Performance enhancement of the Pololu 3pi+ 32U4 robot: exploration of a PID approach to real-time Genetic Algorithm optimisation

Student Number:2381413 Student Number:2334672 Student Number:2386998

Abstract—**This study explores the issue of automated control strategies. In the process of automated production, due to the continuous change of environmental factors and hardware factors, the automatic control of fixed parameters often cannot achieve the accuracy and efficiency of production. Using the Pololu 3pi+ 32U4 as an example, we investigated how to maintain the stability and output power of the power system by adjusting the PID value of the motor output in real time. By using a genetic algorithm (GA) to update the PID compensation in real time, we compared three control strategies: the traditional bang-bang controller, fixed PID compensation, and the method of adjusting the PID using a GA. Experiments were conducted to assess the straight-line traveling ability and PWM compliance by measuring the encoder counts and wheel speeds of the trolley under different control strategies. It was found that the method using GA tuned PID was more effective in keeping the trolley traveling in a straight line and reaching the set PWM value compared to the other two control strategies. This finding is important for improving the stability and efficiency of power systems in autonomous production.**

I. Introduction

1. Background and Challenges: In the field of automated production, ensuring precise movement and orientation of robotic arms requires a sophisticated automatic control system. The key component of such a system is PID control. Proportional (P) control is the part where the controller's response is proportional to the current error, with higher proportional gains speeding up system response and quickly approaching the target value. Integral (I) control eliminates long-term cumulative errors, ensuring accurate achievement of the target value. Derivative (D) control predicts future errors and reduces overshoot, stabilizing the system. However, excessive P and I values can cause system oscillation, while a high D value makes the system overly sensitive to noise. Conversely, values that are too low for P, I, and D can compromise control precision. Therefore, finding optimal PID values is crucial for the best performance of an automatic control system. For instance, in the case of the 3pi robot, its power mechanism often fails to reach the set output values in practical applications.

2. Project Objective:

A. Introduction of GA This project aims to study and optimize the power mechanism of the 3pi+ robot. By deeply analyzing the output from the motor to the drive gears to the wheels, we aim to keep the output on the trolley wheels in a stable and rated state. For this purpose, we intend to use Genetic Algorithm (GA) to update the PID (Proportional-Integral-Derivative) control parameters of the cart motor outputs in real time to achieve a smoother and more reliable running performance.

Genetic Algorithm is a series of simulation evolutionary algorithms proposed by Holland et al and later summarized by DeJong, Goldberg, and others. The general flowchart of the Genetic Algorithm is shown in Fig 1. The Genetic Algorithm first encodes the problem, then calculates the fitness, then selects the parent and the mother by roulette, finally generates the children with high fitness by crossover and mutation, and finally generates the individuals with high fitness after many iterations, which is the satisfied solution or optimal solution of the problem [1].

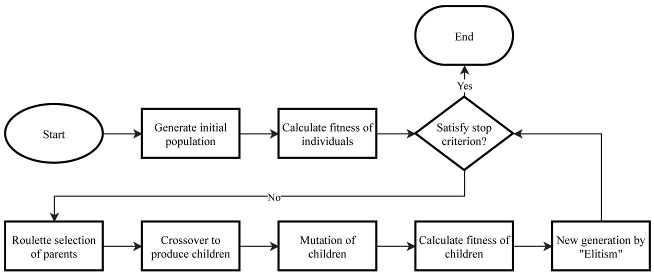

Fig. 1. Fig 1

B. Introduction of EEPROM EEPROM (Electrically Erasable Programmable Read-Only Memory) is a memory that can retain data even when power is off. EEPROM is unique in that it allows data to be electronically erased and rewritten at the byte level, providing flexibility in data management. [2]

Key features: **Electrically rewritable**: When connected to a power supply unit (PSU), the EEPROM module's memory can be completely or byte-by-byte erased. **Programmability:** Similar to PROM (Programmable ROM), EEPROM can be written to memory after being installed in the system, and by applying a voltage, the user can modify or reprogram the content stored on the enclosure. **Read-only memory:** EEPROM is designed for long-term storage of information that does not change dynamically, such as user files, firmware, and certain programs. Unlike RAM, data in EEPROM is retained even after the device is turned off. [3]

In the project, the ability of EEPROM to retain data even after a power outage was critical to storing the robot's critical system information, such as the Pololu 3pi+ 32U4 robot's wheel speed, encoder readings, and PID control parameters. This feature ensures that the robot can return to the same settings after a reboot, which is essential for debugging and optimizing algorithms. [4]

3. Research Methodology and Technical Background: In terms of controlling the output on the trolley wheels, the usual approach is to set a fixed PWM (Pulse Width Modulation) value through programming. However, in our previous experiments, we found that the trolley was unable to travel in a straight line even when the same PWM value was set for the left and right wheels, showing an imbalance in the output. Although the compensation method with a fixed PID value can ameliorate this problem to some extent, the applicability of the fixed PID value is limited under different lighting conditions, battery power, driving conditions and mechanical wear conditions.

Therefore, we propose the use of a genetic algorithm to update the PID value in real time to keep it always in the optimal state. In this way, not only can we ensure that the output on the wheel reaches the rated value, but also reduce the influence of hardware conditions and external environment changes that the cart is subjected to during the driving process.

4. Hypothesis Statement: Core hypothesis: We hypothesize that using a genetic algorithm to adjust the PID values in real time will be more effective in achieving a straight line traveling of the trolley and reaching the rated PWM values compared to traditional methods such as bang-bang controllers or fixed PID compensation. We expect that this approach will show significant advantages in overcoming environmental variations and hardware wear and tear.

II. Experiment Methodology

1. Applications of Genetic Algorithms: *Genetic Algorithm is a search algorithm that mimics the process of natural selection for solving optimization problems. In this project, we use Genetic Algorithm (GA) to optimize the PID controller parameters (Proportional, Integral, Derivative) of a 3pi trolley to improve the trolley's operational performance. The implementation process is described below:*

A. Initialisation: In the initialization phase of the genetic algorithm, the population size is set to 30 individuals. Each individual in this population represents a set of PID controller parameters (Proportional, Integral, Derivative). These parameters are randomly generated to form the initial population, providing a diverse starting point for the genetic algorithm.

B. Evaluation: The performance of each set of PID parameters is checked by controlling the trolley. Where the fitness function is used to evaluate the performance of each PID parameter combination (individual). The key performance indicators of the fitness function are cumulative wheel speed error and peak overshoot. [5]

$$f_{i,k}(t) = \gamma_1 \cdot Mp + \gamma_2 \cdot \sum_{n=1}^{10} e^2(n) \tag{1}$$

*Cumulative error(e^2):*The sum of error terms (e.g. speed error) accumulated over time. In PID control, this is usually associated with the integral term (I term), which accumulates deviations and adjusts them over time. The accumulated error is useful in assessing the performance of a control system over time, particularly in terms of long-term stability and accuracy.

Peak overshoot (Mp): Peak overshoot is the maximum extent to which the system response exceeds its final stabilized value. In PID control, it can be expressed as the maximum percentage or absolute value by which the output of a control system (e.g., speed) exceeds a target or setpoint value. Peak overshoot is an important measure of system stability and degree of overshoot. These criteria are quantitatively evaluated by assigning a performance score to each set of parameters that reflects their control effectiveness. Our goal is to minimize wheel speed error and overshoot.

C. Selection (Roulette Selection): In roulette selection, the cumulative probability is calculated with the selection probabilities of all individuals. The cumulative probability of the no. k individual can be calculated by formula (2).

$$p_x(a_x) = \sum_{j=1}^{k} p_x(a_j) \tag{2}$$

Then random number e, which is between 0 and 1, is generated and compare with px(ak) to determine selection individual. If $ak - 1 < e < ak$, the no. k individual is selected. Repeated n rounds are conducted to generate n individuals of offspring generation [6].

A roulette selection method is used to select parameter combinations from the current population to go to the next generation. In roulette selection, the probability of each group of parameters being selected is directly proportional to its performance score, i.e., better performing parameter combinations have a greater chance of being selected.

D. Crossover (uniform crossover): A uniform crossover method is used to generate new PID parameter combinations. In the uniform crossover, two parent parameter combinations are exchanged with equal probability for their respective genes (PID parameters) to generate new offspring parameter combinations. This method helps to maintain the genetic diversity of the population.

E. Mutation: After generating new parameter combinations, small variations are randomly applied to some of the parameters to introduce new genetic variations and promote the exploratory power of the algorithm.

F. Elite strategies: The elite strategy is an integral part of the implementation of our genetic algorithm, designed to protect the best-performing individuals over generations. This strategy ensures that a certain number of the best-performing individuals (the elite) are passed directly to the next generation without going through the process of selection, crossover and mutation. At the end of each generation, individuals in the current population are evaluated based on their fitness scores. The individuals with the highest scores are preserved for direct replication into the next generation of the population. This strategy ensures that a certain number of the best-performing individuals (the elite) are passed directly to the next generation without going through the process of selection, crossover and mutation.

G. Iteration: In the GA-PID methodology, each cart loop is a generational iteration, evaluating fitness from environmental data, and using roulette selection for crossover and mutation, introducing new individuals to prevent algorithm performance degradation. The process runs until a 20,000ms runtime is reached, aiming to optimize PID parameters for consistent straight-line motion and target PWM adherence, with results verified in following sections.

2. EEPROM:

Algorithm 1 Genetic Algorithm for PID Parameter Optimization

$populationSize \leftarrow$ Size of the population
$maxGenerations \leftarrow$ Maximum number of generations
$mutationRate \leftarrow$ Probability of mutation
$eliteSize \leftarrow$ Number of elite individuals to retain
$population \leftarrow$ Initialize population with random PID
$fitness \leftarrow$ Array to store fitness values
for $generation = 1$ to $maxGenerations$ **do**
  for $i = 1$ to $populationSize$ **do**
    $fitness[i] \leftarrow$ Evaluate($population[i]$) ▷ Evaluate each individual
  end for
  $eliteIndividuals \leftarrow$ SelectElite($population, fitness, eliteSize$) ▷ Select elite individuals
  $newPopulation \leftarrow$ Initialize with eliteIndividuals
  while size of $newPopulation < populationSize$ **do**
    $parent1, parent2 \leftarrow$ RouletteWheelSelection($fitness$)
    $child \leftarrow$ UniformCrossover($parent1, parent2$)
    $Mutate(child, mutationRate)$
    Add $child$ to $newPopulation$
  end while
  $population \leftarrow newPopulation$
end for
return The best parameter set from $population$

A. Key input processing: The status of button A (digitalRead(BUTTON-A-PIN)) is read to determine whether the data in the EEPROM needs to be read.

B. Periodic data logging: Data is recorded every 500 milliseconds (by judging currentMillis - lastRecordTime ¿= 500). This periodic recording process involves the read speed (readSpeed_L() and readSpeed_R()), the encoder count (count_leftEncoder and count_rightEncoder), and calculating the difference between the two encoder counts.

C. EEPROM data storage: The acquired data is stored in the EEPROM. Make sure there is no overflow by checking the remaining space of the EEPROM (eepromAddress ¡ EEPROM.length() - size of (RobotData)) and write data using the EEPROM.put() method.

D. Read data from EEPROM: The readDataFromEEPROM() function reads each block of data from the EEPROM and prints it out via the serial port. This includes left and right wheel speeds, left and right encoder counts, and the difference in encoder counts.

E. Conclusion: The use of EEPROM allows for long-term storage and access to critical operating data, which is essential for debugging and optimizing algorithms.

III. IMPLEMENTATION

A. *Overview of the implementation*

This experiment aimed to test and validate the efficacy of the PID controller in optimizing the powertrain of a 3pi cart. A two-stage experimental design was used:

Benchmarking: Using the bang-bang control method, the data from the operation of the cart without PID intervention was recorded. This phase aims to establish a performance benchmark against which the results of subsequent PID control can be compared.

PID optimisation test: ***Manual tuning phase:*** The PID parameters are set manually for the left and right wheels separately to find the optimal parameters for the current state and environment. ***Genetic Algorithm Optimisation Stage:*** The PID parameters were automatically adjusted using a genetic algorithm to achieve the best performance in different environments.

In this two-stage experiment, we set up two different experimental processes.

The first procedure is to set a fixed running time, based on the encoder running time of 20000ms, the purpose of this part of the experiment is to have a clear understanding of the effect of the three control strategies on the dependent variable.

The second procedure is to set a fixed running distance, and we set a fixed length: the length of five sheets of A3 paper (1470mm). Place the trolley in the starting position of the first sheet of A3 paper. In kinematics, we set the distance traveled by the trolley to X = 1470 and Y = 0. Observe the trajectory of the trolley. To make the observation clearer and more accurate, we use EEPROM to read the data while driving and generate a trajectory map. The purpose of this part of the experiment is to record and compare the actual improvement of the three control strategies on the driving of the trolley.

B. *Discussion of Variables*

- **Controlled Variables**: Experimental environment: Ensure that the environmental conditions (e.g., room temperature, humidity, light conditions) are consistent from one experiment to the next to avoid environmental factors influencing the results.
  The initial state of the trolley: The starting position of the trolley, battery level and mechanical state (e.g. tyre wear) should be consistent before the start of each experiment.
  Test surface: The type and condition of the surface on which the experiment is conducted (e.g., flat, clean, coefficient of friction, etc.) should be kept constant to prevent the ground conditions from influencing the performance of the trolley.
  Experimental Distance: The distance traveled by the trolley or encoder value should remain fixed in each experiment to ensure consistency of testing.
  Experimental equipment: Use the same 3pi trolley for all experiments to ensure that the experimental results are not affected by the difference in performance of different equipment.
  Software and control algorithms: Except for the PID parameters, other software settings and control algorithms of the trolley should remain unchanged to ensure that any performance changes are caused by the adjustment of the PID parameters.
  By controlling these variables, you can ensure the reliability of your experiments so that the results more accurately reflect the effects of the PID parameter adjustments.

- **Independent Variable**: Proportional (P): The portion of the controller response that is proportional to the current error. Integral (I): the part used to eliminate long-term accumulated errors. Differential (D): the part that predicts future error and reduces overshoot.
- **Dependent Variable(s)**: The operational performance of the trolley is assessed by the following indicators: L Speed and R Speed: the speed of the left and right wheels. Diff: the difference between the left and right wheel speeds or encoder counts, used to assess the ability of the trolley to travel in a straight line.

C. *Discussion of Selection Indicators*

Basic Principle of Indicators

1. Encoder Count:

Principle: Encoder counting is used to measure the number of times the wheels of the trolley are rotated, thus indirectly reflecting the distance traveled by the trolley. This is an important indicator to evaluate the straight-line driving ability of the trolley and the speed control.

Advantages: Directly reflects the rotation of the wheel, sensitive and accurate.

Disadvantages: Sensitive to mechanical errors and tire wear, which can lead to inaccurate readings.

2. Speed Differential:

Principle: The speed difference between the left and right wheels is compared to evaluate whether the trolley can keep driving in a straight line. Ideally, the speed of the two wheels should be the same or very close to each other.

Advantages: It intuitively reflects the ability of the trolley to travel in a straight line.

Disadvantages: Speed readings can be affected by fluctuations in motor performance and changes in battery level.

3. PWM Compliance:

Principle: Evaluate whether the actual wheel speed of the trolley reaches the preset PWM value. The PWM value is a direct control measure of the motor speed, so this indicator directly reflects the response accuracy of the control system.

Advantage: Direct evaluation of the effectiveness of the control system.

Disadvantages: This may be affected by motor characteristics and battery status—the necessity of multiple measures.

Complementarity: A single metric often doesn't provide a complete picture of system performance. For example, the encoder count may show that the trolley is traveling in a straight line, but it does not necessarily mean that it has reached the preset PWM speed. By combining different metrics, we can evaluate system performance from multiple dimensions.

Reduce misunderstandings: Relying on a single metric can lead to misunderstandings about system performance. For example, even if the PWM is met, if the speed difference is large, the car will still not be able to drive in a straight line effectively.

Comprehensive Performance Evaluation: The effectiveness of PID control can be more comprehensively evaluated using multiple metrics, including the accuracy of speed control, the ability to travel in a straight line, and the responsiveness of the overall system.

D. *experimental procedure*

Setting up the experimental environment: Ensure the test environment is stable, including road and light conditions.

Conducting benchmark tests: Conduct tests using bang-bang control and record metrics such as Diff.

Manually adjust the PID: Manually adjust the PID parameters, conduct multiple trials, and record the best performance indicators.

Apply Genetic Algorithm: Start the Genetic Algorithm program to automatically adjust the PID parameters and record the improved performance.

Data collection: Data collection is done by using the Electrically Erasable Programmable Read-Only Memory (Electrically Erasable Programmable Read-Only Memory) in the Arduino. It can then be used for analysis and charting.

Data analysis: Compare the performance data at different stages and analyze the optimization effect of the PID controller.

Repeat test: To ensure the reliability of the results, repeat the experiment several times until the data are stable.

IV. Results

Part 1: Fixed Running Time

A. The Speed of Left and Right Wheel:

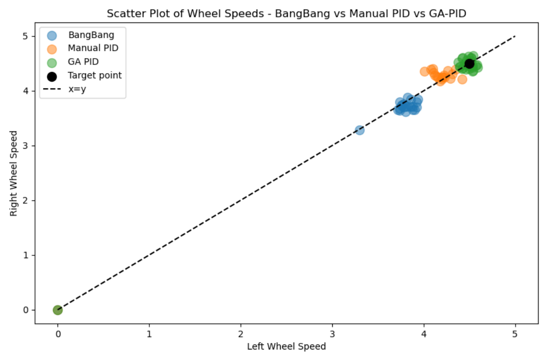

Fig. 2. Left and Right Wheel Speed Comparison of Differences - BangBang vs Manual PID vs GA-PID

Regarding the relationship graph between the left and right wheel speeds. Ideally, if the car achieves perfect straight-line driving, all points should be on the y = x line, meaning the left wheel speed equals the right wheel speed.

With the method of adjusting PID in real-time through GA, the speed points are more concentrated on y = x and closer to the set rated linear speed value of 4.5. This indicates that the speeds of the left and right wheels of the car are close to the same and close to the rated value.

The speed points through the bang-bang controller are more dispersed and further from the rated value, suggesting that the speeds of the left and right wheels are not equal and have not reached the rated value.

The speed points through the optimal fixed PID controller are more concentrated than those of the bang-bang controller but more dispersed than those adjusted by GA PID, and the speed values are closer to the rated value than those of the bang-bang controller but further from the rated value than those adjusted by GA PID.

Therefore, in the adjustment and calibration of left and right wheel speeds, the control of speed is best through GA-adjusted PID, followed by the optimal fixed PID, and the worst performance is using the bang-bang controller for speed adjustment.

B. The average speed of the left and right wheels

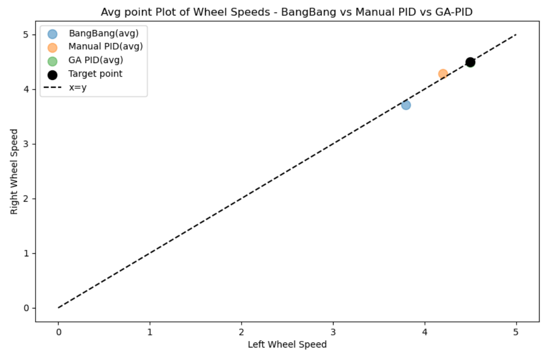

Fig. 3. Left and Right Wheels Average Speed Comparison of Differences - BangBang vs Manual PID vs GA-PID

We averaged the above values of the left and right wheel speeds and compared them with the desired average speed values.

In Bang-Bang control, the Left Wheel Speed = 3.7962, and the Right Wheel Speed = 3.7856. In fixed PID control, the Left Wheel Speed = 4.1985, and the Right Wheel Speed = 4.2830. In GA-PID control, the Left Wheel Speed = 4.4970, and the Right Wheel Speed = 4.4850, which is close to the target point of Left Wheel Speed = Right Wheel Speed = 4.5, coinciding almost exactly. The comparison reveals that under GA-PID control, the average speed difference between the left and right wheels is the smallest and closest to the rated speed.

C. The Difference Between Left and Right Encoder Counts

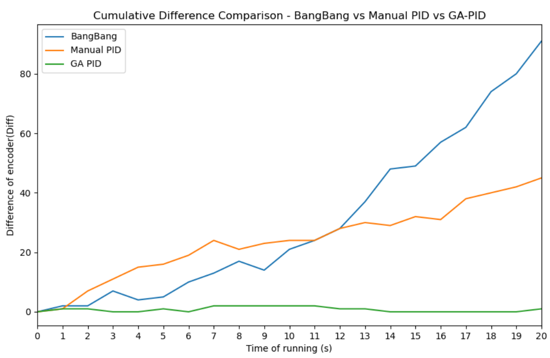

Fig. 4. Left and Right Encoder Counts Comparison of Differences - BangBang vs Manual PID vs GA-PID

The difference value controlled by the bang-bang controller increases over time in an almost exponential manner, indicating that the difference in encoder counts between the left and right wheels is increasing as the operation progresses.

The difference controlled by a fixed PID controller also increases over time but in an approximately linear fashion. This shows that the difference in encoder counts between the left and right wheels is increasing, but at a smaller rate than with the bang-bang controller.

In contrast, the difference in the GA (Genetic Algorithm) real-time adjusted PID control fluctuates around zero over time, showing no significant disparity or increasing/decreasing trend in the encoder counts of the left and right wheels.

Therefore, in adjusting and calibrating the encoder counts of the left and right wheels, the control through GA-adjusted PID is the most effective, followed by the optimal fixed PID. The least effective performance is seen with speed adjustments using the bang-bang controller.

D. The Parameters of GA-PID Control

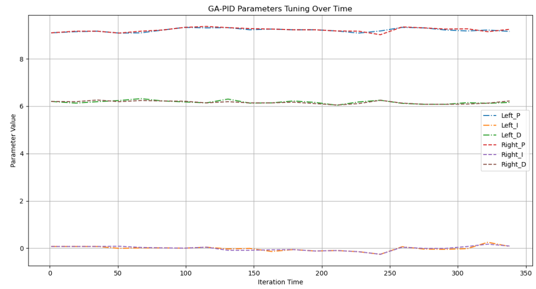

Fig. 5. GA-PID Parameters Tuning Over Time

This chart shows the change of the three values of PID in the left and right wheels with the increase in the number of iterations under the GA-PID control strategy. The P, I, D values are all optimized in real time, and the optimal PID used by the left and right wheels is approximately the same.

Part 2: Fixed running distance In actual experimental tests, The data we need to read during running are the values of X and Y in the kinematics of the left and right wheels, and the speed of the left and right wheels (the value is recorded once every 1 s) From the above two sets of recorded data, it can be calculated:

1. Deviation Value: The angle between the straight line formed at the end and the start of the drive and the straight line of the ideal trajectory.

2. Average Speed: The average value of all the speed values recorded by the left wheel and the average value of all the speed values recorded by the right wheel, and then the value obtained by averaging the two.

We conducted the above practical experiments on the three control strategies of bang-bang, fixed PID, and GA-PID, obtained the relevant values, and then analyzed the data and simulation charts.

The black line is the ideal driving trajectory, and the red line is the actual driving trajectory.

A. Bang-bang Running Trajectory From the first part of the experiment, we know that when using the bang-bang control strategy, the red line is offset to the right compared to the black line. Therefore, the speed of the left wheel is faster than the speed of the right wheel, resulting in the count value of the left encoder being greater than the count value of the right encoder. If the deviation value is large, it means that the speed of the left wheel is quite different

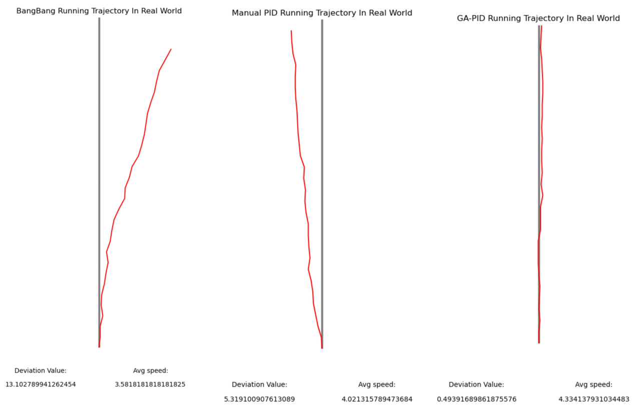

Fig. 6. Running Trajectory In Real World

from that of the right wheel. The AVG speed is smaller, indicating that the speed of the left and right wheels is quite different from the rated speed.

B. Manual PID Running Trajectory When using the fixed PID control strategy, the red line is offset to the left compared to the black line. Indicates that the speed of the right wheel is faster than that of the left wheel, causing the count value of the right encoder to be greater than the count value of the left encoder. The deviation value is smaller, indicating that the difference between the speed of the left wheel and the speed of the right wheel is smaller than that of the Bang-Bang control strategy. The AVG Speed is larger but does not reach the rating. It shows that the difference between the speed of the left and right wheels and the rated speed is smaller than that of the bang-bang control strategy.

C. GA-PID Running Trajectory When using the GA-PID control strategy, the red line and the black line are approximately coincidental, so the left and right wheel speeds are approximately the same, and the left and right encoder count values are approximately the same. If the deviation value is small, it means that the GA-PID has the best performance for the improvement of the straight-line driving of the trolley. Avg speed=4.33413 is close to the rated value of 4.5, which also indicates that GA-PID has the best performance for the trolley to achieve rated speed improvement.

V. Discussion and Conclusion

Main Discovery

By setting a fixed working time to observe the working conditions of three groups of 3Pi+ cars with different control strategies, we can get two underlying conclusions:

1. Whether the speed of the left and right wheels of the trolley is the same

2. Whether the speed of the left and right wheels of the trolley can reach the rated value

Summarizing the above experiments, it is most likely to deviate from the straight line when driving under the Bang-Bang control strategy, and the speed of the left and right wheels is the most different from the rated speed. Under the fixed PID control strategy, the performance of the trolley in a straight line is moderate, but there is a certain gap between the speed of the left and right wheels and the rated value. Under the GA-PID control strategy, the driving path of the trolley is the straightest, and the speed of the left and right wheels is closest to the rated speed.

Significance

GA-PID can improve the above two control strategies, and can effectively make up for the shortcomings of the fixed PID method, such as better handling of nonlinear problems, better adaptability to changes in the environment and hardware facilities, and reducing manual intervention (requiring manual time-consuming and laborious adjustment of PID parameter values) The control system in automated production can be controlled by GA-PID, which can improve the efficiency and accuracy of the robotic arm. There is no need to manually adjust the PID regularly due to changes in the production environment (e.g., temperature, humidity) and mechanical wear, which is beneficial to the efficiency of production and the reduction of labor costs.

Areas for improvement

The GA-PID method is not used to prove the practicability of the method in a complex control system by using the GA-PID method for the speed control of the left and right wheels of the trolley. There are only three independent variables in this work: P, I, and D. Fewer independent variables are easier to control in experiments. As for the problem of wheel speed, the speed on the wheel is still not up to the rated speed, which can reflect that GA should find a better solution in the iterative update of PID. Meanwhile, Fuzzy logic can be used to deal with uncertainty and ambiguity in GA-PID parameter tuning.

Future research directions

Future research direction: Through the control method of GA-PID, it has been possible to realize the straight line driving of the trolley at an approximate rated speed. The future research direction is to integrate the GA-PID method with the line patrol function so that the trolley can also have stable driving performance in the process of line patrol. If the two are fused, there will be more and more difficult challenges, such as more changes in external environmental conditions (lighting conditions) and more changes in hardware conditions (line sensors)

References

[1] H. Y. Y. C. Dingming Yang, Zeyu Yu, "An improved genetic algorithm and its application in neural network adversarial attack," 2022. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0267970 [Accessed 22-05-2022].

[2] https://en.wikipedia.org/wiki/EEPROM.

[3] https://uk.rs-online.com/web/content/discovery/ideas-and-advice/eeprom-guide.

[4] https://www.electricity-magnetism.org/eeprom-programmers/.

[5] J.-H. L, "Genetic algorithm pid control of second-order analog circuit system," *Proceedings of International Conference on System Science and Engineering 2022 (ICSSE 2022)*, 2022.

[6] F. Y. X. F. H. L. G. Dong, "Improved roulette wheel selection-based genetic algorithm for tsp," *2016 International Conference on Network and Information Systems for Computers*, vol. 978-1-4673-8838-2/16, p. 2, 2016.