Lab C: Modeling Robot Sensors & Feedback

Introduction

Single Sentence Overview

Model a SONAR sensor and use it to control the movement of a robot.

Overview

This is a four-part lab in which you will model a standard sensor used on mobile robots: SONAR. Sometimes called ultrasonic sensors, SONARs are used to measure the distance between a robot and objects around it, like the Polar Bear robot does in this video: https://www.youtube.com/watch?v=M4gXGQTkNJc. In this lab you need to:

a) Model a simple SONAR sensor and show that you can capture the time at which the SONAR pulse begins transmission, capture the time at which a returned signal arrives and to calculate the time between those two events. Assuming that you know the speed of sound through air at sea level, this will yield the distance from SONAR to object.
b) Add random noise to the SONAR sensor which would look like shifts in target distance of 0m to 0.1m.
c) Modify the robot model from the previous lab (with motor and battery) so that it can approach a target at 5m in less than 100 seconds. The desired final position of the robot is 5m from its starting position and its desired final speed is 0m/s. It is not allowed to overshoot the target (i.e. to "crash" into it). You are permitted to place the robot between 4.70m and 4.99m. The target is to be detected using SONAR (including noise model). Use a Proportional-Derivative (PD) controller to control the movement of the robot to the target. First, implement the PD control without SONAR, then with SONAR.

Learning Outcomes

The student will know how to

1. Use Simulink to capture the time at which a pulse started
2. Add noise to a simulated sensor signal
3. Adjust the simulation settings to use a "stiff" solver and to set a maximum solver step size that is useful for mobile robots and SONAR
4. Integrate a sensor into a larger model and close a control loop.

Success Criteria

Review the grading rubric and evaluate how it relates to achieving these criteria:

1. Demonstrate a stand-alone model of SONAR without noise
2. Demonstrate a stand-alone model of SONAR with noise
3. Demonstrate that you can control the position of a robot with a simple PD controller

4. Demonstrate that the PD controller can include the SONAR sensor in its feedback loop. Grading Rubric

During this session make sure to conduct all the required **demonstrations** to the lab instructor. Also, submit a **short report** with relevant graphs and screen captures of your models.

- *Part 1 demo (Simple SONAR):*
  - *0 pts: No model produced or no graph produced.*
  - *5 pts: Model and graph produced. Not correct.*
  - *10 pts: Model and graph produced and correct.*
- *Part 2 demo (SONAR with noise):*
  - *0 pts: No model produced or no graph produced.*
  - *5 pts: Model and graph produced. Not correct.*
  - *10 pts: Model and graph produced and correct.*
- *Part 3 demo (PD controller, no SONAR)*
  - *0 pts: No model produced or no graph produced.*
  - *5 pts: Model and graph produced. Not correct.*
  - *10 pts: Model and graph produced and correct.*
- *Part 4 demo (PD controller, with SONAR)*
  - *0 pts: No model produced or no graph produced.*
  - *5 pts: Model and graph produced. Not correct.*
  - *10 pts: Model and graph produced and correct.*

Background

Mobile robots have often used SONAR sensors for navigation, including the Polar Bear robot built by students at the University of Alberta in the 1990s: https://www.youtube.com/watch?v=M4gXGQTkNJc.

Ultrasonic sensors typically have three components: a transmitter, a receiver and a circuit board for regulation of power and signals. Also known as "SONAR" (SOund NAvigation and Ranging) these active sensors employ the "time of flight" method to determine distance to a target. A short burst of sound is transmitted by the sensor's transmitter, the sound travels at the speed of sound to the target and then reflects off of it. If the target has the right characteristics, a portion of this reflecting sound will be detected by the SONAR's receiver. By measuring the time between transmission and reception, and by assuming that the speed of sound is constant, the distance to the target can be calculated.

Your Simulink model needs to do the following:

1. It needs to transmit a continuous stream of pulses
2. It needs to provide a received pulse for each transmitted pulse, proportional to the distance between sensor and target.
3. It needs to be able to record the transmission time, reception time, and to calculate the distance from those values.

The SONAR is assumed to be able to detect objects from 0.1m to 10m, similar to the classic Polaroid 6500 model. In addition, it is assumed to have noise equivalent to varying target distance of 0m to 0.1m.

To implement a PD controller, remember that it is desired to output a voltage into your battery model and that its inputs are the position and the velocity of your robot. The Position error (difference between the measured robot position and its desired final position) is multiplied against the proportional constant. The proportional constant has units of [Volt/m]. Likewise, the Velocity error is the difference between the actual speed and the desired final speed. The Derivative constant has units of [Volt/(m/s)].

Suggestions for simulation

The default simulation settings in Simulink will not provide realistic results with SONAR sensors as the sending and reception of pulsed signals require the use of "stiff" solvers to take into account the rapid transitions in their signals.

Robots tend to require "stiff" solvers during simulations to model two major scenarios: digital / pulsed signals (either electronic or physical like SONAR) and for physical impacts and collisions. In both cases the typical state doesn't change very much, or at all. Then, all of a sudden, a change occurs: the signal changes from "off" to "on" or the robot crashes into an object. The standard ODE45 solver in Simulink doesn't deal with these settings very well. On the other hand, the "s" series of solvers, like ODE23s, do. Furthermore, by changing the maximum step size setting to something in the micro- to milli-second resolution range (for many physical impacts or sensor signals) or nano- to micro-second resolution range (for many electronic signals) can help resolve these changes in a realistic manner without making the simulation time too onerous.

Prerequisites

You are expected to have done the following

- All previous labs, including the battery modeling and motor modelling labs

More Resources and Information

- The classic Polaroid 6500 SONAR sensor
  - Adaptation for robotics : http://www-personal.umich.edu/~johannb/Papers/paper93.pdf
  - Original usage: https://www.youtube.com/watch?v=35y8BRlh-fQ
- The newer Devantech SRF05 SONAR
  - https://www.robotshop.com/ca/fr/module-sonar-srf05-devantech.html

Lab Exercises

The following are the four exercises to work on for this lab.

Model the Basic SONAR

Create a basic SONAR model. **Produce a plot** that illustrates the transmitted and received SONAR pulse, with a phase difference similar to a target distance of 1 to 5 meters.

Model Noise in the SONAR signal.

Add random noise to the signal that adds a delay equivalent to a target distance of 0m to 0.1m (assume 343 m/s speed of sound). Simulink provides at least two blocks for this: a random signal block and a white noise signal block. **Produce a plot showing distance measurements and noise.**

Produce a plot that shows this noise.

Integrate a PD controller in your robot model

Use the difference in final position, difference in final velocity and two constants (proportional and derivative) to create feedback controller to get your robot to a position of 5m with 0m/s final velocity. Don't include the SONAR sensor yet. **Produce a graph to show that the robot approaches the 5m target without overshooting it. Show your model, too.**

Add the SONAR *model to your PD-controlled robot model.*

Take the SONAR model that you developed earlier and use it to sense the distance to the target at 5m. Modify the PD controller to utilise this measured distance value. **Produce a graph to show that the robot approaches the 5m target without overshooting it. Show your model, too.**

Part 4: Wrap-up

Reflection on Learning

Talk to your lab partner and other students in the lab. How will you be able to integrate this 1D model of robot sensing and movement into a 2D or 3D model?

Communication – Reporting

Submit a document with screenshots of your model (including expansions of any subsystems) and relevant graphs.