Simulation Lab #1:
Dynamic Simulation of Jumping

Professor B.J. Fregly
Mechanics of the Human Locomotor System
EML 5595 - Fall 2005

Lab and Jumping Software Developers: Jeff Reinbolt and B.J. Fregly
Computational Biomechanics Lab, University of Florida
Derived from a similar simulation lab developed at Stanford University by
Clay Anderson, Allison Arnold, Silvia Blemker, Darryl Thelen, and Scott Delp

I. Introduction

In the study of human movement, experimental measurement is generally limited to the kinematics of the body segments, external reaction forces, and electromyographic (EMG) signals from the muscles. While these data are essential for characterizing movement, they do not help us understand how individual muscles or groups of muscles are coordinated to produce movement. However, knowledge of muscle coordination is essential for quantifying the stresses placed on bones and understanding the functional roles of muscles in normal and pathological movement. Dynamic simulation of the musculoskeletal system provides a means for understanding multijoint coordination as well as a framework for investigating how the various components of the musculoskeletal system interact to produce movement.

The purpose of this lab is to introduce you to some basic issues involved in dynamic modeling and simulation of the human musculoskeletal system. You will use an interactive jumper model to perform repeated dynamic simulations of jumping with the goal of maximizing the jump height of the model. The interactive model gives you manual control over the initial joint angles in the model and the time histories of the net muscle joint torque actuators controlling the motion of each joint. These torque actuators represent the net control effect of all muscles spanning the toe, ankle, knee, and hip joints in the model. In addition, you can turn on and off passive joint torques that account for foot contact with the ground and ligament restraint of joint hyperextension. In both cases, if a joint goes beyond a physically realistic limit (e.g., the foot rotates through the ground or the knee hyperextends), a passive control torque is generated in the opposite direction to counteract the undesirable joint motion. Jumping was chosen as the activity for this lab because it possesses s a well-defined objective (i.e., jump as high as possible) and, although still complex, its muscular coordination is relatively simple compared to walking.

The musculoskeletal model used in this lab was taken from a study published by Pandy *et al*. (1990). While the joint angle definitions in the present model differ from those in that study (we use relative rather than absolute joint angles), the basic dynamical model is almost identical with two exceptions. First, we model the influence of muscles using net torque actuators instead of individual muscle forces, and second, we include the influence of passive ligamentous restraining torques. By working through this lab, you will get a feel for the computational cost of repeated dynamic simulation, develop an understanding of how passive torques can affect the predicted motion, learn about some of the problems that can occur with forward dynamic simulations, and gain insight into how a net torque applied at one joint can influence the motion of all joints in the model. By performing repeated simulations and examining the predicted motions, you will get some exposure to the available from dynamic musculoskeletal models.

II. Objectives

The purpose of this lab is to give you hands-on experience with a simple dynamic model of the human musculoskeletal system. In the course of this lab, you will:

- Find a set of muscle torque controls that produce a well-coordinated jump. The specific aim is to make the musculoskeletal model jump as high as possible while satisfying two constraints at take off: 1) the joints cannot hyper-extend, and 2) the ground reaction forces must be zero.

- Investigate how changing joint torque actuators in isolation and in conjunction with other joint torque actuators affects the predicted jumping motion.

- Compare the maximum jump height and ground reaction forces predicted by your simulation with the forces predicted by Pandy *et al.* (1990) using optimization.

- Investigate the difficulties involved in producing realistic forward dynamic simulations by performing simulations with and without passive ground contact and ligament torques.

- Discuss potential problems that one would encounter if the jumper model was used in an optimization that sought to predict the maximum jump height.

III. Background

Dynamic models of the musculoskeletal system are typically comprised of four important components: 1) the equations of motion for the body, or skeletal dynamics, 2) a representation of musculoskeletal geometry, 3) a model of muscle-tendon mechanics, and 4) a model of activation dynamics. Figure 1 illustrates how these components are combined to execute a forward dynamic simulation. Based on a set of initial states, which include the muscle activations $\vec{a}(t)$, the muscle forces $\vec{f}(t)$, the generalized speeds $\dot{\vec{q}}(t)$, and the generalized coordinates $\vec{q}(t)$, differential equations (See Eqs. (1)-(3) below) are used to compute the time rate of change of the states. Then a numerical integration is performed to compute the states at time $t + dt$. The new states are fed back and the forward dynamics process repeats, advancing the states in time until the final time of the simulation is reached. In the simulations you will conduct in this lab, a variable-step, 5th order implicit integrator is used (ode15s in Matlab).

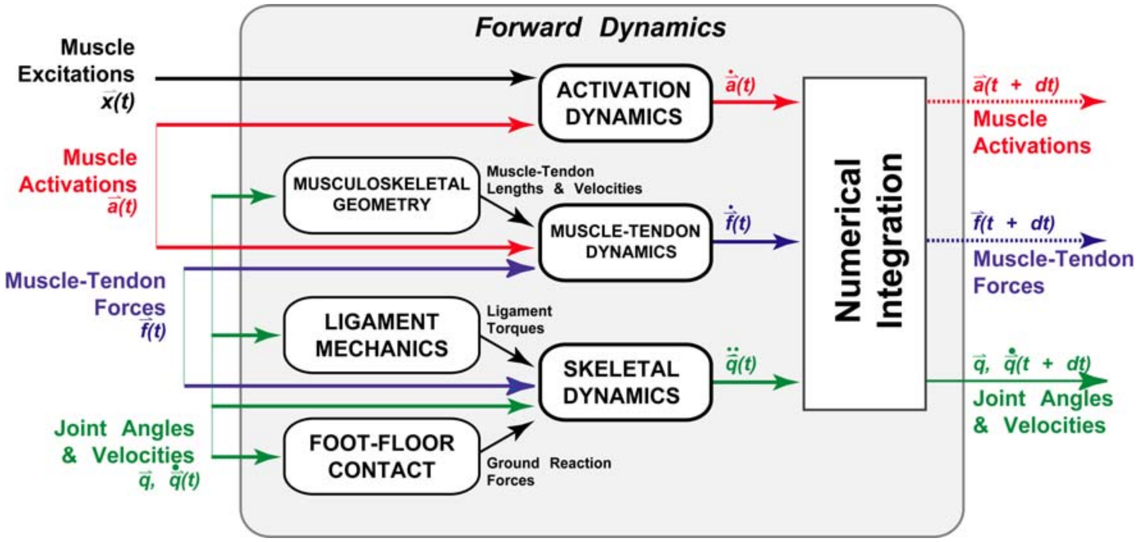

Figure 1. Schematic of a forward dynamic simulation.

Skeletal dynamics

The equations of motion for the body allow one to compute the accelerations of the body segments when forces and torques are applied to the body. The equations of motion can be expressed as follows:

$$\ddot{\vec{q}} = \overleftrightarrow{I}(\vec{q})^{-1} \cdot \{\vec{C}(\vec{q},\dot{\vec{q}}^{2}) + \vec{G}(\vec{q}) + \overleftrightarrow{R}(\vec{q}) \cdot \vec{f}_M + \vec{E}(\vec{q}) \cdot \vec{f}_E\}. \tag{1}$$

Eq. (1) is simply an elaboration of Newton’s third law for a multi-link system, rearranged so that one can compute acceleration (i..e., $a = m^{-1} \cdot f$). The vector of generalized coordinates, $\vec{q}$, is used to specify position and orientation of the body segments. The time derivatives of $\vec{q}$, $\dot{\vec{q}}$ and $\ddot{\vec{q}}$, therefore represent the velocities and accelerations of the segments. Depending on how one chooses to model the body, elements of $\vec{q}$ may be translational displacements, orientations of segments with respect to the lab frame (segment angles), or orientations of segments with respect to other segments (joint angles). Implicit in one’s choice of generalized coordinates are one’s assumptions about the how the joints of the body function. For example, one often models the hip joint as a three degree-of-freedom ball-and-socket joint, which requires three generalized coordinates: flexion-extension (q_1), ab-adduction (q_2), and internal-external rotation (q_3). The system mass matrix, $\overleftrightarrow{I}(\vec{q})$, characterizes the inertial properties of the body (i.e., masses and moments of inertia). The remaining terms in Eq. (1) express the generalized forces or torques that act on the body. $\vec{C}(\vec{q},\dot{\vec{q}}^{2})$ represents centripetal forces that arise from the angular velocities of the segments; $\vec{G}(\vec{q})$ represents gravitational forces; $\overleftrightarrow{R}(\vec{q}) \cdot \vec{f}_M$ represents the moments applied at the joints by the muscles, and $\vec{E}(\vec{q}) \cdot \vec{f}_E$ represents external forces applied to the body such as the ground reaction force. The matrix $\overleftrightarrow{R}(\vec{q})$ is a matrix of moment arms that transform the muscle forces, $\vec{f}_M$, into joint torques. The matrix $\vec{E}(\vec{q})$ performs a similar function for the external forces, $\vec{f}_E$.

Even for simple models, it is often difficult to derive the equations of motion by hand. Consequently, for most biomechanical models, the equations of motion are generated on a computer. The jumping model used in this lab has only 4 degrees of freedom (Pandy *et al.*, 1990), and the equations of motion were generated using Autolev (www.autolev.com), a commercially available symbolic manipulator program developed specifically for deriving dynamics equations. In subsequent labs, you will use SD/Fast and SIMM (*Software for Interactive Musculoskeletal Modeling*) with its Dynamics Pipeline to generate equations of motion for models you develop (Delp *et al*., 1990).

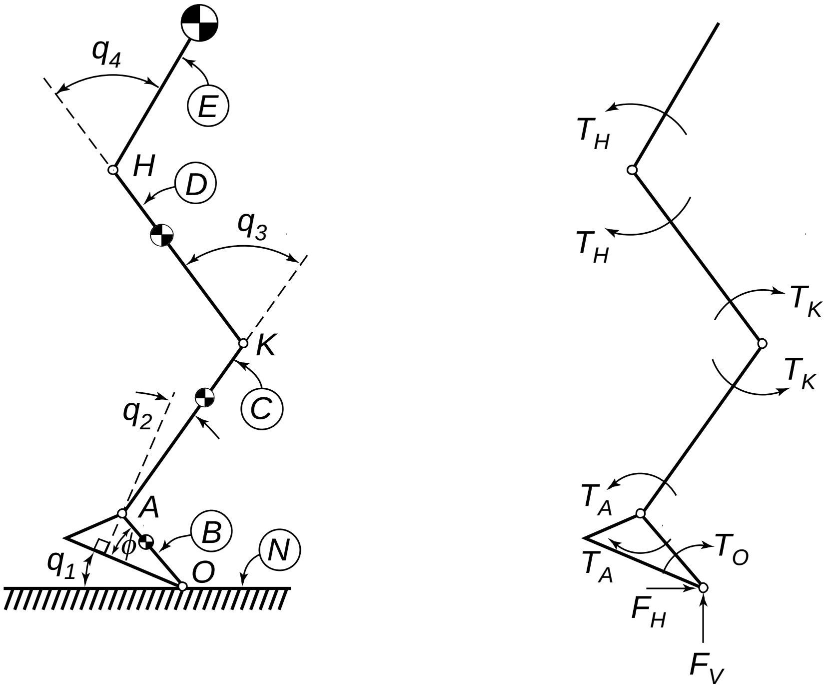

Figure 2. Schematic representation of the four degree-of-freedom planar dynamic jumper model used in this lab. The foot *B* is connected to the ground *N* at point *O* (toes), the shank *C* to the foot *B* at point *A* (ankle), the thigh *D* to the shank *C* at point *K* (knee), and the trunk *E* to the thigh *D* at point *H* (hip). All joints are modeled as frictionless pin joints. q_1 is the toe angle, q_2 the ankle angle, q_3 the knee angle, and q_4 the hip angle, defined as positive in the flexor direction as shown above. The motion of the jumper is actuated by four joint torque controls, a toes torque T_O, ankle torque T_A, knee torque T_K, and hip torque T_H, each positive in the extensor direction as shown in the figure. F_V and F_H are the vertical and horizontal, respectively, components of the ground reaction force vector acting on the foot, which are computed outputs of a forward dynamic simulation performed with the model.

Musculoskeletal geometry

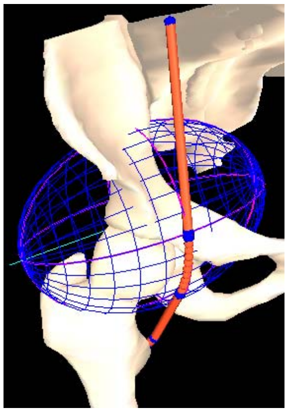

Figure 3. Path geometry of psoas.

Accurately representing the path of a muscle from its origin to its insertion is one of the more challenging aspects of modeling the musculoskeletal system. Sometimes a muscle can be represented as a straight-line path between its origin and insertion. Other times it is adequate to approximate the path as a series of straight-line segments which pass through a series of via points (Delp *et al*., 1990). When modeling muscle paths in three dimensions, it is often necessary to simulate how muscles wrap over underlying bone or musculature. Cylinders, spheres, and ellipsoids have been used as wrapping surfaces (Van der Helm *et al*., 1992; Garner and Pandy, 2000; Arnold *et al*., 2000) (Fig. 3). The jumping model developed for this lab does not use individual muscle actuators. However, In Lab 2, you will use SIMM to specify, alter, and visualize musculoskeletal geometry for a kinematic model you develop.

Muscle-tendon mechanics

The force producing properties of muscle are complex and nonlinear (see McMahon (1984) for review) (Fig. 4). For simplicity, lumped-parameter dimensionless muscle models, capable of representing a range of muscles with different architectures, are most commonly used in dynamic simulation of movement (Zajac, 1989). In a complex musculoskeletal model, the model can be actuated by 50 or more musculotendinous units, each of which is represented as a Hill-type contractile element in series with tendon. The parameters used to characterize each muscle are maximum isometric force F_o^M, optimal muscle fiber length l_o^M, tendon slack length l_S^T, maximum shortening velocity V_{max}^M, and pennation angle α. For a table of the muscle-tendon parameters, consult Anderson and Pandy (1999). During a forward dynamic simulation, muscle force is treated as a state and integrated forward in time using a first-order differential equation of the form

$$\dot{f}^{MT} = \Phi(f^{MT}, l^{MT}, v^{MT}, a), \tag{2}$$

where f^{MT}, l^{MT}, and v^{MT} are the force, length, and velocity of the muscle-tendon actuator, respectively, and Φ is a non-linear function (Zajac, 1989). In Lab 4, you will concentrate specifically on modeling muscle.

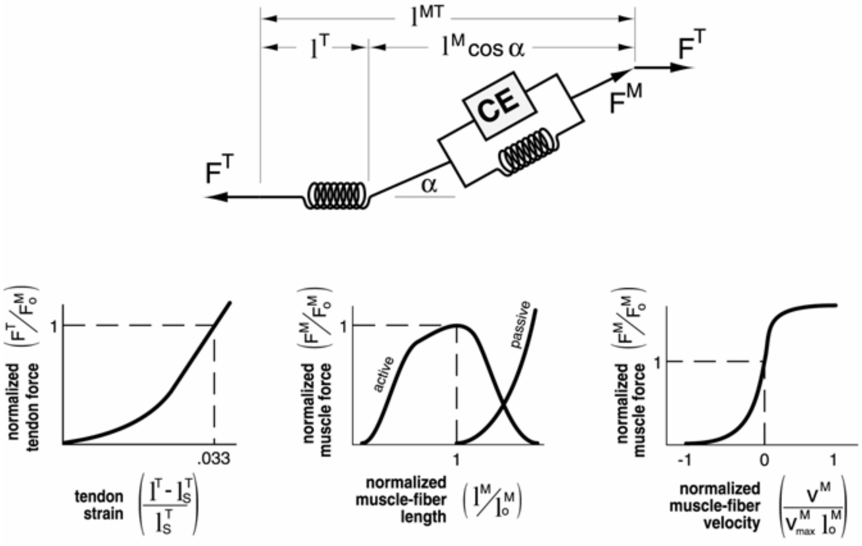

Figure 4. Dimensionless model of muscle and tendon used in our simulations. Muscle properties are represented by an active contractile element (CE) in parallel with a passive elastic element *(top)*. Muscle force is dependent on muscle fiber length *(middle plot)* and velocity *(right plot)*. Muscle is in series with tendon, which is represented by a nonlinear elastic element *(left plot)*. Pennation angle ( ) is the angle between the muscle fibers and the tendon. The forces in muscle and tendon are normalized by peak isometric muscle force (F_o^M). Muscle fiber length (l^M) and tendon length (l^T) are normalized by optimal fiber length (l_o^M). Tendon slack length (l_S^T) is the length at which tendons begin to transmit force when stretched. Velocities are normalized by the maximum contraction velocity of muscle (V_{max}^M). For a given muscle-tendon length (l^{MT}), velocity, and activation level, the model computes muscle force (F^M) and tendon force (F^T).

Activation dynamics

A muscle is not capable of generating force or relaxing instantaneously. The development of force is a complex sequence of events that begins with the firing of motor units and culminates in the formation of actin-myosin cross-bridges within the myofibrils of the muscle. When the motor units of a muscle depolarize, action potentials are elicited in the fibers of the muscle and cause calcium ions to be released from the sarcoplasmic reticulum. The increase in calcium ion concentrations then initiates the cross-bridge formation between the actin and myosin filaments (see Guyton (1986) for review). In isolated muscle twitch experiments, the delay between a motor unit action potential and the development of peak force has been observed to vary from as little as 5 milliseconds for fast ocular muscles to as much as 40 or 50 milliseconds for muscles comprised of higher percentages of slow-twitch fibers. The relaxation of muscle depends on the re-uptake of calcium ions into the sarcoplasmic reticulum. This re-uptake is a slower process than the calcium ion release, and so the time required for muscle force to fall can be considerably longer than the time for it to develop.

In the forward dynamic simulations you will conduct in a subsequent lab, activation dynamics is modeled using a first-order differential equation to relate the rate of change in activation (i.e., the concentration of calcium ions within the muscle) to excitation (i.e., the firing of motor units):

$$\dot{a} = \frac{(x^2 - xa)}{\tau_{rise}} + \frac{(x - a)}{\tau_{fall}}, \tag{3}$$

where a is the activation level of a muscle, x is the excitation level of a muscle, and τ_{rise} and τ_{fall} are the rise and fall time constants for activation, respectively. In the model, activation is allowed to vary continuously between zero (no contraction) and one (full contraction). In the body, the excitation level of a muscle is a function both of the number of motor units recruited and the firing frequency of the motor units. Some models for excitation-contraction coupling distinguish these two control mechanisms (Hatze, 1976), but it is often not computationally feasible to use such models when conducting complex dynamic simulations.

IV. Deliverables

At the completion of each lab, you will need to turn in a written report created from a pre-formatted template that will be provided to you. The written reports are where you will summarize your findings and address questions posed as part of the lab. In addition, for the first lab, you will need to create plots using screen shots taken of the interactive jumper software.

V. Input Files

No special input files are required for this lab. All work can be done interactively using the graphical user interface provided with the jumper model. However, you can save your manual settings (i.e., initial joint angles, values of nodal points defining net muscle joint torques, passive torque settings) from the software and read them back in later to pick up where you left off previously.

VI. Getting Started

Computer work

All labs for the class will be hands-on and require that you use your EML5595 account on the PC workstations in the NEB 109 computer lab. For this lab only, an interactive software program will be downloaded from the course web site and run on your home PC.

Interactive dynamic simulator and muscle torque editor

In this lab, you will use an interactive dynamic simulation program that allows you to edit the time histories of the control torques generated by muscles so as to produce a maximum height jump. The simulator is laid out with four panel areas. The two panel areas on the left hand side contain inputs that can be edited by the user, while the two panel areas on the right hand side contain outputs from each forward dynamic simulation performed with the jumper model.

The top right input panel is the control panel where the user can specify whether to perform a Simulation or an Optimization (only the Simulation selection currently works), whether to include passive ground contact and/or ligament torques, and the initial joint angles for the hip, knee, and ankle. The initial toes angle is always zero to specify that the foot is flat on the ground. Similarly, the initial angular speed of each joint is zero to represent a static initial pose for the model. The five buttons below these selections allow the user to initialize the torque controls, run a forward dynamic simulation, animate the simulation results as many times as desired, save all user settings in separate files, or quit the interactive jumper program. The model can be initialized using default torque values that maintain the jumper in a static pose consistent with the selected passive torques and the default initial joint angles. Alternatively, it can be initialized using previously saved control torque and initial joint angle values written to the hard disk using the save button.

The bottom right input panel allows you to adjust the net muscle control torques. Each curve represents the time history of a joint torque in units of Newton-meters. The red circles are cubic spline nodal points with 10 points per curve. Clicking near (not on) one of the red circles moves the nodal point to that location, changing the shape of the control torque curve in the neighboring region.

The top right output panel provides an animation of the simulated motion generated using the user-specific muscle torque controls and initial joint angles. Just to the right of the animation window is a reporting area that displays important information calculated from the final time frame of the simulation, specifically the normal and tangential contact force and the maximum jump height. Maximum jump height is calculated from the equation below,

$$H_{Max} = H(t_f) + sign(\dot{H}(t_f))\dot{H}(t_f)/2g - H_{Standing} \tag{4}$$

where H_{Max} is the maximum height of the center of mass above its height $H_{Standing}$ when the body is in the upright standing posture (i.e., all joint angles are zero), $H(t_f)$ is the height of the center mass at the final time t_f of the simulation, $\dot{H}(t_f)$ is the first time derivative of $H(t_f)$, and g is the acceleration due to gravity. For all forward dynamic simulations, the total simulation time is a constant of 0.5 seconds, similar to the optimal result obtained by Pandy *et al.* (1990).

The bottom right output panel provides plots of the simulated joint angles (left side) in units of degrees, ground reaction forces (right side top) in units of Newtons, and height of the center of mass (right side bottom) in units of meters. For a realistic jump, the normal component of ground reaction force F_{Norm} should remain positive throughout the simulation, while both components of ground reaction force should go to zero at the final time, which represents the point at which the toes leave the ground. The final height shown in the height plot is typically lower than H_{Max} since $H(t_f)$ is only one of several quantities that goes into the calculation of H_{Max}.

VII. Muscle Torque Control Strategies for Maximizing Jump Height

Below is a series of tasks for you to perform and questions for you to answer. The answers to the questions constitute your written report for Lab 1.

1. Starting from the static control torque curves created by initializing the model with the default values, manually adjust the initial joint angles and muscle torque nodal points to produce the highest jump that you can. Try to get the final normal and tangential components of the ground contact force vector to be as close to zero as possible. Leave the Ground and Ligament passive torque checkboxes deselected. Iterate your guess for the inputs until you produce a well-coordinated maximum height jump that closely satisfies the condition of zero ground contact force at the end of the simulation.

Record your maximum jump height and final values for normal and tangential ground contact force in the space provided below, along with a screen shot of the jumper software graphical user interface showing the input and output curves corresponding to your maximum height jump.

Maximum jump height:
Final normal force:
Final tangential force:

Screen shot of jumper software:

2. What were the most significant problems that you encountered while trying to generate a maximum height jump in step 1. above? What could you do to try to eliminate those problems? How might those problems prevent you from using optimization to find a maximum height jump?

3. Repeat step 1. but with the Ground passive torque checkbox selected. Provide the same results as in step 1.

Maximum jump height:
Final normal force:
Final tangential force:

Screen shot of jumper software:

4. What were the most significant problems that you encountered while trying to generate a maximum height jump in step 3. above? Explain which problems were resolved by adding a ground contact torque to prevent foot penetration into the floor and which new problems were introduced by this addition. How might these new problems hinder your ability to use optimization to find a maximum height jump?

5. Repeat step 1. but with the Ligament passive torque checkbox selected. Provide the same results as in step 1.

Maximum jump height:
Final normal force:
Final tangential force:

Screen shot of jumper software:

6. What were the most significant problems that you encountered while trying to generate a maximum height jump in step 5. above? Explain which problems were resolved by adding ligament torques to prevent hyperextension of each joint and which new problems were introduced by this addition. How might these new problems hinder your ability to use optimization to find a maximum height jump?

7. Repeat step 1. but with the both the Ground and Ligament passive torque checkboxes selected. Provide the same results as in step 1.

Maximum jump height:
Final normal force:
Final tangential force:

Screen shot of jumper software:

8. To produce a maximum height jump as quickly and easily as possible through manual adjustments to the initial joint angle and torque control nodal points, which passive torque checkboxes would you select? Ground only, Ligament only, both, or none? Provide a brief explanation for your answer. To produce a maximum height jump using optimization, which passive torque checkboxes would you select? Ground only, Ligament only, both, or none? Again provide a brief explanation for your answer.

9. Try to produce the highest jump that you can by changing control nodal points only for the knee and hip torque curves. Select any combination of passive torques that you desire, but keep the initial joint angles at their default values. Also leave the toe and ankle torque control nodal points at their default values. Compared to the maximum height jump that you produced earlier for the same passive torque selections, characterize the changes in the knee and hip torque curves that you needed to make to produce a maximum height jump. Are you able to get the heel to rise above the floor even though you cannot change the toe or ankle torque nodal points? Explain briefly why or why not. Record your maximum height jump results below, along with your passive torque selections.

Maximum jump height:
Final normal force:
Final tangential force:
Passive torque selections:

Screen shot of jumper software:

10. For any of the steps above, if you are able to get the model to jump anywhere near the jump height predicted by the optimal solution (i.e., over 0.33 meters) found by Pandy *et al*. (1990), give yourself a pat on the back! It is not easy to do. In the more likely event that your solution was not as high as the optimal solution, explain why. If it was higher than the optimal solution, provide a possible explanation for that result as well. In addition, for the best maximum height jump that you achieved in steps 1. through 9, discuss briefly how your ground reaction force

curves compare to those predicted by Pandy *et al*. (1990). Use the differences to suggest how you might modify your best control strategy found thus far to produce an even higher maximum height jump.

VIII. References

Anderson FC and Pandy MG (1999). A dynamic optimization solution for jumping in three dimensions. *Computer Methods in Biomechanics and Biomedical Engineering*, **2**, 201-231.

Arnold AS, Salinas S, Asakawa DJ, Delp SL (2000). Accuracy of muscle moment arms estimated from MRI-based musculoskeletal models of the lower extremity. *Computer Aided Surgery*, **5**, 108-119.

Atkinson LV, Harley PJ, Hudson JD (1989). Numerical methods with FORTRAN 77. Addison-Wesley Publishing Company, Menlo Park.

Delp SL, Loan JP, Hoy MG, Zajac FE, Topp ET, Rosen JM (1990). An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Transactions in Biomedical Engineering*, **BME-37**, 757-767.

Garner BA, Pandy MG (2000). The obstacle-set method for representing muscle paths in musculoskeletal models. *Computer Methods in Biomechanics and Biomedical Engineering*, **3**, 1-30.

Guyton AC (1986). Textbook of medical physiology, Seventh Edition. W. B. Saunders Company, Philadelphia.

Hatze H (1976). The complete optimization of human motion. *Mathematical Biosciences*, **28**, 99-135.

McMahon TA (1984). Muscles, Reflexes, and Locomotion. Princeton University Press, Princeton, New Jersey.

Pandy, MG, Zajac, FE, Sim, E, and Levine, WS (1990). An optimal control model for maximum-height human jumping. *Journal of Biomechanics* **23**, 1185-1198.

Symbolic Dynamics, Inc. (1996). SD/FAST User's Manual, Version B.2. Mountain View, CA.

Van der Helm FCT, Veeger HEJ, Pronk GM, Van der Woude LHV, Rozendal RH (1992). Geometry parameters for musculoskeletal modeling of the shoulder system. *Journal of Biomechanics*, **2**, 129-144.

Zajac FE (1989). Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *CRC Critical Reviews in Biomedical Engineering* (Edited by Bourne JR), **17**, 359-411. CRC Press, Boca Raton.