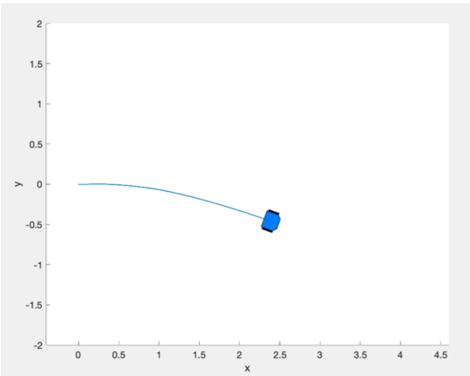
Coursework 1

*Your main coursework submission is a text document (docx or pdf). **Provide a text answer for all questions, explaining your reasoning**. In addition, you will be asked to produce Matlab plots that you should add as figures in the submitted text document, and Matlab scripts that you should submit as separate .m files.*

A simulator of a differential drive robot with 2 DC motors and wheel encoders is provided with this coursework assignment.

Open the script `DifferentialDriveDCMotorRobot.m` in Matlab and run it. Each time this is run, a robot with a slightly different behaviour is generated. You should see a plot similar to this:



These are the **known constant** parameters in the simulation:

- The motor encoders have 64 ticks per complete wheel revolution
- The DC motors operate between -6V and +6V
- The robot's initial position is the origin (0,0) and its initial orientation is 0 rad.

These are **unknown but constant** parameters in the simulation:

- The radius of each robot wheel (they are both the same)
- The distance between the two wheels

There are also some unknown parameters that can change with each simulation run.

`DifferentialDriveDCMotor.m` has two placeholders for implementing a custom controller: an initialisation routine and a script to run on every robot cycle.

```
%%%%%%%%%%%%%%%%%%%%%%%% IMPLEMENT THIS SCRIPT %%%%%%%%%%%%%%%%%%%%%%%%
init_question_x;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%% IMPLEMENT THIS SCRIPT %%%%%%%%%%%%%%%%%%%%%%%%
controller_question_x;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

These scripts are defined in two separate files `init_question_x.m` and `controller_question_x.m` which in the unaltered code just assigns a constant input for the left and right DC motors:

```
u = [4;4];
```

When `DifferentialDriveDCMotor.m` is run with the unaltered script `cw1_question_x.m`, the control input is the same (4V) for both wheels, however, the robot trajectory is not a straight line.

After a simulation is run, the variable `csim.Log.Output` contains a 5xN matrix Log, which each row containing the following data (for each sampled time):

- Robot orientation
- Robot horizontal position
- Robot vertical position
- Encoder 1
- Encoder 2

PS: Similar logs exist for the robot states and inputs and you can explore the stored variables `csim.Log.State` `csim.Log.Input`, but these are less relevant to the coursework.

Question 1 [5 marks]

Assume that the DC motor of each wheel is simulated with the LTI model presented in the lectures.

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} \frac{di}{dt} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_e}{L} \\ \frac{K_m}{J} & -\frac{b}{J} \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} u(t)$$

When the unaltered simulation script is run the input for each DC motor is the same, but the left and right wheels have different speeds. Explain and justify which parameters in this LTI model could be causing this.

Question 2 [10 marks]

Implement a controller (`init_question_2.mat ,controller_question_2.mat)` that makes the robot move in a straight line. The controller should produce a 2D vector `u` that sends a voltage input to each DC motor. Assume that you only have access to the robot odometry measurements which are stored and updated in the Matlab 2D vector `y`.

Question 3 [15 marks]

Implement a controller (`init_question_3.mat ,controller_question_3.mat)` to move the robot to a desired position in 2D cartesian coordinates (which can be changed by the user). This should be implemented with a single feedback control loop (**no separate tasks** for “turning” and “moving straight”). Once again, only the odometry measurements are accessible to the controller.

Question 4 [5 marks]

Implement a controller (`init_question_4.mat ,controller_question_4.mat)` to move the robot to a desired position (which can be changed by the user) and come back to the origin (0,0) without stopping. Once again, only the odometry measurements are accessible to the controller.

Question 5 [15 marks]
Consider a line in cartesian space $p_y = mp_x + b$ (which can be changed by the user). Implement a controller (`init_question_5.mat ,controller_question_5.mat`)so that the robot moves along that line with increasing p_x (from left to right in the simulator). Once again, only the odometry measurements are accessible to the controller.

Question 6 [15 marks]
Consider the controllers implemented in questions 2, 3, 4, 5. For each of them, run the simulator and using `csim.Log.Output produce a` plot with the following trajectories: (1) the desired robot trajectory, (2) the robot trajectory estimated from the odometry measurements (3) the true robot trajectory.

Question 7 [45 marks]

Implement your own version of a Differential Drive Robot model with 2 equal DC Motors (same parameters) using a state-space representation. You do not need to implement a graphical interface, but you should produce a Log similar to `csim.Log.Output` so that the robot trajectories can be plotted after the model is run.

Run your model during 20 seconds, with a sampling time of 0.02 seconds, with the following robot inputs:

- Constant, equal input voltage to both motors
- Left motor voltage is a constant V_L, right motor voltage is V_R = 2*V_L
- Left motor voltage is a constant V_L, right motor voltage is: V_R = V_L+V_L*sin(t)

For each of them, plot the robot trajectory