expressed as follows.

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u}) \\ \mathbf{y} = \mathbf{h}(\mathbf{x}) \end{cases}$$

where x denotes the state vector, u denotes the control input vector, y denotes the observation vector, and f and h denote the state transfer function and observation function, respectively.

Extending the Kalman filter principle

Extended Kalman filtering (EKF) is a nonlinear filtering method based on linearization, which is widely used for state estimation of nonlinear systems. EKF achieves the optimal estimation of the system state by linearizing the nonlinear system model in the vicinity of the current estimation value, and by combining the system model and the observed data. The basic principle of EKF consists of the prediction step and the update step [6].

In the prediction step, the EKF predicts the state at the next moment using the state transfer function f based on the current state estimate and the control inputs. At the same time, the covariance matrix of the state estimate is updated according to the noise characteristics of the system. The prediction step can be expressed as follows.

$$\begin{cases} \hat{\mathbf{x}}_{k|k-1} = \mathbf{f}\left(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k\right) \\ \mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \end{cases}$$

Among them. $\hat{\mathbf{x}}_{k|k-1}$ denotes the predicted value of the state at moment k, and $\mathbf{P}_{k|k-1}$ denotes the covariance matrix of the state prediction, the $\mathbf{F}_k$ denotes the state transfer matrix, the $\mathbf{Q}_k$ denotes the process noise covariance matrix.

In the updating step, the EKF corrects the predicted states based on the observed data and the observation function h. The EKF also updates the covariance matrix of the state estimates according to the characteristics of the observation noise. At the same time, the covariance matrix of the state estimation is updated according to the characteristics of the observation noise. The update step can be expressed as follows.

$$\begin{cases} \mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k\right)^{-} \\ \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{h}\left(\hat{\mathbf{x}}_{k|k-1}\right)\right) \\ \mathbf{P}_{k|k} = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_{k|k-1} \end{cases}$$

where K_k denotes the Kalman gain, H_k denotes the observation matrix, and R_k denotes the observation noise covariance

3 Specific realization processes

3.1 Hardware platform construction

The hardware platform of the balancing robot is the basis for realizing its functions, which mainly includes mechanical structure, sensors, controllers and other parts. The following is the detailed construction process of the hardware platform.

Mechanical Structure The mechanical structure of a balancing robot needs to be designed for stability, flexibility and scalability. Lightweight and high-strength materials, such as aluminum alloy or carbon fiber, are usually used to construct the main frame of the robot. For a two-wheeled balancing robot, its mechanical structure mainly includes the body, wheels, motor mounts and balancing support structure. The body needs to have enough space to accommodate components such as controllers, sensors, and power supplies, and at the same time ensure a low center of gravity to improve the stability of the robot. The choice of wheels is also important, usually using high-torque, low-inertia motor-driven wheels to achieve fast response and precise control. The motor mount needs to hold the motor securely and

ensure that the motor shaft is coaxial with the wheel. The counterbalance support structure can use additional support rods or frames to prevent the robot from tipping over when starting or stopping [7-8].

Sensors Sensors are key components for balancing robots to obtain information about the environment and their own state. Commonly used sensors include gyroscopes, accelerometers, encoders and infrared sensors. Gyroscopes and accelerometers are usually integrated in one chip, such as MPU6050, which are used to measure the robot's attitude angle and acceleration. Encoders are mounted on the motor shafts to measure the rotational speed and positional information of the motors, thus realizing precise speed control and positional feedback. The infrared sensor can be used to detect black lines or obstacles on the ground to realize the tracing or obstacle avoidance function. The selection and installation of sensors need to be determined according to the specific application scenarios and control requirements of the robot to ensure that the measurement accuracy and response speed of the sensors can meet the requirements of the system [9].

Controller The controller is the core component of the balancing robot, responsible for processing sensor data, executing control algorithms and outputting control signals. Commonly used controllers include microcontrollers (e.g., STM32 series), digital signal processors (DSPs) and microcontroller units (MCUs), which are widely used in balancing robot control systems because of their high performance, low power consumption and rich peripheral interfaces. The controller needs to have enough processing power to run real-time state estimation and control algorithms, and at the same time, it should be able to communicate with various sensors and actuators. On the hardware platform, the controller is usually mounted in the center of the vehicle body to facilitate the connection and wiring with various components [10].

Hardware Platform Construction The construction of the hardware platform mainly includes the assembly of the robot, the installation and debugging of the sensors, the selection and installation of the controller and other steps. First of all, according to the design drawings or assembly guidelines, the mechanical structure of the robot is assembled to ensure that the connection between the components is firm and reliable. Then, various sensors are installed, such as gyroscope and accelerometer in the center of the body, encoder on the motor shaft, and electrical connection. Next, choose a suitable controller and install it on the vehicle body, and connect the controller with the sensors, motors and other components through the wires or data lines. Finally, debug the whole hardware platform, check whether the connection of each component is correct, whether the sensor can work properly, and whether the controller can read the sensor data and output control signals correctly.

Hardware Platform Physical Drawings and Connections Show the physical drawings or schematic diagrams of the hardware platform, and describe in detail the connections and functions of each component. For example, the structural frame of the vehicle body is made of aluminum alloy plate, the motor bracket is made of 3D printed parts, and the controller is mounted in the center of the vehicle body and connected to the sensors and motors through DuPont wires. The installation position and orientation of the sensors need to be optimized according to the measurement principle and the motion characteristics of the robot to ensure the accuracy and reliability of the measurement data. Through the physical drawings and schematic diagrams, the structure and connection of the hardware platform can be more intuitively demonstrated, which is easy for readers to understand and reproduce.

3.2 Software system design

The software architecture of balance robot control system is the key to realize its function, which mainly includes operating system, control algorithm, data processing and other modules. The following is the detailed design process of the software system.

Software architecture The software system adopts a layered architecture design, mainly including the bottom driver layer, intermediate data processing layer and upper control decision-making layer. The bottom driver layer is responsible for communicating with hardware devices, including sensor data reading and motor control signal output. The intermediate data processing layer is responsible for pre-processing, filtering and fusion of sensor data to obtain accurate robot state information. The upper control layer executes the control algorithm according to the state information, generates control commands and sends them to the bottom driver layer. This layered architecture design makes the software system modular and scalable, which is easy to develop and maintain.

Development Environment and Programming Language Selection The development environment and programming language of the software system need to be determined according to the hardware platform and development requirements of the controller. For STM32-based controllers, commonly used development environments include Keil uVision, STM32CubeIDE, etc., and the programming language is usually C or C++ , which has high efficiency and portability, and can make full use of hardware resources to realize high-performance control algorithms, while C++ provides object-oriented programming features, which is easy to organize and reuse the code. In the development process, you also need to use some auxiliary tools, such as debuggers, simulators, etc., in order to improve the development efficiency and code quality.

Functions and Interactions of Each Module The state estimation module is responsible for real-time estimation of robot

state information, such as attitude angle, velocity, position, etc., based on sensor data. The module reads the data from gyroscope, accelerometer and encoder, combined with the extended Kalman filter algorithm, to realize the accurate estimation of the robot state. Based on the state information provided by the state estimation module, the control decision module executes the control algorithm and generates control commands. For example, according to the attitude angle and velocity of the robot, the control signals of the motors are calculated to realize balance control and motion control. The data processing module is responsible for pre-processing, filtering and fusion of sensor data to improve the accuracy and reliability of the data. For example, the data from gyroscope and accelerometer are fused by Kalman filtering algorithm to eliminate noise and error and obtain accurate attitude angle information. The modules interact with each other through function calls and data structures, for example, the state estimation module stores the estimated state information in a global variable or data structure, and the control decision-making module reads the variable or data structure to obtain the state information and execute the control algorithm.

3.3 State Estimation and Control Algorithm Implementation

The state estimation and control algorithm is the core part of the balanced robot control system, and its performance directly affects the stability and control accuracy of the robot. The following is the detailed realization process.

Extended Kalman Filter Algorithm Implementation The Extended Kalman Filter Algorithm is a nonlinear filtering method based on linearization, which is suitable for dealing with the state estimation problem of nonlinear systems. The implementation of the algorithm mainly includes state initialization, prediction step and update step. In the state initialization stage, the initial values of the state vector and covariance matrix are set according to the initial state of the robot and the initial measurement values of the sensors. In the prediction step, the state and covariance matrix at the next moment are predicted based on the nonlinear model of the system using the state transfer function and control inputs. In the update step, the Kalman gain is calculated and the predicted state and covariance matrix are corrected based on the observation function and sensor measurements. The following pseudo-code implementation of the extended Kalman filtering algorithm is presented.

Reproduction
Initialization.
Initialize the state vector x0 and the covariance matrix P0
Initialize the process noise covariance matrix Q and the observation noise covariance matrix R, the
Prediction Steps.
For each time step k.
x_k|k-1 = f(x_k-1|k-1, u_k) // state prediction
P_k|k-1 = F_k * P_k-1|k-1 * F_k^T + Q // covariance prediction
Update Steps.
For each time step k.
K_k = P_k|k-1 * H_k^T * (H_k * P_k|k-1 * H_k^T + R)^-1 // Kalman gain calculation
y_k = z_k - h(x_k|k-1) // residual computation
x_k|k = x_k|k-1 + K_k * y_k // state update
P_k|k = (I - K_k * H_k) * P_k|k-1 // covariance update

Stabilization Control Algorithm Design The design of stabilization control algorithm according to the state estimation results is the key to realize the stable operation of the balanced robot. Commonly used control algorithms include PID control, adaptive control, fuzzy control, etc. PID control algorithm is simple and easy to realize, by adjusting the three parameters of proportional, integral and differential, it can realize the control of the robot's attitude and speed. Adaptive control algorithm can automatically adjust the control parameters according to the dynamic characteristics of the system to improve the adaptability and robustness of the system. The fuzzy control algorithm realizes the control of complex nonlinear system by dealing with the uncertainty of the system through fuzzy logic. In practical application, according to the specific characteristics of the robot and control requirements, the appropriate control algorithm can be selected or used in combination. The following is the pseudo-code implementation of the stabilization control algorithm based on PID control.

Reproduction
Initialization.
Initialize the parameters of the PID controller:Kp, Ki, Kd
Initialize the error variables: error, integral, derivative
Control Cycle.
For each time step k.
error = target - current_state // Calculate the error

integral += error * dt //

4 The main algorithms

4.1 Extended Kalman Filter Algorithm

Extended Kalman filter (EKF) is a nonlinear filtering method based on linearization, which is widely used for state estimation of nonlinear systems. By linearizing the nonlinear system model around the current estimated value, it combines the system model and the observed data to achieve the optimal estimation of the system state. The following are the specific steps of the extended Kalman filter algorithm.

State Prediction At each time step k, according to the nonlinear model of the system, the state transfer function f and the control input uk are used to predict the state at the next moment. The state prediction equation is $\hat{\mathbf{x}}_{k|k-1}=\mathbf{f}\left(\hat{\mathbf{x}}_{k-1|k-1},\mathbf{u}_k\right)$

Among them.$\hat{\mathbf{x}}_{k|k-1}$denotes the predicted value of the state at moment k, and$\hat{\mathbf{x}}_{k-1|k-1}$denotes the state estimation at moment k-1, and uk denotes the control input at moment k. The state estimation at moment k-1 is the state estimation at moment k-1, and uk denotes the control input at moment k.

Covariance Prediction While predicting the state, the covariance matrix Pk | k-1 of the state estimate needs to be updated to reflect the uncertainty of the predicted state. The covariance prediction formula is

$$P_{k|k-1}=F_kP_{k-1|k-1}F_k^T+Q_k$$

where F $_k$is the state transfer matrix, which represents the linearized relationship of the system state; Q $_k$is the process noise covariance matrix, which represents the uncertainty of the system model.

Kalman Gain Calculation The Kalman gain, Kk, is a key parameter used to weigh the reliability of the predicted state against that of the observed data.

The Kalman gain is calculated as follows.

$$\mathbf{K}_k=\mathbf{P}_{k|k-1}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T+\mathbf{R}_k\right)^{-1}$$

where H $_k$is the observation matrix, which represents the linearized relationship between the observed data and the state of the system; R $_k$is the observation noise covariance matrix, which represents the uncertainty of the observed data.

State update The predicted state is corrected based on the observed data yk and the observation function h. The state update formula is as follows The state update equation is

$$\hat{\mathbf{x}}_{k|k}=\hat{\mathbf{x}}_{k|k-1}+\mathbf{K}_k\left(\mathbf{y}_k-\mathbf{h}\left(\hat{\mathbf{x}}_{k|k-1}\right)\right)$$

where y $_k$is the observed data at time k, and$\mathbf{h}\left(\hat{\mathbf{x}}_{k|k-1}\right)$ is the observed value calculated from the predicted state, y $_k$-h$\left(\hat{\mathbf{x}}_{k|k-1}\right)$is the observation residual.

Covariance Update While updating the state, the covariance matrix of the state estimate, Pk|k, needs to be updated to reflect the uncertainty of the updated state. The covariance update formula is

$$\mathbf{P}_{k|k}=\left(\mathbf{I}-\mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_{k|k-1}$$

where I is the unit matrix.

Physical significance and calculation of algorithmic parameters

The state transfer matrix Fk: represents the linearized relationship of the system state, which is usually obtained by performing Jacobi matrix computation on the state transfer function f. The state transfer matrix Fk: represents the linearized relationship of the system state, which is usually obtained by performing Jacobi matrix computation on the state transfer function f .

The observation matrix Hk: represents the linearized relationship between the observed data and the state of the system, and is usually obtained by performing Jacobi matrix calculations on the observation function h. The observation matrix Hk: represents the linearized relationship between the observed data and the state of the system.

The process noise covariance matrix Qk: represents the uncertainty of the system model and is usually set according to the statistical properties of the system noise.

The observation noise covariance matrix Rk: represents the uncertainty of the observation data and is usually set according to the statistical properties of the sensor noise.

Mathematical formulas and diagrams are used to show the running process of the algorithm to help readers better understand the principle of the algorithm. For example, the state estimation accuracy of the extended Kalman filtering algorithm at different time steps can be demonstrated by plotting the comparison between the state estimation value and the real value.

4.2 Stabilization control algorithms

Stabilization control algorithms based on state estimation are the key to achieve stable operation of balanced robots. Commonly used control algorithms include PID control, adaptive control, fuzzy control and so on. The following are the design ideas and realization methods of these control algorithms:.

PID control PID control is a classical control algorithm that realizes the control of robot attitude and velocity by adjusting three parameters, namely, proportional (P), integral (I), and differential (D). The basic equation of PID control algorithm is.

$$(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt}$$

where u(t) is the control input, e(t) is the error signal, and K_p, K_i and K_d are the proportional, integral and differential parameters, respectively.

In balancing robots, PID control can calculate the control signals of the motors according to the robot's attitude angle and velocity errors to realize balance control and motion control. For example, for attitude angle control, the error signal can be defined as.

$$e(t) = \theta_{\text{target}} - \theta_{\text{current}}$$

where,θ_{target} is the target attitude angle andθ_{current} is the current attitude angle.

Adaptive control Adaptive control algorithm can automatically adjust the control parameters according to the dynamic characteristics of the system to improve the adaptability and robustness of the system. The basic idea of adaptive control is to estimate the parameters or characteristics of the system online, and adjust the control parameters dynamically to realize the optimal control.

For example, adaptive PID control can automatically adjust the proportional, integral and differential parameters according to the dynamic response of the system to adapt to different operating conditions. The basic formula of the adaptive control algorithm is.

$$K_p(t) = K_{p0} + \Delta K_p(t), K_i(t) = K_{i0} + \Delta K_i(t), K_d(t) = K_{d0} + \Delta K_d(t)$$

K_{p0}, K_{i0} and K_{d0} are the initial control parameters, and ΔKp(t), ΔKi(t) and ΔKd(t) are the parameter increments adjusted according to the dynamic characteristics of the system. ΔKp(t), ΔKi(t) and ΔKd(t) are the parameter increments adjusted according to the system dynamic characteristics.

Fuzzy control The fuzzy control algorithm realizes the control of complex nonlinear system by dealing with the uncertainty of the system through fuzzy logic. The basic idea of fuzzy control is to represent the input-output relationship of the system with fuzzy rules and realize the control through fuzzy reasoning.

For example, for attitude control of a balancing robot, fuzzy rules can be defined as follows.

If the attitude angle error is large and the rate of change is large, the control input is increased.

If the attitude angle error is small and the rate of change is small, the control input is reduced.

The basic equation of the fuzzy control algorithm is.

$$u(t) = \frac{\sum_{i=1}^{n} \mu_i(t)\, u_i}{\sum_{i=1}^{n} \mu_i(t)}$$

Where,$\mu_i(t)$ is the affiliation function of the fuzzy rule and u_i is the control input corresponding to the fuzzy rule.

Control Algorithm Performance Indicators The performance indicators of the control algorithm include stability, convergence speed and control accuracy. Stability refers to whether the system can recover to the equilibrium state after being disturbed; convergence speed refers to the time required for the system to arrive at the target state from the initial state; control accuracy refers to the size of the error of the system in the target state.

Through comparative analysis, the performance differences of different control algorithms can be demonstrated. For example, the PID control algorithm is simple and easy to realize, but has poor adaptability to the nonlinear system; the adaptive control algorithm can dynamically adjust the parameters, which is more adaptable, but the computational complexity is higher; the fuzzy control algorithm can deal with the uncertainty of the system, but it needs to design the appropriate

5 Concluding remarks

This paper investigates the state estimation and stability control method of balancing robot based on extended Kalman filter. The effectiveness of the method in different environments is verified through experiments, and the results show that the state estimation based on extended Kalman filter can significantly improve the stability and control accuracy of the balancing robot. Compared with the traditional control methods, this method shows better adaptability and robustness in complex environments.

Nevertheless, some problems and shortcomings were found during the experiment. For example, the state estimation accuracy decreases in complex environments such as uneven ground, which is mainly due to the influence of sensor noise and model error. In addition, the control algorithm still has some limitations in the high dynamic environment, which needs to be further optimized and improved.

About the Author

Boying Zhang (October 2003), Sex: Male, Origin (Zhangjiakou, Hebei Province), Ethnicity: Han, Education: Bachelor's Degree, Workplace: Liaoning University of Science and Technology, Location: Liaoning Province/Anshan City/Zip Code: 114000.

References

[1] On the design and implementation of control system for two-wheeled self-balancing robot. Liu Zhicheng. Science and Technology Perspectives,2017(09).

[2] Design and implementation of control system for two-wheeled self-balancing robot. Liu, J.; Xiao, J. B.; Wang, X.; Qian, W Electronic Technology and Software Engineering,2019(17).

[3] Design of a self-balancing robot control system based on electromagnetic guidance. Shen T; Wang Q; Chen Yankai; Qin Heting. Industrial Control Computer,2020(07).

[4] Design of control system for ball pickup robot for tennis. Chen Huan. Automation and Instrumentation, 2023(11).

[5] Design of control system for an underwater aquaculture robot. LIANG Guo-Xiang; QUAN Kai-Jie; LIANG Ivana-Ling; HUANG Li-Xia. China Science and Technology Information, 2024(03).

[6] Design of control system for field seeding robot. TAN Fusheng; LI Lianzheng; SUN Longlong; YUAN Guoqing; MA Yaoming; WANG Yumeng; CHEN Jiahe. Robotics Technology and Application,2023(06).

[7] Research on the control system of container automatic palletizing robot. Deer Yebo. Equipment Management and Maintenance,2024(02).

[8] Research on practical teaching mode of embedded robot control system. Pu Xi; Chen Yu; Ruan Tao. Development and Innovation of Electromechanical Products,2024(04).

[9] Design and research on control system of intelligent water surface trash cleaning robot. Wang, Yun-Liang; Zhao, Yi-Wen; Wu, Yan-Juan. Computer Application and Software,2024(09).

[10] Multi-sensory detection robot control system. Lin, Jia-Rui; Mo, Qi; Fu, Hantao; Luo, Shijun; Zhang, Jintong. Software,2024(08).