LAB 5:
Mobile robots -- Modeling, control and tracking

Overview

In this laboratory experiment, a wheeled mobile robot will be used to illustrate

- Modeling
- Independent speed control and steering
- Longitudinal and lateral tracking
- Designing a complex task

Preparation

1. Review what you observed in the ON/OFF robot control experiment and the motor speed control experiment. Your experiences in those experiments will be extensively utilized here.

2. Study the Ziegler-Nichols formulas and their use of the step response.

3. Review the two motor robot transfer function equations discussed in class. Derive them yourself and make sure you understand independent longitudinal speed control and steering control.

Task Description and General Overview

A series of goals are provided below. These are in sequence and it is expected that about half can be accomplished in the first week in the lab.

1. Performing a step-input test to determine an approximate first order model for the robot motor.
2. Using the Ziegler-Nichols formulas to generate an initial design for a PI controller for the motor.
3. Implementing the designed speed controller on the two motors of the robot and making necessary adjustments to controller parameters.
4. Following a given speed profile.

5. Designing a basic steering controller for the robot.
6. Stopping at an obstacle (with forward sensor measurements).
7. Going around an obstacle (designing a complex operation).

Model Determination

We are going to assume that the motors of the robot (input to speed) can be represented by first order dynamics and a small delay. This implies that we can model the system with three parameters: the steady state gain, the time constant and the delay.

Design a step-input test so as to measure the above parameters. You may wish to perform the test on each individual motor on the robot to make sure that they are indeed the same.

PI Controller Design Using Ziegler-Nichols

Design a PI controller for robot speed-control using the Ziegler-Nichols formulas provided in class, and test it in simulation (using Matlab). The Ziegler-Nichols formulas can be found in most introductory control textbooks.

Obtain discrete versions of the gains and implement on the robot. Adjust if necessary. Obtain data and compare (in your report) to your simulation results. The formulas needed for the conversion were provided in class and can be found in many textbooks, such as Discrete Time Control Systems by Katsuhiko Ogata on page 116.

Following a Speed Profile and Tracking a Virtual Target

Test your robot with a trapezoidal (ramp up- constant- ramp down) speed profile.

Structure your software so that the speed profile can be provided in real-time separately. (We want to lead to a situation where the robot follows a moving target, and we can measure or calculate the target speed.)

Create and demonstrate the following scenario:

1. Your robot is moving at a given constant speed.
2. It encounters a slower robot ahead. (This will be our "virtual" target, with software provided absolute speed.)
3. Calculate the relative speed and slow down to match speeds.
4. Make your virtual target slow down and demonstrate that your robot will also slow down as it follows.

Steering Control

Steering control with two motors has been discussed in class and also considered in the ON/OFF control experiment.

Design a simple controller for rotational motion. (Accomplishment of rotational motion is the basic step in steering, but is not fully steering yet. To do true steering, we have to have longitudinal motion too.)

Demonstrate your rotational motion by trying a few given angles.

Complete your full steering controller where you have two separate controllers for speed control and angular orientation. (This was discussed in class as a decoupled controller.)

Follow the wall. (the distance needs to be within range of your side-sensors.)

Structure your software so that you can add different biases to your distance measurement from the wall. Demonstrate this by gradually steering away from the wall, then gradually steering back to it.

Obstacle Sensing and Avoidance

We have two goals in this last task(s).

The first is recognizing an object in front of our robot. The robot will be moving, the object/target will not. The robot has to recognize this fact and stop its forward motion. (Your speed matching "module" should accomplish this.)

You will be dealing with a sensor, looking forward. You should exercise this sensor and know its limitations.

The second goal is to develop software for obstacle avoidance, which will get the robot to complete a trajectory around the fixed obstacle. You can do this with stops and starts, but it is preferable to accomplish obstacle avoidance with a smooth trajectory (if your sensors and timing budget allows it.)

At this point your software should be getting fairly complicated! Make sure you develop and provide a clean flow chart, especially explaining the relationship between software modules accomplishing different **tasks.** Specifically you should have modules for:

- sensing wall
- sensing object ahead
- speed control
- tracking
- decision making for stopping or obstacle avoidance
- turn
- .
- .
- etc.

Your over-all goal of following the wall and avoiding an obstacle can be accomplished with a two-level (or more) hierarchical structure. Make sure you follow a clean logical process.

Analyze and explain smooth obstacle avoidance in detail in your report.

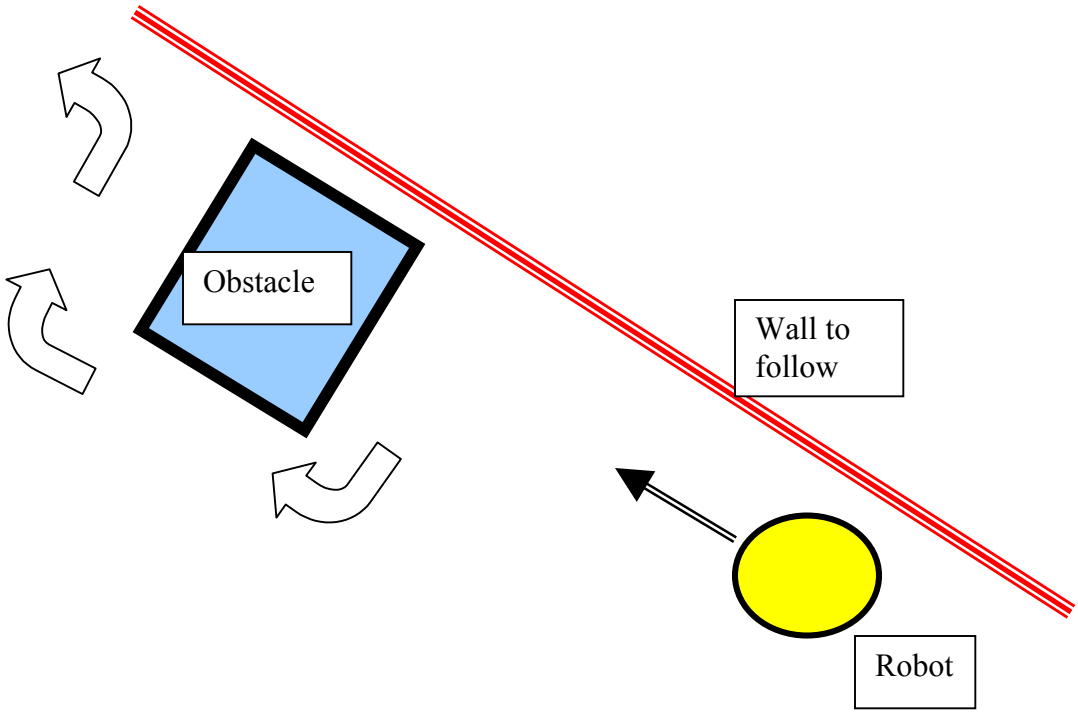

Figure 1. Wall following and obstacle avoidance.

Experiment Procedure

1- Perform a step-input test to determine an approximate first order model for each robot motor. Design a PI controller for robot speed-control using the Ziegler-Nichols formulas provided in class, and test it in simulation (using Matlab).

2- Implement the designed speed controller on the two motors of the robot and make necessary adjustments to controller parameters. Obtain a reasonable response and explain it in your report. (See "tracking a virtual target" explained above.

3- Design a basic steering controller for the robot. Demonstrate your rotational motion by trying a few given angles. Obtain a reasonable response and explain it in your report.

4- Complete your full steering controller where you have two separate controllers for speed control and angular orientation. Select a reference angle and keep the mobile platform moving in that direction until an obstacle is detected.

5- Write a program to enable the mobile platform to perform the task shown in Figure 1. Verify the operation of your software by giving the trajectory, followed by the mobile platform, in your report.