CS 7639 O01 Project 1: Robotarium and Feedback Control

The Robotarium is an open access, remotely accessible robotics testbed housed at Georgia Tech, which was designed with the purpose of enabling robotics research by providing users with free, remote access to state of the art hardware and infrastructure. It consists of a large planar arena, occupied by dozens of small differential drive robots, called GRITSBots. After prototyping their algorithms in the provided simulator, users take control of the robots by submitting their code online. The Robotarium web site is www.robotarium.gatech.edu.

The purpose of this first project is to help you learn the basics of how to operate individual robots with various levels of accuracy, and to stress the knowledge you have acquired during the first few lectures (from introduction to low-level/mid-level designs).

The GRITSBot is controlled using *unicycle dynamics*,

$$u = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{x}_B \\ \dot{\theta} \end{bmatrix}$$

where θ is the angle of the *body frame* of the robot with respect to the frame defined on the testbed. The units of the velocity inputs v and ω are m/s and rad/s respectively.

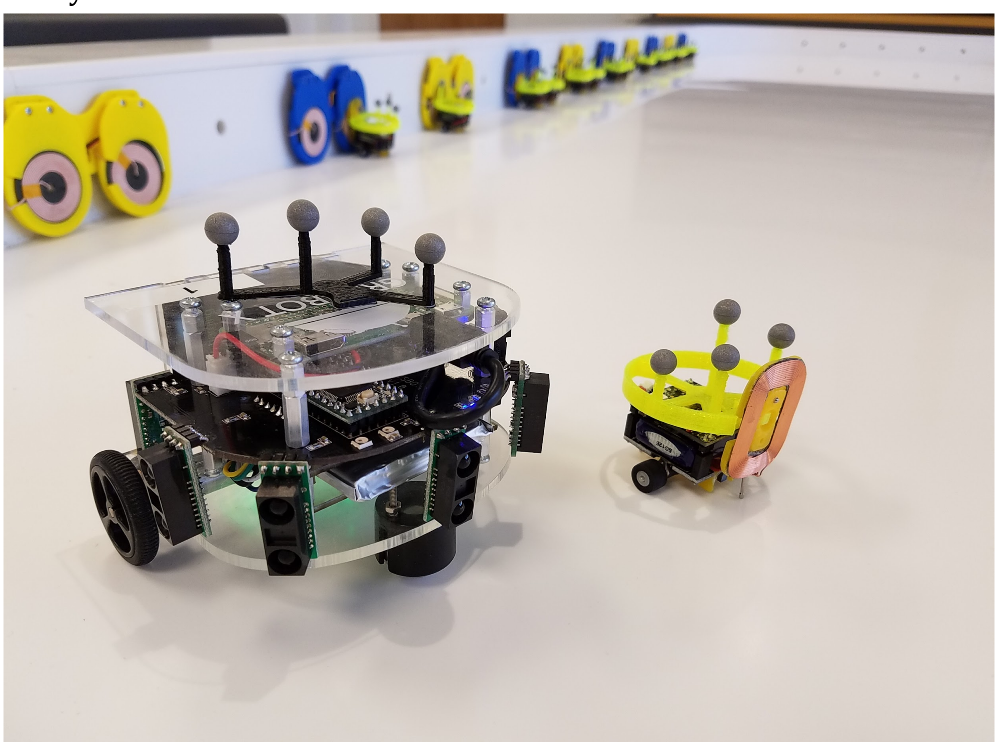


Figure 1. The GRITSBot X (the bigger one on the left).

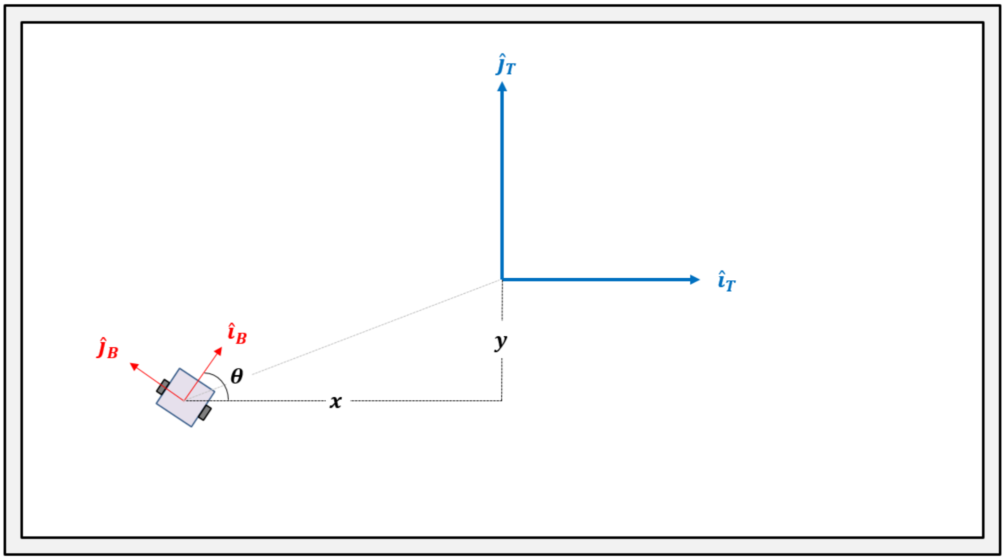


Figure 2. Drawing of the arena.

An overhead sensor is used to return *pose* data for the robot,

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}.$$

The position coordinates (x, y) are given in meters and θ is in radians, with a range of [-pi, pi].

The Robotarium Matlab simulator, along with more information on the GRITSBot X and testbed can be found on the Robotarium website (https://www.robotarium.gatech.edu/faq).

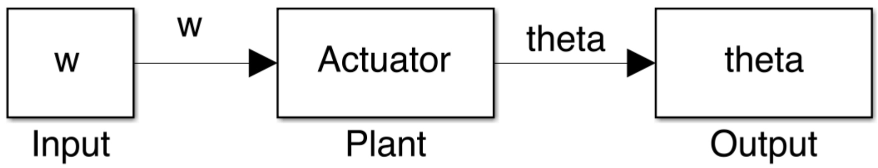
Modeling the Kinematics of the Robots

1. Learn how to perform basic operations on the robots (e.g. move from base to a point with a given orientation, return to a base point etc.) on the Robotarium simulation.
   a. Find out the equation describing the relationship between the **rpm** of the wheels, ω_{wheel}, and the forward velocity of the robot, v, with the given parameters. The radius of the wheels is 1.5cm.
   b. Find out the relationship between the translational velocity of the wheels, v_{wheel}, and the angular velocity of the robot, ω, assuming the wheels are rotating in opposite directions and with the same speed.

2. Describe the feedback control loop for controlling the orientation, θ, of the robots along a path used in the Robotarium **using a block diagram**. That is, a desired orientation is commanded and thus should be your input. Make sure to start with the input and end with the output in order to have a complete diagram.

   This could be helpful: https://en.wikipedia.org/wiki/Control_theory

   Make sure to label each block as sensor/controller/actuator and label on each line what variable is passed to the next block. An open loop block diagram of the robot is like this:



3. Give brief descriptions of the sensor, actuator and controller. What do they do? Relate these components to the actual relevant functions in the Matlab code.

Simulation Activities

Note: Do **NOT** use any pre-defined functions in the Robotarium Matlab simulator (/examples and /utilities).

Prepare a Matlab script for your robot to do the following list of tasks:

1. Open the file, *Template_Project_1.m,* which contains two 3x1 pose vectors (i.e. $[x; y; \theta]$) with the names *target1* and *target2*.

2. Command one robot to *target1*.

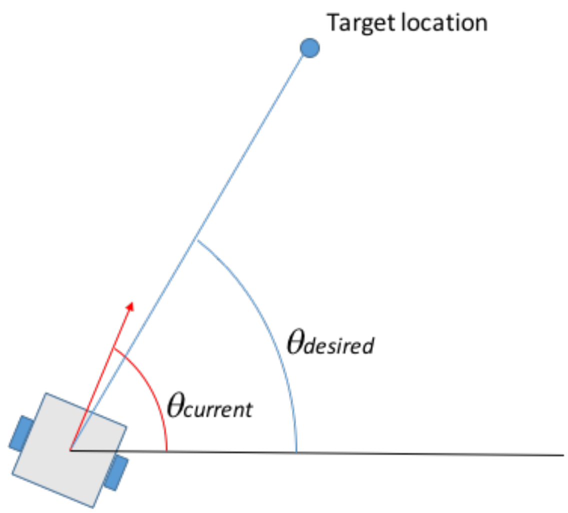
3. Once Task 2 is completed, command the robot to *target2*, with constant velocity 8 cm/s and *no angular velocity* (open loop control). This can be done by first pointing the robot to a desired direction at its current location, and then calculating the **number of iterations needed** to reach *target2* at the given speed. The controller is running at 30 Hz (i.e. one iteration corresponds to about 0.033 seconds).

4. Command the robot to *target1*.

5. Command the robot to *target2* with constant 8 cm/s and a feedback strategy where the angular velocity is adjusted (closed loop control) based on the angular deviation from the direction to the target position. Stop the experiment when the robot reaches the target position, within 1 cm and 0.1 radian accuracy. The feedback strategy can be expressed like the following:

$$\omega = k(\theta_{desired} - \theta_{current}),$$

where k is a gain value of 1.



6. Command the robot to *target1*.

Submit Your Experiment

Once you are satisfied with the simulation, submit the script, Template_Project_1.m, to the Robotarium through your account. Data will be accessible in several days.

Data Analysis

1. Plot the experiment position data of Task 3 and Task 5 on the same Y vs X plot.
2. Plot the distance to the target position of Task 3 and Task 5 against the number of iterations (normalized from 0) on the same plot.
3. Discuss and **explain** what you see in the plots.

Report and Grading

All the tasks are accomplished by the robot. **(50 pts)**

In your report, please do the following:
1. Answer the 3 questions in the Modeling section. **(30 pts)**
2. Indicate the number of iterations needed in Simulation Task 3. **(5 pts)**
3. Include 2 plots and the discussion from the Data Analysis section. **(15 pts)**

Deadline

1. A single .m file on Canvas and Robotarium **by 3:00 pm EST on 1/27**. This is the first attempt to submit your code so that it can be tested on the Robotarium. Experiment data will also be returned after the run.
2. A single .m file on Canvas and Robotarium **by 3:00 pm EST on 2/3**. This is the **optional** second attempt in case the first run fails for any reason.
3. A final report (pdf) on Canvas **by 3:00 pm EST on 2/10**.

Matlab Installation

You can download Matlab from http://matlab.gatech.edu/.