Two-D Anthropomorphic Walking Model

Two segment system, rolls on curved feet

Initialization

```
Needs["DynamicsWorkbench`"]
```

Equations of Motion

Degrees of freedom: stance: u[1], swing: u[2]

Start a new model. Add the first body, which is the stance leg, which is attached to the ground with a hinge joint for now (we will eventually change this, but a hinge joint is close to the same situation as having a curved foot rolling on the ground). The mass and moment of inertia are specified, as are two vectors: InbToJnt is the vector from the center of mass of the inboard joint (ground) to the new body, and BodyToJnt is the vector from the new body's center of mass to the same joint.

```
NewModel[];
AddBody[sta, ground, Hinge, Axis -> ground[3], Mass -> M,
  Inertia -> {0, 0, Il}, InbToJnt -> 0, BodyToJnt -> -(C - R) sta[2] - R ground[2]];
```

Since a hinge joint is not the same as a curved foot, we need to make sure that the acceleration of the center of mass is correct. After AddBody, that acceleration, AccCOM[sta], is automatically defined and is correct for the hinge joint. The following command explicitly sets the acceleration to what it should be for the curved foot.

```
AccCOM[sta] = -R u[1]' ground[1] +
   AngAcc[sta] × ((C - R) sta[2]) + AngVel[sta] × (AngVel[sta] × ((C - R) sta[2]));
```

Now add the pelvis to the stance leg, with specified mass and moment of inertia.

```
AddBody[pelv, sta, Fixed, Mass -> Mp,
  Inertia -> {0, 0, 0}, InbToJnt -> (L - C) sta[2], BodyToJnt -> 0];
```

Finally, add the swing leg to the pelvis. By default, most angles are defined relative to the previous body (joint angles). The RelativeTo is used to make the angle absolute.

```
AddBody[swi, pelv, Hinge, RelativeTo -> ground, Axis -> pelv[3],
  Inertia -> {0, 0, Il}, Mass -> M, InbToJnt -> 0, BodyToJnt -> (L - C) swi[2]];
```

Apply gravitational forces to each body, at the center of mass. The 0 is the location of the force relative to c.o.m.

```
AppFrc[sta, Mass[sta] grav, 0];
AppFrc[pelv, Mass[pelv] grav, 0];
AppFrc[swi, Mass[swi] grav, 0];
```

```
grav = g (-Cos[gamma] ground[2] + Sin[gamma] ground[1]);
```

Generate the equations of motion

```
eom = EOM[]
```

$$
\begin{aligned}
\{&(-g\,M\,R+g\,M\,(-C+R)\,\mathrm{Cos}[q_1])\,\mathrm{Sin}[\mathrm{gamma}]+(-g\,M\,R+g\,M\,(-L+R)\,\mathrm{Cos}[q_1])\,\mathrm{Sin}[\mathrm{gamma}]+\\
&\quad(-g\,Mp\,R+g\,Mp\,(-L+R)\,\mathrm{Cos}[q_1])\,\mathrm{Sin}[\mathrm{gamma}]-g\,M\,(-C+R)\,\mathrm{Cos}[\mathrm{gamma}]\,\mathrm{Sin}[q_1]-\\
&\quad g\,M\,(-L+R)\,\mathrm{Cos}[\mathrm{gamma}]\,\mathrm{Sin}[q_1]-g\,Mp\,(-L+R)\,\mathrm{Cos}[\mathrm{gamma}]\,\mathrm{Sin}[q_1]+\\
&\quad((-C+L)\,M\,R+(-C+L)\,Mp\,R+2\,M\,(C-R)\,R+Mp\,(C-R)\,R)\,\mathrm{Sin}[q_1]\,u_1^2+\\
&\quad((C-L)\,M\,(-L+R)\,\mathrm{Sin}[q_1-q_2]+(C-L)\,M\,R\,\mathrm{Sin}[q_2])\,u_2^2+\\
&\quad\big(-Il-2\,M\,R^2-Mp\,R^2+M\,(C-R)\,(-C+R)+(-C+L)\,M\,(-L+R)+(-C+L)\,Mp\,(-L+R)+\\
&\qquad M\,(C-R)\,(-L+R)+Mp\,(C-R)\,(-L+R)+((C-L)\,M\,R+(C-L)\,Mp\,R-2\,M\,(C-R)\,R-\\
&\qquad\quad Mp\,(C-R)\,R+M\,R\,(-C+R)+M\,R\,(-L+R)+Mp\,R\,(-L+R))\,\mathrm{Cos}[q_1]\big)\,u_1'+\\
&\quad((C-L)\,M\,(-L+R)\,\mathrm{Cos}[q_1-q_2]+(-C+L)\,M\,R\,\mathrm{Cos}[q_2])\,u_2'==0,\\
&g\,(-C+L)\,M\,\mathrm{Sin}[\mathrm{gamma}-q_2]+\big(-(-C+L)^2\,M+(C-L)\,M\,(C-R)\big)\,\mathrm{Sin}[q_1-q_2]\,u_1^2+\\
&\quad\big(\big((-C+L)^2\,M+(-C+L)\,M\,(C-R)\big)\,\mathrm{Cos}[q_1-q_2]+(-C+L)\,M\,R\,\mathrm{Cos}[q_2]\big)\,u_1'+\\
&\quad(-Il+(C-L)\,(-C+L)\,M)\,u_2'==0\}
\end{aligned}
$$

Energy

Here are the potential and kinetic energies of the system

```
PE = -(Mass[sta] grav.(PosCOM[sta] - R q[1] ground[1]) +
     Mass[swi] grav.(PosCOM[swi] - R q[1] ground[1]) +
     Mass[pelv] grav.(PosCOM[pelv] - R q[1] ground[1]))
```

$$
\begin{aligned}
&-M\,(-g\,R\,\mathrm{Cos}[\mathrm{gamma}]-g\,(C-R)\,\mathrm{Cos}[\mathrm{gamma}]\,\mathrm{Cos}[q_1]-\\
&\qquad g\,(C-R)\,\mathrm{Sin}[\mathrm{gamma}]\,\mathrm{Sin}[q_1]-g\,R\,\mathrm{Sin}[\mathrm{gamma}]\,q_1)-\\
&\ Mp\,(-g\,R\,\mathrm{Cos}[\mathrm{gamma}]-g\,(L-R)\,\mathrm{Cos}[\mathrm{gamma}]\,\mathrm{Cos}[q_1]-\\
&\qquad g\,(L-R)\,\mathrm{Sin}[\mathrm{gamma}]\,\mathrm{Sin}[q_1]-g\,R\,\mathrm{Sin}[\mathrm{gamma}]\,q_1)-\\
&\ M\,(-g\,R\,\mathrm{Cos}[\mathrm{gamma}]-g\,(L-R)\,\mathrm{Cos}[\mathrm{gamma}]\,\mathrm{Cos}[q_1]-g\,(C-L)\,\mathrm{Cos}[\mathrm{gamma}]\,\mathrm{Cos}[q_2]-\\
&\qquad g\,(L-R)\,\mathrm{Sin}[\mathrm{gamma}]\,\mathrm{Sin}[q_1]-g\,(C-L)\,\mathrm{Sin}[\mathrm{gamma}]\,\mathrm{Sin}[q_2]-g\,R\,\mathrm{Sin}[\mathrm{gamma}]\,q_1)
\end{aligned}
$$

```
KE = 1/2 (Mass[sta] VelCOM[sta].VelCOM[sta] + Mass[swi] VelCOM[swi].VelCOM[swi] +
    Mass[pelv] VelCOM[pelv].VelCOM[pelv] + AngVel[sta].Inertia[sta].AngVel[sta] +
    AngVel[swi].Inertia[swi].AngVel[swi] + AngVel[pelv].Inertia[pelv].AngVel[pelv])
```

$$\frac{1}{2}\Big(\mathrm{Il}\,u_1^2 + M\left((C-R)^2 u_1^2 + R^2 u_1^2 + 2\,(C-R)\,R\,\mathrm{Cos}[q_1]\,u_1^2\right) + \mathrm{Mp}\left(R^2 u_1^2 - 2\,R\,\mathrm{Cos}[q_1]\,u_1\,(-(-C+L)\,u_1-(C-R)\,u_1) + (-(-C+L)\,u_1-(C-R)\,u_1)^2\right) + \mathrm{Il}\,u_2^2 + M\left(R^2 u_1^2 - 2\,R\,\mathrm{Cos}[q_1]\,u_1\,(-(-C+L)\,u_1-(C-R)\,u_1) + (-(-C+L)\,u_1-(C-R)\,u_1)^2 + 2\,(C-L)\,R\,\mathrm{Cos}[q_2]\,u_1\,u_2 - 2\,(C-L)\,\mathrm{Cos}[q_1-q_2]\,(-(-C+L)\,u_1-(C-R)\,u_1)\,u_2 + (C-L)^2 u_2^2\right)\Big)$$

Contact Point

Pnt2 is the location of the point right under the swing foot

```
Pnt2 = PosPnt[(R - C) swi[2], swi] - R ground[2]
```

$(L-R)\,\underline{\mathrm{sta}_2} + (-L+R)\,\underline{\mathrm{swi}_2}$

This is the height of that point above the ground

```
height = Pnt2.ground[2]
```

$(L-R)\,\mathrm{Cos}[q_1] + (-L+R)\,\mathrm{Cos}[q_2]$

Angular Momentum Calculations

Of whole machine

Angular momentum of the whole machine right before impact. The command asks for the angular momentum of the specified bodies about Pnt2, which was previously defined to be the point right under the swing leg.

```
amb = AngMom[{sta, pelv, swi}, Pnt2].ground[3] // Simplify
```

$$\mathrm{Il}\,u_1 + M\,(L-R)\,(R\,\mathrm{Cos}[q_1] + (C-R)\,\mathrm{Cos}[q_1-q_2])\,u_1 + \mathrm{Mp}\,(L-R)\,(R\,\mathrm{Cos}[q_1] + (L-R)\,\mathrm{Cos}[q_1-q_2])\,u_1 + M\,(C-R)\,(C-L+R\,\mathrm{Cos}[q_1] + (L-R)\,\mathrm{Cos}[q_1-q_2])\,u_1 + M\,R\,(R + (C-R)\,\mathrm{Cos}[q_2])\,u_1 + \mathrm{Mp}\,R\,(R + (L-R)\,\mathrm{Cos}[q_2])\,u_1 + M\,R\,(R + (C-L)\,\mathrm{Cos}[q_1] + (L-R)\,\mathrm{Cos}[q_2])\,u_1 + \mathrm{Il}\,u_2 + (C-L)\,M\,(C-R+R\,\mathrm{Cos}[q_2])\,u_2$$

Angular momentum of the whole machine right after impact. Right after impact, the old swing leg is now called the stance leg. To find the angular momentum about the same point as before, we now have to ask for the point right under the new stance leg.

```
ama = AngMom[{sta, pelv, swi}, 0].ground[3] // Simplify
```

$$\mathrm{Il}\, u_1 + M R (R + (C - R) \mathrm{Cos}[q_1]) u_1 + \mathrm{Mp}\, R (R + (L - R) \mathrm{Cos}[q_1]) u_1 + M (C - R) (C - R + R \mathrm{Cos}[q_1]) u_1 + \mathrm{Mp} (L - R) (L - R + R \mathrm{Cos}[q_1]) u_1 - M (L - R) (-L + R - R \mathrm{Cos}[q_1] + (-C + L) \mathrm{Cos}[q_1 - q_2]) u_1 + M R (R + (L - R) \mathrm{Cos}[q_1] + (C - L) \mathrm{Cos}[q_2]) u_1 + \mathrm{Il}\, u_2 + (C - L) M (C - L + (L - R) \mathrm{Cos}[q_1 - q_2] + R \mathrm{Cos}[q_2]) u_2$$

The two angular momenta above should be equal. The angular momentum before impact, amb, is computed from the state of the system immediately before impact. The angular momentum after impact is given in terms of the new state of the system.

Of trailing leg about the hip

Angular momentum of the trailing leg about the hip, right before impact. The command asks for the angular momentum of the stance leg (which is the trailing leg before impact), about the hip (specified as a vector from the stance leg center of mass to the pelvis).

```
amb2 = AngMom[sta, PosPnt[(L - C) sta[2], sta]].ground[3] // Simplify
```

$$(\mathrm{Il} + (C - L) M (C - R) + (C - L) M R \mathrm{Cos}[q_1]) u_1$$

Angular momentum of the trailing leg about the hip, right after impact. Note that after impact the trailing leg is now called the swing leg, and the hip is now specified as a point from the center of mass of the swing leg. Note that instead of PosPnt[...] shown, another exactly equivalent vector is Pos-COM[pelv]. Note also that all of the items in this new equation are in terms of the state of the system after impact.

```
ama2 = AngMom[swi, PosPnt[(L - C) swi[2], swi]].ground[3] // Simplify
```

$$(C - L) M (L - R) \mathrm{Cos}[q_1 - q_2] u_1 + (C - L) M R \mathrm{Cos}[q_2] u_1 + \mathrm{Il}\, u_2 + (C - L)^2 M u_2$$

Here is how to use these equations. ama and ama2 are two expressions that are linear in the unknowns u1 and u2 (after impact). amb and amb2 are two expressions that depend on the state before impact, and are known. By setting {ama,ama2}=={amb,amb2}, you can solve for u1 and u2 (after impact).