Simulation of Falling in the Elderly and Motor-Impaired

Team Members

- Michael Raitor

Description

Use an OpenSim Model controlled with MPC to simulate falls that cannot be simulated experimentally in a laboratory setting. Additionally, use the model to determine which components of aging and disability most significantly contribute to difficulty in control and balance. In humans many factors change simultaneously such as strength and sensory ability, with a predictive model, it would be possible to determine which aspect of the human needs to be augmented to reduce or eliminate falls and fall-related injuries.

I plan to make the following modeling choicest represent the human euro muscular system

Reality	**Model Representation**
Proprioception	Noise injected into position state variables
Muscle Strength	Actuator Limits
Range of Motion	Joint constraints
Reaction Time	Time Delay in control loop
Temporary, spontaneous vestibular loss	Initial Time Delay before controller reacts to falling body
Incorrect understanding of bodily limitations	Internal model constraints differ from simulated physical model

Research Questions

- Can I recreate common types of falls using this simple model?
    - I will determine this when validating the model against existing data sets of balanceing/falling
- Is there a significant difference between torque-actuated and muscle-actuated model performance when simulating falls?
- How different are results in fall mechanics when pulling the floor or pushing the model vs adding a “vestibular loss”?
- Which parameters (outlined above) are most sensitive to perturbations that result in falls?

Progress

- Created Simple inverted pendulum model controlled with an non-limited torque actuator.
- Verified the model and dynamics simulation method to be used with three separate simulations of the pendulum swinging freely from rest:
    - Ran forward simulation in OpenSim
    - Ran forward simulation in Matlab using hand-derived equations of motion of system
    - Ran forward simulation in Matlab using State information from OpenSim, updating at each timestep
- Implemented a PID controller to balance the torque-actuated pendulum
    - Ran forward simulation in Matlab, communicating from OpenSim at each tilmestep
    - Output motion file of the data in order to run in OpenSim
    - Verified output motion by running forward dynamics simulation in OpenSim using control file generated in MatLab

      Video of PID-controlled pendulum in OpenSim
-
- Implemented MPC controller in Matlab successfully controlling torque-actuated pendulum using hand-derived State equations
- Implemented MPC in Matlab/OpenSim that uses state information from OpenSim "plant" model. Control signals are generated from this state information using Matlab and the system is integrated forward in time using approximated system dynamics in Matlab. System dynamics equations generated using finite differences.
  Software Architecture:

Matlab_OpenSim_Arch.pdf

Video of MPC-controlled pendulum in OpenSim

-

- Currently working on making software architecture more robust to begin testing perturbations/altering parameters as described above in *Description.*

Video Presentation

-

Home: BIOE-ME 485 Spring 2017