Segway Design Meeting #1 Notes – June 16, 2007

Overall Design Goals ("Functional Requirements"):

- Build something you can ride…
- Make it safe enough to ride without worrying about getting thrown off. (Still wear a helmet/pads, see safety notes.)
- Beat TLB.org scooter in the following categories:
  - weight (goal < 50 lbs with battery, < 35 lbs without…portable, fold-up?)
  - cost (goal < $1,000 total components, some of which we will scavenge)
  - size (goal < 30" wide to fit through doors, this will be tough…see drive notes)
  - Google hits (I'm not sure how we go about doing this.)

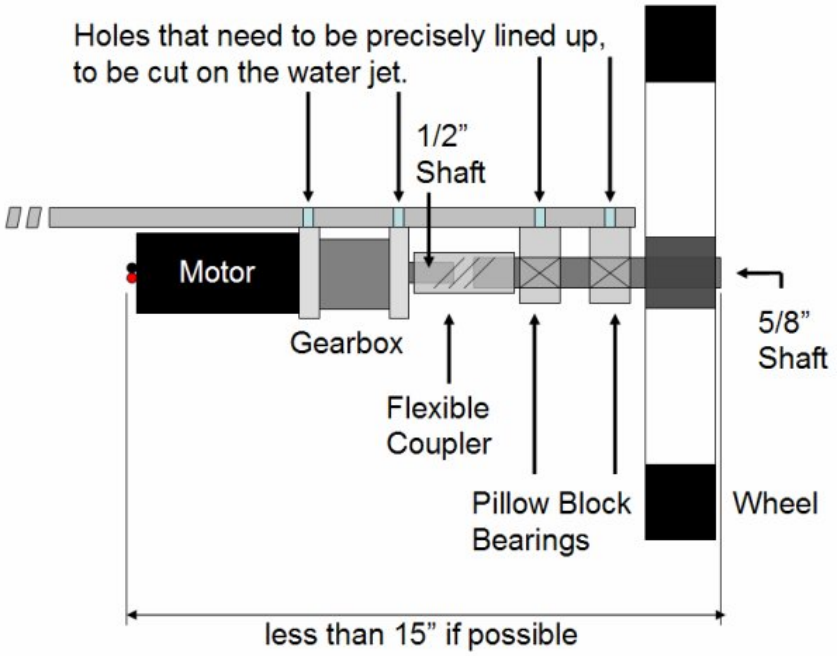
Drive System Discussion Summary: Direct Drive from CIM Motors

- CIM Motors (FIRST kit motor) specs:
  - Voltage: 12V
  - No-Load Speed: 5280 RPM
  - Cost: $28 ea.
  - http://www.banebots.com/pc/FIRST/M4-R0062-12
- BaneBots CIM Planetary Gearbox specs:
  - Ratio: 16:1
  - No-Load Speed: ~(5280 RPM ÷ 16) = 330 RPM
  - Output shaft: ½" keyed steel, probably not enough to directly support the full weight of the segway + rider
  - Cost: $89.50 + $19.50 = $109 ea. for gearbox and 16:1 modification
  - http://www.banebots.com/pc/FIRST/GP-56012
- Wheel specs:
  - Skyway 12.5" or 14" Tuffwheels
  - Absolute top speed (should be 50% or more above the actual speed limit for safety factor):

    $$330\frac{\text{rev}}{\text{min}}\cdot 12.5\text{in}\cdot\pi = 12.272\frac{\text{mi}}{\text{hr}}$$

    $$330\frac{\text{rev}}{\text{min}}\cdot 14\text{in}\cdot\pi = 13.744\frac{\text{mi}}{\text{hr}}$$
  - Hub: 5/8" keyed?
  - Cost: TBD, FIRST discount?
  - http://www.skywaywheels.com/products_006.htm
- Coupler/Bearing Design:
  - Coupler to a 5/8" shaft to support rider weight up to 250 lbs. (Need to do computer stress analysis on this when we do the CAD model.)
  - Flex coupler to avoid transmitting load to gearbox shaft.
  - Two pillow-block bearings to support 5/8" drive shaft, need to be spaced at least 3 shaft diameters apart, so at least 1-7/8" between bearings.
  - Nevan/Shane/Matt will spec out bearings and couplers before next meeting.

- Added width: Can we still fit through a 30” door? Need to do a solid model assembly.
- Cost: TBD



Quick PowerPoint sketch of the proposed direct-drive assembly. The bearings support the full weight of the segway and rider so that the gearbox shaft has very little sideways force on it. We will need to do a SolidWorks model of this next meeting.

Control System Discussion Summary:

- Accelerometer: ADXL203
  - Senses the effect of gravity, can sense the angle to vertical, like a pendulum
  - When horizontal: 0g on x-axis, -1g on y-axis
  - When tilted, x-axis feels some of the gravity, increases or decreases from 0g depending on the direction of tilt.
  - Sensor output: 0-5V analog, no tilt = 2.5V, 1g ≈ 3.5V, -1g ≈ 1.5V
  - Didn’t talk about how to read into microcontroller, will do this next time maybe.
  - http://www.analog.com/en/prod/0,2877,ADXL203,00.html
  - Cost: $36 on Digi-Key, but we have two already
- Gyroscope: TBD
  - Senses angular velocity (rate of tilting, in degrees per second).
  - Can be used to “dampen” control system, keep the motors from overshooting the neutral point.
  - Sensor output: 0-5V analog, stationary = 2.5V, -75 deg/sec ≈ 0V, 75 deg/sec ≈ 5V (need to test this)
  - Tried old BEI Gyro from FIRST kit, but it was broken. Possible Replacement: ADXRS401
  - Cost: $56 on Digi-Key

    - http://www.analog.com/en/prod/0%2C2877%2CADXRS401%2C00.html
- General Control Strategy:
    - Inputs: Tilt Angle (accelerometer + gyro), Angular Velocity (gyro)
    - Outputs: motor speeds (or accelerations)

	Angle Forward	**Angle Zero**	**Angle Back**
Angular Velocity Forward	Accelerate Forward FAST	Accelerate Forward	???
Angular Velocity Zero	Accelerate Forward	Do Nothing	Accelerate Back
Angular Velocity Back	???	Accelerate Back	Accelerate Back FAST

    - What do we do in ??? situations?
    - Could use if statements, but a simple, smoother approach could be to use the + or – signs from the sensors to determine the + or – signs to the motors:

      $$\textit{Motor Acceleration} = (K1 \times \textit{Angle}) + (K2 \times \textit{Angular Velocity})$$

      K1 and *K2* are "gains" or "speed factors" that you can tweak to get the desired response. But in general, this formula follows the same rules as the above table. (Example: If *Angle* is positive and *Angular Velocity* is positive, *Motor Acceleration* will be (very) positive. For the ??? situations, the direction of the motor depends on *K1* and *K2*.

    - Things we didn't talk about yet:
        - How much do we trust our sensors? Are the different situations where we trust one more than the other?
        - What about a speed limit?
        - How do we handle steering?
        - How do we control the motors with the microcontroller?

Safety Discussion:

- Cushion handlebar.
- Speed limit.
- "Training wheels" for testing.
- Turning rate limit (for not tipping over sideways at high speed).
- "Dead man's switch" to turn off motors if rider lets go.
- Wiring tied down well to prevent accidental disconnections.
- No matter what, somebody will fall off at some point, so wear a helmet and maybe pads.

Next Meeting: Saturday, June 30, 10-3 again

- Full list of parts to order (Nevan/Shane/Matt will spec out bearings/couplers and report back to all. Everyone should think about other miscellaneous parts we will

need besides what we talked about today. (Handlebar, steering mechanism, mounting hardware for sensors, extra aluminum for bracing/support, etc.)

- We will do a CAD model of the important drive elements. Individual part models to be done in advance. (Some from www.firstcadlibrary.com, others to be spec-ed and modeled by Nevan/Shane/Matt/Max. This way next time we can focus on the full assembly model and specifically getting a model of the plate to be cut on the water jet.
- I will get a working gyroscope, I promise. We can play around a bit more with the sensors and think about some of the missing links in the control system setup. Also how to control the motors with the microcontroller, which we didn't talk about. We eventually have to start actually writing code, but that will depend on how I feel that day. (I hate writing code.)