Final Project (ME-190, Fall 2016)

Implementation due date: Last lab (Dec. 6-8)

Report due date: Monday, Dec. 12, 2016, 5:00 pm

Project Objectives:

1- Develop equations of motion (for the MinSeg robot) using the fundamental laws of physics
2- Design, simulate, and implement a controller from the free body diagram to a working system
3- Integrate theory, hardware, and software for a mechatronic system
4- Learn the value of model-based mechatronic design
5- Write a technical report

Part 1. Modeling the MinSeg robot (Expected to be done by Nov. 29, 2016)

This section focuses on developing a complete **State-Space** model for the MinSeg robot. Figure 1 shows the electrical circuit as well as the rigid-body diagram of the robot. Using Kirchhoff's voltage law and the rigid body kinematics and kinetics analysis, we can develop a set of differential equations representing the system's electromechanical dynamics.

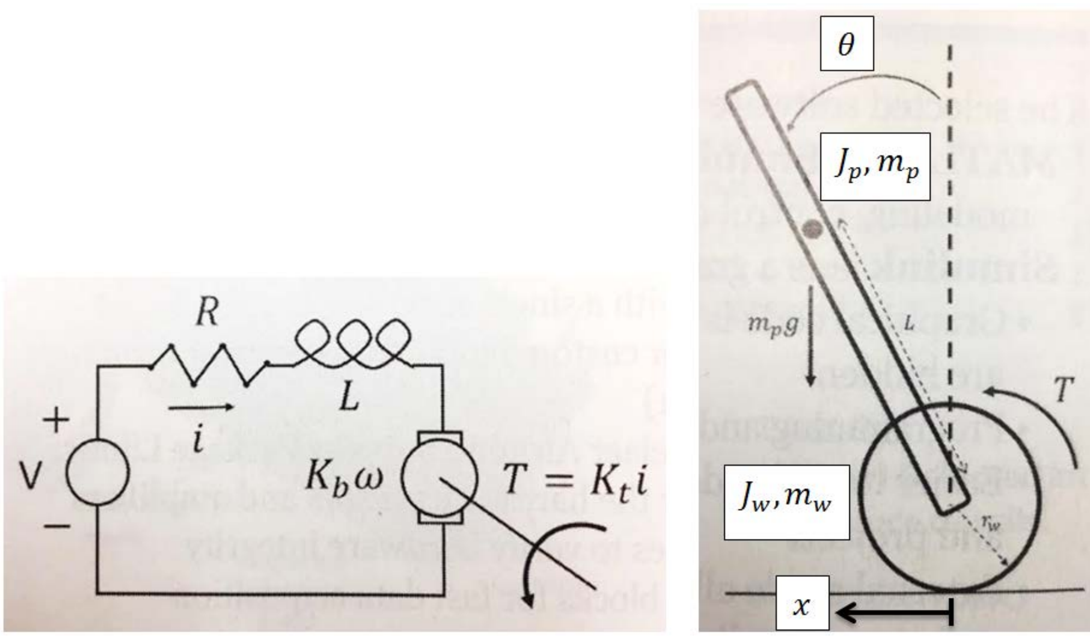

Figure 1. Electro-Mechanical diagrams of the MinSeg robot.

Tasks to be addressed in the report:

1.1. Sketch the free body diagrams of the mechanical sub-systems (the wheel and the body), and apply Newton's law to derive the equations of motion. Carry out the relative acceleration analysis from the rigid-body kinematics to derive the equations of motion for the mechanical system.

Note: Write the equations in terms of the position of the wheel center, x, and the angular deflection of the robot's body, α.

1.2. (Optional) Derive the equations of motion for the mechanical system using Lagrange's method, and verify with those obtained from part 1.1.

1.3. Use Kirchhoff's voltage law and write the differential equation of the electrical circuit.

1.4. Combine the mechanical and electrical equations assuming the coil inductance is zero, i.e., $L_m = 0$.

1.5. Assume the deflection angle Θ will remain small during the robot's control in the upright position. Replace $sin(\Theta)$ with Θ; $cos(\Theta)$ with 1; and $\dot{\Theta}^2$ with 0. This process will convert the nonlinear model to a linear one which is only accurate for small deflection angles. Write the linearized equations in the following matrix form:

$$M\ddot{X} + D\dot{X} + KX = Fu$$

where

$$X = \begin{bmatrix} x \\ \Theta \end{bmatrix}.$$

M , D, and K are 2×2 matrices representing the system's inertia, damping, and stiffness.

F is a 2×1 input coefficient vector.

u is the system's input (i.e., the applied voltage).

Hint: Your linear equations will look something like these:

$$\left(m_p + m_w + \frac{J_p}{r_w^2}\right)\ddot{x}(t) + (??)\ddot{\theta}(t) + \cdots = (??)V$$

$$(??)\ddot{x}(t) + \left(J_p + m_p L^2\right)\ddot{\theta}(t) + \cdots = (??)V$$

1.6. Show that one can convert the above matrix equation to the state-space form using the following formula:

$$\dot{q}(t) = Aq(t) + Bu(t)$$

where

$$q(t) = \begin{bmatrix} X \\ \dot{X} \end{bmatrix}, \qquad A = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ -M^{-1}K & -M^{-1}D \end{bmatrix}, \qquad B = \begin{bmatrix} \mathbf{0} \\ M^{-1}F \end{bmatrix}$$

where $\boldsymbol{0}$ and $\boldsymbol{I}$ are zero and identity matrices of appropriate dimensions.

Hint: start from $q(t)$ and take its time derivative and … .

1.7. Write a Matlab script that converts the MinSeg robot equations to the state-space form using the above formula for a set of arbitrary system parameter values. Use some of the parameter values from

Lab 8, and carry out a simple engineering analysis or use your best guess for the other parameters for a preliminary simulation.

1.8. Simulate the state-space system in simulink for a small initial angle α (e.g., $\alpha = \pi/60$) for zero input voltage. For the output equation use:

$$y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} q(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(t)$$

Plot and report the output response, and discuss whether the response makes sense based on the physics of the system and the eigenvalues of the state matrix, *A*.

Part 2. Designing a Linear Quadratic Regulator (LQR) Controller (To be updated)

Part 3. Implementation of the LQR controller (To be updated)