AMME5520 Major Project and Lab Report

Ian R. Manchester and Max Revay

The project is worth 20% of your final grade. You may work with other students to understand the problems, and to discuss and compare different solution methods, but each submission must reflect your personal understanding of the problems and your own implementation of solutions.

The assignment report is to be submitted as a on TurnItIn by **Sunday of Week 13 (11th of June) at 11:59pm**. This assignment should take an average student 35-40 hours to complete.

The lab report is worth 10% and is to be submitted by **Sunday of Week 11 (28th of May) at 11:59pm**

1 Major Project

The first part of this assignment builds upon your work in Assignment 1. Now you must integrate the motion planning and control modules, along with a PRM and state estimation, to create a complete system that guides a ball balancing robot through an obstacle field. The dynamical model for the robot is outline in appendix A. **Note that there has been an addendum to the dynamics in the previous assignment**. Marks are listed out of 100.

Download the file **Project.zip** from Blackboard. This contains a function that generates a random obstacle field, and stubs of functions for you to modify to achieve your goal. You can adjust various parameters, including the number of obstacles. With other settings at default, you could take 10, 20, 30, and 40 obstacles as Easy, Medium, Hard, and Extreme difficulty levels.

The following is a suggested outline of the different components. You may vary this if you choose.

1. (5 marks) Write a collision checking function that answers the following question: Does a line segment between two points x_1 and x_2 come within distance d of an ellipse defined by the inequality $(x - c)^T A(x - c) \leq 1$. This should go in `CheckCollision.m`.

2. (30 marks) Write a PRM combined with Dijkstra's algorithm or $A^\star$ to find an approximately-optimal collision-free path, in terms of distance travelled. Hint: you may use the matlab function `knnsearch` to select candidate connection points.

   Make sure to allow an appropriate "buffer zone" around each obstacle, in case your path tracking is not perfect (there are various more or less precise ways to do this).

   This should go in `ComputePath.m`.

3. (40 marks) Design a state-feedback LQR controller to regulate the vehicle to your planned trajectory from Question 1. Describe how you linearised about

the trajectory, and motivate your choice of controller weights, and evaluate the resulting tracking performance.

Remark on the interaction between tracking performance and the design of the "buffer region" for the trajectory planning. Evaluate performance of the complete system for a variety of numbers and sizes of obstacles.

This should go in `ComputeControl.m`.

4. (20 marks) Assume now that the measurements available to the ball balancing robot include noisy values of: linear acceleration of the ball, θ_x, θ_y from encoders and bearing measurements of the obstacles relative to the robot. Use realistic values for a low-cost IMU. You may assume that the robot has prior knowledge of the locations of 4 obstacles. Construct an output-feedback controller that combines state estimation and state feedback.

   This can be done by modifying the `meas.m` function to change the measurements available, and also modifying ComputeControl.m to include a state estimator (you may want to use the `params` variable to keep track of the state estimate).

5. (10 marks) The dynamic models (2) and (3) assume that the robot is travelling along flat ground. We can introduce the effects of bumpy terrain as a disturbance acting on the states $\ddot{x}$ and $\ddot{y}$ as follows

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B(\tau) + D(q) \qquad (1)$$
$$D(q) = [-m_b g h', 0]^T$$

   where $h(x,y)$ is the terrain height and h' is the terrain gradient; $\frac{\partial h}{\partial x}$ in equation (3) and $\frac{\partial h}{\partial y}$ in equation (2). You have been provided with a function that will randomly generate terrain and it's gradient.

   Design a controller that will minimize the effect of the terrain on the tracking error of the robot.

6. **(Up to 15 Bonus marks)** Simultaneous Localization and Mapping. Suppose now that the map available to the robot has only approximate obstacle locations. Construct an extended Kalman filter that uses the bearing measurements to refine the map and navigate through. Investigate the effects of errors in the angles, e.g. due to camera calibration.

Your submission should be a professional-grade report justifying the design decisions you have made with a thorough analysis of the results.

You must also separately upload all matlab files you produced for this assignment. Key elements of the code can be quoted in the assignment for explanation purposes if desired.

2 Lab Report

For each pair of students working on the lab, submit one short report (max 5 pages) on your findings for the laboratory questions. This should discuss your derivation of the linearised model, motivate your decision in control design, and evaluation of different controllers.

A Robot Dynamics

At low speed the dynamics of the ball balancing robot can be described by two decoupled equations in the x-z and y-z planes. These equations of motion are similar to those in the previous assignment, however, they describe the system in 3 dimensional space. Assuming that the system does not rotate around the z axis, there are 4 degrees of freedom with (x, y) describing the position of the center of the ball and (θ_x, θ_y) describing the rotations about the x and y axes respectively from the vertical.

The system parameters remain the same from the previous assignment. These are provided in table 1 for reference.

Dynamics In Y-Z Plane

The motion in the y-z plane are described by the following equation

$$M(q_x)\ddot{q}_x + C(q_x, \dot{q}_x)\dot{q}_x + G(q_x) = B_x(\tau_x) \tag{2}$$

with the vector $q_x = [y, \theta_x]^T$ and matrices defined below

$$M(q_x) = \begin{pmatrix} m_b + \frac{I_b}{r_b^2} + m_h & -m_h r_h \cos\theta_x \\ -m_h r_h \cos\theta_x & m_h r_h^2 + I_h \end{pmatrix}, \quad C(q, \dot{q}) = \begin{pmatrix} 0 & m_h r_h \dot{\theta}_x \sin\theta_x \\ 0 & 0 \end{pmatrix}$$

$$G(q_x) = [0, -m_h r_h g \sin\theta_x]^T, \quad B_x(\tau_x) = [\tau_x, r_b\tau_x]^T$$

Dynamics In X-Z Plane

The dynamics in the x-z plane take a similar form.

$$M(q_y)\ddot{q}_y + C(q_y, \dot{q}_y)\dot{q}_y + G(q_y) = B_y(\tau_y) \tag{3}$$

Where the vector $q_y := [x, \theta_y]^T$, and the matrices are defined as

$$M(q_y) = \begin{pmatrix} m_b + \frac{I_b}{r_b^2} + m_h & -m_h r_h \cos\theta_y \\ -m_h r_h \cos\theta_y & m_h r_h^2 + I_h \end{pmatrix}, \quad C(q, \dot{q}) = \begin{pmatrix} 0 & m_h r_h \dot{\theta}_y \sin\theta_y \\ 0 & 0 \end{pmatrix}$$

$$G(q_y) = [0, -m_h r_h g \sin\theta_y]^T, \quad B_y(\tau_y) = [\tau_y, r_b\tau_y]^T$$

Symbol	value	units	definition
I_h	0.0256	$kg \cdot m^2$	Moment of inertia of head around it's center of mass.
m_h	3	kg	Mass of Head.
r_h	0.33	m	Distance between the center of mass of the ball and the head.
I_b	0.25	$kg \cdot m^2$	Moment of inertia of body around center of mass.
m_b	6	kg	Mass of body.
r_b	0.25	m	Radius of Body.

Table 1: Constant values and definitions.

Motor Dynamics

The motors have the same dynamics as in the previous assignment.

$$\frac{T(s)}{R(s)} = \frac{2}{s+2}$$