Botball robot running calibration
Mingxiao Song
Dalian No. 24 High School

Botball Robot Running Calibration

1. Introduction

Robots are highly repeatable but not accurate. Therefore, robot accuracy can be improved through robot calibration. During construction and debugging of the Botball robot, lack of accuracy of components and problems with its operating system may cause robot to deviate from its planned motion trajectory and cause serious mistakes in the final task. These errors happen from time to time in robot operation, because it’s difficult to achieve perfect angular orientation with the assembled structure and the performance of the components are also being influenced by each other as the robot moves. Therefore, how to use dexterous means to prevent or reduce errors in construction and operation to help robot complete its own calibration and adjustment has gradually become the focus of attention. This article studies the calibration of the Botball robot during its operation, analyzes the causes of the error, discusses the purpose of implementing the correction, and provides a feasible solution for the correction of the robot, so as to provide reference for the operation of other robots.

2. Botball robot running deviation overview

2.1 Deviation in robot operation

Even if the underlying program is absolutely accurate, the uncertainty of robot itself will still cause some random route problems during debugging and assembly. Below we present some problems we have encountered.

2.1.1 Deviation of robot motion route

The inability of a robot to move along a predetermined route will affect every task of the contest and is one of the most important problems we need to avoid. During first debugging run, although our programming was correct, the robot car could not move straight along the black line and the deviation was large. In addition, unexpected position of the car may have also caused the car to crash into the plastic piping around the playing field or hinder the operation of the big car. These discrepancies have made it impossible for us to carry out the next task smoothly, such as the task of fetching date palms and Botguy, which required precision in angles and positions.

2.1.2 Deviation of the robot turning angle

Since the execution of each task of the Botball requires the robot to be in a relatively accurate position, deviation of the turning angle of robot is also a crucial problem to be solved. In debugging, it often occurs that the rotation angle of the car does not reach the expected value, which will affect the success rate of the task. Deviation of the car's turning angle is due to the cumulative deviation of the car's initial position (the position before the curve is different from preset value) following the previous task. On the other hand, it is due to the structural installation problems of the car itself, such as the loose screws of the wheel and the diverged rubber pads of tire etc.

2.1.3 Robot arm opening and closing angle deviation

This kind of problem mainly appears in the task of capturing objects. The degree of opening and closing of the clips is often difficult to reach preset angles. For example, the clips cannot clamp the objects, the clips open and close at inappropriate angles etc., which will easily damage the steering gear and components especially when there is obstruction. The clip itself is driven by a small steering gear, and the power provided by it is limited. It is not easy to grasp the object firmly with clips. If the opening and closing angle of the clip cannot be accurately controlled, then the clip will loosen when gripping the object and the object will fall off, eventually causing the task to fail.

2.2 Analysis of causes of deviation in robot operation

Through analysis, the causes of deviations in the operation of the Botball robot involve component material factors and indirect external factors.

2.2.1 Influence of component materials factors

The influence of component materials are the errors belonging to the nature of the components themselves during installation or production, including the materials, the applicable range，the quality and performance etc. These are factors determined by the material which we can do nothing about.

To further illustrate the impact of component materials, an example is described below:
Deviation of the wheels prevents the robot from moving in a straight line. We have tried to twist the wheels or change their structure with relatively tough measures. However, the results prove that these methods are not feasible. First, manual correction cannot guarantee that each wheel is adjusted to the same angle symmetrically. Second, the operation itself can be destructive to the equipment. In the design of the machine, we need to protect the resources and make it reusable. Third, the operation is not durable. The rubber of the wheels is elastic, so the wheel may return to its original angle after moving for a while. Therefore, the objective factors are difficult to change. We need to switch the operation to human intervention.

2.2.2 Influence of indirect factors

Opposite to the component materials factors, the influence of external indirect factors refers to the deviation that can be avoided by debugging. Such factors that can be manually controlled are subjective influence factors.

For example, human can set up the initial position of the robot. However, the accuracy is hard to control. If the initial position is wrong, the robot will find itself difficult to go back to the normal route. Another example is that some teams were not familiar with the rules and did not set up robot position before the game started. This will directly affect the code and data during the debugging process and in turn affect the actual performance later on. Clearing obstacles and control of the site safety of the robot are also very important steps. If the car moves irregularly, accuracy of data measurement will be affected. The probability of damage due to a sudden collision during the operation of the robot will also increase. The program code is the core of the entire robot operation. If the robot fails to set off, then the preparation of hardware and documentation are in vain. During previous runs, after each straight move, left turn and right turn command execution, the car would continue to move for a short period of time. At first we thought it was due to inertia or differential problem and tried to change the speed of the code but we later discovered that this was because we did not explicitly code in the duration of a “freeze” in the program.

In addition, the reinforcement of the car structure is also important. If the wheels of the car are not stable, it will constantly change the car’s direction. Therefore, the car is unable to run in a straight line. If loosely installed, the robot arm will not be able to clamp palms through stable force even if it is in preset angle. If the tail tracking device of the car shakes back and forth with the car, the gray-scale we used to correct it will not help.

The above are two factors that can lead to deviations in robot operation: unchangeable internal factors and controllable external factors.

3. Purpose of the Botball robot running calibration

3.1 Purpose of robot calibration

Since robots need to automatically complete preset tasks without any human intervention, the deviation of the running route will not only have an impact on current task, but will also have cumulative effects on many subsequent tasks. Through robot calibration, we can avoid these errors in time and provide higher accuracy for the next task.

3.2 Difficulty analysis of robot calibration

With the limited and unchangeable components, we need more innovative and divergent thinking to develop solutions for assembling and designing robot. Predicting all conditions and problems that the robot may encounter is the most difficult part for robot calibration. Because the independent operation of the robot has uncertainties, it is necessary to consider all possible problems, although constantly trying and simulating various possible situations is a tedious and time-consuming job. However, if we want to ensure that robots can perform tasks reliably with little probability of error in the formal contest, we need to confirm the calibration.

4. Specific measures for Botball robot calibration

According to Wikipedia, robot calibration is the process of identifying certain parameters in the kinematic structure of an industrial robot, such as the relative position of robot links. In our context, we define robot calibration as successfully completing tasks against referred standard and continuously adjusting the robot's operation angle so that it can reach the correct position and angle. The specific Botball robot calibration measures are as follows:

4.1 Collision calibration

Collision calibration is to achieve the purpose of self-calibration by colliding surrounding plastic pipe. The deviation can be calibrated in that the car will keep moving for some distance after it hits surrounding fence. This solution requires the installation of a relatively strong structure in the front and the rear of the robot for calibration, such as horizontal beams or symmetrical protruding structures.

This kind of calibration is applicable to the situation where the car is close to the surrounding area and the car clip is not on the running line of the car, and there is no interference with the surrounding pipe. Collision calibration is very accurate and can adjust the running equipment well. We use the collision calibration while the car is driving back from the middle of the venue to the side. Usually, the car needs to run for 4200 millimeters, however, we set it to run 200 millimeters more. As a result, the left or right side of the car which gets to the edge first will wait for the other side since the car is still moving on. This is an effective method of automatic calibration, but it is not appropriate to use when the car is in the middle. However, the car may not be able to achieve the purpose of rotation correction because some structures are stuck by the pipe.

The problem needs to be addressed in this type of calibration. If simply allowing robot to continue moving after hitting the wall, you may damage the motor and the mechanical structure of the robot. Therefore, it is necessary to set up the normal position of the robot, and try to increase moving length on this basis.

4.2 Patrol line calibration

Through the gray sensor, the gray-scale data is transmitted in time to detect the location. The patrol line calibration is to correct according to the black line on the site, enabling the car to walk straight along the black line. At the same time, via gray sensor at the rear, the car can still move forward or turn back according to different conditions.

The advantage of patrol line calibration is that it can be applied to a wide range of situations. No matter where the car is, the robot can retreat to the black line for adjustment. The car runs relatively slowly during the retreat, and even if encountering obstacles, it will not seriously affect the car. The disadvantage of patrol calibration is that it needs to be measured continuously during programming. Only by considering in advance the response of the cart when it gets different data at different locations, can it flexibly deal with potential situations. At the same time, it takes a lot of time to detect the gray level when the cart slowly retreats.

The following is an example of code that we designed in the competition program.

```
If(analog (0)<2000&&analog(1)<2000){
Mav(1,-100);
Mav(2,-100);
}
If(analog(0)<2000&&analog(1)>2000){
Mav(1,-100);
Mav(2,0);
}
```

```
If(analog(0)>2000&&analog(1)<2000){
Mav(1,0);
Mav(2,-100);
}
If(analog(0)>2000&&analog(1)>2000){
Mav(1,100);
Mav(2,100);
}
```

The blue bar structure shown in Figure 1 is the hardware structure used for collision correction and play a crucial role during the car’s turning while the black gray-scale sensors on both sides of the blue bar are used. To detect gray-scale components for patrol line correction purposes, such structure can also provide relatively accurate positioning of the car's initial position.

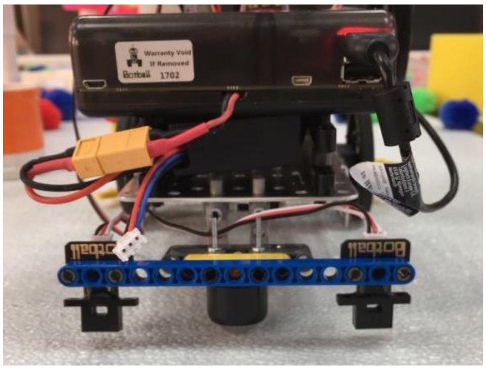

Figure 1

4.3 Differential speed calibration

Differential speed calibration is one of the most widely used calibration methods and can be used to assist the robot to run in a straight line. Differential speed values also require constant debugging, but they can be used repeatedly after relatively stable data are available. The differential speed calibration can overcome problems in situations when the car cannot run in a straight line or when it cannot turn right angle due to the objective factors mentioned above.

At the same time, differential speed calibration also involves some techniques in coding. Since the speed of differential calibration is often changed, it is very time consuming to change main function or each customized function. So it is necessary to set the speed parameter used for differential calibration as a global variable at the beginning of the code. Thus, we only needs to change the global variable when circumstance changes. This makes it easier in case of speed adjustment and emergency.

At the end of our game preparation, the motor of the car wheel was suddenly damaged and could not move. The replacement of the motor may cause all of our previously measured data to be invalid, and all distances and angles to be readjusted. In this case, we have designed the most efficient remedial measures, that is, to make the car run in a straight line in a differential speed manner. On this basis, we measure the difference in the straight line distance when the car starts to run. If the deviation is small, the corresponding value can be added or subtracted directly in the subsequent program. In this way, the differential speed saves us a lot of time in data recovery and achieves high efficiency.

4.4 Install speed sensor

A speed sensor installed on the wheel can record the time taken for each rotation of the wheel and estimate the change of the speed of the car by two adjacent time points, and then automatically adjust the speed in the program.

Assuming that the circumference of the wheel is d, that is, the distance traveled by the wheel after one rotation, the running time of the wheel rotation is t. So the speed of wheel rotation is $v=d/t$, and the obtained data v is stored in the speed sensor. The car will measure the speed of the left and right wheels at the same time and adjust the speed by comparing two data. If the speed of the right wheel is greater than the speed of the original differential, the program will slow down the wheel speed.

This solution can be time-saving during robot debugging. When the robot is running, it may often change the wheel speed. In this case, we no longer need to measure the speed and adjust the data manually. Instead, the robot automatically adjusts itself. This may be the ideal level of robot calibration.

4.5 Measures to reduce debugging errors

In order to reduce debugging mistakes, we should carefully confirm and check code. First of all, coding requires absolute precision following constant standard. Program should be compiled separately on the computer and the controller before it can be debugged. Secondly, the data in the program needs to be feasible. We need to consider the limits of the equipment and control the data within its limits, such as the height at which the robotic arm can lift and the maximum angle at which the clip opens, so as to avoid mis-operations and damage to the car components. At the same time, we must pay attention to the placement of the car's starting position and meet the requirements of program. We cannot afford subjective error before game starts. Finally, we must keep an eye on the progress of the program. Sometimes there will be problems with the distance or height of the running machine which can easily cause damage. Therefore, we should rescue the robot quickly to protect the device.

As shown in Figure 2, installing the baffle in front of the car can effectively prevent small balls or obstacles from entering the bottom of the car, thus ensuring its smooth operation.

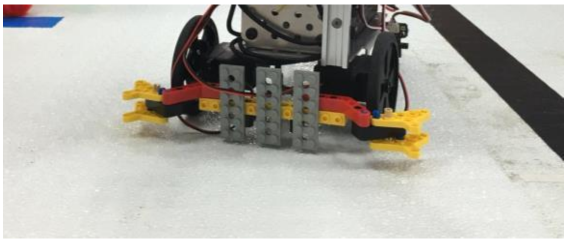

Figure 2

5. Conclusion

Nowadays, robots are increasingly needed to perform high-precision positioning tasks with almost zero errors. Even doing daily tasks, robot precise coordination among components is required. This paper discusses several calibration solutions. If the above calibration methods are applied in the development of small robots such as Botball, the robot's running accuracy will be greatly improved. At the same time, the damage of robots caused by route deviation to various components can be avoided. Thus the success rate of task completion can be greatly enhanced.

References

- Wikipeadia: Robot calibration definition
- Botball robot build guide
- Botball manual
- Botball match description