JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Department of Electrical and Computer Engineering
520.454/654 — **Control Systems Design** — Fall 2018

Project Description

1 Goal

The goal of this project is to design a controller to stabilize a Lego model of a Segway. This will be done by creating a nonlinear model of the Segway, designing a linear controller to stabilize it, and downloading the controller into the Lego Mindstorms EV3 hardware package.

The basic steps are as follows:

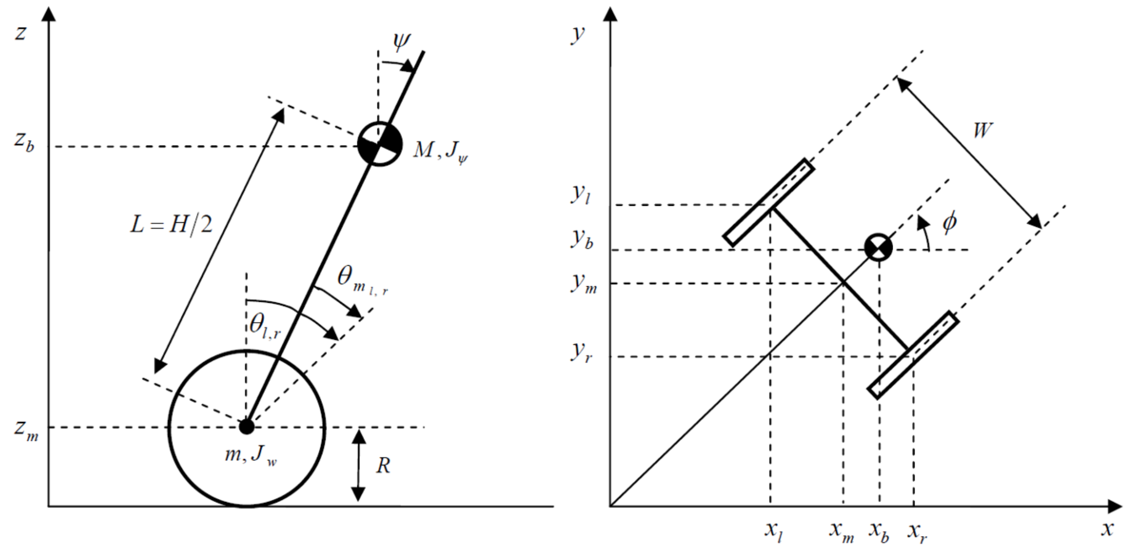
1. Create a state-space nonlinear model for the Segway in Simulink.
2. Linearize the model so as to enable control systems design.
3. Design a controller for the system and implement the controller in Simulink testing it against the nonlinear model.
4. Construct an observer that estimates the drift from a gyroscope.
5. Download the controller and test it on the Lego Segway robot.
6. Design a way for the robot to turn, go forward/backwards, and detect obstacles (if time permits).

2 Constructing the Lego Segway

A quick search on the internet will reveal a number of different ways of constructing Lego Segways. I recommend that you follow the model known as the "Balanc3r," see `https://www.youtube.com/watch?v=9uVuaqVLIq0`. Its main sensor is the EV3 Gyro, which measures the angular rate. The actuation that you have are the two motors that drive the wheels. More about these is given below.

3 Model introduction

You will base your mathematical model on the description based on the Fig. 1. The equations of motion describe the changes in the body pitch (ψ) and yaw (φ) angles, and



ψ : body pitch angle $\theta_{l,r}$: wheel angle (l, r indicates left and right) $\theta_{m_{l,r}}$: DC motor angle

Fig. 1. Diagrammatic representation of the segway model.

the left and right wheel angles (θ_l and θ_r, respectively). Our primary concern is stabilization of the robot, so we will assume that the two wheels move at the same rate. Thus, we define

$$\theta = \tfrac{1}{2}\left(\theta_l + \theta_r\right) \quad \text{and} \quad \theta_m = \tfrac{1}{2}\left(\theta_{m_l} + \theta_{m_r}\right)$$

to be the average of the two wheel and motor angles, respectively. The equations of motion for these three variables are then given by

$$\begin{aligned}
F_\theta &= \ddot{\theta}\big((2m+M)R^2 + 2J_w + 2n^2 J_m\big) + \ddot{\psi}(MLR\cos\psi - 2n^2 J_m) - MLR\dot{\psi}^2 \sin\psi \\
F_\psi &= \ddot{\theta}(MLR\cos\psi - 2n^2 J_m) + \ddot{\psi}(ML^2 + J_\psi + 2n^2 J_m) - ML\sin\psi(g + L\dot{\varphi}^2\cos\psi) \\
F_\varphi &= \ddot{\varphi}\left(\frac{mW^2}{2} + J_\varphi + \frac{W^2}{2R^2}(J_w + n^2 J_m) + ML^2\sin^2\psi\right) + 2ML^2\dot{\psi}\dot{\varphi}\sin\psi\cos\psi
\end{aligned}$$

where the various Fs are the forces acting on these angles. They are given by

$$\begin{aligned}
F_\theta &= \alpha(v_l + v_r) - 2(\beta + f_w)\dot{\theta} + 2\beta\dot{\psi} \\
F_\psi &= -\alpha(v_l + v_r) + 2\beta\dot{\theta} - 2\beta\dot{\psi} \\
F_\varphi &= \alpha(W/2R)(v_r - v_l) - (W^2/2R^2)(\beta - f_w)\dot{\varphi}
\end{aligned}$$

with v_l and v_r are the voltages applied to the left and right motors, respectively, and

$$\alpha = n\frac{k_t}{R_m} \quad \text{and} \quad \beta = n\frac{k_t k_b}{R_m} + f_m,$$

Table 1: Parameter and their respective values.

Symbol	Parameter description	Value	Units
M	Mass of whole robot	0.6	kg
m	Mass of wheels	0.03	kg
L	Distance to center of mass	0.125	m
R	Wheel radius	0.028	m
W	Body width	0.126	m
D	Body depth	0.07	m
g	Acceleration due to gravity	9.81	$\mathrm{m/s^2}$
R_m	Resistance	6.83	Ω
J_ψ	Moment of inertia of robot	$ML^2/3$	$\mathrm{kg\,m^2}$
J_φ	Moment of inertia of body yaw	$M(W^2+D^2)/12$	$\mathrm{kg\,m^2}$
J_w	Moment of inertia of wheels	$mR^2/2$	$\mathrm{kg\,m^2}$
J_m	Moment of inertia of DC motors	10^{-5}	$\mathrm{kg\,m^2}$
k_t	Torque constant	0.305	Nm/A
k_b	EMF constant	0.556	V/(rad/s)
f_m	Friction coefficient between DC motor and body	0.0022	
f_w	Friction coefficient between wheel and floor	0	
n	Gear ratio	1	

The various parameters are there corresponding values are given in Table 1.

3.1 Creating a Simulink representation of the nonlinear model

The way to represent a general second order model like that described above, is to use two integrator elements and to connect them in series. For example, in the general equation

$$F = m\ddot{x} + \gamma\dot{x} + kx \tag{1}$$

we would implement two integrators. The first represents the equation

$$\dot{x} = \int \ddot{x}\,\mathrm{d}t$$

and the second is

$$x = \int \dot{x}\,\mathrm{d}t.$$

The first integrator connects directly to the second as shown in Fig. 2:

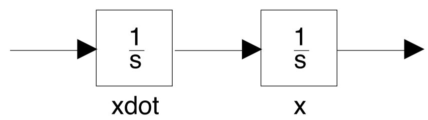


Fig. 2. Simulink representation the integrator cascade. The first integrator takes $\ddot{x}$ as input and gives $\dot{x}$ as output. The second takes the latter as the input and outputs x. Note that this makes signals $\dot{x}$ and $\ddot{x}$ available.

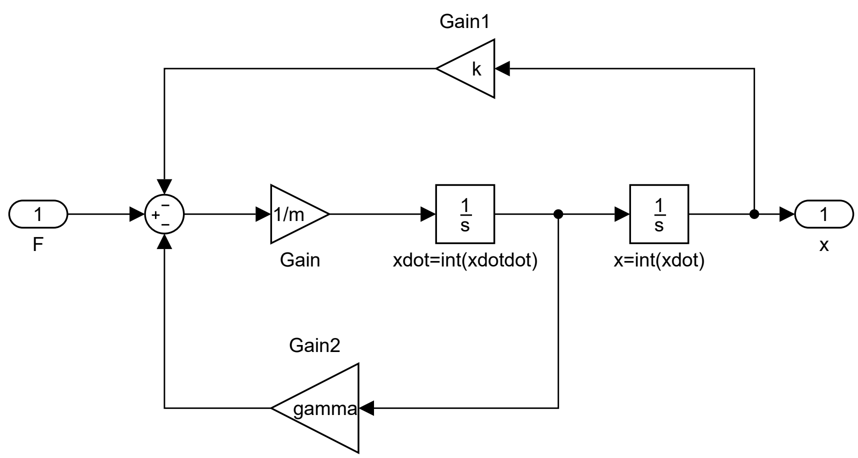


Fig. 3. Simulink representation of the equation $F = m\ddot{x} + \gamma\dot{x} + kx$. The summation element takes $F - kx - \gamma\dot{x}$. This is then fed through the gain $1/m$ to generate $\ddot{x}$ which goes through two integrators, leading to x. Note how the terms $\gamma\dot{x}$ and kx are generated through feedback loops.

As, the input to the first integrator equals $\ddot{x}$, this can be obtained by isolating this term in (1) as follows:

$$\ddot{x} = \frac{1}{m}\big(F - \gamma\dot{x} - kx\big).$$

The terms on the right-hand side represent the output of a summation (of the three terms: F, $\gamma\dot{x}$ and kx) which then go through a gain equal to $1/m$, as shown in Fig. 3.

Now, in the example above the terms are all linear. This can be easily changed to accomodate nonlinear terms. For example, if we have

$$F = m\ddot{x} + \gamma\big(\dot{x}\big)^2 + k\sin x \tag{2}$$

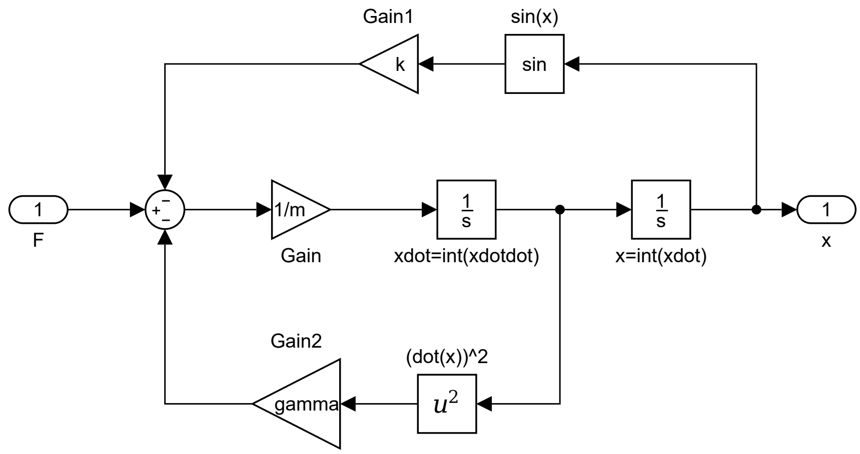


Fig. 4. Simulink representation of the equation $F = m\ddot{x} + \gamma\big(\dot{x}^2 + k\sin x\big)$. Note how the two terms in the right continue to involve feedback connections of the states x and $\dot{x}$ though these now go through nonlinearities.

all we need to do is to pass through the signals through their respective nonlinear components, as shown in Fig. 4.

Segway. To describe the Segway model requires six integrators, for the six states θ, $\dot{\theta}$, ψ, $\dot{\psi}$, φ and $\dot{\varphi}$. However, you will quickly see that while you can rewrite the equation for $\ddot{\varphi}$ you can't do this for $\ddot{\theta}$ and $\ddot{\psi}$ since they depend on each other. To circumvent this, rewrite these two equation as

$$\begin{bmatrix} m_1 & m_2(\psi) \\ m_2(\psi) & m_3 \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} F_\theta + MLR\dot{\psi}^2 \sin\psi \\ F_\psi + ML\sin\psi(g + L\dot{\varphi}^2\cos\psi) \end{bmatrix}$$

The matrix

$$\begin{aligned} \tilde{M} &= \begin{bmatrix} m_1 & m_2(\psi) \\ m_2(\psi) & m_3 \end{bmatrix} \\ &= \begin{bmatrix} (2m+M)R^2 + 2J_w + 2n^2J_m & MLR\cos\psi - 2n^2J_m \\ MLR\cos\psi - 2n^2J_m & ML^2 + J_\psi + 2n^2J_m \end{bmatrix} \end{aligned}$$

has determinant

$$\begin{aligned}\Delta_M &= \left((2m+M)R^2 + 2J_w + 2n^2J_m\right)\left(ML^2 + J_\psi + 2n^2J_m\right) - \left(MLR\cos\psi - 2n^2J_m\right)^2\\ &= \left(2mR^2 + 2J_w + 2n^2J_m\right)\left(ML^2 + J_\psi + 2n^2J_m\right) - 4n^4J_m^2\\ &\quad + MR^2\left(ML^2 + J_\psi + 2n^2J_m\right) - (MLR\cos\psi)^2 + 4MLRn^2J_m\cos\psi\\ &= \left(2mR^2 + 2J_w\right)\left(ML^2 + J_\psi\right) + 2n^2J_m(2mR^2 + 2J_w + ML^2 + J_\psi)\\ &\quad + MR^2\left(J_\psi + 2n^2J_m\right) + M^2R^2L^2(1-\cos^2\psi) + 4MLRn^2J_m\cos\psi.\end{aligned}$$

If is invertible, then its inverse is given by

$$\tilde{M}^{-1} = \frac{1}{m_1m_3 - (m_2(\psi))^2}\begin{bmatrix} m_3 & -m_2(\psi) \\ -m_2(\psi) & m_1 \end{bmatrix}$$

and we can now write

$$\begin{bmatrix}\ddot{\theta}\\ \ddot{\psi}\end{bmatrix} = \frac{1}{m_1m_3 - (m_2(\psi))^2}\begin{bmatrix} m_3 & -m_2(\psi) \\ -m_2(\psi) & m_1 \end{bmatrix}\begin{bmatrix} F_\theta + MLR\dot{\psi}^2\sin\psi \\ F_\psi + ML\sin\psi(g + L\dot{\varphi}^2\cos\psi)\end{bmatrix}$$

to isolate $\ddot{\theta}$ and $\ddot{\psi}$. This now allows one to create a Simulink model of the Segway.

3.2 Linearizing the Model and Creating a State-Space Representation

The next step is to linearize the above equations in order to implement a control system. This can be done by approximating the values of $\sin(\psi)$ and $\cos(\psi)$ as ψ and 1, respectively, and assuming that $\dot{\psi}$ and $\dot{\varphi}$ are 0. While the linear model is not accurate for large angles of ψ, it *is* valid in this case because it is accurate for small values of ψ which is generally what we are dealing with. This yields the equations

$$\begin{aligned} F_\theta &= ((2m+M)R^2 + 2J_w + 2n^2J_m)\ddot{\theta} + (MLR - 2n^2J_m)\ddot{\psi}\\ F_\psi &= (MLR - 2n^2J_m)\ddot{\theta} + (ML^2 + J_\psi + 2n^2J_m)\ddot{\psi} - MgL\psi\\ F_\varphi &= (mW^2/2 + J_\varphi + (W^2/2R^2)(J_w + n^2J_m))\ddot{\varphi}\end{aligned}$$

Then going through the same process as the nonlinear model, we can isolate $\ddot{\theta}$ and $\ddot{\psi}$ and find that

$$\begin{bmatrix} m_1 & m_2 \\ m_2 & m_3\end{bmatrix}\begin{bmatrix}\ddot{\theta}\\ \ddot{\psi}\end{bmatrix} = \begin{bmatrix} F_\theta \\ F_\psi + MgL\psi\end{bmatrix}$$

where

$$\tilde{M} = \begin{bmatrix} m_1 & m_2 \\ m_2 & m_3\end{bmatrix} = \begin{bmatrix} (2m+M)R^2 + 2J_w + 2n^2J_m & MLR - 2n^2J_m \\ MLR - 2n^2J_m & ML^2 + J_\psi + 2n^2J_m\end{bmatrix}$$

We then can take the inverse, which is the same as before, and find that

$$\begin{bmatrix}\ddot{\theta}\\ \ddot{\psi}\end{bmatrix} = \frac{1}{m_1m_3-(m_2)^2}\begin{bmatrix}m_3 & -m_2\\ -m_2 & m_1\end{bmatrix}\begin{bmatrix}F_\theta\\ F_\psi + MgL\psi\end{bmatrix}$$

meaning that

$$\begin{aligned}\ddot{\theta} &= \frac{m_3F_\theta - m_2(F_\psi + MgL\phi)}{m_1m_3-(m_2)^2}\\ \ddot{\psi} &= \frac{-m_2F_\theta + m_1(F_\psi + MgL\psi)}{m_1m_3-(m_2)^2}\\ \ddot{\varphi} &= \frac{F_\varphi}{(mW^2/2) + J_\varphi + (W^2/2R^2)(J_w + n^2J_m)}.\end{aligned}$$

We can now use these equations to create a state-space representation of the system of the form

$$\begin{aligned}\dot{x} &= Ax + Bu\\ y &= Cx + Du\end{aligned}$$

where the states are

$$\begin{bmatrix}\theta & \dot{\theta} & \psi & \dot{\psi} & \varphi & \dot{\varphi}\end{bmatrix}^T$$

and the inputs are

$$\begin{bmatrix}v_l & v_r\end{bmatrix}^T.$$

By substituting the values of F_θ, F_ψ, and F_φ into the equations, we find that after a little manipulation

$$\ddot{\theta} = -\tfrac{2\beta(m_2+m_3)+2m_3f_w}{m_1m_3-(m_2)^2}\dot{\theta} - \tfrac{m_2MgL}{m_1m_3-(m_2)^2}\psi + \tfrac{2\beta(m_2+m_3)}{m_1m_3-(m_2)^2}\dot{\psi} + \tfrac{\alpha(m_2+m_3)}{m_1m_3-(m_2)^2}(v_l+v_r) \quad (3)$$

$$\ddot{\psi} = \tfrac{2\beta(m_1+m_2)+2m_2f_w}{m_1m_3-(m_2)^2}\dot{\theta} + \tfrac{m_1MgL}{m_1m_3-(m_2)^2}\psi - \tfrac{2\beta(m_1+m_2)}{m_1m_3-(m_2)^2}\dot{\psi} - \tfrac{\alpha(m_1+m_2)}{m_1m_3-(m_2)^2}(v_l+v_r) \quad (4)$$

$$\ddot{\varphi} = \tfrac{-W^2(\beta-f_w)}{2R^2m_4}\dot{\varphi} + \tfrac{W\alpha}{2Rm_4}(v_r - v_l) \quad (5)$$

where

$$m_4 = \frac{mW^2}{2} + J_\varphi + \frac{W^2}{2R^2}(J_w + n^2J_m).$$

We can then easily find that the A and B matrices are

$$A = \begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0\\ 0 & -\frac{2\beta(m_2+m_3)+2m_3f_w}{m_1m_3-(m_2)^2} & \frac{-m_2MgL}{m_1m_3-(m_2)^2} & \frac{2\beta(m_2+m_3)}{m_1m_3-(m_2)^2} & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & \frac{2\beta(m_1+m_2)+2m_2f_w}{m_1m_3-(m_2)^2} & \frac{m_1MgL}{m_1m_3-(m_2)^2} & \frac{-2\beta(m_1+m_2)}{m_1m_3-(m_2)^2} & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 0 & \frac{-W^2(\beta-f_w)}{2m_4R^2}\end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ \frac{\alpha(m_2+m_3)}{m_1m_3-(m_2)^2} & \frac{\alpha(m_2+m_3)}{m_1m_3-(m_2)^2} \\ 0 & 0 \\ \frac{-\alpha(m_1+m_2)}{m_1m_3-(m_2)^2} & \frac{-\alpha(m_1+m_2)}{m_1m_3-(m_2)^2} \\ 0 & 0 \\ -\frac{W\alpha}{2m_4r} & \frac{W\alpha}{2m_4r} \end{bmatrix}$$

Now that we have the A and B matrices, we can move on to designing a controller for our system.