Simulation Lab #2:
Joint Torque Driven Simulation of the Swing Phase of Gait

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

Lab Developers: Darryl Thelen, Clay Anderson, Scott Delp
Neuromuscular Biomechanics Lab, Stanford University

I. Introduction

In this class, you are learning how to develop and use biomechanical models to simulate motion. This process involves developing models of musculoskeletal geometry and coupling them with differential equations that describe the dynamics of the system. The differential equations include the equations of motion, muscle activation dynamics, and musculotendon mechanics. Biomechanical models often have many degrees of freedom, and therefore it is often infeasible to derive the equations of motion by hand. Instead it is convenient to use a software package such as SD/FAST to generate the code that describes the equations of motion of a system. Such code can be used to numerically solve inverse and forward dynamic problems that arise in the study of human motion.

II. Objectives

The purpose of this lab is to demonstrate the use of SD/FAST to generate and numerically solve differential equations of motion for a biomechanical system. As an example, you will develop a dynamic model of the lower extremity and use SD/FAST routines to analyze the dynamics of the swing phase of gait.

By working through this tutorial, you will:

- Learn how to generate an SD/FAST system description file for a mechanical system
- Create a four link model of the lower extremity
- Use SD/FAST to generate source code that describes the equations of motion of your four link model
- Use the generated code to conduct both inverse and forward dynamic simulations of the swing phase of gait
- Explore the relative contributions of initial conditions and joint torques to the motion of the lower extremity during the swing phase of gait

III. Deliverables

At the completion of this lab, you will need to turn in a written report created from a pre-formatted template that will be provided to you. In the report, you will summarize your findings and address the questions that are posed. In addition, for this lab you will need to turn in your SD/FAST C code used for generating your solutions, along with plots created from SIMM motion files generated by your SD/FAST code.

Please turn in:

1.	A written report	A Microsoft Word template for the report called `Simulation_Lab2_Report.doc` available on the course web site.
2.	`leg_model.sd`	A well-commented SD/FAST system description file for the four link model of the lower extremity
3.	Joint torque plots from `inv_swing.mot`	Joint torque plots generated by an inverse dynamics simulation using driver program `main.c` provided to you
4.	Joint angle plots from `inv_swing.mot` and `for_swing.mot`	Joint angle plots for the hip, knee, and ankle generated by inverse and forward dynamic simulations using driver program `main.c` provided to you
5.	`main_mod.c`	Modified driver program in which the motion of the pelvis and ankle joints are prescribed and no joint torques are applied
6.	Joint angle plots from `inv_swing.mot` and `for_pass_swing.mot`	Joint angle plots for the hip, knee, and ankle generated by inverse and forward dynamic simulations using driver program `main_mod.c` created by you

IV. Input Files

All the necessary input files for the lab are provided in the zip file `Lab2.zip` on the course web site.

1. Data Files

`gait.mot`
SIMM motion file of the gait kinematics of a young adult walking at a natural cadence. Data file includes the generalized coordinates describing pelvis and leg motion during one gait cycle.

`gait_swing.mot`
SIMM motion file of the kinematics of the swing phase of gait. File includes the generalized coordinates, generalized speeds, and accelerations of the pelvis and lower limb during the swing phase of gait.

`gait_swing.dat`
Same as gait_swing.mot without the motion file header information. This data file is read by the main executable program and used to conduct the inverse dynamics simulation.

2. **SIMM Model Files**

   `leg_model_2d.jnt`
   SIMM joint file used to animate the leg motion. A 3D leg model (Delp et al. 1990) included in SIMM was adapted for this model by adding generalized coordinates for the pelvis and removing non-planar degrees of freedom.

3. **Program Files**

   `main.c`
   Driver program that performs inverse and forward dynamic simulations of the swing phase of gait using SD/FAST generated code. Reads in experimental gait data and outputs motion files for visualization in SIMM.

   `sdlib.c`
   SD/FAST general purpose library that will be used in all of your simulations.

   `Lab 2.dsw`
   `Lab 2.dsp`
   Microsoft Visual C++ project files for the driver program `main.c` and associated SD/FAST-generated C files.

4. **SD/FAST**

   `sdfast.zip`
   Zip file containing all files referred to in tutorials 1-4 of the SD/FAST manual. Note that only the `fbar` and `sphere` tutorial problems come with example C rather than Fortran driver programs.

V. SD/FAST Tutorials

Using SD/FAST to simulate a model of a dynamic system involves the following basic steps:

1. Generating an SD/FAST system description file for your model.
2. Running SD/FAST to generate code that describes the equations of motion of your system.
3. Combining SD/FAST generated code and SD/FAST routines within a driver program that you write to numerically simulate the system under user-defined conditions.

Your should read and complete tutorials 1 and 2 in the SD/FAST user's manual to become familiar with these steps:

- Tutorial 1: Introducing SD/FAST
- Tutorial 2: Simple Pendulum

A pdf version of the manual is available on the course web site. Tutorial 1 provides a broad overview of SD/FAST along with a sample C driver program for the model `fbar.sd`. Tutorial 2 asks you to write, compile, and run Fortran code to simulate a dynamical system. All SD/FAST routines have both Fortran and C versions, with the syntax for each type of call described in the manual. When completing Tutorial 2, you should write your driver program in C rather than Fortran. See pages R-1 through R-3 of the manual for a summary of differences in the C and Fortran implementations of SD/FAST routines, and use the `fbar.c` driver program from Tutorial 1 as an example. Recall that throughout the class, you will be using C code for all your programs. As you complete each section of Tutorial 2, feel free to try variations to cover aspects of SD/FAST or the dynamical model that interests you. If you have time, you may find it useful to look through SD/FAST Tutorials 3 and 4 to get an idea of some of the more advanced modeling capabilities of the software.

SD/Fast has a specific, compact notation for defining body segments, describing joint locations, and creating joints. Body segments are defined based on their inertial properties, specifically the mass and mass moments of inertia. Joint locations are defined via body-fixed local vectors relative to the body centers of mass (Figure 1). A number of pre-defined joints (e.g. pin, slider, ball) exist in SD/FAST and can be used to describe relative motion between bodies. Please see the manual for a complete description of the types of joints that can be modeled.

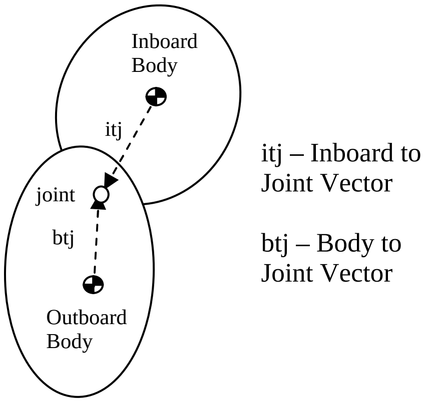
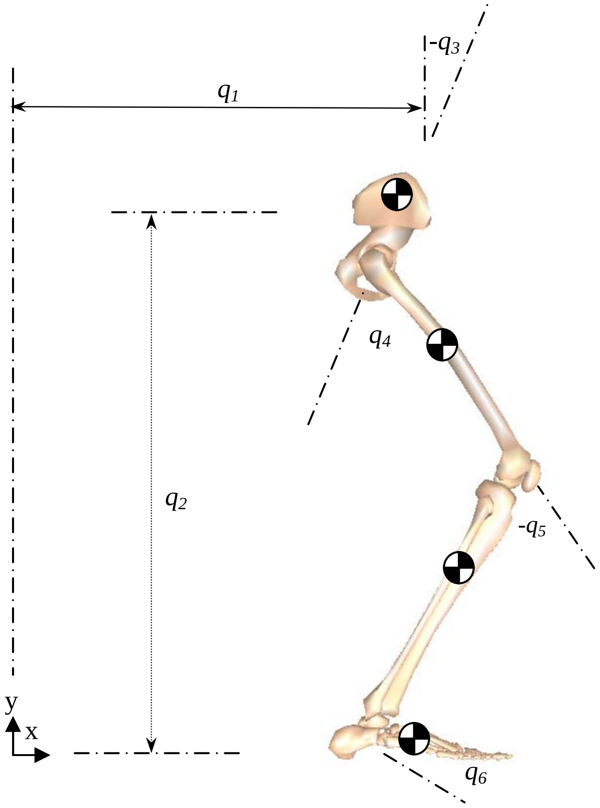


Figure 1. In SD/Fast, each body articulates with respect to an inboard body by a joint of a particular type. The position of a joint relative to the center of mass of the body is defined by the **Body-To-Joint** vector. An **Inboard-To-Joint** vector gives the position of the joint relative to the Inboard Body's center of mass.

VI. Planar Leg Model

In this lab, you will be using a planar leg model to simulate the swing phase of gait. The model has 4 segments, 4 joints and six generalized coordinates as described in Figure 2. The four segments are the pelvis, femur, tibia and calcaneus (foot). The segment names are selected to correspond with the SIMM motion file that will be used to animate your simulations. The pelvis is connected to ground via a planar joint that allows translation and rotation in the *xy* plane. The hip, knee and ankle joints are modeled as pin joints with rotation axes aligned with the global z axis (Figure 2).



q_1, q_2 – horizontal and vertical position of pelvis, midway point between ASIS
q_3 – pelvis angle with respect to the vertical
q_4 – hip flexion angle
q_5 – knee extension angle
q_6 – ankle dorsiflexion angle

Figure 2. Four segment model of the lower extremity, which includes the pelvis, femur, tibia and calcaneus (foot). The pelvis is attached to ground by a 3 dof planar joint which allows translation and rotation in the plane. The hip, knee and ankle joints are assumed to be frictionless pin joints. Six generalized coordinates are used to uniquely specify the joint positions.

VII. Inertial Properties

Creating the SD/FAST description file for the leg model requires estimating the length and inertial properties of the body segments. Specifically you need to specify the segment masses, locations of the joints relative to the mass centers and the mass moments of inertia. Since you cannot directly measure segmental inertia properties of humans, it is necessary to estimate these properties through indirect methods. Researchers in the field of anthropometry, the study of the physical dimensions of the human body, have accumulated estimates of the length and inertial properties of human body segments. These estimates have been obtained using various techniques including direct measurements from cadavers and the coupling of volumetric measurements with body density tables. The inertial parameters of the pelvis and foot segments have been estimated for you from the anthropometric data reported by McConville et al. 1980 (Table 1). A body mass of 75 kilograms and a height of 1.77 meters were assumed to scale the anthropometry data.

You are to complete the missing values in Tables 1 and 2 by scaling the anthropometric data accumulated by Winter (1990), which is provided to you in the Appendix of this tutorial. Note that your model is planar so only the I_z compoments of the mass moments of inertia will be important. However, the system description file requires values of I_x and I_y as well. Please make and document appropriate assumptions used in estimating I_x and I_y.

Table 1. Segmental inertia parameters are relative to a local xyz reference frame fixed at the segmental center of mass. In an upright reference configuration, both the local and global reference frames are aligned such that x points anteriorly, y points superiorly and z points laterally.

Segment	M (kg)	I (kg-m^2)		
		I_x	I_y	I_z
Pelvis	11.77	0.1030	0.0871	0.0579
Femur	7.76			
Tibia	3.71			
Foot	1.45	0.0015	0.0040	0.0040

Table 2. Joints used in the planar leg model. Pelvis is attached to ground via a planar joint. Hip, knee and ankle are assumed to be pin joints.

Joint	Type	Body	Inboard body	btj (m)			itj (m)		
				x	y	z	x	y	z
Pelvis_ground	Planar (3 dof's)	Pelvis	Ground	0.0707	0.000	0.000	0.000	0.000	0.000
Hip	Pin (1 dof)	Femur	Pelvis						
Knee	Pin (1 dof)	Tibia	Femur						
Ankle	Pin (1 dof)	Foot	Tibia	-0.034	0.020	0.000	0.000	-0.243	0.000

itj – inboard to joint position vector, btj – body to joint position vector (See Figure 1)

VIII. Create and Process SD/FAST Description File

Create an SD/FAST system description file that describes your planar leg model. Save the model description file as `leg_model.sd`.

The next step is to generate code that can be used to simulate the system. Since we only have one SD/FAST license for code generation, this will only work on the computer located in the right front corner of NEB 109. Once you have your SD/FAST-generated C code, you may move it to any computer that has Microsoft Visual C++ 6.0 on it to complete the assignment.

To generate your SD/FAST C code, open a DOS Command Prompt and change to the directory where your model description file is located. Then enter the following command to cause SD/FAST to generate C code describing your system:

```
sdfast –d –lc –ge leg_model.sd
```

Notes on arguments:		
	`–d`	specifies double precision code
	`–lc`	specifies output language as C
	`-ge`	generate everything, including library file sdlib.c
	`–h`	another useful argument – help

SD/FAST will create three model-specific files by processing your system description file. Routines in these files will be called during inverse and forward dynamic simulations:

`leg_model_d.c`	Dynamic simulation code
`leg_model_s.c`	Simplified analysis routines
`leg_model_i`	Information file

In addition, SD/FAST will create a file called `sdlib.c` containing library routines that are not model specific.

IX. Torque Driven Simulation of the Swing Phase of Gait

In this lab, SD/FAST generated code and routines will be used to perform inverse and forward dynamic simulations of the swing phase of gait.

- Inverse dynamics analysis: given $q_1 \ldots q_6$ and the first and second time derivatives, calculate the torques necessary at each of the joints to generate the prescribed motion.
- Forward dynamics analysis: use the initial generalized coordinates $q_1 \ldots q_6$ and initial generalized speeds $u_1 \ldots u_6$ together with the joint torques determined via inverse dynamics to numerically integrate the equations of motion and predict the motion.

Complete the following steps to perform an inverse and forward dynamics analysis of the swing phase of gait.

1. Use SIMM to visualize lower extremity kinematics during the swing phase of gait.

   The following files should be loaded into SIMM:

Bone files:	SIMM defaults
Joint file:	`leg_model_2d.jnt`
Gait cycle motion file:	`gait.mot`
Swing phase motion file:	`gait_swing.mot`

Then animate the motion files and use the SIMM plotmaker to create a plot of the hip, knee and ankle angles during the swing phase of gait.

Note: The leg model you are using to animate and plot the simulation results has a 2D planar knee joint that allows for sliding-rolling motion between the femur and tibia. While it can be effectively used to visualize the simulated motions, the model does not strictly correspond to the SD/FAST model you are creating and simulating. In future labs, you will learn how you can use SIMM with the Dynamics Pipeline to include more complex joint kinematics within your dynamic simulations.

2. Use Microsoft Visual C++ to set up the Lab 2 project environment.

   Start Microsoft Visual C++ 6.0, select `File` → `Open Workspace`, change to the directory where your downloaded Lab 2 files are located, select the workspace file `Lab 2.dsw`, and click on `Open`. To see the C files included in this project, click on the `FileView` tab in the lower left corner and then on the + next to the `Source Files` folder. You will see that the following files are automatically included in the project:

   ```
   leg_model_d.c
   leg_model_s.c
   main.c
   sdlib.c
   ```

   The file `main.c` is a sample C driver program provided to you as a starting point for your own programming work. Read through the steps of the program to begin to familiarize yourself with the layout of the program. You will need to modify the code later so it is worthwhile to begin understanding it.

3. Generate an executable for your model.

   The executable will be generated by compiling and linking your C driver program `main.c`, the SD/FAST-generated C files `leg_model_d.c` and `leg_model_s.c`, and the SD/FAST library routines in `sdlib.c`. In Microsoft Visual C++, select `Build` → `Build Lab2.exe`. Ignore all the warnings produced by the compilation and linking stages, but make sure that no errors are reported.

4. Run the executable and generate results.

   To run your executable, open a DOS Command Prompt, change to the `Debug` folder in the directory where all your C files are located, and type `Lab2.exe.`

   Alternatively, from within Microsoft Visual C++, select `Build` → `Execute Lab2.exe.` This executes your simulation and outputs the quantities specified in your driver program `main.c`.

5. Load the results into SIMM.

   Two output motion files are created by the simulation. You should load both of these into SIMM along with the `leg_model_2d.jnt` file.

   `inv_swing.mot` – contains the kinematics (generalized coordinates, generalized speeds, and accelerations) and also the joint torques that were calculated using inverse dynamics

   `for_swing.mot` – contains the kinematics (generalized coordinates, generalized speeds, and accelerations) that were simulated from the joint torques using forward dynamics

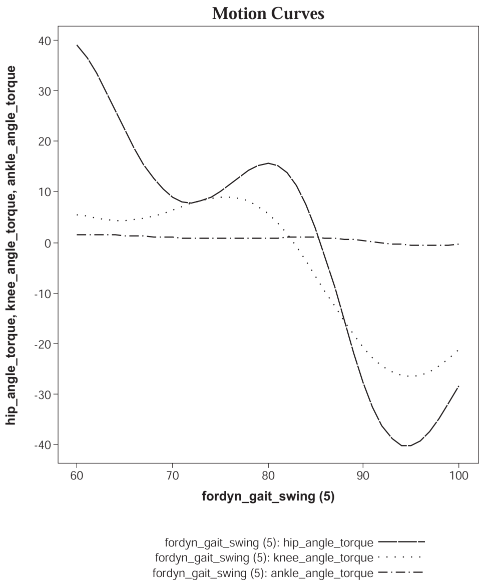


Figure 3. Hip, knee and ankle torques calculated from an inverse dynamics analysis of the swing phase of gait.

6. Analyze the results of the inverse dynamics simulations.

   - Plot the hip, knee and ankle joint torques from inverse dynamics. Submit these plots with your final report. Your results should be similar to those shown in Figure 3.
   - Intuitively, how do you believe the joint torques are contributing to the observed motion? (e.g., how does the hip moment flex or extend the hip, knee and ankle?)
   - What is the physical meaning of the generalized torques associated with the pelvis degrees of freedom (pelvis_tx_torque, pelvis_ty_torque, pelvis_rotation_torque)?

7. Compare the results of the inverse and forward dynamic simulations.

   - Use the SIMM plotmaker to overlay plots of the inverse and forward dynamic hip, knee and ankle angles. Submit these plots with your final report.
   - Do they agree? Why or why not? Hint: Please see Risher et al. [*J Biomech Eng* 119:417-422, 1997].

- After determining the underlying source of the discrepancy, describe appropriate modifications that would be necessary to ensure that the forward dynamics solution agrees with experimental data.

X. Passive Simulation of the Swing Phase of Gait

Many investigators have assumed that the swing phase of gait can be achieved via a relatively passive leg during normal speed gait [e.g. Mochon and McMahon, Ballistic Walking, *J Biomech* 13:49-57, 1980]. Your task is to turn off joint torques, prescribe pelvis and ankle motion, and then determine a set of initial velocities for the hip and knee that can achieve as nearly as possible swing motion passively. In the program `main.c`, you will need to modify the code such that the generalized coordinates associated with the pelvis (_tx, ty, rotation) and ankle (_angle) have prescribed motion. Specifically, the positions, velocities and accelerations of these degrees of freedom should be set to experimental values. We recommend that you first implement a simulation that uses prescribed motion at the pelvis and ankle together with applied hip and knee torques to recreate the forward dynamic simulation. Doing this involves the following steps:

- Copy the driver program and rename it `main_mod.c`.
- In the `FileView` window (left side), select `main.c` and press `Delete`. Replace it by selecting `Project → Add To Project → Files` and picking `main_mod.c`.
- Change the code to write the forward dynamics simulation to a motion file named `for_pass_swing.mot`.
- Within `SDUMOTION`, prescribe the positions, velocities, and accelerations of pelvis and ankle q's to experimental values read in from the file `gait_swing.dat` (see commands `sdpres` and `sdpresacc`).
- Within `SDUFORCE`, modify the code to apply torques only about the hip and knee.
- Rerun your simulation.

This simulation should reproduce the results you got when you applied torques for each of the generalized coordinates. After verifying this, you should turn off the hip and knee torques and then vary the initial hip and knee velocities to try to achieve the swing motion passively.

- Within `SDUFORCE`, comment out any applied torques.
- Prior to the forward dynamic simulation, alter the initial hip and knee angular velocities in the state vector *y*.
- Manually iterate trying different initial knee and hip angular velocities until you get a swing motion that is similar to the original motion.

How good of an agreement were you able to obtain between the actual and passive swing motions? Are there any physical factors not included in your model that might need to be included to get your passive swing kinematics to agree better with experimental data? Submit your modified driver program and plots comparing your passive swing joint angles with the original prescribed joint angles as described in the deliverables section.

XI. Stretches

If you have time, you can use the model to try to achieve the following:

- Implement appropriate corrections such that the joint angles determined from the forward dynamics simulation agree exactly with experimental data (hint: consider feedback control).
- Allowing passive motion at the ankle, determine a set of initial hip, knee and ankle velocities that result in passive motion that agrees with experimental data.
- Implement passive ligament torques at the knee to restrict hyperextension.

XII. References

Delp SL, Loan JP, Hoy MG, Zajac FE, Topp EL and Rosen JM (1990) "An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures", IEEE Trans Biomed Eng, 37:757-67.

Hollars MG, Rosenthal DE and Sherman MA (1994) SD/FAST User's Manual, Version B.2, Symbolic Dynamics, Inc., Mountain View, CA.

McConville JT, Clauser CE, Churchill TD, Cuzzi J and Kaleps I (1980) "Anthropometric relationships of body and body segmenet moments of inertia," Technical Report AFAMRL-TR-80-119. Air Force Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio.

Mochon S, McMahon TA. (1980) "Ballistic Walking," J Biomech 13:49-57.

Piazza SJ and Delp SL (1996) "The influence of muscles on knee flexion during the swing phase of gait," J Biomech 29: 723-733.

Risher DW, Schutte LM and Runger CF (1997) "The use of inverse dynamics solutions in direct dynamics simulations," J Biomech Engin 119:417-422.

Winter DA, 1990, Biomechanics and Motor Control of Human Movement, 2nd Ed., Wiley, New York, NY.