## Fall Quarter Project Status Report for Period Ending 11/7/2012

To

Andy Chang, National Instruments

Dr. Robert Throne, Rose-Hulman Institute of Technology

From

Spencer Carver

Kevin Collins (Leader)

Ander Solorzano

Ruffin-White Magner

Report Date

10/31/2012

## Executive Summary

The project to build a self-balancing unicycle for National Instruments is proceeding along, and while progress is being made, not everything has gone smoothly. Since work began on the project two months ago, there has been a large amount of designs, revisions, and models developed in order to accurately portray the unicycle problem. While the majority of these issues were minor, one of the major concerns is that it is currently unknown if a mathematical model for the system can be developed at all. While this model is unnecessary for the actual prototype to be completed, the model would be a great benefit in constructing the robot’s code base later on. There has also been a large amount of parts ordered for the project, including the robot’s control unit: a NI cRio real-time controller. There has also been ongoing negotiations with several other sources, such as the motor supplier in China, in order to obtain the remaining parts necessary for the robot.

The next step for the unicycle project is to continue collecting parts for the physical prototype, and develop a controls system within LabVIEW that can work with the simulation that has been developed. This will (theoretically) allow for a much easier transfer once the physical unicycle has been built.

In terms of the past two months, there has not been very many scope changes, with a few clarifications coming out regarding the design, and a few more regarding travel and transportation for the product at the end of the school year. Since there have not been many changes, we are currently very much on budget. We have already obtained the LabVIEW Robotics Suite software, the cRio, and half of our motor controllers. Of our total budget of roughly $27,000 we have gone through about $24000 in terms of orders, though many of these components are from our sponsor, and as such do not actually cost as much as listed.

## 1. Introduction

### 1.1 Project Overview

The client for this project is Andy Chang, a representative of National Instruments. The project has been proposed as a stage demonstration for NI Week 2013, a large conference put on by National Instruments to showcase applications of their products and introduce new ones. This project in particular is designed to show how NI products (especially LabVIEW) can be applied to solve challenging controls problems, and a balancing unicycle is something this is easy to see on. Within the scope of the project, our final goal is to deliver to the client a physical prototype capable of balancing on its own. The prototype should be able to correct for its own motions, and be able to correct if slightly knocked as well. In order to accomplish this, we have divided the time we can spend on this project into two parts. The first part of this project involves accurately describing the unicycle problem within a simulator, specifically the National Instruments Robotics Simulator. Once our LabVIEW code is working on the simulator, we will then move to the physical prototype. The difficulties in getting the prototype to function should be mitigated by the use of the simulator, but we have also allotted the entire spring term to getting the prototype functioning properly in order to further our goal of a completed prototype.

### 1.2 Technical Background

This project is a robotic application of a 2-dimensional inverted pendulum problem, and is an application of high-precision, real-time controls on a robotic platform. This problem is far more complex than 1-dimensional inverted pendulums, and as a result far less work has been done on this variant than the less complex one. With regard to the 1-dimensional case, there has been much notable work, most popularly related to that of the Segway[1] [2], that accurately describes this case. For the 2-dimensional case, there are much fewer existing works, and while a few have a similar design concept as our robot[3], none utilize NI technologies in the way that we plan for our project.

In a 1-dimensional inverted pendulum, the robot is balanced through rotation of the balancing wheel in order to counter-torque the effect of the falling mass located above the wheel. In the 2-dimensional case, there is the added issue of side-to-side motion, which most current systems negate by creating a wide wheelbase (effectively neutralizing the 2nd dimension from the problem). For a unicycle, this simplification is impossible, as the single wheel is too thin to effectively negate side-to-side motion. Because of this, the 2-dimensional inverted pendulum has remained a difficult controls problem even today, and is a worthwhile investigation of any platform.

## 2. Project Planning Summary

In order to actually complete this project within the three quarter time limit of the capstone design course, the team decided (with our client’s recommendation) to pursue an ambitious modeling plan for the fall and beginning of winter quarters, and to use the remainder of winter quarter to get the robot’s simulation functioning in the LabVIEW Robotics Simulator. The spring term is nearly completely set aside to work on the physical prototype of the robot, and get it functioning in-line with what is on the simulator.

### 2.1 Plan of Approach and Timeline

Our basic approach to the project can be outlined as follows

Fall Quarter

* Model Design
* Model Controller
* Order Parts

Winter Quarter

* Physical Construction
* Model and Controls Adjustment
* Robot Programming

Spring Quarter

* Tuning and Debugging

During the fall, we will design a very detailed model of our planned physical robot and import it into the LabVIEW Robotics Environment Simulator. We will then create a control algorithm in LabVIEW to control our model. While we are working on this, we will be ordering parts for the physical robot to be built in the winter.

During the winter, we will build our robot. Then, we will adjust our virtual robot model to more closely represent the actual robot and adjust our control algorithm so that it works for the new model. Once we are confident in our algorithm, we will program it to our physical robot.

The entire spring quarter is reserved for adjusting our control algorithm and debugging the system. We anticipate that it will take quite a bit of work to tune the controller to a point that it will achieve acceptable results.

### 2.2 Training Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Topic | Member(s) Tasked | Relevance | Date to be completed |
| Mechanical Modeling | Kevin | taken mechanical modeling classes (Mechanical Engineer) | end fall |
| System Modeling in NI software | Ander and Spencer | Experience using NI software | end of fall |
| Sensors, Motors and Parts | Ruffin | Previous experience working with sensors | end of fall |
| System Control in NI software | Ander | Previous experience working with NI software. Taken multiple control classes | end of fall |

### 2.3 Division of Labor

**Controls** - Controls consists of writing a control algorithm within LabVIEW to control our simulated robot, and eventually our physical robot. Ander is our primary controls engineer since he has had the most experience with controls in LabVIEW. Spencer is also helping develop our control algorithm and is in charge of making sure that our controller output interfaces well with our mechanical and analytical models.

**Analytical and Simulated Models** - Ruffin and Kevin have been working more on the mechanics side of things. Right now, they have created both an analytical, mathematical model created using first principles learned in the mechanical engineering curriculum, and a virtual 3d model of the robot created in the LabVIEW robotics environment simulator. Ruffin has taken charge of the simulator, and Kevin has taken charge of the analytical models and setting up the analytical and simulated models in such a way that they can be manipulated with controls in LabVIEW.

**Purchasing** - Spencer and Ruffin have been involved in purchasing components. They have dealt with lots of sensors through their experience on the robotics team, and know where to look for components.

### 2.4 Budget and Key Expenditures

Since National Instruments is our client, we are using NI hardware and software wherever possible. Our main expense for this project is the LabVIEW Robotics Suite 2012, which is nearly $15,000 on its own. Our second largest expense is the cRIO 9024 real time controller, which is nearly $9000. National instruments has provided us with both the robotics suite, and the cRIO, so our actual expense has been drastically minimized. However, if someone wanted to replicate our project, they would need to take these expenses into account.

Additional expenses include our Inertial Measurement Unit (IMU) that we are able to get a student discount on, but will still be $2660. This IMU is necessary due to its high level of precision, as for this controls system, the faster the platform can collect data about its imbalance, the faster it can correct and maintain the balanced state with minimal output. We will also require two hub motors from alibabba electronics which will be ~$100 apiece, as they must be ordered directly from a manufacturer in China .

## 

## 3. Design/Research Accomplishments

### 3.1 System Description

The entire system design that we are dealing with consists of a non-linearized system that we are controlling in discrete-time. For our purposes, we decided to use a state variable feedback system description since poles can be placed in the system to achieve a controlled (in our case balanced) state.

To use a state variable description, we will need to linearize the non-linear system. The state variable system can be described as follows:

|  |
| --- |
|  |
| State variable feedback system description with prefilter and controllability matrix *K*. |

From the above system, a generalized system of equations that will be used to describe system can then be derived as:

|  |
| --- |
|  |
| In the given state variable equations, **G**,**H**, **C**, **D** are matrices of constants that define the system’s equation. **G** is a matrix that contains position, velocity, angular position, and angular velocity. **H** is a matrix that contains the unknown states. **K** is the controllability matrix that contains the controller gain equation. |

### 3.2 Challenges and Open Issues

One of our main challenges is the difficulty in solving the analytical model of the system. We originally assumed that we could simply create two analytical models in each dimension and then control each dimension simultaneously, but this only works if you assume that the robot is completely balanced in one direction in order to control the other. If there comes a point where the robot is unbalanced in both dimensions, which is inevitable, then the model is different, and there is a chance that the robot will respond differently to our control algorithm. How much of a difference this will make remains to be seen, but it is crucial that we complete our labview simulation as soon as possible to learn what expect from the system.

## 4. References

[1]<http://www.google.com/patents?hl=en&lr=&vid=USPAT6543564&id=69sMAAAAEBAJ&oi=fnd&dq=inverted+pendulum+robot+unicycle&printsec=abstract#v=onepage&q=inverted%20pendulum%20robot%20unicycle&f=false>

[2]<http://www.google.com/patents?hl=en&lr=&vid=USPAT6581714&id=oJIMAAAAEBAJ&oi=fnd&dq=inverted+pendulum+robot+unicycle&printsec=abstract#v=onepage&q&f=false>

[3]<http://lawww.epfl.ch/webdav/site/la/users/139973/public/repports/Kappeler.Rapport.pdf.pdf>