Agenda

14:00 – 14:15	Welcome and goals of workshop *– Jen Hicks*
14:15 – 14:35	Inverse dynamics and static optimization: how it works, exercises, & practice *– Jeff Reinbolt*
14:35 – 15:00	Guided analyses and exploration on your own *– Jeff Reinbolt & You*
15:00 – 15:20	Forward dynamics simulation: how it works, exercises, & practice *– Jeff Reinbolt*
15:20 – 15:45	Guided simulation and exploration on your own *– Jeff Reinbolt & You*
15:45 – 16:05	**Interfacing OpenSim models with MATLAB/Simulink** ***– Jeff Reinbolt***
16:05 – 16:55	Guided control system design and exploration on your own *– Jeff Reinbolt & You*
16:55 – 17:00	Closing remarks *– Jen Hicks*

Interfacing OpenSim models with MATLAB® / Simulink®

OpenSim Workshop

Acknowledgements

Home
About Simtk.org
How to Contribute

Search Simtk.org
Go
Advanced Search

News | Log Out
Create Project | My Page

Dynamic Simulation of Movement Based on OpenSim and MATLAB®/Simulink®

Project Overview

Overview
- Update
- Statistics
- Geography of use

Team
Downloads
Documents
Publications
Advanced

Downloads & Source Code

This project has no public downloads, but makes source code available.

Description: Numerical simulations are playing an increasingly important role in solving complex engineering problems, and have the potential to revolutionize medical decision making and treatment design. Musculoskeletal diseases cost the United States economy an estimated $849 billion a year (equal to 7.7% of the gross domestic product) and place great demands on the healthcare system. This research area could greatly benefit from computational tools that offer greater understanding of neuromuscular biomechanics, and predictive capabilities for optimal surgical and rehabilitation treatment planning.

The MATLAB®/Simulink® package is the world's leading mathematical computing software for engineers and scientists in industry, government, and education. Although Simulink® extends MATLAB® with a graphical environment for rapid design, control, and simulation of complex dynamic systems, this powerful package has limited resources for simulations of neuromusculoskeletal systems. On the contrary, OpenSim is a popular open-source platform for modeling, simulating, and analyzing neuromusculoskeletal systems, but it is lacks the robust design and control tools of Simulink®.

This project is an interface between OpenSim and MATLAB®/Simulink® that combines relevant strengths (e.g., neuromusculoskeletal dynamics, rapid model-based design, control systems, and numerical simulation) of each individual software package. The foundation of this interface is a MATLAB® S-function (system-function) based on an OpenSim model as a Simulink® block written in C++ and compiled as a MEX-file using the MATLAB® mex utility.

Project Lead

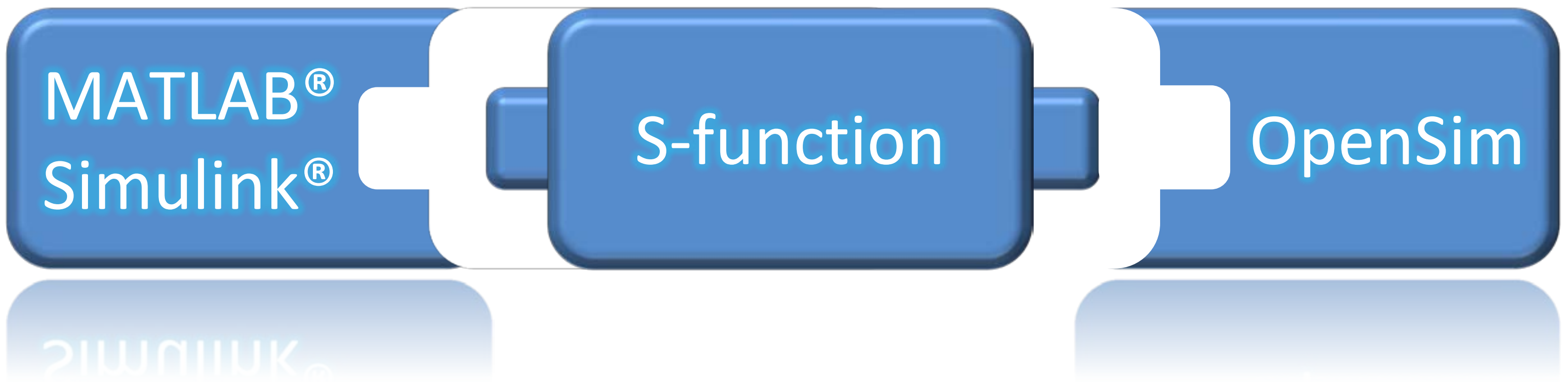
misagh mansouri
Contact

Jeff Reinbolt
Contact

Driving Biological Problems

This project is part of Neuromuscular Biomechanics

Why do we care about MATLAB and OpenSim?



Pros

- World's leading ***mathematical computing software***
- Used by engineers and scientists in ***industry, government, and education***
- Rich resources for ***numerical integration***
- Simulink extends MATLAB with a graphical environment for ***rapid design, control, and simulation*** of complex dynamic systems

Con

- Limited resources for simulating and analyzing neuromusculoskeletal systems

Pros

- Popular ***open-source software*** maintained on ***SimTK.org***
- Used for simulating and analyzing ***neuromusculoskeletal systems***
- Offers ***tracking algorithms*** (CMC), ***actuators*** (muscles), and ***analyses*** (muscle-induced accelerations).
- GUI provides tools for ***viewing models, editing muscles, generating muscle-actuated simulations, and plotting results***

Con

- Limited resources for rapid design and control of dynamic systems

Key Concepts

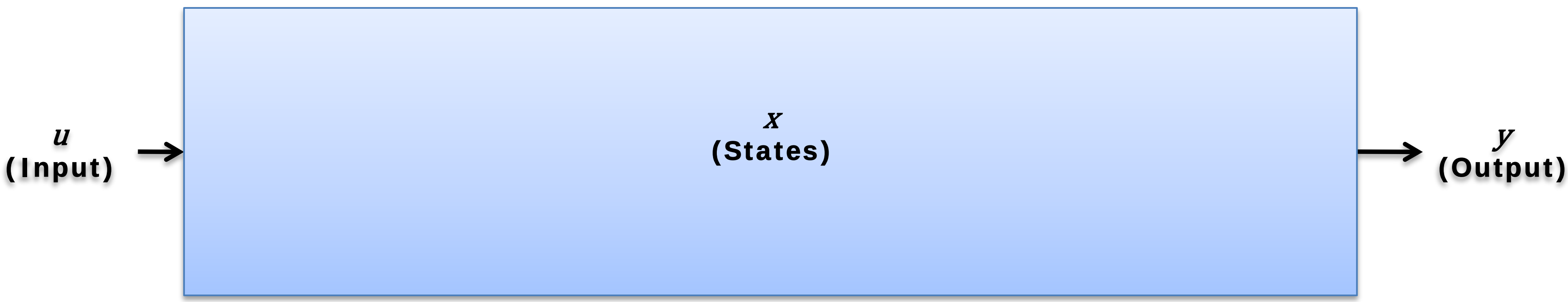
• **S-function**	system function
• **Simulink block**	inputs, states, and outputs
• **Forward dynamics**	controls, states, and integration
• **Feedback**	closed-loop control

S-Function: System Function

- S-functions are ***system-functions*** that extend the capabilities of Simulink
- An *S-function* is a ***computer language description of a Simulink block*** written in MATLAB®, C, C++, or Fortran and compiled as MEX-files
- S-functions are ***dynamically linked subroutines*** that MATLAB can automatically load and execute
- The S-function API enables you to ***interact with the Simulink engine*** very similar to built-in Simulink blocks
- By following a set of simple rules, you can ***implement an algorithm in an S-function (e.g., interface with OpenSim)*** and use the S-Function block in a Simulink model
- After you write your S-function, you can ***customize the user interface using masking***

Simulink Block: Inputs, States, and Outputs

Basic Mathematics of Simulink Blocks



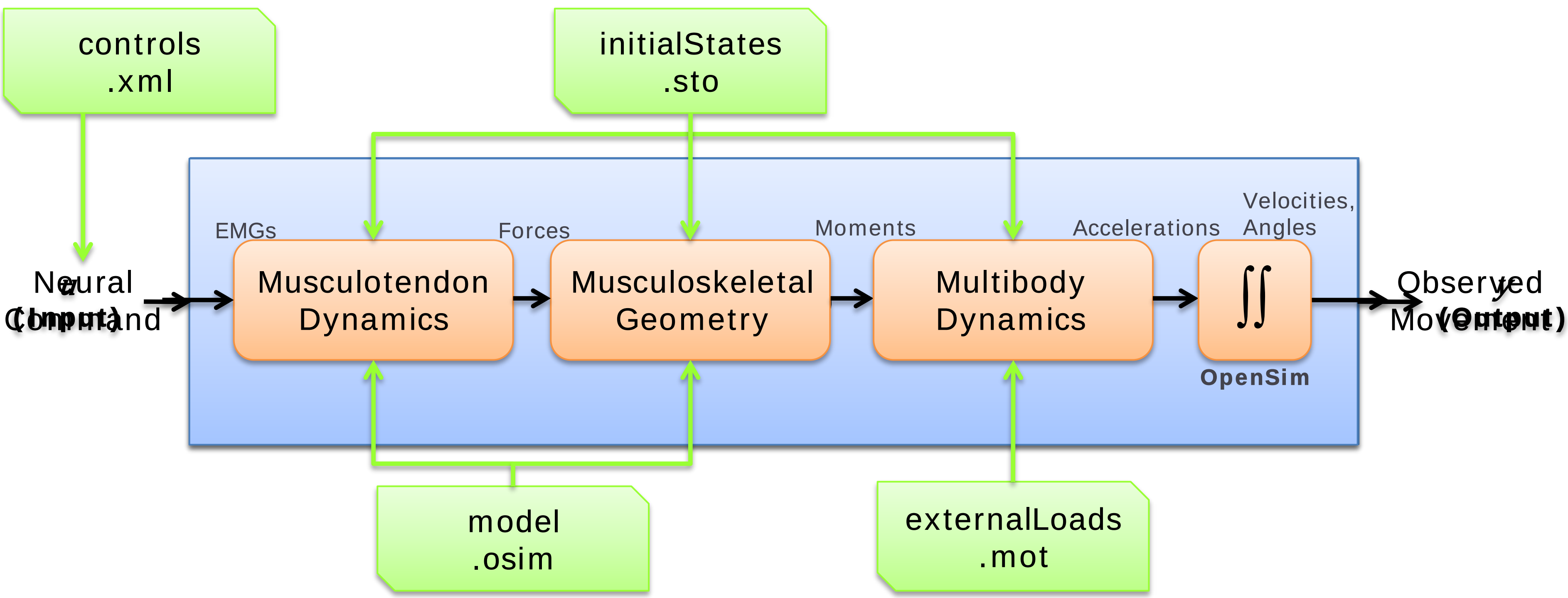
$$y = f_o(t, x, u) \qquad \text{(Output)}$$

$$\dot{x} = f_d(t, x, u) \qquad \text{(Derivatives)}$$

Forward Dynamics: Controls, States, and Integration

Simulink
OpenSim
Files

OpenSim Forward Dynamics



Forward Dynamics: Controls, States, and Integration

Simulink

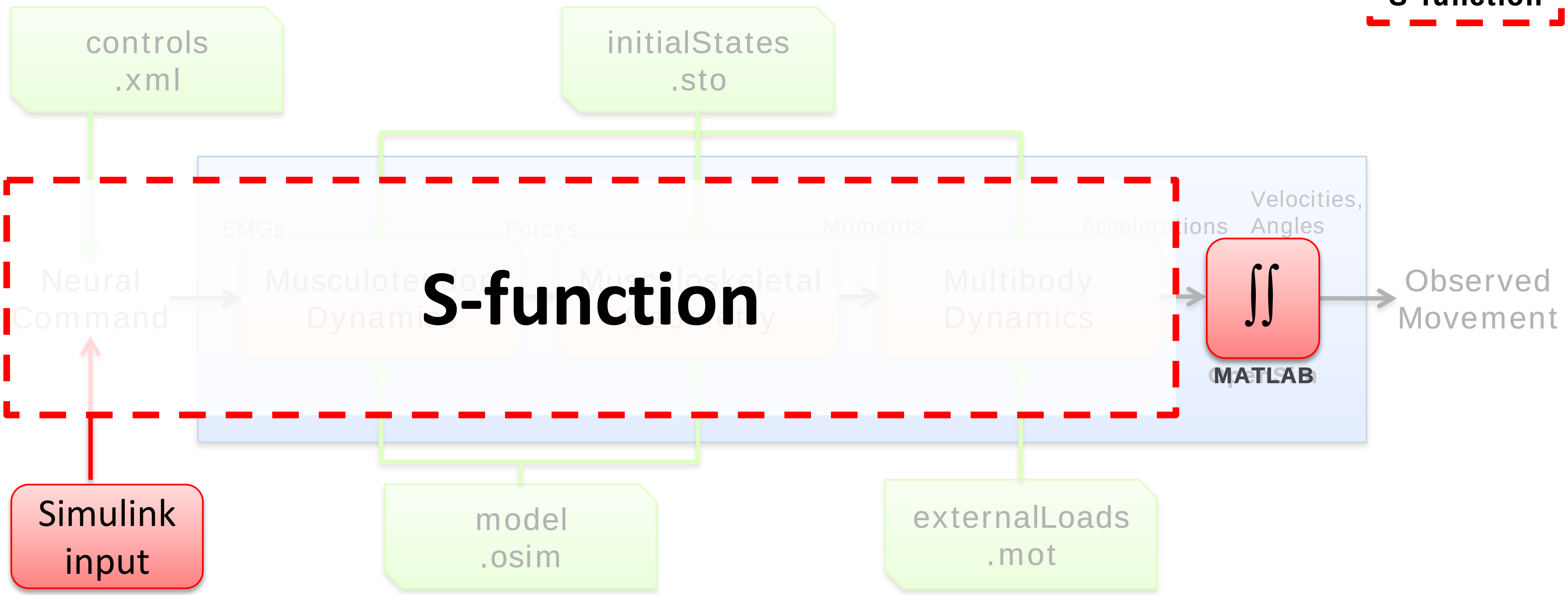
OpenSim

Files

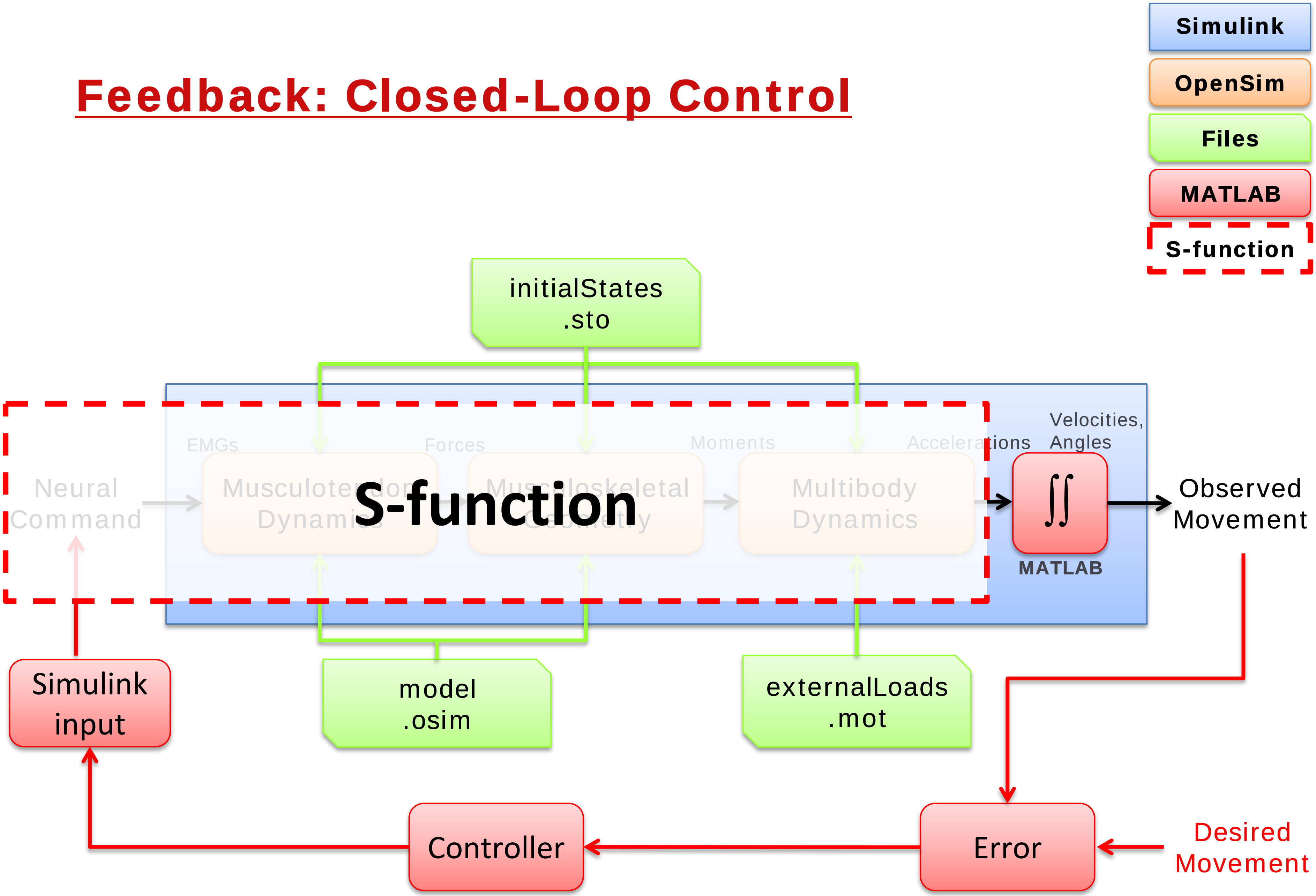
MATLAB

S-function

MATLAB/Simulink Forward Dynamics



Feedback: Closed-Loop Control



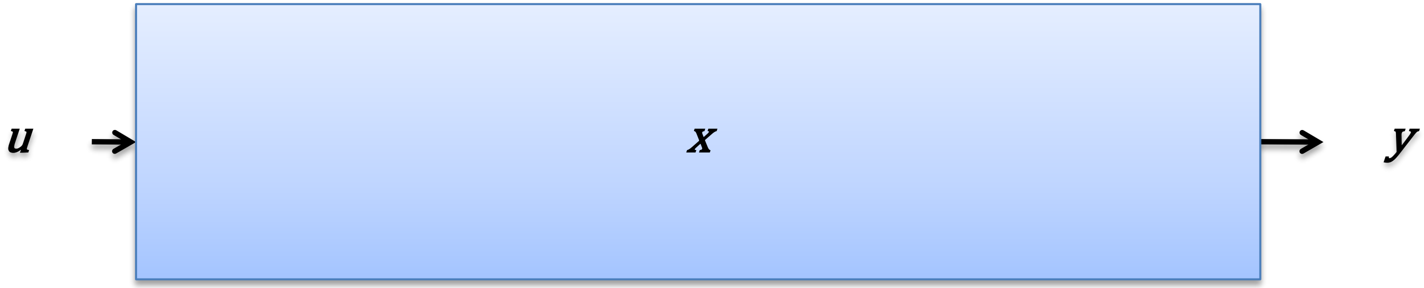
Exercise

1. What computer language can an S-function be written in?

   A. MATLAB
   B. C
   C. C++
   D. All of the above and Fortran

Exercise

2. What are the three basic parts of every Simulink block?



A. Coordinates, velocities, and accelerations

B. Muscle force, length, and velocity

C. Inputs, states, and outputs

D. Controls, states, and integration

Exercise

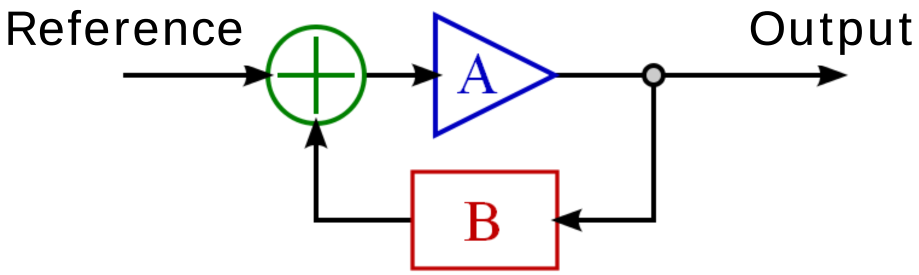
3. Which software provides the integrator to solve the dynamical equations for the model states when using the OpenSim interface block in Simulink?

A. OpenSim
B. MATLAB
C. Simulink
D. None of the above

Exercise

4. What control system design feature allows a controller to take the difference between the reference set point and the block output to change the inputs to the system?

   A. Closed loop
   B. Open loop
   C. Feedback
   D. A and C



Agenda

14:00 – 14:15	Welcome and goals of workshop *– Jen Hicks*
14:15 – 14:35	Inverse dynamics and static optimization: how it works, exercises, & practice *– Jeff Reinbolt*
14:35 – 15:00	Guided analyses and exploration on your own *– Jeff Reinbolt & You*
15:00 – 15:20	Forward dynamics simulation: how it works, exercises, & practice *– Jeff Reinbolt*
15:20 – 15:45	Guided simulation and exploration on your own *– Jeff Reinbolt & You*
15:45 – 16:05	Interfacing OpenSim models with MATLAB/Simulink *– Jeff Reinbolt*
16:05 – 16:55	**Guided control system design and exploration on your own** ***– Jeff Reinbolt & You***
16:55 – 17:00	Closing remarks *– Jen Hicks*