

**MSE 450**  
**Project Proposal**  
**Real-Time Embedded Control System**  
**for an Inverted Pendulum**

Self Proposed Project Group 1  
Rachel George - 301288581  
Ryan Fielding - 301284210

March 9<sup>th</sup>, 2020

# Abstract

This report outlines the project proposal for the MSE 450 (Real-time and Embedded Control Systems) course project, which will be developed in conjunction with the MSE 483 (Modern Control Systems) course project. In short, the system will implement a state space model feedback control system via the course Tiva C TM4C123GH6PM microcontroller, with the goal of stabilizing an inverted pendulum across various small impulses. This inverted pendulum system will be built from the parts of a broken Canon printer donated by a past high school teacher. The main components taken from this printer include the printer cartridge track (for the cart), a DC motor, linear optical encoder. The pendulum will be mounted to a potentiometer to measure the angle of the pendulum, which will be mounted to the cart, whose position will be measured (inc/decremented) by the optical encoder. To effectively implement this system, the Tiva C microcontroller initialization must be completed early on, in order to allow for state model controller testing and to ensure a stabilized pendulum by the project deadline. The initialization will consist of:

- 2 digital inputs for the optical quadrature encoder outputs, A and B, which will increment and decrement the cart position via interrupts.
- 1 analog input for the angle of the pendulum via a  $10k\Omega$  potentiometer.
- 2 PWM outputs for the actuation and control of the 12V, 1A DC motor through an H-bridge (TA7267BP) interface board, connected to the cart via a belt drive.

Initialization of the microcontroller and all working components will be completed by March 15<sup>th</sup>, 2020, to allow enough time for state model development and testing. This includes but is not limited to, measurement of  $\theta$  and  $x$ , ensuring that the position increments properly at high speeds, and adequate position control of the DC motor powered cart. At this point, the observer based state feedback controller can be implemented. The final control system will be complete and tested by March 30<sup>th</sup>, 2020.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Overview . . . . .	4
1.2	Control System Background . . . . .	4
1.3	Problem Investigation . . . . .	5
1.4	Expected Outcome(s) . . . . .	5
<b>2</b>	<b>Design Goals and Objectives</b>	<b>6</b>
2.1	Objectives . . . . .	6
2.2	Nice to Have's . . . . .	6
<b>3</b>	<b>Requirements and Methods</b>	<b>7</b>
3.1	Control Model . . . . .	7
3.2	Experimentation . . . . .	8
3.3	Requirements . . . . .	8
<b>4</b>	<b>Platform and Tools</b>	<b>8</b>
4.1	Hardware . . . . .	8
4.2	Software . . . . .	10
<b>5</b>	<b>Implementation Plan and Timeline</b>	<b>10</b>
5.1	Timeline . . . . .	10
5.2	Software Implementation Approach . . . . .	11
5.2.1	Concerns . . . . .	11
<b>6</b>	<b>Conclusions</b>	<b>12</b>

# List of Figures

1	Open Loop System Versus a Closed Loop System [1] . . . . .	4
2	System Measurement - System Response to Disturbances [1] . . . . .	5
3	System Concept - Inverted Pendulum on a Cart [2] . . . . .	6
4	Flow Chart - Predicted State Feedback Model to be Implemented [3] . . . . .	7
5	DC Motor Belt and Feedback . . . . .	9
6	System Setup . . . . .	9
7	Custom Built Low Friction Linear Guided Cart . . . . .	10
8	Quadrature Encoder Outputs [4] . . . . .	11

# 1 Introduction

## 1.1 Overview

The system discussed from a controls perspective, involves the concept of an inverted pendulum; an unstable system which if otherwise not intervened with by engineered calculated actuation, will become unbalanced due to gravity and the displacement of the pendulum's centre of gravity from a set point. This experiment encapsulates a true real-time system which integrates an embedded computer system to react within stringent response-time constraints.

Physically the system will be built of reused spare parts from a broken Canon printer, donated from one of the team member's previous high school teachers. Parts salvaged include a linear guide rail, a 12V, 1A DC motor and a linear optical encoder. A basic potentiometer will be used to measure the angle of the pendulum. The platform chosen to control this system is the Tiva TM4C123GH6PM microcontroller. It will be integrated with an H-bridge motor interface board to drive the DC motor, and the outlined sensors. Software development will be done in Code Composer Studio written in C.

## 1.2 Control System Background

The controller's commands will be dependent on the feedback of the state system; this ultimately entails a closed loop system. As exemplified below (Figure 1), it is the difference between a closed loop system and an open loop system. Simply put, an open loop system has no indication of its perpetuated error that it may be incurring, therefore feedback loops or closed loop systems can respond accordingly to disturbances that deviate the output from the target value.

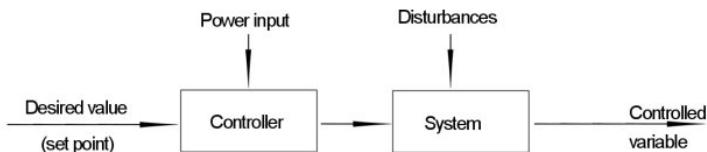


Figure 1A

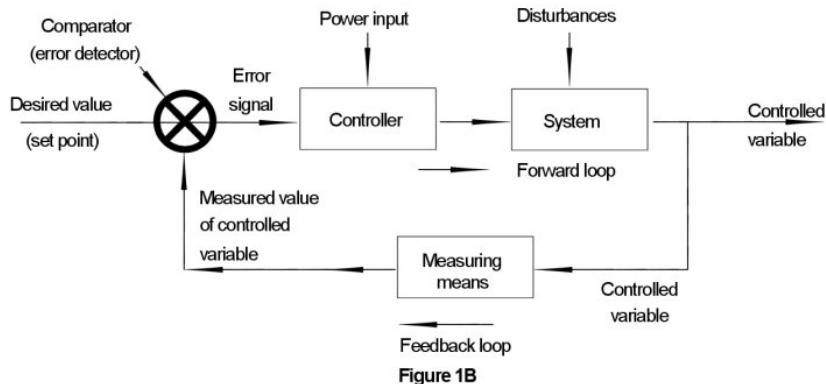


Figure 1B

Figure 1: Open Loop System Versus a Closed Loop System [1]

### 1.3 Problem Investigation

The measure of a successful closed-loop system is merited by its degree to which the measured final value compares to its target value after the system responds to a disturbance. This success can be measured by multiple characteristics of the system response. One such characteristic is the maximum error, a value which corresponds to the initial overshoot of the system response [1]. Another valuable criteria is the response time and the settling time of the system; the time from which an error is detected to when a corrective action is taken, and the time taken for the controller to complete its corrective path, respectively. After the final value is obtained, the steady state error is observable - the absolute difference of the final measured value and the ideal value, see the figure below for this visual representation (Figure 2).

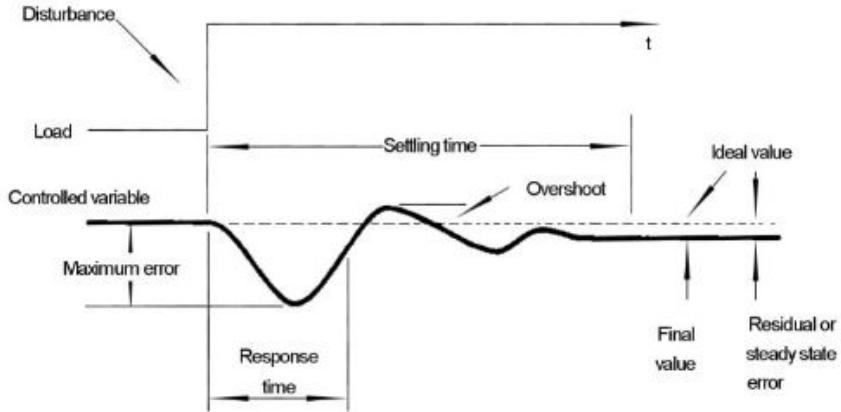


Figure 2: System Measurement - System Response to Disturbances [1]

State Space Modelling will be implemented as the chosen model, ergo using the "notations of state, inputs, outputs and dynamics to describe the behavior of a system" [5]. From the perspective of an unstable system, state modelling will allow us to predict the behavior of the system. Partner this with real time feedback, and one is able to implement a state feedback controller to achieve a desired objective. In our case, stabilization of an inverted pendulum.

### 1.4 Expected Outcome(s)

We expect to obtain a system response and from observing its characteristics previously explained, we plan to manipulate the system to minimize the steady state error. This error responds to a system gain, however it is important to balance the increase of gain to the system as it impacts the sensitivity of the response and may additionally result in the increase of the maximum error and/or settling time [1]. A critically damped system with a medium gain is most desirable. This can be observed through an ideal physical system producing minimum oscillation about the set point to counteract the moment created by the movement of the pendulum's center of mass. From this experiment we also aim to form a solidified concept of embedded computer control system programming; designing and building a system, integrating hardware and software for an embedded control application.

## 2 Design Goals and Objectives

### 2.1 Objectives

The goal of this controller aims to maintain the pendulum in a state of inversion, that is, an angle of  $\theta = 180^\circ$ , constantly and for various impulse forces to the pendulum, by changing the position ( $x$ ) of the cart. See figure 3 below.

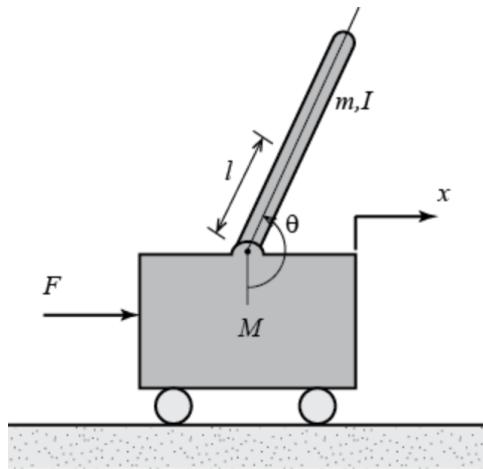


Figure 3: System Concept - Inverted Pendulum on a Cart [2]

#### Variables

- $x$  the position of the cart
- $\theta$  the angle of the pendulum
- $F$  applied forces to the cart
- $M, m$  masses of the cart, pendulum
- $l, I$  length and inertia of the pendulum

### 2.2 Nice to Have's

In addition to the objectives outlined in the previous section, there are many sub-features that can be developed in the software of this system that present complex control theory problems, 2 of which are outlined below.

- Side-stepping of cart to a desired position ensuring constant pendulum stability.
  - For example, move cart from  $x = 0\text{cm}$  to  $x = 10\text{cm}$  with minimal overshoot, rise time, steady state error.
- Swing-up and swing-down commands
  - Enables the system to move the cart from one state of stability to another, ie. from  $\theta = 0^\circ$  to  $\theta = 180^\circ$  and stabilize, or vice versa with minimum oscillation.

### 3 Requirements and Methods

#### 3.1 Control Model

MSE 483 - Modern Control Systems / Intro to State Space Control Systems - effectively outlines the process of implementing a state space model based control system. Physical systems possess many non-linearities and thus more practical models and tuning will need to be implemented. Such as the problem of the nonlinear control of an inverted pendulum, which sees to balance the pendulum from an unstable equilibrium position. Such controls methods include Linear Quadratic Regulation (LQR), observer based state feedback control and more. The following flowchart displays the proposed observer based state feedback control model, implementing an observer due to the fact that the system does not provide full state feedback, which would require the purchase of 2 tachometers to measure the rate of change of  $\theta, x$ , that is,  $\dot{\theta}, \dot{x}$ . However, we will be implementing a state estimator to estimate the values of these states instead.

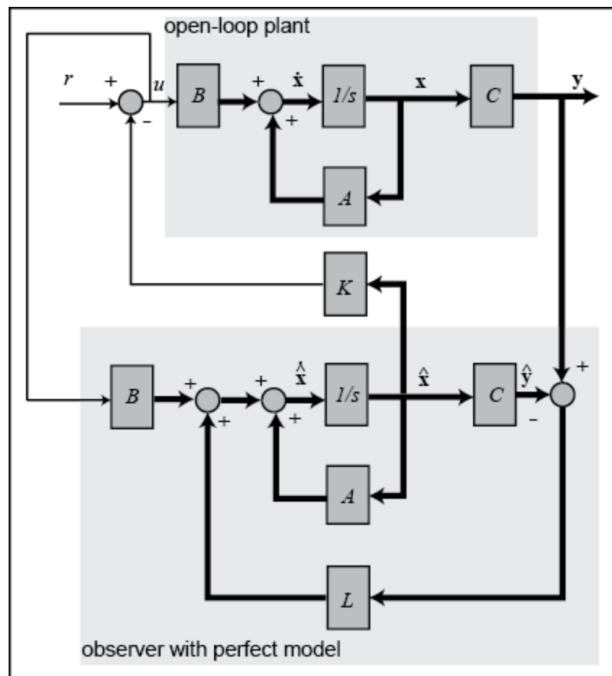


Figure 4: Flow Chart - Predicted State Feedback Model to be Implemented [3]

The above control system will not be elaborated too in depth as it is out of the scope of this report, however it presents the overall concept. The most important variables and what they represent are discussed in the following:

- $x$  the states of the system -  $x, \dot{x}, \theta, \dot{\theta}$ , measured and estimated through observer based state feedback.
- $y$  the demanded outputs of the controller,  $x, \theta$
- $u$  the inputs to the system, which would be the force to the cart,  $F$ .

- All other matrices,  $A, B, C, D$  and so forth, represent the complete state model for the linearized system, full form shown below:

$$\dot{x} = Ax + Bu \text{ and } y = Cx + Du \quad (1)$$

And for the state estimator:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \text{ and } \dot{e} = \dot{x} - \dot{\hat{x}} = \dots \implies \dot{e} = (A - LC)e \quad (2)$$

## 3.2 Experimentation

To evaluate the proposed control models, experimentation will roughly proceed as follows:

1. Power on the microcontroller with latest code developments.
2. Normalize the cart position and pendulum angle (move cart to center of guide rail, hold pendulum vertically).
3. Release the pendulum and evaluate controller effectiveness / stability.
4. Introduce minor disturbances to the pendulum and the cart, evaluate system response.

## 3.3 Requirements

The requirements for this problem are short and sweet - to balance the pendulum vertically as opposed to letting it fall over due to gravity. To elaborate on these requirements, we can set the following parameters:

- $\Delta\theta < 1^\circ$  for small impulses to the pendulum or cart.
- Settling time of  $< 4$  seconds.
- Cart position must be settled consistently at a constant location, ie.  $x = 0cm$ .

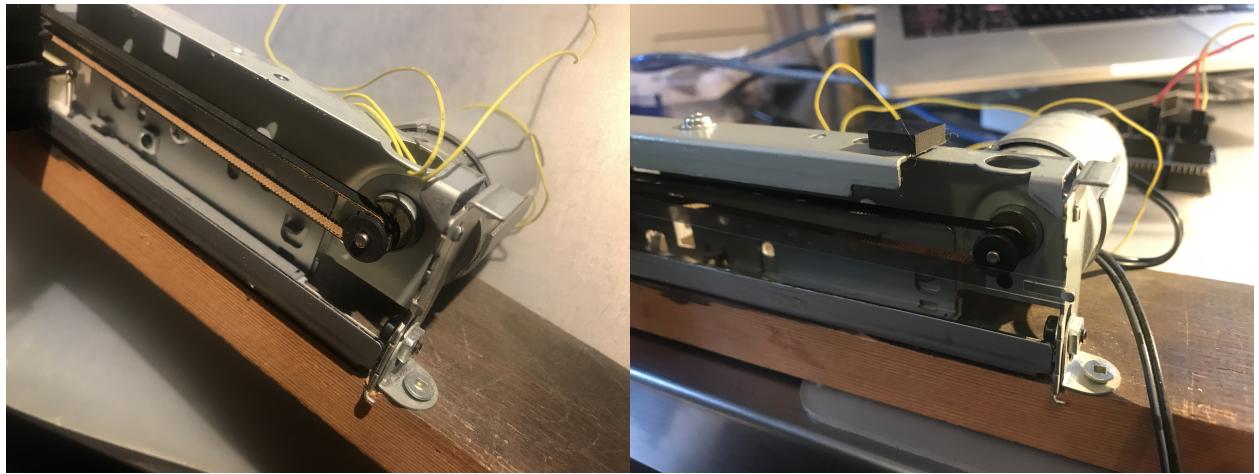
Further requirements may be set for the Nice to Have's outlined in section 2.2, depending on the project's completion schedule. Assuming all prior requirements are complete, we will investigate said problems and construct design requirements.

# 4 Platform and Tools

## 4.1 Hardware

The Tiva TM4C123GH6PM microcontroller from Texas Instruments will be used as the real-time embedded controller. Code Composer Studio 9.3.0 for Mac OS X will be used to program and debug the board.

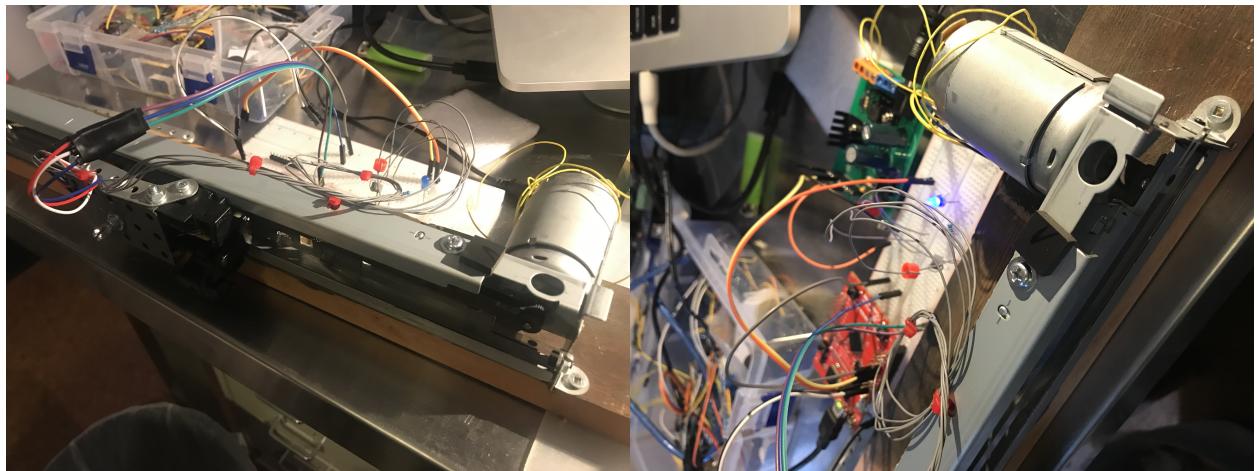
As for the hardware of the inverted pendulum system, the re-purposed Canon printer can be seen throughout the following figures.



(a) DC Motor Belt Drive

(b) Optical Encoder Linear Strip

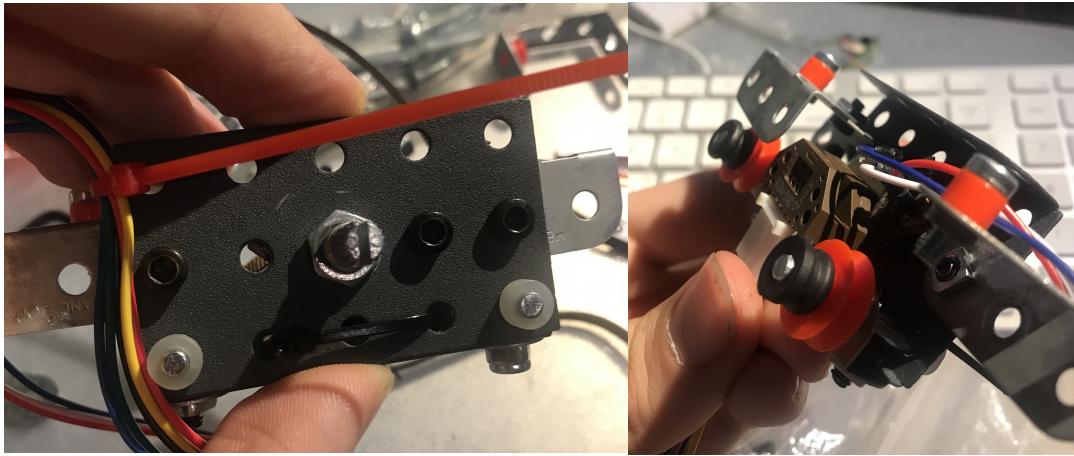
Figure 5: DC Motor Belt and Feedback



(a) System Setup with Wiring

(b) System Electronics

Figure 6: System Setup



(a) Potentiometer on Cart

(b) Optical Encoder on Cart

Figure 7: Custom Built Low Friction Linear Guided Cart

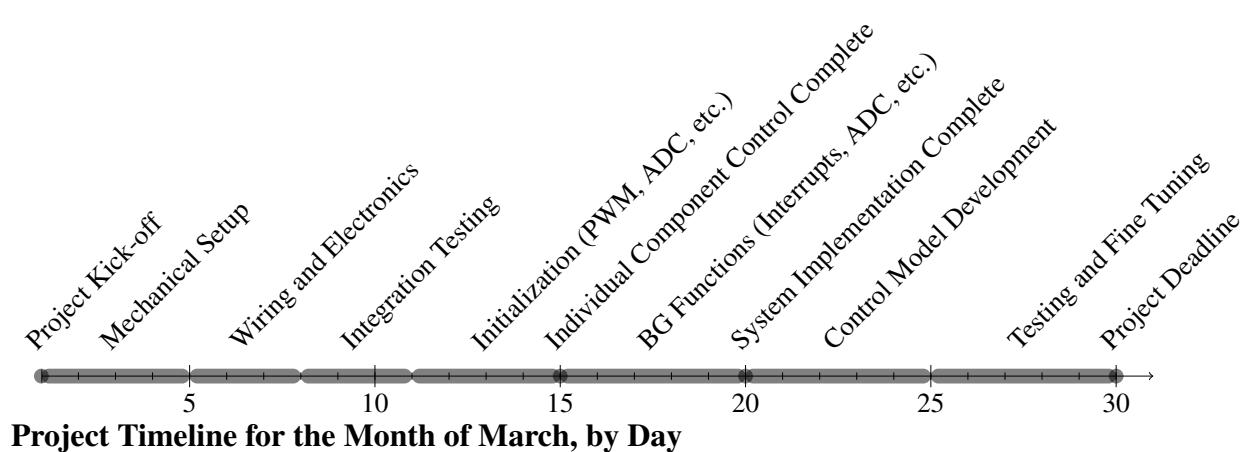
## 4.2 Software

To develop the state space model analysis, software tools MATLAB and SIMULINK will be used. MATLAB to develop the system state model, test and implement observer based feedback control, and SIMULINK to simulate and test the system, as well as fine tune the control gains. Furthermore, the controllability and observability of the system can be assessed in MATLAB, and will be discussed in the final project report.

# 5 Implementation Plan and Timeline

## 5.1 Timeline

It has been decided upon that the project duration will last approximately 1 month, to ensure completion of the working control system before the due date. The following timeline shows the approximate estimated dates for each major phase of the project. It should be noted that some tasks will be completed in parallel rather than in series, when possible.



## 5.2 Software Implementation Approach

An approximate approach to the functions that will run on the microcontroller have been discussed as follows:

- Interrupt handler for optical encoder, triggered by channel A and B outputs. This ISR will increment or decrement the cart position according to the following figure values:

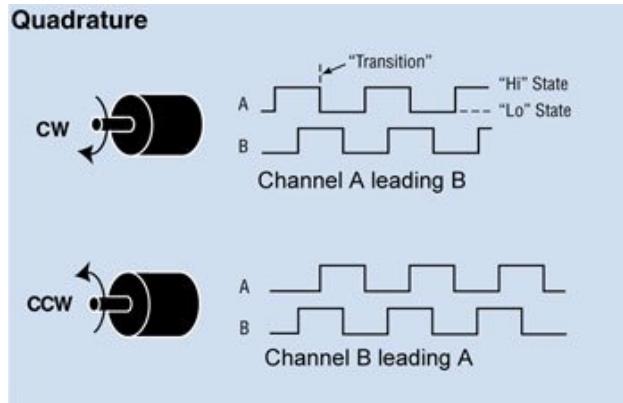


Figure 8: Quadrature Encoder Outputs [4]

- Read ADC function for potentiometer input.
- Duty cycle set function to control the DC motor.

The aforementioned functions will be executed in the approximate following order, while the ISR is running in the background to vary the cart position:

1. Check cart position.
2. Read pendulum angle.
3. Evaluate state feedback model control.
4. Write PWM outputs for motor position.
5. Repeat.

### 5.2.1 Concerns

There are a few potential roadblocks. Such issues may cause difficulty in programming the overall control system, especially due to the fact that our control model implies a substantial amount of pure theory, unlike a simple but effective PID controller. Said concerns are mentioned below:

- Extremely fine optical encoder strip will result in rapid interrupts at high cart speeds. Could inhibit the control model from running in real time.

- Non-linearities such as friction will sufficiently impact the control model.
- Gain tuning takes time and testing, and producing a critically damped system will require accurate SIMULINK models and iterative physical testing.

## 6 Conclusions

Overall, this system is completely feasible, controllable and observable. Thorough testing has been completed with all proposed components to ensure satisfactory working requirements to implement the state controller. Furthermore, proposed GPIO for the microcontroller has been completed and checked in the course lab experiments, hence all the required knowledge is possessed. Moving forward, there are a few concerns that may cause potential problems in the execution of the project, as highlighted in section 5.2.1. Such problems will be addressed as soon as possible once the integration of the functioning mechanical and electrical system before any controls development takes place.

All to date development can be seen on Github [6], currently most development is on branch **init**. Folder **matlab/** discusses system analysis, controllability and observability. Folder **tiva/** contains all code for the microcontroller board and Code Composer Studio.

## References

- [1] *OPEN LOOP AND CLOSED LOOP SYSTEMS*. [Online]. Available: <https://www.pesquality.com/blog/open-loop-and-closed-loop-systems>
- [2] *Inverted Pendulum: System Modelling*. [Online]. Available: <http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=SystemModeling>
- [3] *Inverted Pendulum: State-Space Methods for Controller Design*. [Online]. Available: <http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=ControlStateSpace>
- [4] *Quadrature Encoder Overview*. [Online]. Available: [https://www.dynapar.com/technology/encoder\\_basics/quadrature\\_encoder/](https://www.dynapar.com/technology/encoder_basics/quadrature_encoder/)
- [5] *Chapter 2: System Modelling*. [Online]. Available: [https://www.cds.caltech.edu/~murray/courses/cds101/fa04/caltech/am04\\_ch2-3oct04.pdf](https://www.cds.caltech.edu/~murray/courses/cds101/fa04/caltech/am04_ch2-3oct04.pdf)
- [6] R. Fielding and R. George, *Inverted Pendulum Github Repository*. [Online]. Available: <https://github.com/ryanfielding/inverted-pendulum>