Advances in User Behavior Prediction for Smart Home Systems Based on IoT and Intelligent Control

Yifei Sun

Department of School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan, China

sunyifei12341@outlook.com

Abstract. Smart home systems, as a pivotal application scenario of the Internet of Things (IoT), have elevated user behavior prediction to a core functional pillar. This paper comprehensively reviews the current research status and prospective development trends of user behavior prediction in smart home systems. Firstly, it conducts an in-depth analysis of behavior recognition and prediction methodologies based on sensor data. Additionally, it explores long short-term memory (LSTM) network-based behavior prediction models, which excel at capturing temporal dependencies in user behavior sequences, enabling precise forecasts of subsequent actions such as adjusting lighting or temperature. Then, the paper discusses smart home control strategies, with a focus on the practical application of intelligent control methods. Fuzzy PID control is widely used in temperature regulation due to its strong adaptability to non-linear and uncertain environments, ensuring stable indoor temperature. Neural network PID control demonstrates superior performance in force-control scenarios such as smart door locks and curtain systems, enhancing operational accuracy and responsiveness. Furthermore, it delves into the application of machine learning methods, including decision trees, support vector machines, and deep learning, in optimizing user behavior analysis, device fault diagnosis, and energy management within smart home systems. Finally, the paper systematically analyzes the challenges faced by current smart home systems, such as data privacy and security risks, poor compatibility between heterogeneous devices, and the low accuracy of behavior prediction in complex scenarios. The advancement of smart home systems will further drive the innovation of IoT technology, delivering more intelligent, convenient, and personalized services to users.

Keywords: Smart home systems, User behavior prediction, Fuzzy PID control, Machine learning.

1. Introduction

Smart home systems, as an important application of Internet of Things technology in household environments, have received extensive attention from both academia and industry in recent years. With the rapid development of sensor, communication, and artificial intelligence technologies, smart home systems have shown great potential in enhancing the quality of life at home and improving energy efficiency. Currently, smart home systems have been preliminarily applied in areas such as lighting control, security monitoring, and environmental regulation, but there is still significant room for improvement in user behavior prediction and intelligent decision-making.

Traditional smart home systems only adopt rules or use simple sensors to model and analyze the environment data and implement intelligent control, which makes the system not dynamically and flexible adjusted according to user requirements and habits. With the development of machine learning methods, behavior modeling and prediction-oriented smart homes have become the hotspot. However, these methods based on sensor and the collected data still need users to label for predicting human behavior, but the tedious labeling is not acceptable at all, specially making it impossible for systems to learn through observing and perceiving things themselves automatically. So how to develop smart home systems that can identify and predict the user behaviors becomes a problem.

Smart home systems have become a hot spot application scenario of the Internet of Things in recent years. In terms of user behavior prediction, Zhang et al. presented a novel Automatic Annotation User Behavior Prediction (AAUBP) model, which combined a discontinuous solution sequence mining (DVSM) behavior recognition model and a behavior prediction model based on Long Short-Term Memory (LSTM) networks [1].

Yang et al. proposed neural PID controllers for robotic manipulation systems with uncertainties [2]. Lu and Lin studied the PID control design of planar nonlinear uncertain systems with actuator saturation, transformed the system into an optimization problem to be solved by an optimal controller and then solved it via linear matrix inequalities constraints, realizing the stable control of the systems at the set-points that are both possible and accessible [3]. In terms of temperature control systems, Wang et al. proposed a two-degree-of-freedom Smith internal model controller based on fuzzy control, the mathematical model of the temperature control system was first established, then based on this model they designed the two-degree-of-freedom Smith internal model controller of the temperature control system by design a new set-point tracking controller with good characteristics of disturbance rejection capability [4]. In terms of adaptive control in the field of smart homes, Paul et al. designed a fuzzy-tuned PID controller [5]. Samavat et al. proposed adaptive maximum power point tracking (MPPT) method based on metaheuristic-enhanced fuzzy method for optimizing standalone photovoltaic systems [6]. In terms of system security, researchers have developed a damage state identification method based on machine learning. Zheng et al. proposed a new method for damage state identification based on machine learning, used for vulnerability analysis of nuclear power plants considering structural uncertainties [7]. Kun Chen et al. proposed a constant force control strategy that combines fuzzy control and PID [8]. Wang et al. (2023) studied the fault-tolerant control problem of high-speed trains and proposed a sliding mode fault-tolerant control strategy [9]. Liu Peng research on the development and control of all-electric actuators in complex environments [10].

Therefore, this study designs a smart home system using a combination of machine learning and fuzzy PID control and using MATLAB simulation to analyze and verify the results [11].

2. Methods

2.1. Machine Learning

2.1.1 Data collection

To adapt to household habits, the machine learning system is divided into two modules: data collection and machine learning models.

The data collection is divided into passive data and active data.

Passive data: collected by sensors (e.g., temperature/humidity preferences, lighting usage times, appliance activation/deactivation patterns, and motion trajectories via infrared sensors).

Active data: user interactions (e.g., manual adjustments to air conditioners/lighting via mobile apps, feedback on comfort levels).

All data is processed locally on the MM32 chip (leveraging its local computing capability) to protect privacy, with only non-sensitive insights uploaded to the cloud for long-term model optimization.

2.1.2 Machine learning model

In order to satisfy the design requirements of the smart home system, machine learning must have three major functions: Preference Prediction, Activity Recognition, and Reinforcement Learning.

Preference Prediction: A Random Forest regression model is trained on previous records to predict temperature, humidity, and brightness preferences. For example, the system has learned that Member A likes the bedroom temperature at 22:00 to be 24℃, but Member B does not want the living room temperature in the evening (around 19:00) to be less than 26℃ [12].

The Random Forest regression model constructs multiple independent decision trees, taking the average of all trees' prediction results as the final output to reduce the risk of overfitting from a single decision tree and to improve prediction stability. The core steps include:

Sampling: Randomly select several samples from different subsets of the original dataset with replacement.

Decision Tree Construction: Build a new decision tree based on the subsets drawn in each of the above sampling steps (Random Forest can also randomly extract some features at each node, further increasing the diversity of the trees).

Ensemble Prediction: For each new sample to be predicted, obtain the prediction values from all of the above-built decision trees. Then, take the average as the final result for this sample.

The formula is: Let the input feature vector be X = (x_1, x_2, …, x_d) (such as time, room, etc.). The model includes K decision trees. The predicted value of the result for the *i*-th tree is $h_i(X)$. Thus, the final prediction result of the Random Forest is:

$$f(X)=\frac{1}{K}\sum\nolimits_{i=1}^{K} h_i(X) \tag{1}$$

Activity Recognition: A Convolutional Neural Network (CNN) is adopted to process motion and time data and accurately determine home scenes and scenarios, such as sleeping, cooking, or being away, to control devices. This enables smart home automation for different scenarios (for example, turning on the lights or dimming them during “watching TV” activities) [13].

Convolutional Neural Networks (CNNs), a type of feedforward network suitable for handling and processing spatial data as well as time series or sequence-structured data, are widely used for activity recognition. CNNs apply filter banks to extract local features of images through convolutional layers, compress the dimension after pooling layers, and finally output the activity types through fully connected layers.

The formula is as follows:

Convolution operation: Feature detectors whose parameters are automatically learned and calculated through algorithms such as forward propagation, regression, and gradient descent are used to detect and extract local features.

Let the input feature map be X∈ℝ^(H·W·C), where H represents height, W represents width, and C represents the number of channels (which can also indicate the length of time series data, feature dimensions, etc.).

Let the feature detectors be W∈ℝ^(h·w·C), where h and w represent the size. The output elements of the feature map Y are given as Y(i, j):

$$Y(i,j)=\sum_{\mathrm{h}-1}^{\mathrm{m}=0}\sum_{\mathrm{w}-1}^{\mathrm{n}=0}\sum_{\mathrm{C}-1}^{\mathrm{c}=0} X(i+m,j+n,c)\cdot W(m,n,c)+b \tag{2}$$

Where (i, j) denotes the position of the output feature map; *m, n,* and *c* represent the spatial indices and the channel index of the convolution kernel, respectively; *b* is the bias term.

Pooling operation: The pooling layer reduces the dimensions of feature maps while preserving key information. Taking the commonly used max pooling as an example, the elements of the output feature map *Y* are:

$$Y(i,j)=\max\{X(i\cdots+m,j\cdots+n)|\ m\in[0,k-1],n\in[0,k-1]\} \tag{3}$$

Where *k* is the pooling window size; *s* is the stride (sliding interval).

Final classification output: After several convolution and pooling layers, the feature map is flattened into a vector and fed into a fully connected layer, which produces the probability distribution overactive categories through the SoftMax function.

The RL agent learns the optimal strategy for scheduling energy use by balancing cost and comfort. For instance, it may learn that the family rarely uses the living room air conditioner after 23:00 and automatically switch it to standby mode unless manually activated.

2.2. PID

PID control achieves accurate system regulation by combining three components: proportional (P), integral (I), and derivative (D) [14].

Let:

- $r(t)$ = set value (target value),
- $y(t)$ = actual system output,
- $e(t) = r(t) - y(t)$ = error (deviation),
- $u(t)$ = control signal output by the controller.

The continuous-time PID control formula is:

$$u(t)=Kp\cdot e(t)+Ki\cdot \int_{t}^{0} e(\tau)d\tau+Kd\cdot \frac{\mathrm{de(t)}}{\mathrm{d(t)}} \tag{4}$$

Where Kp is proportional coefficient; Ki is integral coefficient; Kd is derivative coefficient.

2.3. Fuzzy Control

The basic structure of a fuzzy controller typically consists of four parts: fuzzification interface, rule base, fuzzy inference, and defuzzification interface.

The measured state of the controlled object is first transformed into a fuzzy quantity (in natural language terms) through fuzzification. Using human-like control rules, fuzzy inference generates the fuzzy output of the control variable, which is then defuzzied into a precise control value.

2.4. ML + Fuzzy PID Control

To meet smart home system requirements, machine learning (ML) is integrated with fuzzy PID control [15]. The system forms a closed-loop feedback cycle of “ML Prediction + Fuzzy PID Dynamic Control”, comprising five phases: data input, prediction optimization, parameter adjustment, control output and feedback iteration.

2.4.1 Data Input Layer

Inputs include:

Real-time environmental data (current temperature *T_current*, humidity *H_current*, time *t*, season, family member status, etc.),

Target set value (user-desired temperature),

Historical data.

The key inputs are:

Deviation: $e = T\_set - T\_current$ (if $e > 0$, heating is needed; if $e < 0$, cooling is needed).

Rate of change of deviation: *e_c* (indicates the speed of temperature change, e.g., a high positive value when warming rapidly).

2.4.2 ML Prediction and Optimization Layer

The ML module predicts future needs and dynamically adjusts control baselines, providing predictive parameters for fuzzy PID.

2.4.3 Fuzzy PID Controller: Dynamic Parameter Adjustment

The fuzzy PID controller adjusts *Kp*, *Ki*, and *Kd* in real time:

Fuzzification: Convert e and e_c into fuzzy linguistic variables (e.g., “large deviation,” “small deviation,” “fast change,” “slow change”), mapped to fuzzy sets (e.g., {negative large, …, zero, …, positive large}).

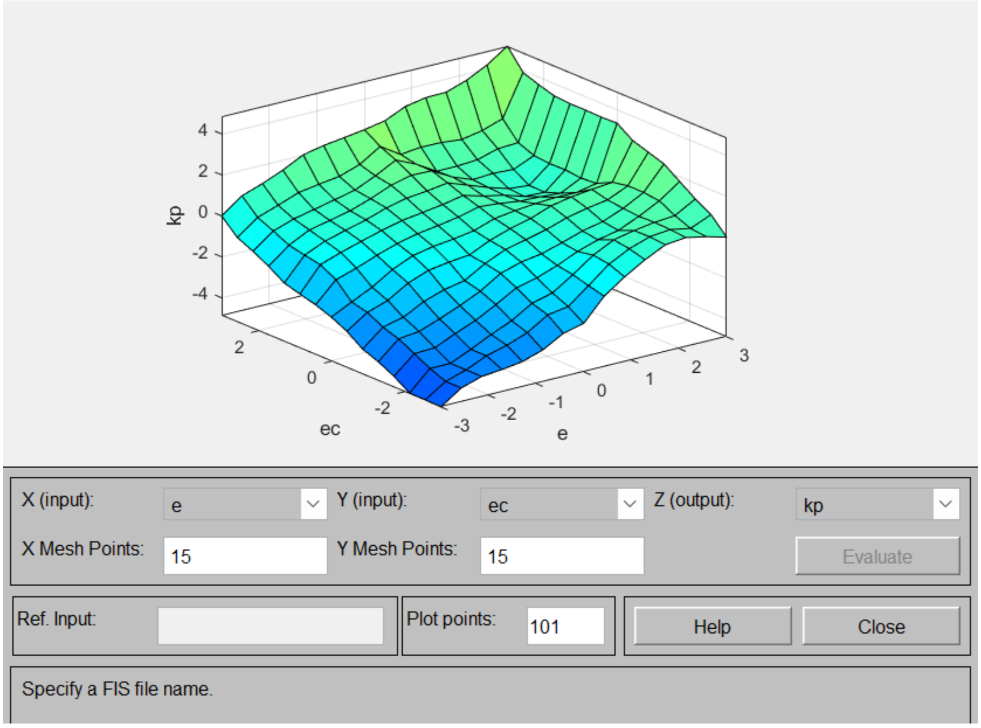
Fuzzy inference: Based on base parameters (Kp_base, Ki_base, Kd_base) from ML and fuzzy rules (e.g., when e is positive large and e_c is positive small, Kp should be large to quickly eliminate deviation), calculate corrections:

$$\begin{aligned} \Delta \mathrm{Kp} &= \mathrm{frule(e, ec)} \\ \Delta \mathrm{Ki} &= \mathrm{frule(e, ec)} \\ \Delta \mathrm{Kd} &= \mathrm{frule(e, ec)} \end{aligned} \tag{5}$$

Defuzzification: Convert fuzzy corrections into precise values to obtain the final control parameters:

$$\begin{gathered}Kp = Kp\_base + \Delta Kp \\ Ki = Ki\_base + \Delta Ki \\ Kd = Kd\_base + \Delta Kd\end{gathered} \tag{6}$$

The relevant information about fuzzy rules is shown in Fig. 1.



X (input): e

Y (input): ec

Z (output): kp

X Mesh Points: 15

Y Mesh Points: 15

Ref. Input:

Plot points: 101

Specify a FIS file name.

Figure 1. Three-dimensional surface graphics based on fuzzy rules

2.4.4 Control Output Layer

The device executes the PID controller to calculate the control quantity based on the adjusted parameters (e.g., air conditioner compressor power, heater output). The formula is:

$$u(t)=Kp *e(t)+Ki*\int_0^t e(\tau)d\tau+kd*\frac{de(t)}{dt} \tag{7}$$

Here, the control variable u(t) acts on the controlled object (e.g., household appliances), driving the actual temperature to approach the target value T_{set}.

2.4.5 Feedback Iteration

Sensors continuously collect real-time control effectiveness data (e.g., actual temperature $T_{current}$, equipment energy consumption) along with user feedback (e.g., “comfort score”). These serve as reward signals for the RL agent, where higher rewards correspond to lower energy consumption and higher comfort levels. The reward signals are then used to update the ML model (improving prediction accuracy) and the fuzzy PID rule base (enhancing the rationality of parameter adjustment), forming a closed-loop iterative process.

2.4.6 Transfer Function

The transfer function of this system must be analyzed by considering both the controlled object (room temperature system) and the controller (ML + fuzzy PID). Because fuzzy PID parameters change dynamically and ML introduces time-varying factors, the system is essentially nonlinear and time-varying. However, under specific operating conditions, it can be approximated as a linear model.

Room temperature regulation can be modeled as a second-order system, with its transfer function expressed as:

$$G(s) = \frac{K}{as^2+bs+c} \tag{8}$$

For a traditional PID controller, the transfer function is:

$$G_{PID}(s) = Kp + Ki/s + Kd*s \tag{9}$$

Due to parameters Kp，Ki，Kd dynamically changing with deviation e and ec, the transfer function becomes time-varying:

$$G_{fuzzy\_PID}(s, e, ec) = Kp(e, ec) + Ki(e, ec)/s + Kd(e, ec)*s \tag{10}$$

After integrating ML, the parameters additionally depend on prediction results (e.g., season, time, usage patterns). Thus, the transfer function can be expressed as:

$$G_{ML\_fuzzy\_PID}(s, e, ec, x) = Kp(e, ec, x) + Ki(e, ec, x)/s + Kd(e, ec, x)*s \tag{11}$$

Where x is the prediction vector from the ML module (e.g., representing different usage habits of various household users in different scenarios).

The transfer function of the overall closed-loop control system, with input as target temperature T_{set} and output as actual temperature $T_{current}$ is given by:

$$\Phi(s)=\frac{GML\_fuzzy\_PID(s,e,ec,x\hat{})\cdot G(s)}{1+GML\_fuzzy\_PID(s,e,ec,x\hat{})\cdot G(s)} \tag{12}$$

3. Simulation Result Analysis

To compare the performance of fuzzy PID with traditional PID control, MATLAB simulation experiments were conducted for both approaches. The control circuit diagram is shown in Fig. 2.

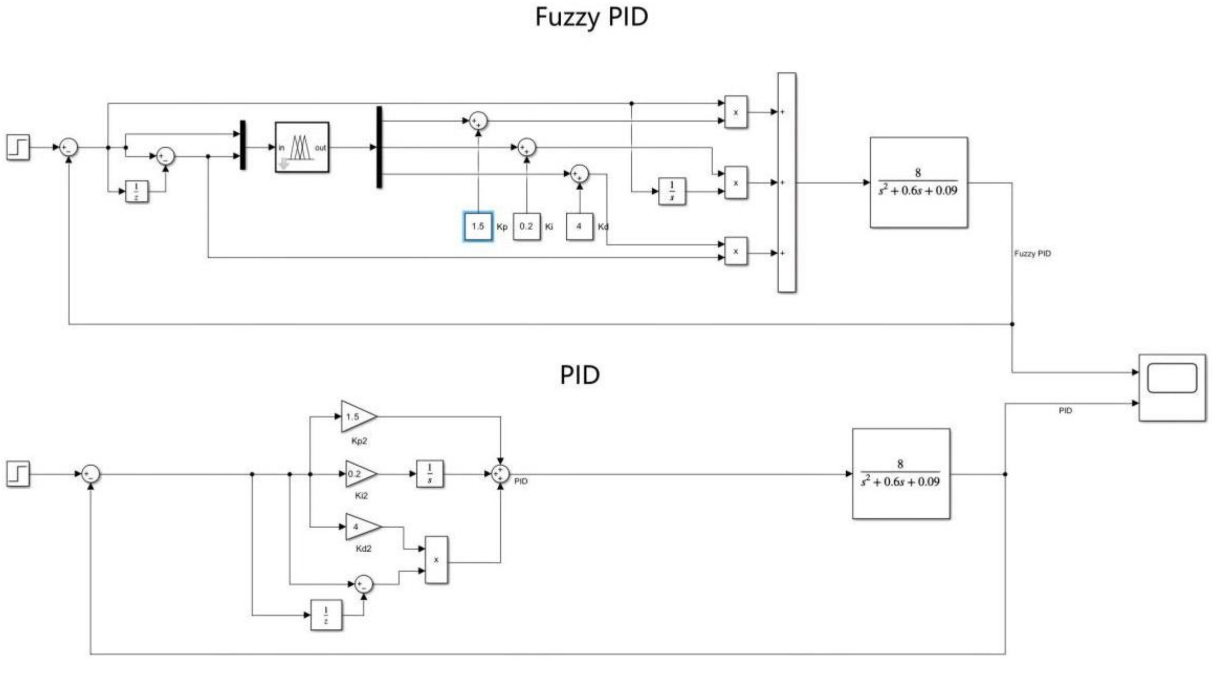


Figure 2. The control circuit

3.1. The input is a step function

When the input is a step function defined as:

$$e(t) = 100t \quad t\geq 0s \tag{13}$$

Where t denotes time, the simulation results are illustrated in Fig. 3.

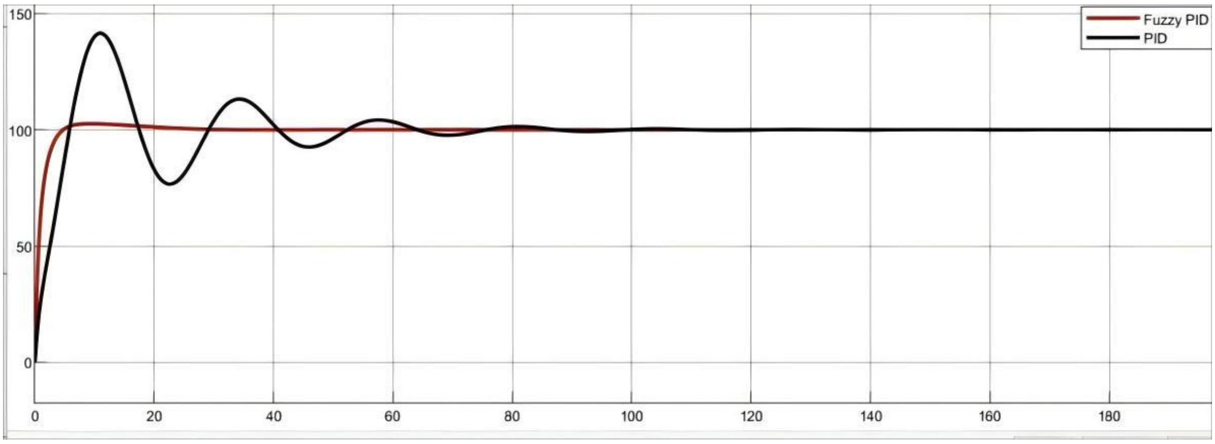


Figure 3. The result when the input is a step function

When a step input is applied, it can be observed that the overshoot of the fuzzy PID controller is significantly optimized. The traditional PID curve (black) exhibits a large initial peak with substantial overshoot at the beginning of the step response, whereas the fuzzy PID curve (red) shows a nearly smooth and gradual climb. As shown in the figure, the fuzzy PID greatly reduces overshoot. This improvement occurs because the fuzzy PID dynamically and automatically adjusts its parameters using fuzzy rules, thereby avoiding the excessive overshoot caused by fixed PID parameter values. As a result, the fuzzy PID is more suitable for precise control in smart home applications, such as air conditioning systems.

For temperature or humidity regulation systems, the fuzzy PID offers better performance by reducing equipment wear and meeting energy-saving requirements. Additionally, as seen in Fig. 3, the fuzzy PID achieves a faster steady-state response. It converges to the steady-state value much more quickly than the traditional PID. In control systems where the objective is to reach a stable operating point where sensors and actuators function efficiently, the traditional PID would require a longer oscillatory period before settling. This implies that in practical applications, devices such as air conditioners and lights can reach a stable state faster with fuzzy PID, allowing users to experience immediate comfort, such as a desired room temperature, sooner. In contrast, the traditional PID may take several minutes to stabilize, consuming more energy during the adjustment process. Thus, the fuzzy PID contributes effectively to energy-saving goals.

3.2. The input is a trigonometric function

When the input is a step function defined as:

$$e(t) = 100\sin(t) \quad t \geq 0s \tag{14}$$

Where t denotes time, the simulation results are illustrated in Fig. 4.

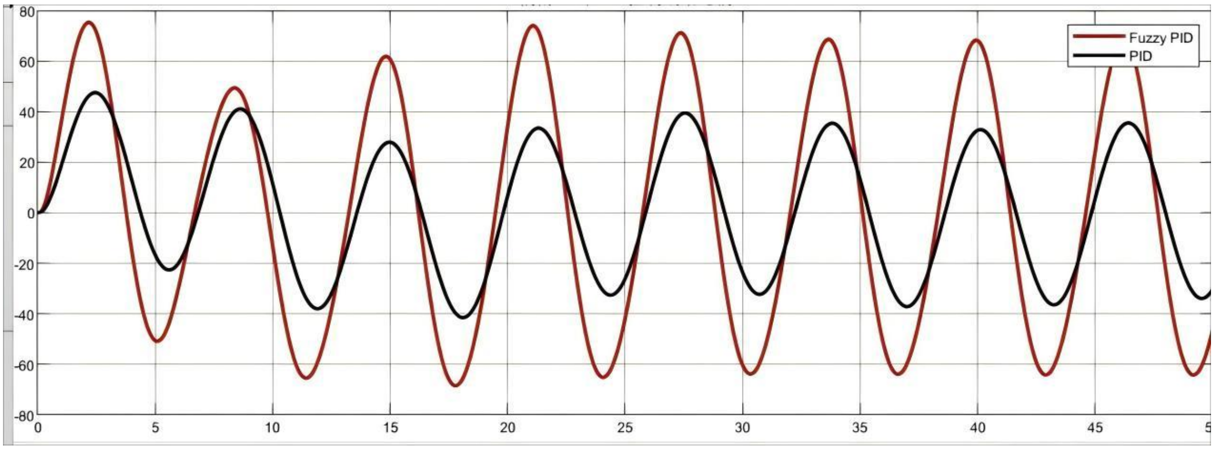


Figure 4. The result when the input is a trigonometric function

In order to further evaluate the superiority of fuzzy PID under different input types, the original step input was replaced with a triangular input. The simulation results are shown in the figure below. It can be observed that the fuzzy PID controller exhibits better tracking performance and accuracy, along with a smaller steady-state error. When tracking periodically varying triangular inputs, the red fuzzy PID curve follows the ideal trajectory more closely and maintains a lower steady-state error, whereas the traditional PID (black curve) shows significant deviation.

In real-world home environments, user demands often vary over time or with changing conditions. For example, temperature requirements may differ throughout the day—such as when family members wake up or prepare to sleep. Traditional PID controllers may struggle to adapt quickly due to error accumulation, resulting in slow response and inefficient operation, which compromises comfort and increases energy waste. In contrast, fuzzy PID can dynamically adjust to varying demands, enabling more stable and efficient device operation. It also responds more quickly; for instance, under one full cycle of a triangular input, the fuzzy PID reaches steady state within about 20 seconds, while the traditional PID requires approximately 30 seconds and exhibits waveform distortion.

In smart home applications where human needs frequently change—such as for lighting, air conditioning, curtains, and water systems—users often require automated or scheduled adjustments. Fuzzy PID can anticipate upcoming demands and adjust devices in advance, allowing the controlled variable to follow setpoints more accurately and rapidly.

In summary, compared to traditional PID, fuzzy PID not only improves overshoot and response speed under step inputs but also enhances tracking capability under triangular inputs. It achieves higher tracking accuracy and lower steady-state error across different types of input. With its faster and more precise response, fuzzy PID better supports the dynamic, comfort-driven, and energy-efficient requirements of smart homes, making it a superior upgrade over traditional PID control.

3.3. Limitations and Future Outlook

However, the current machine learning models still have certain limitations that need to be addressed. The three ML models—preference prediction, behavior recognition, and anomaly detection—may not perform effectively without sufficient high-quality training data. For example, when new users move in or visitors stay, the system lacks historical data for learning, resulting in a "data accumulation period" of about 2–4 weeks. During this time, users may need to frequently adjust settings such as room temperature manually since the system cannot yet infer their preferences, potentially reducing user satisfaction.

To mitigate this, an active feedback mechanism could be introduced. For instance, users could be prompted to tag specific situations, such as "guest mode” when someone stays overnight, enabling the system to quickly learn and adapt to new preferences. This would help accelerate the learning process for individual behaviors and reduce the need for manual adjustments.

Additionally, the parameter selection in the simulation does not fully reflect real-world conditions. For example, the amplitude of the input function was set to 100 in this experiment, which may not match practical scenarios. Moreover, using only step and trigonometric functions as inputs may limit the generalizability of the results. Therefore, future studies should include more diverse input functions and conduct comparative analyses under a wider range of conditions.

4. Conclusion

This study focuses on overcoming the limitations of conventional smart home systems, which often rely on fixed rules or basic feedback control, resulting in energy inefficiency, limited security, and insufficient personalization. A machine learning-enhanced smart home automation system is proposed to improve energy savings, security through proactive anomaly detection, and personalized user experiences.

The core methodology integrates ML (including data collection—both passive from sensors and active from user interactions—and models such as random forest regression for preference prediction, CNN for activity recognition, and reinforcement learning for device scheduling), PID control, and fuzzy logic control, combined into an ML-based fuzzy PID control strategy.

The control system operates as a closed-loop feedback system consisting of five stages: Data Input (real-time environmental data, target values, and deviation metrics), ML Prediction (adjusting control benchmarks), Fuzzy PID Parameter Adjustment (dynamically optimizing Kp, Ki, and Kd via fuzzy rules), Control Output (device execution), and Feedback Iteration (updating models based on sensor data and user feedback).

Simulations in MATLAB with step and triangular inputs demonstrate that the fuzzy PID controller outperforms traditional PID in reducing overshoot, achieving faster steady-state response, improving tracking accuracy, and minimizing steady-state error.

Nevertheless, some limitations remain, including the dependency of ML models on extended data collection and the imperfect alignment of simulation parameters with real-world conditions. Future work will incorporate an active feedback mechanism for rare events and include more diverse input types in simulations to enhance robustness and practical applicability.

References

[1] Zhang Ningbo, Yan Yajie, Zhu Xuzhen, Wang Jing. A novel user behavior prediction model based on automatic annotated behavior recognition in smart home systems [J]. China Communications, 2022, 19 (9): 122-138.

[2] Yang Qinchen, Zhang Fukai, Wang Cong. Deterministic learning-based neural PID control for nonlinear robotic systems [J]. IEEE/CAA Journal of Automatica Sinica, 2024, 11 (5): 156-167.

[3] Lyu Xujun, Lin Zongli. PID control of planar nonlinear uncertain systems in the presence of actuator saturation [J]. IEEE/CAA Journal of Automatica Sinica, 2022, 9 (1): 94-102.

[4] Wang Zhigang, He Meng. Temperature control based on fuzzy logic two-degree-of-freedom Smith internal model [J]. Instrumentation, 2020, 37 (2): 3-10.

[5] Paul Satyam, Arunachalam Ajay, Khodadad Davood, Andreasson Henrik, Rubanenko Olena. Fuzzy tuned PID controller for envisioned agricultural manipulator [J]. International Journal of Automation and Computing, 2021, 18 (4): 68-80.

[6] Samavat Tina, Nazari Mostafa, Lin Fuhong, Yang Lei. Optimizing stand-alone PV systems: A metaheuristic-enhanced fuzzy approach for adaptive MPPT [J]. China Communications, 2025, 22 (1): 67-80.

[7] Zheng Zhi, Wang Yong, Pan Xiaolan, Ji Duofa. Machine learning based damage state identification: A novel perspective on fragility analysis for nuclear power plants considering structural uncertainties [J]. Earthquake Engineering and Engineering Vibration, 2025, 24 (1): 204-225.

[8] Chen Kun, Zhang Yawei, Zhang Zhen, Gui Zhiwei. Proportion integration differentiation (PID) control strategy of belt sander based on fuzzy algorithm [J]. Journal of Donghua University (English Edition), 2023, 40 (2): 54-61.

[9] Wang Qianling, Ma Caiqing, Lin Xue. Disturbance observer-based fuzzy fault-tolerant control for high-speed trains with multiple disturbances [J]. Chinese Physics B, 2023, 32 (10): 441-449.

[10] Liu Peng, Shen Dagang, Ba Yaoji, Cao Jinfeng, Liu Jie, Wang Lihui. Development and control strategy of subsea all-electric actuators [J]. Journal of Ocean University of China, 2022, 21 (5): 70-83.

[11] Yu Meng, Wang Shuya, Ren Xibiao, Xue Han, Yue Xuejun, Jin Chuanhong, Lin Shanggang, Lin Fang. Combining machine learning algorithms with traditional methods for resolving the atomic-scale dynamic structure of monolayer MoS_2 in high-resolution transmission electron microscopy [J]. Chinese Physics B, 2025, 34 (1): 166-174.

[12] Zhang Bofei, Yu Haipeng, Hu Zeyong, Yue Ping, Tang Zunye, Luo Hongyu, Wang Guantian, Cheng Shanling. A machine learning-based observational constraint correction method for seasonal precipitation prediction [J]. Advances in Atmospheric Sciences, 2025, 42 (1): 38-54.

[13] He Haijun, Meng Xi, Tang Jian, Qiao Junfei. Event-triggered-based self-organizing fuzzy neural network control for the municipal solid waste incineration process [J]. Science China (Technological Sciences), 2023, 66 (4): 200-213.

[14] Zhang Xin, Zhou Hao, Liu Jinguo, Ju Zhaojie, Leng Yuquan, Yang Chenguang. A practical PID variable stiffness control and its enhancement for compliant force-tracking interactions with unknown environments [J]. Science China (Technological Sciences), 2023, 66 (10): 130-144.

[15] Ding Min, Tao Zili, Hu Bo, Ye Meng, Ou Yingxiong, Yokoyama Ryuichi. A fuzzy control and neural network based rotor speed controller for maximum power point tracking in permanent magnet synchronous wind power generation system [J]. Global Energy Interconnection, 2023, 6 (5): 44-56.