

Simulation Lab #5: Muscle Actuated Simulation of Kicking

Professor B.J. Fregly
Mechanics of the Human Locomotor System
EML 5595/BME 6938 - Fall 2003

¹Laboratory Developers: Darryl Thelen, Silvia Blemker, Scott Delp, Clay Anderson
Neuromuscular Biomechanics Lab, Stanford University

I. Introduction

In the previous labs, you learned how to use SD/FAST to conduct dynamic simulations, how to model musculoskeletal geometry accurately, and how to model the dynamic actions of muscle-tendon actuators. In this lab, you will learn how to couple all of these components together to produce a muscle-actuated forward dynamic simulation of human movement. The software packages SIMM, Dynamics Pipeline, and SD/FAST will be used to simulate kicking. After producing the model, you will explore some of the complexities involved in defining a set of muscle excitations that can produce a coordinated, purposeful movement.

II. Objectives

The purpose of this lab is to learn how to create a muscle-actuated forward dynamic simulation of movement.

By working through this lab, you will:

- Become familiar with the implementation of inverse and muscle-driven forward dynamic simulations using SIMM, Dynamics Pipeline, and SD/FAST
- Investigate the sensitivity of joint moments to perturbations in task kinematics
- Explore the complexity of neuromuscular control for an apparently simple task
- Get familiar with a muscle-actuated leg simulation model in preparation for your project

¹ This laboratory was derived from a set of simulation exercises developed by Professor Maury Hull at U.C. Davis.

III. Deliverables

Please turn in:

1. A written report

A Microsoft Word template for the report called `Simulation_Lab5_Report.doc` is available on the course web site.

IV. Input Files Needed

All of the necessary input files for the lab are provided in the zip file `Lab5.zip` on the course web site.

1. SIMM Files

`kick.jnt`

SIMM joint file for the lower extremity. This model includes a planar knee joint that incorporates the sliding/rolling motion between the femur, tibia, and patella that occurs during knee flexion.

`kick.msl`

SIMM muscle file that includes a select subset of muscles of the lower extremity.

`bones/*.*`

Bone files for the lower extremity.

2. Pipeline Code

Main Driver Routine for Forward and Inverse Dynamic Simulations

`dynsim\src\main.c`

This file contains the main driver routine for a forward and inverse dynamic simulations and several other utility routines that are independent of the musculoskeletal model in the simulation. This file can be generated by SIMM/Pipeline. You will probably want to generate it once and then customize it to your specific simulation. This file contains the subroutines `main()`, `sdumotion()`, `sduforce()`, and `init_motion()`.

General Purpose Source Files (generated by SIMM Pipeline when you Save Dynamics)

`dynsim\src\gmc.c`

General muscle code – a collection of routines to calculate muscle-tendon model quantities (e.g. passive muscle force, tendon force and pennation angle) from simulation state information.

`dynsim\src\mathtools.c`

General-purpose mathematics routines.

`dynsim\src\pipetools.c`

Utility routines for conducting dynamic simulations.

`dynsim\src\readmuscles.c`

Routines to read in the musculotendon model information from a muscle file. Attachment points, wrapping surfaces, and muscle-specific parameters (e.g. maximum isometric force, tendon slack length, etc...) are input and used during the simulation. Muscle files are read at runtime allowing the changing of muscle parameters and muscle excitations without altering the simulation code.

`dynsim\src\readtools.c`

General purpose utility routines used to read in data from input muscle files and kinetics data files.

Muscle-Tendon Model Source Files

`dynsim\src\assigns.c`

`dynsim\src\derivs.c`

`dynsim\src\inits.c`

Code implementing dynamic muscletendon models. The code is set up such that you can use one of the existing models described in the Pipeline manual or can alternatively use templates that are provided to create your own model. Note that these three files are modified versions of the Pipeline-generated code created specifically for this lab.

Header Files

`dynsim\src\basic.h`

Contains #defines and enum that are used in many source files.

`dynsim\src\functions.h`

Contains prototypes for many of the functions in the source files.

`dynsim\src\structs.h`

Contains definitions of all of the structures that are used in the Dynamics Pipeline.

`universal.h`

Contains `#includes` for all of the standard header files. This file should be included at the top of every source file in the Pipeline.

Libraries

`lib\acpp.lib`

General purpose routines used for parsing files on input.

`lib\wrap.lib` and `lib\wrapd.lib`

Precompiled library files that account for muscle wrapping in the calculation of muscletendon length and velocity during a dynamic simulation.

3. Pipeline Inputs

`inverse.ktx`

Initial inverse dynamics kinematics file containing time histories for the hip and knee flexion angles and their time derivatives.

`params.txt`

Nominal input parameters file for your Pipeline code.

V. Getting Started

From the class website, download and unzip Lab5.zip, which contains all the files necessary for this lab.

Important: In this lab, you will be using a different version of the muscle-tendon model routines (`assigns.c`, `derivs.c`, `inits.c`) than the ones created automatically by SIMM Pipeline. A copy of these files has already been saved in the `\dynsim\src` directory. These files contain muscle model 8, which is slightly modified version of model 4 described in the Pipeline tutorial.

VI. Muscle-Actuated Forward Dynamic Models

Dynamic models of the musculoskeletal system are typically comprised of four components: 1) the equations of motion for the body, or skeletal dynamics, 2) a representation of musculoskeletal geometry, 3) a model of activation dynamics, and 4) a model of muscle-tendon mechanics. 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 $\bar{a}(t)$, the muscle forces $\bar{f}(t)$, the generalized speeds $\bar{u}(t)$ and the generalized coordinates $\bar{q}(t)$, differential equations (see equations below) are used to compute the time rate of change of the states. Then 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.

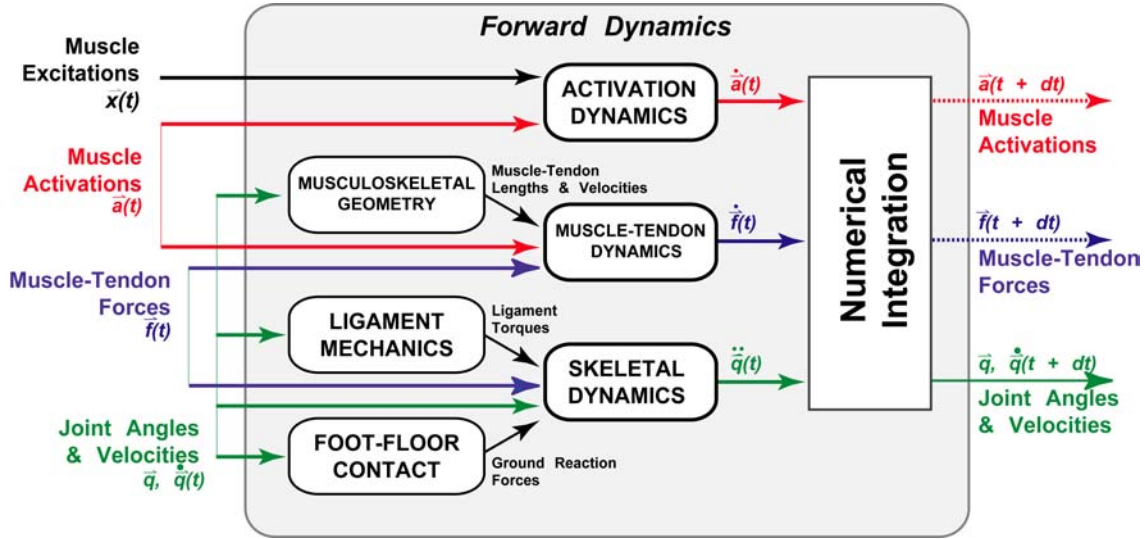


Figure 1. Schematic of a forward dynamic simulation.

The state equations of motion for the system can be written as a set of first order differential equations in the general form:

$$\begin{bmatrix} \dot{\bar{q}} \\ \dot{\bar{u}} \\ \dot{\bar{a}} \\ \dot{\bar{l}}^M \end{bmatrix} = \begin{bmatrix} \bar{u} \\ \bar{I}(\bar{q})^{-1} \cdot \{ \bar{C}(\bar{q}, \bar{u}^2) + \bar{G}(\bar{q}) + \bar{R}(\bar{q}) \cdot \bar{f}_M(\bar{a}, \bar{l}^M) + \bar{E}(\bar{q}) \cdot \bar{f}_E \} \\ (\bar{x} - \bar{a}) / \bar{\tau}(\bar{a}, \bar{x}) \\ \bar{v}^M(\bar{q}, \bar{u}, \bar{l}^M, \bar{a}) \end{bmatrix}$$

where the states consist of:

\vec{q}	generalized coordinates
\vec{u}	generalized speeds
\vec{a}	muscle activations
\vec{l}^M	muscle fiber lengths

In this lab, you will be using numerical integration to solve for the states as a function of time given initial states and muscle excitation-time histories $\vec{x}(t)$.

VII. Inverse Dynamics

The software packages SIMM, Dynamics Pipeline, and SD/FAST may be used to perform an inverse dynamic simulation. The software package SIMM is used to generate the rigid body model (skeletal system and joints). SD/FAST is a software package that generates equations of motion using Kane's method or, for larger systems, a more efficient Order-N method. Dynamics Pipeline is a third software package that links SIMM and SD/FAST so that inverse dynamic simulations may be performed. Using the rigid body model (`kick.jnt` file), the software generates C code to perform the inverse dynamic simulation.

You will use the SIMM Pipeline software to generate the inverse dynamic code for analyzing a kicking motion. For these analyses, you will generate a kicking motion based on a nominal two degree-of-freedom model with hip and knee flexion angles being the generalized coordinates. This model is similar to one used in a previous lab.

1. Start SIMM and load the joint and muscle files from your Lab 5 directory:

Joint file: `kick.jnt`

Muscle file: `kick.msl`

Motion file: `kick.mot`

Animate the kicking motion

2. Use SIMM Pipeline to write dynamics files for your system. This step has been done for you since we do not have Pipeline code generation licenses for the NEB 109 computers. Click on `File > Save Dynamics`, which will generate C code for both forward and inverse dynamics simulations. SIMM Pipeline will create a Microsoft Visual C++ project called `pipeline.dsw` containing all files necessary to create a single executable capable of performing forward or inverse dynamics simulations. Two important files generated by the Pipeline are the following:

`model.sd` SD/FAST system description file that is used to generate code that describes the system equations of motion

`sdfor.c` Model-specific C code that contains the body segment parameters and joint kinematic information used in forward or inverse dynamic simulation

3. Compile the inverse dynamics simulation program. Start Microsoft Visual C++ and open the project workspace file `pipeline.dsw`. Select `Build > Build simulation.exe` to create an executable. Note that SIMM Pipeline will automatically call SD/FAST to process your system description file `model.sd`. All the files necessary for this step are included in your directory.
4. The file `inverse.ktx` contains the experimental kinematics data (angles, angular velocities, and angular accelerations) that will be used in the inverse dynamics analysis. Use a text editor to open the file and examine its contents. Note that the extensions `_gcv_10` or `_bogus` have been added to each of the angles and their time derivatives (e.g., `hip_flexion_gcv_10` and `hip_flexion_vel_bogus`). The extension `_gcv_10` tells the Pipeline code to use a **generalized cross-validated spline** to smooth the angular data using a 10 Hz cut-off frequency. The extension `_bogus` (or any other name the Pipeline code does not recognize) turns a name understood by the Pipeline code (e.g., `hip_flexion_vel`) into a name it does not understand and so will not process. The number of such columns is indicated next to `otherdata` (4 in this case) in the header. If the `_gcv_10` and `_bogus` extensions were removed, the Pipeline code would use the position, velocity, and acceleration data reported in the file without applying any smoothing or filtering. If the estimated accelerations are not smooth (as with the original data), this can produce large unrealistic “bumps” in the joint torques calculated via inverse dynamics.
5. Open the file `params.txt` and make the following changes so that it can be used to perform an inverse dynamics analysis. Note that the `#` sign is the comment character.
 - Comment out the line that starts with `muscle_file` so that muscle forces will not be calculated based on the excitation histories stored in the muscle file.
 - Change the name of the `output_motion_file` from `results.mot` to `inverse.mot`.
 - Uncomment the line that starts with `kinetics_file` and change the name of the input file from `forces.ktx` to `inverse.ktx`.
 - Change the name `output_kinetics_file` from `results.ktx` to `inverse_out.ktx`.
 - Change `end_time` to 0.99 so that 100 frames will be simulated, consistent with the number of frames in `inverse.ktx`.

- Change `output_muscle_activations` and `output_muscle_forces` to `no` since no muscle activations or forces are computed by inverse dynamics.
 - Change `output_hinge_torques` to `yes` to specify output of inverse dynamics joint torques.
6. Run the inverse dynamics simulation. Select `Build > Execute simulation.exe` or press `Ctrl+F5`.

The program will create an output file called `inverse.mot`. Open the file with a text editor and examine its contents. Note that the line

```
range 0.000000 1.000000
```

scales the range of all quantities in the file to be between 0 and 1. This will allow you to overlay and compare results from the inverse dynamics simulation with those from the forward dynamic simulation to be performed in the next section of this lab

7. View the simulation results in SIMM. Load the motion file `inverse.mot` and animate the motion. You may need to set the gear (speed) of the animation slower for the animation to proceed at a speed that you can visualize. Use the `motion curve >` command in the `Plot Maker` to make graphs of the net hip and knee torques generated by inverse dynamics.

Determine at what time of the motion cycle that peak moments occur. Compare this with the kinematic data (joint positions, velocities, and accelerations). What events in the kinematic data correspond with the timing of the peak moments? Using your knowledge of dynamics, explain your findings.

VII. Forward Dynamics

For these simulations, you will attempt to replicate a nominal two degree of freedom kicking motion by specifying the activation patterns for a limited set of muscles over a one second time period. Based on the visual analysis of the motion and moment plots, guess the activation patterns of the rectus femoris (`rect_fem`), gluteus maximus2 (`glut_max2`), and medial gastrocnemius (`gas_med`) muscles. If you are not familiar with the functions of these muscles, then use the `Model Viewer` window to turn off and on these muscles to observe how they cross the hip and knee joints. You must specify an onset time, offset time, and activation level. To deactivate a particular muscle, enter 0 for both the onset and offset time. See Section 3.3 SIMM

Muscle File in the Dynamics Pipeline manual for how to specify onset time, offset time, and activation level for each muscle in your muscle file.

You are now ready to perform the forward dynamic simulation. Previously, when the `Save Dynamics` function was performed, both the inverse and forward dynamic models were generated using SIMM, SD/FAST, and the Dynamics Pipeline. For the forward dynamic simulations, the file `kick.msl` will also be used.

1. Modify the parameter values in `params.txt` to specify a forward dynamics simulation. `params.txt` contains the names of the input and output files that the simulation needs. The following file names should be set:

- Uncomment the line that starts with `muscle_file` so that muscle forces will be calculated and applied to the model. Make sure that the specified muscle file is `kick.msl`.
- Change the name of the `output_motion_file` from `inverse.mot` to `forward.mot`.
- Comment out the line that starts with `kinetics_file` so that prescribed motions or inverse dynamics joint torques will not be read in by the Pipeline code. Note that if you wanted to perform a forward dynamics simulation driven by your inverse dynamics joints torques (which is a good check to perform), you would leave this line uncommented and would copy your inverse dynamics output motion file `inverse.mot` to a forward dynamics input kinetics file `forward.ktx`. The motion file header would be deleted and replaced with a header for an input kinetics file (i.e., frames, elements, and otherdata). To avoid prescribing any motions as in inverse dynamics, an extension such as `_bogus` would be added to the column labels `hip_flexion` and `knee_flexion` with `otherdata` set to 2. The `muscle_file` line would also remain commented out so that joint torques and muscle forces would not be applied simultaneously.
- Change the name of your `output_kinetics_file` from `inverse_out.ktx` to `forward_out.ktx`.
- In the `Gencoord Initialization` section, uncomment `hip_flexion` and `knee_flexion` and enter the correct initial position and velocity next to each one based on their values in the output motion file from inverse dynamics (i.e., the smoothed and filtered values).
- Change `output_muscle_activations` and `output_muscle_forces` to `yes` to output these quantities in the output kinetics file.
- Change `output_hinge_torques` to `no` since no inverse dynamics joint torques are to be calculated.

2. Run the forward dynamics simulation. Press `Ctrl+F5` from the `Build` menu.
3. A data file of the simulated motion, `forward.mot`, is generated containing the values of the generalized coordinates as a function of time. Load the motion file and observe the simulated motion by clicking `start/stop` and selecting `sd_motion` in the `Model Viewer` window. The `gear` may need to be set at around `0.05` in order to observe the animation. Alternatively, the `sd_motion` slider bar may be used to view the motion.
4. Based on the graphical results of your simulation, guess a new set of activation patterns in your muscle file to better replicate the nominal motion `kick.mot`. Then perform another forward dynamic simulation to evaluate your guess. Repeat this produce until you are able to produce a kicking motion similar to the experimental data. Matlab optimization may be (but need not be) used for this step, where `simulation.exe` would be the external analysis function called repeatedly by Matlab.

Muscle	Time of Onset (s)	Time of Offset (s)	Activation Level
<code>rect_fem</code>			
<code>glut_max2</code>			
<code>med_gas</code>			

How do the kinematic results from your muscle-actuated simulation compare to the nominal motion `kick.mot`?

Describe how the muscle forces combine to produce the hip and knee moments calculated in the inverse dynamics phase of this lab.

Why is it difficult to guess the activation patterns that generate the nominal motion?

VIII. Exploration Phase

Muscle actuated dynamic simulations provide access to important biomechanical quantities that underlie human motion. For example, knowledge of muscle force is essential for quantifying the stresses placed on bones and also for understanding the functional roles of muscles in normal and pathological movement.

The remainder of this lab will involve modifying your forward dynamics simulation to estimate the loads at the hip joint during kicking. Your assignment is to use the SD/FAST command `sdreac` to quantify the reaction forces at the hip. You should modify `main.c` to compute the hip reaction loads at the end of each time step and write them to the output motion file. Modify the routines `open_output_motion_file` and `write_motion_frame` to output the horizontal, vertical, and net forces at the hip to the motion file `forward.mot`. Create a plot of the hip reaction forces. Create a second plot of the muscle forces. What portion of the hip reaction loads can be attributed to muscle forces? What other forces in the system contribute to the hip reaction loads?