Balancing EduMiP
MAE144: Embedded Control & Robotics

Due: Friday, Dec 18th 5:00pm

EduMiP Modeling

Referring to Example 17.10 in Numerical Renaissance, the final equations of motion for the MiP are as follows.

$$\left(I_w + (m_w + m_b)r^2\right)\ddot{\phi} + (m_b r l \cos\theta)\ddot{\theta} - (m_b r l \sin\theta)\dot{\theta}^2 = \tau \tag{1a}$$

$$(m_b r l \cos\theta)\ddot{\phi} + \left(I_b + m_b l^2\right)\ddot{\theta} - m_b g l \sin\theta = -\tau \tag{1b}$$

You will notice these are functions of torque, whereas we wish to design a controller that outputs a PWM duty cycle to our motors. Therefore we must also include the dynamics of the motors themselves. We can reasonably model a geared DC motor's output torque as a function of its speed. Since there are two motors we include a coefficient of two in front of the equation. The motor specs listed below are for the simple DC motor without a gearbox, so we must include the gearbox ratio G in the motor model too. You can solve for the torque constant k with the below provided stall torque $\bar{s}$ and free run speed ω_f.

$$\tau = 2G(\bar{s}u - kw_m) \tag{2a}$$

$$\omega_w = \dot{\phi} - \dot{\theta} = \omega_m / G \tag{2b}$$

Where:
$\bar{s}$ = motor stall torque,
k = motor constant,
ω_w = wheel speed,
ω_m = motor armature speed,
G = gearbox ratio, and
u = normalized motor duty cycle $\in [-1, 1]$

Because we don't have the actual EduMiP hardware, we will assume a hypothetical bulked up "Cyber-MiP", which has the following physical properties:

1. Encoder disks have 50 slots and therefore provide 200 counts per motor armature revolution
2. Nominal battery voltage $V_b = 7.4$V
3. Motors (without gearbox) have free run speed of $\omega_f = 3260$rad/s at V_b
4. Motors (without gearbox) have stall torque $\bar{s} = 0.010$N m at V_b
5. Motors are connected to the wheel with a gearbox ratio $G = 20.5$.
6. The motor armature has inertia $I_m = 9.2 \times 10^{-8}$Kg $\cdot$ m^2
7. Wheels have a radius $r = 62$mm
8. Wheels have a mass $m_w = 50$g each
9. Total assembled MiP body has a mass $m_b = 280$g
10. MiP center of mass to wheel axis $l = 107.3$mm
11. MiP body inertia $I_b = 6.23 \times 10^{-4}$ Kg $\cdot$ m^2

Just as we had to compensate for the torque of both motors accounting for the gearbox, we must also take the gearbox into account when modeling the inertia of the wheels. Attaching a rotating inertia through a gear ratio instead of directly attaching it to a shaft increases the effective inertia by the square of the gearbox. By modeling the wheels as disks and adding their inertia to the effective inertia of the motor armatures, we arrive at the following estimate of the combined wheel inertia.

$$I_w = 2\left(\frac{m_w r^2}{2} + G^2 I_m\right) \tag{3}$$

Assignment:

Your final project will consist of a controller design for the mobile inverted pendulum. Your final submission will consist of a written report, and the controller design Matlab code. The submission will detail your controller design and the Matlab code used in that design process. See the *task* sections below for details on what must be included in the report. You may reference Numerical Renaissance but your report should otherwise be self-sustained. Please submit the report as a PDF and all Matlab code as .m files on Canvas. Be sure your Matlab files are well commented. Your code and report should be entirely your own, copying is not allowed. We will be posting a simulation code/app/website you can use to validate your design.

Problem 1 - Stabilizing Body Angle

Task:
Using the physical properties given above and the equations of motion provided in Example 17.6 of Numerical Renaissance, develop a model $G_1(s)$ as a transfer function from normalized duty cycle u of the motors to angle θ of the MiP body in radians. *I suggest plotting the impulse response of $G_1(s)$ as a sanity check to make sure your transfer function is unstable and models the MiP falling over backwards at a realistic speed.*

Now design a stabilizing transfer function $D_1(s)$ to keep the MiP upright.

What to Submit:

1. Transfer function $G_1(s)$ from motor duty cycle u to angle θ. Please evaluate all coefficients and provide numeric constants (with a reasonable number of significant digits).
2. Your design for a stabilizing controller $D_1(s)$.
3. Bode plot of the open-loop system $G_1(s)D_1(s)$ indicating phase and gain margin and any significant features of your design.
4. Step response of the closed-loop system.
5. Discrete time equivalent controller $D_1(z)$ for a 200Hz sample rate.

Problem 2 - Stabilizing Wheel Position

Task:
Design a second controller, $D_2(s)$, to stabilize the wheel position ϕ. Since the dynamics of θ and ϕ are very different, you may use the successive loop closure method described in section 18.3.4 of Numerical Renaissance.

You may initially design the controller $D_2(s)$ assuming the inner loop has a constant gain of 1. This is to say that we assume the feedback of $D_1(s)G_1(s)$ is sufficiently fast that we treat it as 1 for the purpose of simplifying the design of $D_2(s)$ at a much slower timescale.

When you think you have a stabilizing controller, include your model for $G_1(s)$ and controller $D_1(s)$ to check performance. You may need to add a prescaler P to the output of your controller $D_2(s)$ if the feedback of $D_1(s)G_1(s)$ has steady state gain not equal to 1. With the entire successive loop closure system modeled in Matlab, plot the step response for a unit step in wheel position. This should model the MiP wheels moving forward one radian. You should observe non-minimum phase behavior where the wheels must roll backwards briefly before rolling forward.

What to Submit:

1. Your transfer functions $G_2(s)$ and controller $D_2(s)$.
2. Root locus demonstrating stability of outer loop with inner loop gain of 1.
3. Continuous time step response assuming inner loop gain of 1.
4. Continuous time step response with your $G_1(s)$ and $D_1(s)$ inner loop model included.
5. Open-loop bode plot showing gain and phase margins with the inner loop model included.
6. Discrete time transfer function $D_2(z)$ for a sample rate of 20Hz.