Infusing Cyber-Physical Systems Concepts into Computer Science Curricular

Module: Timing Constraints

Release 0.1

August 20, 2013

Contact: huichen@ieee.org

1 Overview

We increase the complexity of this course module by adding a timing constraint to the problem itself. You not only program the robotic vehicle to follow a circular track but also make sure to complete a circle in a given time period (e.g., 10 seconds) with a tolerable error (e.g., ±2 seconds). The module provides you an experience of embedded control with timing constraints. The module is an introduction to the integration of discrete abstraction, such as a logical sequence of instructions (i.e., the program that controls the robotic vehicle), and continuous abstraction, such as a mathematical expression (i.e., a continuous mathematical function) that describes the relationship between the timing property and the controls of the robotic vehicle. Additionally, the module provides an enhanced experience of algorithmic thinking, algorithm efficiency, and solution evaluation of a technical solution.

1.1 Prerequisite

We recommend that you complete the "Trial and Error", "Binary Search", "Simple Model" modules before you start this module. The three modules should provide you with an introduction to the control of a robotic vehicle with a computer program, the concept of algorithm and algorithm efficiency, and simple mathematical modeling aiding the development of the programming solution to a technical problem. It does not preclude that you treat this module as a standalone one without trying the three modules; however, when in doubt, you are encourage to check out the three modules.

1.2 Lab Requirement

- A *Boe-Bot* robot.
- A PC with the *BASIC Stamp Editor* software installed
- A stop watch. You can also use an online stop-watch, such as those on `http://www.online-stopwatch.com/`
- A large circular track and a small circular track
- Program *circlerun.bs2*

- Gridded paper[1]

1.3 Programming Boe-Bot

See the "Trial and Error" module.

1.4 Running Circles

We introduce Program *circlerun.bs2* in the "Trial and Error" module and use it also in the "Binary Search" and "Simple Modle" modules. We will use this program in this module as well. For convenience, we list the program in program listing 1. The program controls a roboit vehicle to run along cirular tracks as depicted in Figure 1.

```
1  ' {$STAMP BS2}
2  ' {$PBASIC 2.5}
3
4  DEBUG "Program Running"
5  DO
6      PULSOUT 13, 850
7      PULSOUT 12, 750
8      PAUSE 20
9  LOOP
```

Listing 1: Program circlerun.bs2

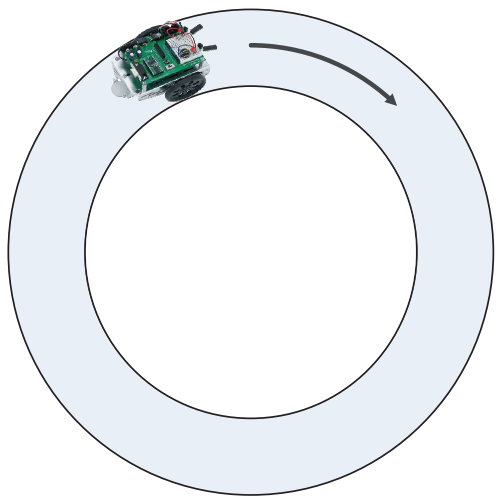

Figure 1: The robot runs within a circular track.

2 Tasks

You are to solve the same problem in the "Trial and Error", "Binary Search", and "Simple Model" modules, i.e., to revise program *circlerun.bs2* so that the robotic vehicle will travel endlessly following a given circular track whose radius is different from those obtained from previous experiments. However, you must control the speed of the robotic vehicle as well so that the robotic vehicle finishes one circle in a given time frame with a tolerable error, e.g., 10 ± 2 seconds.

To meet the requirement, you must adjust the pulse widths for both pins 13 and 12, i.e., the pulse width values in Lines 6 and 7. It becomes non-trivial by using a "Trial and Error" method, a "Binary Search" method, or a "Simple Model" method. The solution to be explored in this module is to construct a mathematical model that describes the relationship among the speed of the vehicle, the radius of the circular path along which the vehicle travels, and the time that it takes to finish a circle. The model is more complex than the "simple model".

[1] You may print the gridded paper PDF file from `http://sysnetgrp.net/cpsedu/`.

2.1 Task 1: Timing Constraint

Through this task, you will use a mathemtical model to guide the development of the revision of the program that controls the robotic vehicle.

1. You shall complete a few test runs and use equation 1 to establish the relationship between the pulse widths in Lines 6 and 7 of program *circlerun.bs2* and the revolution speed of the two driving wheels.

   Make sure that the two servo motors that drive the two drive wheels are well centered. For each test run, use the pulse widths in Table 1, run the program on the robotic vehicle for the amount of time indicated in Table 1, and measure the distance that the robotic vehicle travels. Calculate the revolution speed of the wheels using equation 1. Note that you need measure the radius of the drive wheels. We assume that the two drive wheels are of the same size.

$$V_{test} = \frac{d_{test}}{2\pi r_{wheel} t_{test}} \tag{1}$$

   where r_{wheel} is the radius of the drive wheels, d_{test} is the distance that the robotic vehicle travels, t_{test} is the time spent, and V_{test} is the revolution speed of the drive wheels. The revolution speed is measured in *round/second* (or *r/s*).

Pulse Width for Pin 13 (μs)	Pulse Width for Pin 12 (μs)	Wheel Radius (r_{wheel}) (inches)	Duration (t_{test}) (s)	Distance (d_{test}) (inches)	Revolution Speed (V_{test}) (round/second)
850	650		8		
825	675		8		
800	700		8		
775	725		8		

Table 1: Pulse Width versus Revolution Speed

2. Given timing constraint T, you will obtain V_{in} using equation 2.

$$V_{in} = \frac{r_{in}}{r_{wheel} T} \tag{2}$$

   where V_{in} is the revolution speed of the inward wheel of the robotic vehicle, r_{in} is the radius of the circle along which the inward wheel travels, and T is the timing constraint, i.e., the time that the robotic vehicle needs to finish a whole circle.

   You will place the robotic vehicle at the center of the given circular track and measure the distance between the inward wheel of the vehicle and the center of the circular track.

   You will also measure the distance between the two drive wheels. We denote the distance as $w_{vehicle}$. You can then estimate the desired revolution speed of the outward wheel of the vehicle using equation 3.

$$V_{out} = (1 + \frac{w_{vehicle}}{r_{in}}) V_{in} \tag{3}$$

   Since you have already known the radius of the drive wheels, you will have enough information to fill up Table 2. We denote the radius of given tracks within which the robotic vehicle must run as r_t. We always measure r_t from the center of the path of the track to the center of the circles of the track, i.e., $r_t = r_{in} + r_{vehicle}/2 = r_{out} - r_{vehicle}/2$.

Timing Constraint T (s)	Radius of Circular Track r_t (inches)	Distance between Wheels $r_{vehicle}$ (inches)	Radius of Inner Circle r_{in} (inches)	Radius of Drive Wheel r_{in} (inches)	Inward Wheel Revolution Speed V_{in} (r/s)	Outward Wheel Revolution Speed V_{out} (r/s)
8	12					
12	16					
16	28					

Table 2: Required Revolution Speed of Vehicle Drive Wheels and Timing Constraint

3. Using V_{in} and V_{out}, you can now estimate the required the pulse width values for in Lines 6 and 7 using the model constructed in 1, i.e., Table 1. V_{in} and V_{out} correspond to V_{test} in Table 1. You may use one of the following two methods.

   (a) Read the pulse width values from the graphs.

   You will make two graphs. One shows the relationship between the pulse width value for Pin 13 (i.e., at Line 6) and the revolution speed of the wheel controlled by Pin 13 and the other shows the relationship between the pulse width value for Pin 12 (i.e., at Line 7) and the revolution speed of the wheel controlled by Pin 12. In each of the graph, x-axis or horizontal axis represents the pulse width and y-axis or vertical axis the revolution speed.

   You can now find the required revolution speed from the graph and then read the required pulse width value from each of the graphs.

   (b) Use interpolation.

   In this module, you may use linear interpolation. Assume that the required revolution speed, denoted as $V_{required}$ falls into two adjacent revolution speed values in Table 1. $V_{required}$ corresponds to V_{in} and V_{out} estimated in step 2. We denote them as V_i and V_{i+1}, whose corresponding pulse width values are p_i and p_{i+1}.

   Then estimate the required pulse width value, denoted as $p_{required}$ using equation 4

$$p_{required} = \frac{p_{i+1} - p_i}{V_{i+1} - V_i}(V_{required} - V_i) + p_i \tag{4}$$

   You shall repeat the above estimation for both inward and outward wheels, i.e., based on the estimated V_{in} and V_{out} in step 2, you will obtain p_{in} and p_{out} using equation 4. Then you will fill up Table 3.

4. You will evaluate the technical solution described above using the effort and the accuracy.

   We measure the effort using the number of test runs that have been conducted to find the correct pulse width values.

   The accuracy includes both the accuracy of the circle that the robot follows and the time the robot spent to finish running a circle.

   We measure the former use both absolute error and the relative error. First, we measure the radius of the circular path along which the vehicle travels, then we calculate abosolute error as $e_r = |r_t - r_p|$ and the relative error $e_r\% = |e_r/r_t| = |(r_t - r_p)/r_t|$ where r_p is the measurement of the radius of the circular path along which the vehicle travels and r_t is the radius of the given circular track.

Timing Constraint T (s)	Radius of Circular Track r_t (inches)	Inward Wheel Revolution Speed V_{in} (r/s)	Outward Wheel Revolution Speed V_{out} (r/s)	Pulse Width at Pin 12 (Line 7) p_{in} (μs)	Pulse Width at Pin 13 (line 6) p_{out} (μs)
8	12				
12	16				
16	28				

Table 3: Required Pulse Width Meeting Timing Constraint

We measure the later using both the absolute error and the relative error as well. First, we measure the average time that the vehicle needs to finish a circle using the pulse width values. Denote it as T_c. Denote the timing constraint as T. The abosulte timing error is $|e_T = T_c - T|$ and the relative timing error is $e_T\% = |e_T/T| = |(T_c - T)/T|$. You will also compare it with the tolerable error to see if the error is within the tolerable error.

You will fill up Table 4 using the testing results.

Timing Constraint T (s)	Radius of Circle Track r_t (inches)	Pulse Width for Pin 13 (μs)	Pulse Width for Pin 12 (μs)	Radius of Circle Path (r_p)	e_r	$e_r\%$	Time Spent on a Circle (T_c)	e_T	$e_T\%$
8	12								
12	16								
16	28								

Table 4: Evaluation: Effort and Accuracy

3 Discussion

Where does the model described in equations 1– 3 come from? We discuss it as follows. Let r_{out} be the radius of the circle that the outward wheel travels within the circular track and r_{in} the one that the inward wheel travels. Denote V_{out} and V_{in} as revolution speed of the outward wheel and that of the inward wheel, respectively. Let r_{wheel} be the radius of a wheel, N_{out} be the number of revolutions that the outward wheel rotates to complete a whole circle, and N_{in} be the number of revolutions that the inward heel rotates to complete a whole circle. Then, the distance that the outward wheel travels in a whole circle is $d_{out} = 2\pi r_{out} = 2\pi r_{wheel} N_{out} = 2\pi r_{wheel} V_{out} t$ where $2\pi r_{wheel}$ is the circumference of the wheel and t is the time that the robot vehicle finishes running the whole circle. Similar, for the inward wheel, we have $d_{in} = 2\pi r_{in} = 2\pi r_{wheel} N_{in} = 2\pi r_{wheel} V_{in} t$. Hence,

$$\frac{d_{out}}{d_{in}} = \frac{2\pi r_{out}}{2\pi r_{in}} = \frac{r_{out}}{r_{in}} = \frac{2\pi r_{wheel} V_{out} t}{2\pi r_{wheel} V_{in} t} = \frac{V_{out}}{V_{in}} \tag{5}$$

which yields,

$$V_{out} = \frac{r_{out}}{r_{in}} V_{in} = \frac{r_{in} + w_{vehicle}}{r_{in}} V_{in} = (1 + \frac{w_{vehicle}}{r_{in}}) V_{in} \tag{6}$$

where $w_{vehicle}$ is the distance between the two driving wheels of the robotic vehicle.

We measure the revolution speed of a wheel using "rounds/second" or ("r/s"). Then the revolution speed of the inward wheel can be empirically determined as follows,

$$V_{in} = \frac{N_{test}}{t_{test}} = \frac{\frac{d_{test}}{2\pi r_{in}}}{t_{test}} = \frac{d_{test}}{2\pi r_{in} t_{test}} \tag{7}$$

where N_{test} and d_{test} are the number of revolution and the distance, respectively, that the robot vehicle completes in the test run using time t_{test} and d_{test}.

Then for any given time constraint T, we have

$$V_{in} = \frac{N_{in}}{T} = \frac{\frac{2\pi r_{in}}{2\pi r_{wheel}}}{T} = \frac{r_{in}}{r_{wheel} T} \tag{8}$$

where N_{in} is the number of revolutions that the inward wheel needs to finish running a circle.

4 Submission

The instructor requires you to submit the following items,

- A brief recitation of the problem description and the technical solution;
- Tables 1, 2, 3, and 4 that you have completed.
- The answers to the two evaluation questions, 1) how accurate is the circle? 2) and how many trials have been conducted? For question 1, you will address both the accuracy of the radius of the circular path and the the accuracy of the time spent. For question 2, consider that you are given 1 circular track, 5 circular tracks, and 10 circular tracks and what the effort is.
- Discuss whether or not if the timing constraint is met. What if it is not met?
- Discuss whether or not if the radius of the circular path is sufficiently good. What if it is not?