**Segway Control**

**Introduction:**

The goal of this project was to update the existing Elektor Wheelie, which is a two-wheeled Segway used for short distance transportation, to create a safe and reliable platform to be used as a sound basis for academic research, development and fast deployment of advanced controllers in LA's future projects.

**State of the Art:**

The initial state of the Segway had a few issues:
- Both of the amplifier, ESCON 70/10, were undersized and couldn’t be used to control the motor. They could support only 10A which was not nearly enough for the control of the motors.
- One of the battery was dead
- The acquisition card used for the analog control of the amplifiers and the control of the power relay’s switch had permanent issues and could not be trusted.
- The buzzer wasn’t working properly

**System modeling:**

The equation used are:

\[ \ddot{x} = \frac{1}{m} \sum_{i=1}^{n} F_i = \frac{1}{m}(F - bh\dot{x}) \]

\[ \dot{\theta} = \frac{1}{\tau_s} \sum_{i=1}^{n} \tau = \frac{1}{\tau_s}(-N\cos(\theta) - Pl\sin(\theta)) \]

With variables:
- \( F \): force applied to the cart
- \( M \): the mass of the cart
- \( m \): the mass of the pendulum
- \( l \): the length from the contact point of the pendulum to its center of mass
- \( t \): the moment of inertia of the pendulum
- \( b \): coefficient of friction of the cart
- \( x \): cart position coordinate
- \( \theta \): pendulum angle from vertical
- \( \phi \): \( \theta \) used for our transfer function

We get the following transfer function for our angle:

\[ G_{\text{pendulum}}(s) = \frac{G_{\text{motor}}(s)}{G_{\text{sensor}}(s)} \]

\[ G_{\text{motor}}(s) = 2 \frac{V_{\text{out}}}{I_{\text{in}}} = \frac{1}{K_{\text{motor}}(s)} \]

\[ G_{\text{sensor}}(s) = K_{\text{sensor}}K_{\text{Potentiometer}} \]

\[ \dot{\theta}(s) = G_{\text{pendulum}}(s)G_{\text{sensor}}(s)K_{\text{Sensor}} \]

\[ S_{\theta}(s) = K_{\theta}(1 + T_{\theta} \dot{\theta}) \]

\[ T_{\theta} = \frac{1}{K_{\theta}} \]

With:
- \( K_{\text{sensor}} \): the conversion constant of the PCB op-amp circuit
- \( K_{\text{motor}} \): the conversion constant set on the amplifier
- \( K_{\text{motor}} \): the measured constant of the motor
- \( K_{\text{sensor}} \): the radius of the wheel
- \( \dot{\theta} \): the measured time constant

\[ q = (M + m)(\dot{\theta} + ml^2) - (ml)^2 \]

**Results:**

From Matlab we get a satisfactory response to our unit step input. Our phase margin is 90 degrees and our gain margin is infinite. If the steady state error is not that much of a problem, the overshoot might be unpleasant for the user. More sensors could be added to get a better control over the state of our system. Furthermore, more testing on the device is advised.

**Final State:**

- Both batteries were replaced with new ones with the same reference.
- Both amplifiers were replaced by a single, dual channel one, the SmartDriveDuo-30, an amplifier capable of supporting the 24V and 10A on each channel.
- However this new amplifier doesn’t have an analog and digital output for feedback, and doesn’t have an integrated PD controller which makes our new control design not as smooth as before.
- Furthermore this amplifier is controlled by a 0-5V analog signal and a digital one to indicates the direction of each motor, where the previously used amplifiers where controlled by a ±3 to 5V signal, the code had to be adapted in consequences.
- We tried to take into account the friction in our control model, which hasn’t been done previously.

**LabView programing:**

We kept most of the previously conceived Labview VI for the main loop. The main modifications were done on the part of the program controlling the new amplifier as we needed to add a digital signal controlling the direction of our motors. We also adapted the previously used PDs with the coefficient we found using Matlab.

**Matlab and Simulink simulation:**

We chose to use Simulink in order to modelize our system without the approximations of a linearization around theta. This gave us the sub-system (plant):

Which we used to to simulate our control design:

We needed to add a PD stabilizer due to instability of our system. We also chose to use a PD controller for our design as we didn’t need an integrator on our Segway. The steady state error allows the user to be inclined a bit while moving which is more comfortable for him.

**Authors:**

- Anas Beladda
- Mathieu Berthet

**Supervisors:**

- Professor Christophe Salzmann
- Navid Asmani Saadabad

**Spring semester 2019**