Introduction to Programming with SimTK

Peter Eastman
SimTK Workshop, September 25, 2008
(ExamplePendulum.cpp)

```cpp
#include "SimTKsimbody.h"
#include "SimTKsimbody_aux.h"
using namespace SimTK;

int main() {

    // Create the system.
    MultibodySystem system;
    SimbodyMatterSubsystem matter(system);
    GeneralForceSubsystem forces(system);
    Force::UniformGravity gravity(forces, matter, Vec3(0, -9.8, 0));

    Body::Rigid pendulumBody(MassProperties(1.0, Vec3(0), Inertia(1)));
    pendulumBody.addDecoration(Transform(), DecorativeSphere(0.1));
    MobilizedBody::Pin pendulum(matter.updGround(), Transform(Vec3(0)), pendulumBody,
                                Transform(Vec3(0, 1, 0)));
    system.updDefaultSubsystem().addEventReporter(new VTKEventReporter(system, 0.01));

    // Initialize the system and state.
    system.realizeTopology();
    State state = system.getDefaultState();
    pendulum.setOneU(state, 0, 1.0);

    // Simulate it.
    RungeKuttaMersonIntegrator integ(system);
    TimeStepper ts(system, integ);
    ts.initialize(state);
    ts.stepTo(100.0);
}
```
Create the System

MultibodySystem system;
SimbodyMatterSubsystem matter(system);
GeneralForceSubsystem forces(system);
Force::UniformGravity gravity(forces, matter, Vec3(0, -9.8, 0));

• MultibodySystem is a subclass of System
  – Provides features for multibody dynamics
• SimbodyMatterSubsystem constructs an arbitrary set of bodies
• GeneralForceSubsystem applies forces to the bodies
  – Force::UniformGravity is an example of a force
Add a Body

Body::Rigid pendulumBody(MassProperties(1.0, Vec3(0), Inertia(1)))
pendulumBody.addDecoration(Transform(), DecorativeSphere(0.1));
MobilizedBody::Pin pendulum(matter.updGround(), Transform(Vec3(0)), pendulumBody,
Transform(Vec3(0, 1, 0)));

• Body defines *physical properties* of a body
  – Mass, moment of inertia, appearance (optional)

• MobilizedBody combines *physical properties* (a Body) with *mobilities* (generalized coordinates)
  – Many subclasses for different types of motion
    • MobilizedBody::Pin allows rotation around a single axis
  – Multiple MobilizedBodies can use the same Body (if they have identical physical properties)
Display an Animation

\[\text{system.updDefaultSubsystem().addEventReporter(new VTKEventReporter(system, 0.01));}\]

• VTKEventReporter is an \textit{event reporter}
  – Invoked at regular intervals during the simulation
  – Outputs information (in this case, draws an image of the System)
  – We’ll discuss this much more later!
Initialize the System/State

```java
system.realizeTopology();
State state = system.getDefaultState();
pendulum.setOneU(state, 0, 1.0);
```

- `realizeTopology()` performs initialization
  - Must be called after constructing the System, before creating a State for it
- Create a State by making a clone of the System’s default state
  - All state variables set to default values
- Can modify the state variables before starting the simulation
Simulate It!

```java
RungeKuttaMersonIntegrator integ(system);
TimeStepper ts(system, integ);
ts.initialize(state);
ts.stepTo(100.0);
```

• An Integrator advances the continuous equations of motion
• TimeStepper invokes the Integrator repeatedly and handles events
• We’ll discuss this much more later!
Exercises

• Increase the length of the pendulum to 2 m
• Increase the duration of the simulation to 150 s
• Decrease gravity to 1.6 m/s² (lunar gravity)
• Change the location of the base of the pendulum
• Change the pendulum to a Ball joint
  – It now has three generalized coordinates. Initialize the speed of each one to a different value.
• Change it into a double pendulum (one pendulum attached to the end of another one)
MobilizedBody Transforms

\[ \mathbf{T}_{\text{total}} = (\mathbf{T}_{\text{out}})^{-1} \cdot \mathbf{T}_{\text{mob}} \cdot \mathbf{T}_{\text{in}} \]
A Chain of Bodies

(ExampleChain.cpp)

```cpp
MobilizedBodyIndex lastBody = matter.getGround().getMobilizedBodyIndex();
for (int i = 0; i < 10; ++i) {
    MobilizedBody::Ball pendulum(matter.updMobilizedBody(lastBody), Transform(Vec3(0)), pendulumBody,
                                Transform(Vec3(0, 1, 0)));
    lastBody = pendulum.getMobilizedBodyIndex();
}
```

- MobilizedBodyIndex can be used to refer to a MobilizedBody in the matter subsystem
  - Basically an int, but typesafe
  - Cleaner than referencing a body with a pointer
Randomize the Chain

State state = system.getDefaultState();
Random::Gaussian random;
for (int i = 0; i < state.getNQ(); ++i)
    state.updQ()[i] = random.getValue();

• Sets every generalized coordinate to a random value
• Random is a random number generator
  – Subclasses for uniform and Gaussian distributions
  – Uses a fast, statistically robust algorithm (Mersenne Twister)
Exercises

• Instead of a chain of 10 bodies, make it into 10 independent pendulums, each with its base at a different location
• Attach each body to a randomly chosen existing body (a tree of pendulums)
• Add a spring connecting the end of the chain to ground (see Force::TwoPointLinearSpring)