Legged Mobility Assignment

Th legged mobility assignment is structured in three parts, covering fundamentals, actuation and control of legged robots. The assignment is due one week after the legged mobility lectures have ended. However, we strongly recommend you attack the problem set in parallel to the lectures, which will help you deepen your understanding of the lecture content as well as spread out the time burden of the assignment.

Please follow the TA announcements for submitting your solution. In general, present all intermediate steps of your solution in a clear and logical fashion of your write-up and submit any required Matlab implementation code in addition. Note that the grading points will be scaled at the end of the course to match the scoring scale of all problem sets.

Part 1–Fundamental Models: Linear inverted pendulum model with foot (13+2pts)

Part 1 of the legged mobility assignment will help you deepen and expand your understanding of the linear inverted pendulum model as it is being used in humanoid robotics. After completion of this part, you should have a solid understanding of the model and the associated idea of preview control, and be ready to derive basic control strategies for the center of mass motion of a humanoid in walking.

A linear inverted pendulum model (LIPM) is often at the core of a humanoid controller. In contrast to the LIPM introduced in class, however, the LIPM used in humanoid robotics includes a foot with an actuated ankle joint incorporating the center of pressure in control decisions. Figure 1 shows this extended model. It consists of a point mass m actuated by a massless leg (length l) with one prismatic joint (actuator force f_l) and one revolute joint (angle θ) representing the ankle-foot (actuator torque τ with resulting force f_τ on point mass). Your goal is to derive the equation that describes the model's horizontal motion, assuming the vertical position stays at $y = y_0$ for all times.

1. Use simple vector decomposition of $\boldsymbol{f}_l$, $\boldsymbol{f}_\tau$, and the gravitational force $\boldsymbol{f}_g = m\boldsymbol{g}$ to write down the net forces f_x and f_y acting on the point mass in the x and y directions (2pts).

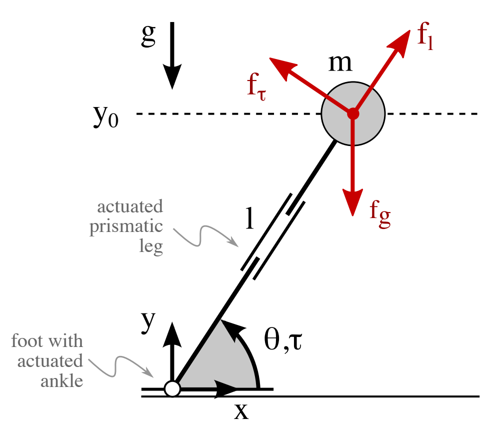

Figure 1: Linear inverted pendulum with ankle-foot complex.

2. Use the definitions of $\cos\theta$ and $\sin\theta$ as well as the relationship between τ and f_τ to represent f_x and f_y as functions of x, y_0, l, mg, f_l and τ. (2pts)

3. Solve for f_l using the horizontal motion constraint requiring zero *net* force in the vertical direction: $f_y = 0$. (The vertical leg or ground reaction force is not zero.) (2pts)

4. Substitute your solution for f_l back into the equation for computing f_x and reduce f_x to a function of x, y_0, mg, and τ. (2pts)

5. With f_x from the previous step, write down the equation of motion in the horizontal direction, $m\ddot{x} = f_x$, and use your knowledge about the relationship between the ankle actuation torque τ and the center of pressure p to resolve the equation to

$$m\ddot{x} = \frac{mg}{y_0}(x - p). \tag{1}$$

(2pts)

6. Use the relationship between kinetic energy and work performed on the mass and calculate the capture point x_T of the model, assuming an initial velocity $v_0 > 0$, a final velocity $v_e = 0$, and a constant center of pressure p. Plot the capture point as a function of initial velocity and center of pressure (3-D surface plot). Use the plot and reasonable assumptions about a humanoid's foot geometry to discuss under what conditions the robot should make a step or rely on the ankle strategy to resume standing balance. (3pts)

7. Optional: Repeat steps one through six for a LIPM with an ankle-foot complex and a hip-flywheel complex. The latter incorporates the hip strategy in control decisions. Use the paper "*Capture Point: A Step toward Humanoid Push Recovery*" by Pratt and colleagues (2006, IEEE Humanoids Conference) for guidance on the hip-flywheel complex and its influence on model dynamics (section V.A-B in the paper). For step 6, consider the influence of the initial velocity, the center of pressure, and the hip torque on the capture point. (+3pts)

Additional resource: The paper "*The 3D linear inverted pendulum model mode: A simple modeling for a biped walking pattern generation*" by Kajita and colleagues (IEEE IROS, 2001) may be a helpful resource for deriving the equation of motion in x with the center of pressure included, although the explicit derivation differs a bit from steps one through five.

In the third part of the legged mobility assignment, you will design a controller for the motion of a humanoid's center of mass. This controller will use the center of pressure p as a control input to achieve a desired motion $[x\,\dot{x}]^T_{des}$ of the robot center of mass. The equation (1) you derived here will be central to the control design.

Part 2–Actuation: Series elastic actuators in legged mobility (13+3pts)

Part 2 focuses on actuation in legged robots. The assignment will help you (1) learn to pick motors for legged mobility applications and (2) revisit the concept of series elastic actuation, a concept widely used in legged mobility as well as manipulation. After completion of this assignment you should have the basic skills for motor selection and for the control of series elastic actuators (SEAs).

Figure 2 shows a humanoid actuation model that reduces the robot to a lumped mass $M = 80kg$ moving only in the vertical direction on a massless, prismatic leg. A DC motor is attached to the mass. The rotation of the motor is converted into linear motion through a rack and pinion gear box with radius $r = 5cm$ and gear ratio N. The output of the gear box defines the length l_m of the motor and actuates a series linear spring with stiffness $k = 20kNm^{-1}$. The spring itself actuates the piston of the leg which pushes on the ground with the force $F_s = k\Delta l$, where Δl is the current spring compression. The equation of motion of the entire robot simplifies to

$$M\ddot{y} = F_s - Mg, \tag{2}$$

where $g = 9.81ms^{-2}$ is the gravitational acceleration. Your goal is to develop a controller for this series elastic actuator that is suitable for walking based on specific motor and gear selections.

1. Start by choosing the motor and modeling its mechanical dynamics <u>without</u> considering the series spring. The mechanical dynamics of a motor are given by

$$J_m\ddot{\theta} = \tau_m - \tau_{ext}, \tag{3}$$

where J_m is the motor inertia, τ_m is the motor torque, and τ_{ext} is the torque generated by the external load (the torque generated by the spring acting through the rack and pinion gear on the motor). You decide on the 48V version of the Maxon EC90 as the motor of your SEA. Find the spec sheet from Maxon's website or catalog. What is the part number of the motor you selected?

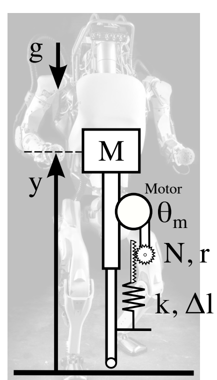

Figure 2: Series elastic actuator of a humanoid. The humanoid is reduced to a lumped mass M moving only in the vertical direction on a massless, prismatic leg. The leg is actuated by a motor connected to the lumped mass and in series with a spring that drives the leg piston. Motor rotation is converted into linear motion through a rack and pinion gear box with radius r and gear ratio N and changes the length of the spring by the amount $\Delta l_m = r\theta_m/N$. Similarly, the spring length changes by the amount $\Delta l_y = y_0 - y$ from the compression of the leg. The resulting spring force $F_s = k(\Delta l_y + \Delta l_m)$ actuates the leg.

What is the motor inertia J_m in units $[kgm^2]$? What is the maximum continuous torque τ_{max} of the motor in units $[Nm]$? (2pts)

2. Next, choose a suitable gear ratio N. In legged mobility, motors are often used cyclically, and the maximum continuous torque can be exceeded. Since the SEA you are developing is actuating one leg, you can expect a 50% duty cycle of motor operation. What maximum torque τ^*_{max} can you expect to generate with the motor given this duty cycle? (The Maxon catalog provides key information on motor selection, operating ranges, and behaviors for dc motors. Specifically, pages 37-57, "Technology short and to the point", discuss among other things the relation between max current (torque) and duty cycle of a motor.) Use τ^*_{max} and derive the gear ratio N that is necessary to develop a peak force of 137% body weight, $1.37Mg$. (3pts)

3. Now add the series leg spring to your motor and gearing to complete your SEA. The spring develops a force $F_s = k\,\Delta l$, where $\Delta l = \Delta l_y + \Delta l_m$ is the net spring compression due to changes in leg length, $\Delta l_y = y_0 - y$, and in motor length, $\Delta l_m = \frac{r}{N}\theta_m$. Relate the linear force of the spring to the external torque τ_{ext} acting on the motor shaft using the gear ratio N and the moment arm r. Substitute the expression for τ_{ext} into the motor equation (3) to get the equation of motion for the motor position θ_m as a function of τ_m and F_s. (2pts)

4. With your actuator mechanics complete, develop a cascaded controller that stabilizes the humanoid robot to the reference height $y = y_0$. First, design an outer-loop <u>behavior controller</u> based on the robot dynamics equation (2). The controller regulates the robot height based on the error $y_{des} - y$ between the desired and current height, and generates a desired leg force F_s^{des} as control output. Second, design an inner-loop <u>SEA controller</u> that commands motor torques τ_m to track the desired force F_s^{des}. This can be done in different ways. For one example, you could use $F_s = k\,\Delta l$ to convert the desired spring force F_s^{des} into a desired motor position θ_m^{des} , and then track this motor position with a P-D control on the error $\theta_m^{des} - \theta_m$. In general, you are free to choose the level of complexity for both the behavior and the SEA controller. You can use simple P-I-D-type feedback control or implement more advanced control schemes that use feedforward dynamics models (for one bonus point). (For the latter, Pratt and Williamson's article "Series elastic actuators" (IEEE International Conference on Intelligent Robots, 1995) is a good resource). Draw a flow diagram of your control and explain its operation. (3+1pts)

5. Finally, simulate the resulting "humanoid" behavior in Matlab and demonstrate the stabilization of the robot for the three target heights $y_{des} = 0.7, 0.8, 0.9m$. Assume initial the conditions $y_0 = 1, \dot{y}_0 = 0, \theta_{m,0} = 0, \dot{\theta}_{m,0} = 0$ and implement Euler integration of the motor dynamics and the dynamics of the lumped mass with your controller active. Make a figure that shows the robot position y and motor torque τ_m over time to demonstrate the stabilization. (3pts)

6. <u>Optional</u>: Develop a model for the thermal dynamics of the motor and use this model to predict the motor temperature over time in the simulation. (The same section "Technology short and to the point" of the Maxon catalog describes a simple model of thermal motor dynamics based on a few key parameters provided in the spec sheet of the EC90.) Does the final steady state temperature stay within acceptable bounds? Does it differ between the target heights? Why is this outcome unrealistic if you consider designing a knee actuator for a real humanoid? (+2pts)

Part 3–Control: Modern Humanoid Control (12pts)

Part 3 focuses on control in legged mobility. The assignment will help you to deepen your understanding of modern humanoid control approaches based on cascaded optimization. You are not required to develop the controller code yourself; it is provided to you along with a simulation of a humanoid. Rather, the goal of the assignment is for you to get some hands on experience with these control approaches in action. After completion of this assignment you should have a more practical understanding of modern humanoid control than can be conveyed in the lectures.

Download and unpack the file humanoid.zip. It contains the control and simulation code. To run the code, make sure the **Control System Toolbox** and the **Optimization Toolbox** are installed in Matlab. Now run the script main.m. You will see the control in action, driving a 5-link humanoid model through one stance phase of its right leg, which lasts about 540ms in this example. Figure 3-a shows this humanoid model at the beginning of the stance phase with some details on segment and joint conventions. Figure 3-b provides an overview of the general control flow.

1. The controller first plans the motion of three elements throughout the upcoming stance phase. The first element is the center of mass (CoM). Its motion is planned using the LIPM without ankle torque. What other two elements are planned for in this section of the code (FP and TR)? (2pts)

2. The controller then progresses through stance time (for loop). At each time step, the three plans are modified reactively. For instance, the CoM plan is modified based on the current humanoid CoM state (position and velocity). Model predictive control (MPC) optimization of the LIPM with ankle control (part 1 of this assignment) is used to bring the CoM back to the originally planned CoM motion should deviations occur. How are the plans of FP and TR modified? (2pts)

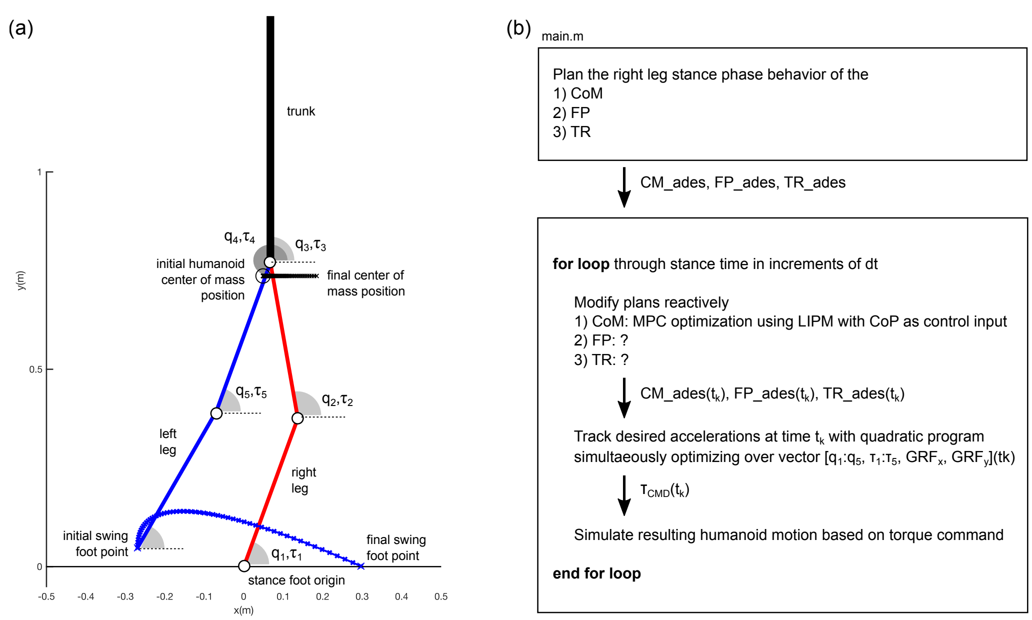

Figure 3: Schematic of humanoid (a) and its control based on optimization (b).

3. The modified plans at time step t_k are then used as desired outcomes in the cost function (H and f terms) of a quadratic program optimization,

$$\min_x \ \frac{1}{2}x^T H x + f^T x,$$
$$s.t. \ \ M(q)\ddot{q} + C(q,\dot{q}) + N(q) = \tau,$$
$$\ldots$$

which finds the 12 element vector $x(t_k) = [q_{1...5}, \tau_{1...5}, GRF_x, GRF_y]^T(t_k)$ that minimizes these costs while simultaneously satisfying imposed constraints. Familiarize yourself a bit more with the idea of a quadratic program optimization using Matlab's documentation on the command `quadprog`. With this knowledge, add an inequality constraint on the horizontal force GRF_x at the bottom of the function `QP_BuildConstraints.m` (in the `qp` subfolder) to satisfy the friction cone requirement

$$-\mu GRF_y \ \leq \ GRF_x \ \leq \ \mu GRF_y$$

with the friction coefficient $\mu = 0.8$.

To do this, consider that Matlab's `quadprog` command expects inequality constraints in the form

$$A_{ineq}x \leq b_{ineq}$$

and first reformulate $GRF_x \leq \mu GRF_y$ into the matrix form $[a_{11}\ a_{12}]\ [GRF_x\ GRF_y]^T \leq b_1$, where a_{11} and a_{12} are matrix elements of A_{ineq} and $b_{ineq} = b_1 = 0$. Then reformulate $-\mu GRF_y \leq GRF_x$ into a similar form $[a_{21}\, a_{22}][GRF_x GRF_y]^T \leq b_2$ with $b_2 = 0$. Finally, combine the two inequality constraints into matrix form and pad with zeros to account for the other 10 elements of the optimization vector x, leading to

$$A_{ineq} = \begin{bmatrix} 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,a_{11}\ a_{12} \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,a_{21}\ a_{22} \end{bmatrix} \qquad b_{ineq} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

(3pts)

4. Study the effect of changing μ on the humanoid behavior. Changing μ is telling the controller what ground condition it can expect. Set μ to 0.1 (slippery) and then to 0.01 (basically don't generate horizontal forces). With vertical GRF of about 400N, confirm with Matlab figure 3 generated by the code that your horizontal GRF max out at about 40N and 4N, respectively. How does this limitation on the horizontal GRF affect the tracking of the originally planned CoM motion (Matlab figure 2) in each case? (2pts)

5. Of the 12 element vector x that the humanoid model optimization generates, only the resulting torques $\tau_{1...5}$ are sent as command to the humanoid joint motors. The resulting humanoid behavior is simulated in the function `SIM_SimulateHumanoid.m` (in the `sim` subfolder). This function also has code for simulating (a) a horizontal disturbance push to the humanoid CoM

at about 200ms into the stance phase (time index 20) and (b) actuator noise (code lines 40 to 50). Set the friction coefficient μ back to 0.8 and then play around with these two sources of uncertainty in humanoid control. What happens if a disturbance impulse of +20Ns is applied: Does the humanoid still follow its intended plan? If so, how is that accomplished (Matlab figure 2). How does actuator noise influence tracking the behavior goals (Matlab figure 3)? (3pts)