11/14/2016



By Tom Scherlis, Crosby Deliman, and Alex Zhang

Term 1 Physics Report SHADY SIDE ACADEMY

Contents

Introduction:
Rationale:2
Objectives:
Plan:
Challenges
Parts search3
CAD3
Materials:6
The Hardware:7
Drive Train7
Wheels:7
Mounting plates
Handlebars
Electronics:
Power electronics9
Control electronics
Jaguar Motor Controllers:
IMU's:
Algorithms:
Challenges:
More pictures of electronics:
Reflections14
Acknowledgements14

Introduction:

We decided to build a Segway this term, a self-balancing single person mobility vehicle. Consumer Segways retail for multiple thousands of dollars, and the "cheap" clones and miniature versions usually retail for upwards of 600\$. Our goal was to create our own Segway, because why not, it will be fun!

Rationale:

First of all, building a Segway has a large inherent motivation in that it will result in the builders having a Segway. Segways are unquestionably fun, and it is always a positive to own one. However, they are usually difficult to get access to outside of tours and other rental services, and owning one for yourself is very expensive.

Second, the process of building a Segway is an incredibly good learning experience. It allows us to learn metal working, high power electrical design, control electrical design, computer modelling, and algorithm design. The Segway is difficult, but as a project it has been done often enough that there are plenty of resources available for those looking to build their own.

Objectives:

- Support adult rider without breaking
- Maintain balance without falling
- Steer by tilting the handlebars
- Be battery powered with reasonable battery life (>15 min)
- Drive at a reasonable speed (>10mph)
- Have an unobstructed foot area for rider to stand on
- Fit through a doorway (be around 2ft wide max)
- Be relatively small and compact
- Be durable

Plan:

Building a Segway requires an organized design to fabrication process in order to end up with something that works well and is safe and well-engineered. Here is the comprehensive 12 step process we mapped out ahead of time to build the Segway:

- 1. Extensive online research of similar projects
- 2. Look for parts that we can already use to make this feasible
- 3. Design in Solidworks
- 4. Waterjet aluminum sheets and base plate
- 5. Build and mount the drivetrain
- 6. Inflate and mount wheels/tires
- 7. Mount handlebars and bearings
- 8. Mount Jaguar motor-controllers and motor controller mounting board
- 9. Wire power electronics for motor controllers, battery, fuse, charger, etc
- Wire control electronics and perform diagnostic checks/firmware upgrades on all motor controllers <- we are here
- 11. Write and program all control algorithms for balance, and steering.
- 12. Test it!

We are approximately at step 10 as of writing this report, but it is unclear how easy it will be to actually be to complete the Segway from there.

Challenges

One of the major challenges of this project was designing the CAD model. We needed to be precise and accurate with our CAD model, which needed to perfectly fit all the equipment such as the batteries, motors, and motor controls. There was no room for error. We also had problems putting the wheels on the tire. The wheels wouldn't fit on the tires.

Parts search

Before starting the projects, we looked to see what parts we already had available to build the Segway. Unfortunately, we realized that gearboxes, motors, and motor controllers together cost a large amount of money, as in several hundred dollars each, so we had to find a way to get them for free.

We contacted Mr. Wizzard, who coaches robotics team, to see what he had. SSA used to compete in FIRST Robotics Competition, so the robotics team had a large amount of unused motors, motor controllers, and high power motor controllers laying around, which Mr. Wizzard kindly let us use for this project.

CAD

The first step was to figure out the drive train, which everything else was built around. We knew which wheels, motors, and gearboxes we were using, as well as that we needed a shaft coupler and bearing to support the weight of the rider. Therefore we made a computer model of just those components first. Here are some screenshots:





Here are some fancy renders:





Materials:

We used an abundance of parts we already owned, including the drivetrain parts and the aluminum plates for the Segway. We also left out parts such as the mounting hardware, because we had most of it already laying around except for the 3/8in and 1/2in bolts.

Thing	Place	Qtty	Qtty owned	price	notes
Wheels	http://www.staton- inc.com/store/index.ph p?p=product&id=1839	2	0	57	‰in axle 3/16 keyway
Motors	http://www.andymark.c om/CIM-Motor-p/am- 0255.htm	4	4	0	From wizzard
Motor controllers	http://www.vexrobotics. com/217-3367.html	4	4	0	From wizzard
Gearboxes	unknown	2	2	0	From wizzard
wires	Thick wires and speaker wire	~	~	~	From wizzard
Axle	http://www.staton- inc.com/store/index.ph p?p=product&id=792	10	0	12	Sold by the inch
Platform	Under feet	1	0		1ftx2ft
Кеу	http://www.staton- inc.com/store/index.ph p?p=product&id=791	2	0	0.48	to lock axle
Tires	https://www.amazon.co m/gp/product/B0039R G6YA/ref=oh_aui_detai lpage_o00_s00?ie=UT F8&psc=1	2	0	21.08	
16x1.75-2.125 Schrader Valve Bicycle Tube, Black	https://www.amazon.co m/gp/product/B00MJY OXTE/ref=oh_aui_deta ilpage_o00_s00?ie=UT F8&psc=1	2	0	10.98	
Shaft Coupler	McMaster	2		120	½ to 5/8 solid shaft coupler
IMU	https://www.adafruit.co m/product/2472	1	0	35	
Flame decals	Did not buy yet	lots	none		Very important
Underglow RGB lights	Did not buy yet	Lots	none		^^^^
Bearing Blocks	Mcmaster 5913K62	2	0	22	

Battery	Amazon	1	0	35	12v x 20Ah SLA
Charger	Direct from Powerlet LLC	1	0	50	Came with plug for BMW motorbikes

The Hardware:

Drive Train

Two motors were attached to each gearbox for extra power. An axle was then attached to each gearbox. The wheels each spun on two separate axles so that they could spin at different speeds allowing the Segway to turn. In order for the wheels to be attached to the axles, we used a tapping kit with oil to thread the axel and bolt the wheel to the axle. The wheels are from a jogger and have a 5/8 in axle with a key. We added new tires with separate inner tubing around them.

The square steel tubing had to be cut and drilled in order for it to be bolted down to the motors. Holes were drilled on each end through the layers of the square tube with the vertical drilling and tapping machine. These were used as angle brackets to attach the motors to the mounting base plate.



Large bearings were mounted to the 5/8 in axle to hold the force of the rider, and hopefully the whole system is frictionless enough for good driving. However, due to some slack in the gearboxes the drivetrains have a little bit of hysteresis, which could be an issue with self-balancing. It's probably 10degrees on the wheels or so, maybe less.

Another possible issue is syncing the left and right halves. Proper shaft encoders are very expensive at this size, so we did not use them and as such we have an open loop control algorithm. Our plan is to run them on a constant voltage supply with a camera or something to measure the speed of the wheel, then cycle through the power levels to create a lookup table of motor power versus actual RPM for the left and right side to prevent them from running at different speeds.

Wheels:

We used inflatable 16 inch tires on Skyway BMX wheels. The hubs were already bored to 5/8in with a precut keyway, which made attaching them relatively straightforward. With Dr. Peterson's help, we

mounted the tires, inserted them onto the shafts, drilled and tapped 6mm holes, then bolted them on the ends to prevent them from sliding them off.

Mounting plates

All mounting plates were cut from ¼ inch aluminum plate on the waterjet cutter at Techshop. The cutting took a few days, but it was definitely worth doing as all of the holes were lined up already because it is CNC. The challenges we faced waterjet cutting were numerous, but ultimately it was worth it. First of all, breaking the waterjet is very bad, and they charge hundreds of dollars if you break the nozzle. The base plate for example was almost as big as the plate we were cutting from, so it was difficult to create a toolpath that didn't intersect the mounting clamp.

Another issue with cutting a plate as small as ¼ inch aluminum is that it is lightweight, and thus the force from the jet can cause it to vibrate on the mounting ribs resulting in a jagged edge. We also had to do a lot of finishing work to clean up the mounting tabs on the parts, which are necessary to prevent them from falling into the garnet pool of the cutter.



Here are some pictures:

Waterjet facts:

- It uses more energy when running than every other tool at Techshop combined
- The pump is 60,000 PSI
- The water jet is supersonic and filled with abrasive garnet
- It costs 3\$ per minute to run.
- It can cut up to 8 inches of granite or 4 inches of titanium

After waterjet cutting the base plates, we had to do extensive finishing work. That included brushing them to clean them and grinding off sharp tabs, but we also had to drill out all the holes to clean them up and file the tabs so they would lock together. Greg at Techshop kindly let us use his big tap wrench so we could tap the 3/8 in holes for the battery plate.

Handlebars

To achieve our goal of tilt steering, we needed a set of handlebars. We mounted high force mounted bearings on the base of the base plate, and attached an approximately 4ft long steel square tube to it with a half inch bolt and a lock nut. The result is a very strong connection, and you can pick up the whole Segway by the handlebars.

The handlebars themselves were just generic 10\$ handlebars from amazon and mounted with a $\frac{3}{20}$ bolt and locknut. We need to reinforce it though because with some force you can rotate them a bit on the mounting bolt.



Images of the handlebar mounting

Electronics:

Power electronics

The requirements of the power electronics were as followed:

- Powered by 12v sealed lead acid (SLA) battery
- Provide up to 100A to each motor controller
- Provide up to 300A for the entire system before tripping the fuse
- Be protected by a 300A fuse
- Power distribution using high current 100A bus bars
- Have hardware battery disconnect for safety and power saving
- Centralized power distribution (no daisy chaining) to remove critical failure points and to cut down on power cable gauge size

We ended up using 12AWG speaker wire for the motor controllers and 2AWG cabling for the battery, with a 300A fuse in series with the battery before the bus bars. Here is a picture of our power distribution scheme before the motor controllers were wired to the bus bars.



Control electronics

Unfortunately, we didn't have time to totally finish the control systems. We did however do extensive planning for them, and we plan to order a PCB for it shortly as we realized that using protoboard would be a massive pain after trying to wire it for a couple hours. Seriously protoboard is awful.

Here is a block diagram:



As you can see, there isn't *that* much to actually do here, and we've already laid out the framework. The key element is controlling the Jaguar motor controllers, and we elected to use another Arduino to do that because we already had a few laying around, and they have a lot of nice features. First of all, all of the hardware register editing code is already handled by the Arduino library, so writing code is incredibly simple. Second, the Arduino dev boards already have oscillators and FTDI chips attached, and a 5v regulator. The advantage of the FTDI chip being built in is that we can communicate over USB without a special adapter or and RS232 cable. The Arduino also has its own bootloader, so reprogramming and debugging can be done without a fancy ISP or Atmel ICE debugger. However, I think using an independent MCU would still be more fun and we might end up doing that to finish this project if we have time. We needed some big decoupling capacitors because the voltage drops from drawing several hundred amps are pretty substantial.

Jaguar Motor Controllers:

The Jaguars come in two types, black and gray. The gray Jaguars rely on using a Controlled Area Network (CAN bus) to communicate, and the black ones have a built in UART to CAN converter on the input side. We had 3 gray and 1 black, so we daisy chained them like shown above. The advantage of CAN is that it is very high speed, and because it uses differential signaling it is resistant to the huge amounts of digital noise created by the massive motors. The UART protocol does not have that, and relies on heavy cable shielding. However, the Arduino cannot talk to CAN without a special CAN transceiver IC, which we didn't want to use so we just used UART.

We created a special UART to RS232 cable to upgrade the Jaguars' firmware using Texas Instrument's program BDC-COM before trying to use them. On the control board we used a pair of MOSFETS to isolate the Arduino and Jaguars from any sort of reverse voltage or voltage spikes across the UART line.

IMU's:

We used I2C to communicate with the dual BNO055 IMU's which each provide high speed orientation and gyroscopic data used to self-balance and determine handlebar orientation. Luckily, the IMU's have a built in 1 bit address pin, so up to two can be used on the same I2C bus without using a chip select to alternate between them. We modified Adafruit's IMU control IMU to support a variable address, because out of the box it defaults to only using the default 0x28 address, but with two IMU's one of them will use the 0x29 I2C address. Luckily the library had the capabilities to use a variable address built in, but the constructor defaulted to 0x28 which meant a slight modification was necessary.

Algorithms:

The software will use a PID loop to balance the Segway when it is complete, but we haven't written that code yet unfortunately. We initially planned on doing extensive torque and angular momentum calculations, but realized that because we have a high speed closed loop system, that wouldn't be necessary. Instead, we decided to use an arbitrary system and just spend time tuning the PID control constants. In this case, the Proportional value would probably be fairly high to enable a quick response time to a tilt. The Integral term would probably be fairly low, to prevent the Segway from accelerating too fast from leaning forwards for a while. The Derivative value would need to be tuned to see what feels comfortable. The P and I values would use the orientation data from the IMU that has already been run through a Kalman filter to get absolute orientation from acceleration and gyroscopic data. The D value would just use the raw gyroscope data.

We would need several safety shutoffs to prevent any sort of driver harm, although we're not sure what that would include quite yet.

Challenges:

We ran out of time before we could finish this system. However, here is the prototype that we made but did not program:



Hopefully we can finish it soon. It takes a long time to work with protoboard, so maybe we will make a PCB with some proper plugs for the UART line and I2C bus with RJ11 or 6p6c connectors.







Partially complete Segway

What it can do:

- Look cool
- Support a person
- Boot up successfully
- Complete a Paul Blart cosplay
- Look cool

What it can't do:

- Actually drive
- Balance
- Work at all









Reflections

Looking back on the project now, there are few things we would have changed. For every two weeks, we should have had a definitive goal. Then by the time two months passed, we would have finished our physics project easily and quite painlessly. Even when our group planned to meet, we had to cancel some meetings due to last minute conflicts including college visits, sports tournaments, disease, and college deadlines. Also we should have planned our CAD model a little bit better. The pieces of metal did not fit correctly, so we spent hours grinding. If we planned better, we would have saved a couple of hours. Additionally, we should have planned out the tires better. Because the tires didn't fit, we needed to wait a couple of days for the new tires to come. This almost led us not to finish our Segway project.

This project was definitely was a great success. Even though we worked extremely hard for the last days because we went down to the wire, we still had a lot of fun and the Segway looks dope. Working together as a group was very rewarding. We learned a lot from each other and from Dr. Peterson. Random discussions during our work times made hours seem like minutes.

Acknowledgements

A good amount of design inspiration was taken from MIT's segway project, although ultimately our design was our original idea. We used their same wheels, however, and built on the issues they encountered, ie adding 2 motors to combat a lack of power issue that they faced.

We would not have been able to finish this project without help from a couple of a people. We would like to thank Mr. Wizard for allowing us to use the Robotics lab and where we built our Segway, and of course the motor controllers and motor units. Additionally, we would like to thank Scott Peterson for advising us and guiding us through this entire process. He also helped us out when building using his metalworking skills. Finally, we would like to thank Eshan Gupta for believing in us. Also Mark.