Feedback and Reactive Control
Evan House
Alaina Kafkes, Wenbo Yu
10/17/17

B

24 hour pass used

ABSTRACT:

Learning how to control automated systems is an essential part of robotics. Two of the most developed methods for controlling automated systems are known as feedback and reactive control. Feedback control uses the robot's current state and compares it to the goal state and acts based on the difference between the two. Reactive control uses the robot's sensors to gather data, and the robot reacts in response to the data gathered by the sensors. [1] In this lab, our group developed a walking and turning gait. With these behaviors, we implemented a feedback and reactive control behavior. We found that it is essential to develop behaviors that will work reliably in changing environments.

I. INTRODUCTION:

Where are your references listed?

In robotics, control theory, or the mathematical study of how to control automated systems (robots) is one of the central areas of study. [1] Two common types of control are both feedback and reactive control.

this section is also missing some concepts we went over in class and in the book

Feedback control is used in order to get a robot to maintain or reach a desired state. This desired state is commonly known as the set point. This is achieved through constantly comparing the robot's current state with the set point. [1] The name feedback naturally refers to the information that is continually sent to the robot. One example of this is a robot maintaining a certain distance from a wall. Reactive control is different from feedback control in that it is used to react to a set of situations instead of trying to reach a desired state. Reactive systems continually collect sensory information and react accordingly. An example of this is when a robot turns to avoid an obstacle. [1] In reactive control, it is ideal to split up the situations or conditions to which the robot reacts into mutually exclusive states, meaning each state is unique. Ideally, these situations can all be detected by the system's sensors. [1]

in this section your text is far too close to the book, if you directly use their words you should use quotation marks

In feedback control, there are both achievement goals and maintenance goals. Achievement goals are those where the robot tries to reach a certain state, and once it reaches the state, the robot is done doing work. [1] Maintenance goals are those which require continuous work. An example of this is a robot walking continuously and maintaining its balance. [1] The difference between the desired state and the current state of the robot is naturally called the error. In feedback control, the error is computed and relayed to the system. If the error is large, then the robot reacts accordingly; if the error is small enough, the desired state has been reached. [1]

There are a few commonly used types of feedback control. The most basic is known as proportional control, where the system reacts in direct proportion to the error. Formally, it can be described by the following equation:

$$o = K_p i \text{ [1]}, \qquad (i)$$

where o is the output, i is the input, and K_p is a proportionally constant particular to each different system. The factors which determine how the system responds to error are known as gains. [1] A second type of feedback control is derivative control. Once again, derivative control is formally described by the following equation:

$$o = K_d \frac{di}{dt} \text{ [1]}, \qquad (ii)$$

where o, i, and K_d are the output, input and proportionality constant respectively. The $\frac{di}{dt}$ is the derivative term which is the derivative of the position (velocity). The idea is that this derivative term corrects for the system's momentum as the system gets closer to the goal state [1]. The third type of feedback control is integral control. We first describe it formally:

$$o = K_f \int i(t)\, dt \text{ [1]} \qquad (iii)$$

Integral control works by keeping track of the system error and integrating these errors over time. If the error reaches a set threshold, the system compensates accordingly. Most commonly, the system tracks steady state errors which are repeatable, fixed errors. [1] Naturally we can combine some or all of these types of control together. Proportional integral derivative control is described by the following equation:

$$o = K_p i + K_d \frac{di}{dt} + K_f \int i(t)\, dt \text{ [1]} \qquad (iv)$$

II. METHODS:

Before we began working on the tasks in this lab, the first thing our group did was attach two new sensors to the robot. We added two additional infrared (IR) sensors to the robot, one on the right and left sides of the robot's body. In total, the robot had three IR sensors, one on the left, right and one facing forward.

Once the additional IR sensors were attached, our first task was to develop a walking gait for the biped robot. Our first step in doing this was to move the robot's legs by hand to see how we could best make it walk. We also considered the robot's walking gait in the demo behavior and sought to simulate this behavior. We decided that we would develop the walking gait by moving its right foot forward, lifting its left leg by standing on the side of its feet and moving forward the left foot. It would repeat this motion with both its right and left feet.

After we decided on this walking gait, we began to attempt to implement it. The first thing we did was to test out each of the actuators and identify which actuators needed to implement the walking gait. The next step was to test out different position values for each actuator between 0 and 1024. After we identified the actuators and had an understanding of what each position value did, we began to write our walking gait function. We found it helpful to split our function into three distinct stages. In the first stage the robot would lift its left leg and move it forward; in the second stage, the robot lifts its right leg and moves it forward; in the third stage, the robot resets its feet into the middle position. Once this is complete, the function returns to stage one and cycles through the three continuously.

One of the problems we had with this initially, is that the robot's feet would often slip as it walked on the ground. We could not account for this completely, as every surface is different, however we found it helpful to decrease the speed of the robot's movements, and cause the robot to sleep in between movements and stages. Although this naturally decreased the speed of the walking gait, it made the walking gait far more reliable. We also later added a ready position function which the robot would move to before walking.

Our second task in this lab was to develop a turning gait for our robot platform. We implemented three different turning functions: turn left 90 degrees, turn right 90 degrees and turn around 180 degrees. Once again, we first looked at the robot and its actuators to determine how we should go about writing these functions. We decided that the robot would turn by taking small steps to either the right or left side, until it had turned 90 or 180 degrees. Once we had identified the correct actuators and position values for the small steps, we tested out our function to see how many small turns the robot should take to reach 90 and 180 degrees. We found this method to work reliably, although in the end, it did depend more on the friction of the surface than if it were to turn 90 degrees in just one step.

Our third task was a relatively straightforward task of ensuring that our robot would be able to detect obstacles to its front, right and left. The additional IR sensors we added to the right and left sides of the robot's body were able to account for this.

Our fourth task in this lab was to use feedback control in order to make the robot follow a wall. More specifically, the robot would walk along the side of the wall maintaining a distance of 15 centimeters between the sensor and the wall. We found it helpful to first identify the range of the sensor value and more specifically, the sensor value at 15cm from the sensor. This was pretty straightforward, as we just had a function return the sensor value continuously, and we placed an object in the path of the IR sensor 15cm away. After testing the sensor values, we identified a range of values corresponding to a distance about 15cm from the wall that the robot should stay within. At this point we were fortunately able to depend largely on our previously written turning and walking gaits. Our function told the robot to check the value of its right sensor, thus identifying how far it was from the wall. If the sensor value was too large, meaning the robot was too far from the wall, the robot would turn partially to its right, in the direction of the wall; If the sensor value was too small, meaning the robot was too close to the wall, the robot would turn partially to its left, in the direction away from the wall.

One issue we found here, was that the time at which the robot would check the sensor value would change as the robot walked. While its right foot was forward, the body was turned in such a way that the sensor value was larger than the specified range, although the robot was still about 15cm from the wall. In order to account for this, our group made a slight adjustment to the walking gait for this function. After the robot completed one walking cycle,

that is, it had moved its left foot forward once and then its right foot, we had the robot shift back to a simple standing position so that the IR sensor was directly facing the wall. Once the robot had moved to this position, it would check the sensor value. In this way, we could reduce the variability in the range of sensor values. We were also certain that we were measuring 15cm between the sensor on the side of the robot's body and the wall.

Our final task in this lab was to use reactive control in order to turn when it was blocked by an object. We implemented three different variations. When the front and left sensors were blocked, the robot would turn right; when the front and right sensors were blocked, the robot would turn left; when the front, right and left sensors were all blocked, the robot would turn around. In this case, we defined the robot being blocked by detection of an obstacle within 15cm of the robot.

As we did in the case of feedback control, we began by measuring the front, left and right sensor values at 15cm away from each individual sensor. After identifying the range of sensor values, we began writing our function, which was rather straightforward. The robot would check its sensor values and if it found its sensors to be blocked it would turn accordingly. As we did with the wall following task, we were able to use the turning gait functions we had previously implemented.

III. RESULTS:

Walking Gait Data:

Measurement	Mean (cm)	Standard Deviation
5s Walking Distance	10.76	0.26
10s Walking Distance	18.06	1.06
15s Walking Distance	25.30	1.41

Measurement	Mean (degrees)	Standard Deviation
5s Change in Heading	N/A	N/A
10s Change in Heading	0.40	3.72
15s Change in Heading	-5.80	8.30

Measurement	Mean (cm)	Standard Deviation
10s Forward Distance	18.02	1.06
15s Forward Distance	24.91	1.42

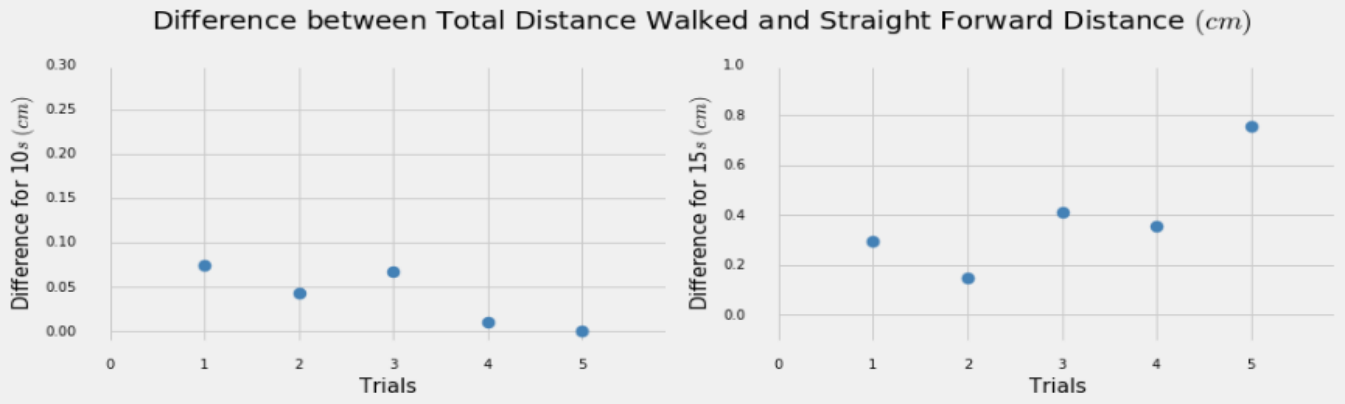
The first set of data we gathered was related to the walking gait our group implemented. We measured the distance which the robot walked after 5 seconds, 10 seconds and 15 seconds. Because there was a few second delay between when we started running the function and when the robot began walking, our group would start the timer right when the robot began moving. Once the timer reached 5 seconds one member of the lab group would immediately stop the program from the command line. We repeated this procedure for times of 10 and 15 seconds as well.

Once the robot had stopped, we measured the total distance the robot walked from its starting point, and we also measured the degree by which it had shifted from the direction it had been facing. We measured the distance walked according to the position of the front most forward foot. Having measured the angle and total distance, we were able to calculate the distance walked along the original direction the robot had been facing. However, for the distance walked after 5 seconds, the degree by which the robot had shifted from the original axis was too small to measure. For this reason, the distance walked along the axis is not calculated for the walking gait after 5 seconds.

in the future include this information in the table

this data is a bit hard to read since the numbers are so small, the plots are nice but a table may be easier to read given the space constraints

Above is the data our lab group gathered for the walking gait after 5, 10 and 15 seconds. Below is a plot of the difference between the distance walked and the distance walked directly forward.



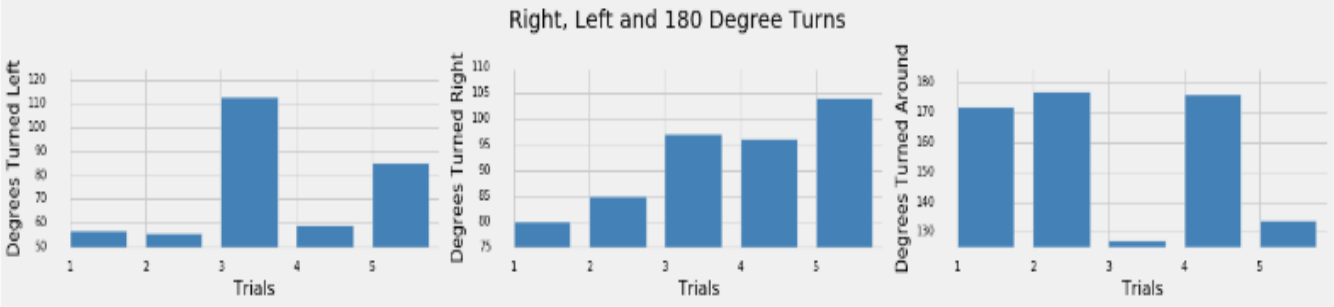
good!

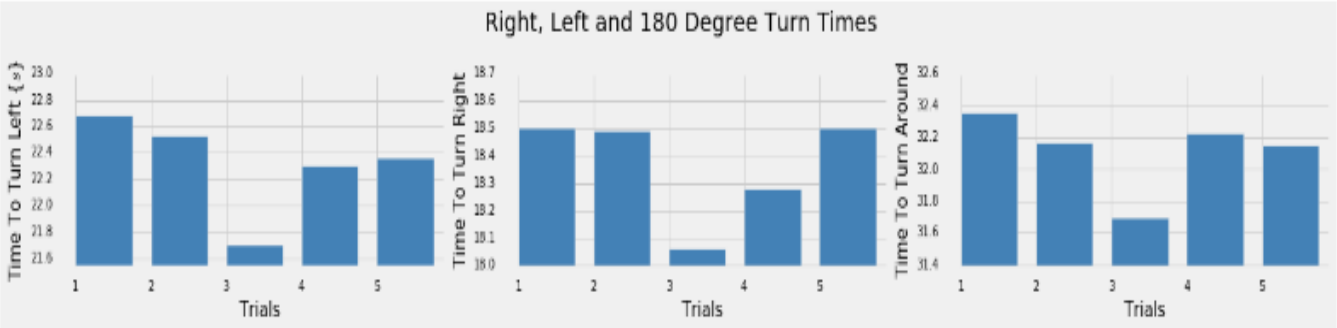
Turning Gait Data:

Measurement	Mean (degree)	Standard Deviation
Left Turn Degree	74.0	22.27
Right Turn Degree	92.4	8.69
Turn Around Degree	157.2	21.98

Measurement	Mean	Standard Deviation
Left Turn Time	22.31	0.34
Right Turn Time	18.37	0.18
Turn Around Time	32.12	0.22

After we had gathered the data for the walking gait, we next gathered data concerning the turning gait. For each turn, the 90 degree right and left turn and the 180 degree turn, we recorded how many degrees the robot actually turned as well as the time it took for the robot to turn. Below are plots of all three, detailing our measurements in each case. We gathered this data by running the function and immediately starting a timer. We laid a meter stick in the initial direction which the robot was facing before it began to turn. Once the robot stopped, we immediately stopped the timer. We laid a second meter stick in the direction which the robot faced after it had stopped. We then used a protractor to measure the angle between the two meter sticks.



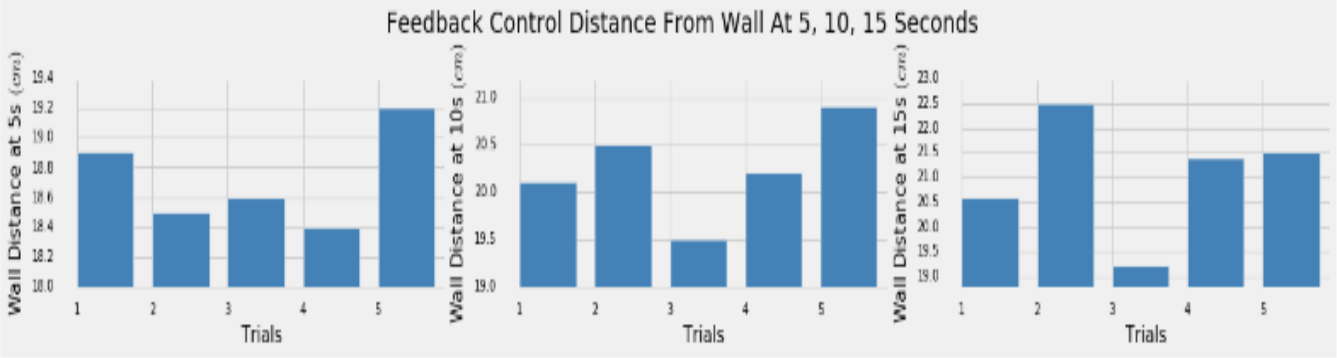


Feedback Control Data:

Measurement	Mean (cm)	Standard Deviation
5s Feedback Control Distance	5.10	1.16
10s Feedback Walking Distance	10.74	0.33
15s Feedback Walking Distance	17.80	0.75

Measurement	Mean (cm)	Standard Deviation
5s Feedback Control Wall Distance	18.72	0.29
10s Feedback Control Wall Distance	20.24	0.46
15s Feedback Control Wall Distance	21.04	1.10

The last set of data we collected concerned the feedback control behavior, where the robot would walk along the wall and adjust its position if it had walked too close or too far from the wall. The goal was to keep the robot about 15 centimeters from the edge of the wall. We let the feedback control function run for 5, 10 and 15 seconds before stopping it and making measurements. We measured the distance the robot had walked and the distance from the robot's sensor to the wall; this measurement was repeated five times. There was a short delay between the start of the function and when the robot started walking, so our group began a timer immediately when the robot started walking. At 5, 10, or 15 seconds, one member of the lab group would quit the function and the robot would stop immediately. We chose to measure the distance to the wall from the robot's sensor closest to the wall. We measured the distance the robot walked by the front most position of its forward foot when it had stopped.



IV. DISCUSSION:

We were able to successfully develop a walking gait that was relatively consistent and accurate. From the data gathered, we found that our robot did not veer too much from a straight path, as the difference from the path was never more than a centimeter. One of the tradeoffs in this case, however, was that our robot walked at a slow pace. One potential improvement for the future would be to increase the walking speed while maintaining the same level of accuracy. There are two main uncertainties involved in measuring the walking distance. First, it is possible that our start and stop timing were not always consistent or precise. Second, we found it quite difficult to measure the change in heading because the degree was quite small, especially considering the size of the robot's body. A few times, the robot stopped in a position so that it lost its balance and fell. We repeated the measurements when this happened. We were not surprised to find that the difference in heading increased the longer the robot walked.

We were also able to develop turning gaits, though they turned out to be less reliable than our walking gait. Our standard deviation for each of the turning gaits turned out to be pretty large. Among the turning gaits, the 90 degree turn to the right was the most accurate with an average turn of 92.4 degrees. For that reason, we chose to have the robot turn to the right when turning 180 degrees. Our group speculated that perhaps the results on the right side were more favorable due to either variance in the left and right motors or due to a difference in friction generated by the right and left feet.

Using the previously developed turning and walking gaits, we successfully produced a feedback control function. However, we found that the robot increased in distance from the wall the longer the robot walked. Even after just 5 seconds, the robot was consistently more than 15 centimeters away from the wall. There may be a couple of reasons that this was the case. One possible reason is that our sensor value was inconsistent so that the robot was not actually 15 centimeters away from the wall when it started. A second reason is that the position of the sensor varied with when the walking gait was stopped so that the sensor was slightly more than 15 centimeters away.

Our reactive control function worked as expected. The sensors detected objects in their path consistently. The only variability in performance was due to the variance in the previously written turning gait functions.

Through this lab we were able to gain greater insight into the field of control theory in robotics. [1] For one, we discovered that there are a lot of factors that change which are largely out of our control that we must account for to the best of our abilities. For example, we often found that our biggest issue was dealing with the variability in the actuators or the friction or lack thereof between the robot's feet and the floor. Clearly an important part of robotics is writing behaviors that can account well for changes in the external environment.

V. CONCLUSION:

In this lab, we developed walking and turning behaviors for our robot. We used these behaviors to implement feedback and reactive control functions. After we had successfully written these functions, we gathered data on the results of our behaviors. The walking gait we developed worked with greater consistency than our turning gait. We found that the uncertainty in the environment led to greater inconsistency in our results. We attempted to write our functions in such a way that would help account for this.

demo,wall :

demo, reactive :