How Forward Dynamics Works

Musculoskeletal Model Dynamics

In contrast to inverse dynamics where the motion of the model was known and we wanted to determine the forces and torques that generated the motion, in forward dynamics, a mathematical model describes how coordinates and their velocities change due to applied forces and torques (moments).

From Newton's second law, we can describe the accelerations (rate of change of velocities) of the coordinates in terms of the inertia and forces applied on the skeleton as a set of rigid-bodies:

$$\ddot{q} = \left[M(q)\right]^{-1}\left\{\tau + C(q,\dot{q}) + G(q) + F\right\}$$ Multibody dynamics

where $\ddot{q}$ is the coordinate accelerations due to joint torques, τ , Coriolis and centrifugal forces, $C(q,\dot{q})$, as a function of coordinates, q , and their velocities, $\dot{q}$, gravity, $G(q)$, and other forces applied to the model, F , and $\left[M(q)\right]^{-1}$ is the inverse of the mass matrix.

$$\tau_m = \left[R(q)\right]f(a,l,\dot{l})$$ Moments due to muscle forces

$$\dot{l} = \Lambda(a,l,q,\dot{q})$$ Muscle contraction dynamics

$$\dot{a} = \mathrm{A}(a,x)$$ Muscle activation dynamics

The net muscle moments, τ_m , in turn, are a result of the moment arms, $R(q)$, multiplied by muscle forces, f , which are a function of muscle activations, a , and muscle fiber lengths, l , and velocities, $\dot{l}$. Muscle fiber velocities are governed by muscle contraction dynamics, Λ , which is dependent on the current muscle activations and fiber lengths as well as the coordinates and their velocities. Activation dynamics, A , describes how the activation rates, $\dot{a}$, of the muscles respond to input neural excitations, x , generally termed the model's controls. These form a set of differential equations that model *musculoskeletal dynamics*.

States of a Musculoskeletal Model

The *state* of a model is the collection of all model variables defined at a given instant in time that are governed by dynamics. The model dynamics describe how the model will advance from a given state to another through time. In a *musculoskeletal model* the states are the coordinates and their velocities and muscle activations and muscle fiber lengths. The dynamics of a model require the state to be known in order to calculate the rate of change of the model states (joint accelerations, activation rates, and fiber velocities) in response to forces and controls.

Controlling a Musculoskeletal Model

The forces (e.g., muscles) in a *musculoskeletal model* are governed by dynamics and have inputs that affect their behavior. In OpenSim, these inputs are called the *controls* of a model, which can be excitations for muscles or torque generators. Ultimately, controls determine the forces and/or torques applied to the model and therefore determine the resultant motion.

Numerical Integration of Dynamical Equations

A simulation is the integration of the musculoskeletal model's dynamical equations starting from a user-specified initial state. After applying the controls, the activation rates, muscle fiber velocities, and coordinate accelerations are computed. Then, new states at small time interval in the future are determined by numerical integration. A 5th-order Runge-Kutta-Feldberg integrator is used to solve (numerically integrate) the dynamical equations for the trajectories of the musculoskeletal model states over a definite interval in time. The Forward Dynamics Tool is an open-loop system that applies muscle /actuator controls with no feedback, or correction mechanism, therefore the states are not required to follow a desired trajectory.

The Forward Tool and CMC

The Forward Dynamics Tool uses the same model and actuator set used in CMC, together with the initial states and controls computed during the CMC step, to run a muscle-driven forward dynamic simulation that aims to reproduce the same motion tracked by CMC. The input of initial states is optional. If not entered, the model assumes default values for the state variables. Similarly, if no controls are specified as input then the integration proceeds with zero value for the controls.

As in CMC and RRA, a 5th order Runge-Kutta-Feldberg integrator is used. In contrast to CMC, which used PD controllers in a closed-loop system to ensure tracking of the desired trajectories, the Forward Dynamics Tool is an open-loop system. That is, it blindly applies the recorded actuator controls with no feedback or correction mechanism to help ensure accurate tracking. In theory, starting forward with the exact same conditions as CMC, and feeding it the exact same controls computed by CMC, it should reproduce the same trajectory computed during CMC. However, even tiny differences in values (due to truncation or round-off) or in the ways these values are used by the Forward Dynamics Tool as compared to CMC will cause the forward simulation to diverge from the expected trajectory. This is particularly a problem for longer simulations, in which small differences have more time to accumulate, and in which divergences can become more noticeable, eventually causing the simulation to become completely unstable.

The discrepancies between the Forward Dynamics Tool and CMC can be reduced in a number of ways, as was done with the OpenSim example subjects gait2354 and gait2392:

- During the CMC step, it is recommended that the output precision be set to 20 to ensure that the values read in and used by the Forward Dynamics Tool match the values from CMC with a high precision.
- The Forward Dynamics Tool sets the model's initial state to the CMC state values corresponding to the Forward Dynamics Tool's initial time. If you set the Forward Dynamics Tool to start at a time for which no CMC state values are available, the initial time is adjusted to the nearest time that does have state values. This is done automatically.
- Through a property in the Forward Dynamics Tool's setup file (<use_specified_dt>), you can also use the same integration time steps as were used in CMC. It is recommended that this property be set to **true** to minimize the divergence of long forward simulations. If set to **false**, the Forward Dynamics Tool's integrator adaptively computes its own time steps based on the desired set error tolerances. As these time steps tend to differ from those used in CMC, the results usually diverge faster.

Next: How to Use the Forward Dynamics Tool

Previous: Getting Started with Forward Dynamics

Home: Forward Dynamics