```
Quit[];

SetDirectory[NotebookDirectory[]]
<< HurToolbox.m
```

```
D:\Dropbox\TAMU\Group\Project\Walker\five_link\AbsAngleJointTorque
```

```
HurToolbox for modeling and analysis of multibody systems 1.0.1.
HurToolbox mainly uses vector manipulation (vectors, dyadics).
Coordinates and matrix representation of the dyadics are also available.
Available methods: Newton-Euler
   Method, Euler-Lagrange Method, Hamiltonian Method, Kane's Method.
Copyright 2019 Pilwon Hur
Department of Mechanical Engineering
Texas A&M University
All rights reserved.
Email questions, comments, or concerns to pilwonhur@tamu.edu.
```

```
HurInitialize[]

HurLoadData["data1.m"]

HurDefineRF[a, b, c, d, e]
HurDefineGeneralizedCoordinates[q1[t], q2[t], q3[t], q4[t], q5[t]]
HurDefineDCM[a, q1[t], {0, 0, 1}]
HurDefineDCM[b, q2[t], {0, 0, 1}]
HurDefineDCM[c, q3[t], {0, 0, 1}]
HurDefineDCM[d, q4[t], {0, 0, 1}]
HurDefineDCM[e, q5[t], {0, 0, 1}]

FootST = 0;
ShankSTCOM = lsa a2;
KneeST = (lsa + lsb) a2;
ThighSTCOM = KneeST + lta b2;
Hip = KneeST + (lta + ltb) b2;
TorsoCOM = Hip + lb / 2 c2;
ThighSWCOM = Hip - ltb d2;
KneeSW = Hip - (ltb + lta) d2;
ShankSWCOM = KneeSW - lsb e2;
FootSW = KneeSW - (lsb + lsa) e2;

HurDefineCOMPos[a, ShankSTCOM]
HurDefineCOMPos[b, ThighSTCOM]
HurDefineCOMPos[c, TorsoCOM]
HurDefineCOMPos[d, ThighSWCOM]
HurDefineCOMPos[e, ShankSWCOM]

HurKinematics[]
```

```
HurDefineMass[a, ms]
HurDefineMass[b, mt]
HurDefineMass[c, mb]
HurDefineMass[d, mt]
HurDefineMass[e, ms]
HurDefineInertia[a, {0, 0, 0, 0, 0, Is}]
HurDefineInertia[b, {0, 0, 0, 0, 0, It}]
HurDefineInertia[c, {0, 0, 0, 0, 0, Ib}]
HurDefineInertia[d, {0, 0, 0, 0, 0, It}]
HurDefineInertia[e, {0, 0, 0, 0, 0, Is}]
```

```
HurDefineVertical[n2]
```

```
Transpose[HurGetJacobian[FootST, a, n]].HurList2Column[{0, 0, 0, 0, 0, tau1}] +
 Transpose[HurGetJacobian[KneeST, a, n]].HurList2Column[{0, 0, 0, 0, 0, -tau2}] +
 Transpose[HurGetJacobian[KneeST, b, n]].HurList2Column[{0, 0, 0, 0, 0, tau2}] +
 Transpose[HurGetJacobian[Hip, b, n]].HurList2Column[{0, 0, 0, 0, 0, -tau3}] +
 Transpose[HurGetJacobian[Hip, c, n]].HurList2Column[{0, 0, 0, 0, 0, tau3}] +
 Transpose[HurGetJacobian[Hip, c, n]].HurList2Column[{0, 0, 0, 0, 0, -tau4}] +
 Transpose[HurGetJacobian[Hip, d, n]].HurList2Column[{0, 0, 0, 0, 0, tau4}] +
 Transpose[HurGetJacobian[KneeSW, d, n]].HurList2Column[{0, 0, 0, 0, 0, -tau5}] +
 Transpose[HurGetJacobian[KneeSW, e, n]].HurList2Column[{0, 0, 0, 0, 0, tau5}]
HurDefineNonConservativeForces[Flatten[%]]
```

```
{{tau1 - tau2}, {tau2 - tau3}, {tau3 - tau4}, {tau4 - tau5}, {tau5}}
```

```
{tau1 - tau2, tau2 - tau3, tau3 - tau4, tau4 - tau5, tau5}
```

```
HurELEquation[]
```

```
HurGlobalELEquation // MatrixForm
```

```
HurGlobalMMatrix // MatrixForm
```

```
HurGlobalCMatrix // MatrixForm
```

```
HurGlobalGVector // MatrixForm
```

```
dyn1 = HurGlobalELEquation[[1]]
HurToJulia[dyn1]
```

```
1/2*(-2*tau1+(2*tau2+(-2*g*lsa*mb*sin(q1)+(-2*g*lsb*mb*sin(q1)+(-4*g*lsa*ms*sin(q1)+(-
  2*g*lsb*ms*sin(q1)+(-4*g*lsa*mt*sin(q1)+(-4*g*lsb*mt*sin(q1)+(2*(lsa+lsb)*(ltb*(mb+(
  ms+mt))+lta*(mb+(ms+2*mt)))*sin((q1+-1*q2))*(q2d)^(2)+(lb*(lsa+lsb)*mb*sin((q1+-1*q3
  ))*(q3d)^(2)+(-2*lsa*lta*ms*sin((q1+-1*q4))*(q4d)^(2)+(-2*lsb*lta*ms*sin((q1+-1*q4))
  *(q4d)^(2)+(-2*lsa*ltb*ms*sin((q1+-1*q4))*(q4d)^(2)+(-2*lsb*ltb*ms*sin((q1+-1*q4))*(
  q4d)^(2)+(-2*lsa*ltb*mt*sin((q1+-1*q4))*(q4d)^(2)+(-2*lsb*ltb*mt*sin((q1+-1*q4))*(
  q4d)^(2)+(-2*lsa*lsb*ms*sin((q1+-1*q5))*(q5d)^(2)+(-2*(lsb)^(2)*ms*sin((q1+-1*q5))*
  (q5d)^(2)+(2*Is*q1dd+(2*(lsa)^(2)*mb*q1dd+(4*lsa*lsb*mb*q1dd+(2*(lsb)^(2)*mb*q1dd+(4
  *(lsa)^(2)*ms*q1dd+(4*lsa*lsb*ms*q1dd+(2*(lsb)^(2)*ms*q1dd+(4*(lsa)^(2)*mt*q1dd+(8*
  lsa*lsb*mt*q1dd+(4*(lsb)^(2)*mt*q1dd+(2*lsa*lta*mb*cos((q1+-1*q2))*q2dd+(2*lsb*lta*
  mb*cos((q1+-1*q2))*q2dd+(2*lsa*ltb*mb*cos((q1+-1*q2))*q2dd+(2*lsb*ltb*mb*cos((q1+-1*
  q2))*q2dd+(2*lsa*lta*ms*cos((q1+-1*q2))*q2dd+(2*lsb*lta*ms*cos((q1+-1*q2))*q2dd+(2*
  lsa*ltb*ms*cos((q1+-1*q2))*q2dd+(2*lsb*ltb*ms*cos((q1+-1*q2))*q2dd+(4*lsa*lta*mt*cos
  ((q1+-1*q2))*q2dd+(4*lsb*lta*mt*cos((q1+-1*q2))*q2dd+(2*lsa*ltb*mt*cos((q1+-1*q2))*
  q2dd+(2*lsb*ltb*mt*cos((q1+-1*q2))*q2dd+(lb*lsa*mb*cos((q1+-1*q3))*q3dd+(lb*lsb*mb*
  cos((q1+-1*q3))*q3dd+(-2*lsa*lta*ms*cos((q1+-1*q4))*q4dd+(-2*lsb*lta*ms*cos((q1+-1*
  q4))*q4dd+(-2*lsa*ltb*ms*cos((q1+-1*q4))*q4dd+(-2*lsb*ltb*ms*cos((q1+-1*q4))*q4dd+(-
  2*lsa*ltb*mt*cos((q1+-1*q4))*q4dd+(-2*lsb*ltb*mt*cos((q1+-1*q4))*q4dd+(-2*lsa*lsb*ms
  *cos((q1+-1*q5))*q5dd+-2*(lsb)^(2)*ms*cos((q1+-1*q5))*q5dd)))))))))))))))))))))))))
  ))))))))))))))))))))))))
```

```
dyn2 = HurGlobalELEquation[[2]]
HurToJulia[dyn2]
```

```
1/2*(-2*tau2+(2*tau3+(-2*g*lta*mb*sin(q2)+(-2*g*ltb*mb*sin(q2)+(-2*g*lta*ms*sin(q2)+(-
  2*g*ltb*ms*sin(q2)+(-4*g*lta*mt*sin(q2)+(-2*g*ltb*mt*sin(q2)+(-2*(lsa+lsb)*(ltb*(mb+
  (ms+mt))+lta*(mb+(ms+2*mt)))*sin((q1+-1*q2))*(q1d)^(2)+(lb*(lta+ltb)*mb*sin((q2+-1*
  q3))*(q3d)^(2)+(-2*(lta)^(2)*ms*sin((q2+-1*q4))*(q4d)^(2)+(-4*lta*ltb*ms*sin((q2+-1
  *q4))*(q4d)^(2)+(-2*(ltb)^(2)*ms*sin((q2+-1*q4))*(q4d)^(2)+(-2*lta*ltb*mt*sin((q2+-
  1*q4))*(q4d)^(2)+(-2*(ltb)^(2)*mt*sin((q2+-1*q4))*(q4d)^(2)+(-2*lsb*lta*ms*sin((q2+
  -1*q5))*(q5d)^(2)+(-2*lsb*ltb*ms*sin((q2+-1*q5))*(q5d)^(2)+(2*lsa*lta*mb*cos((q1+-1*
  q2))*q1dd+(2*lsb*lta*mb*cos((q1+-1*q2))*q1dd+(2*lsa*ltb*mb*cos((q1+-1*q2))*q1dd+(2*
  lsb*ltb*mb*cos((q1+-1*q2))*q1dd+(2*lsa*lta*ms*cos((q1+-1*q2))*q1dd+(2*lsb*lta*ms*cos
  ((q1+-1*q2))*q1dd+(2*lsa*ltb*ms*cos((q1+-1*q2))*q1dd+(2*lsb*ltb*ms*cos((q1+-1*q2))*
  q1dd+(4*lsa*lta*mt*cos((q1+-1*q2))*q1dd+(4*lsb*lta*mt*cos((q1+-1*q2))*q1dd+(2*lsa*
  ltb*mt*cos((q1+-1*q2))*q1dd+(2*lsb*ltb*mt*cos((q1+-1*q2))*q1dd+(2*It*q2dd+(2*(lta)^(
  2)*mb*q2dd+(4*lta*ltb*mb*q2dd+(2*(ltb)^(2)*mb*q2dd+(2*(lta)^(2)*ms*q2dd+(4*lta*ltb*
  ms*q2dd+(2*(ltb)^(2)*ms*q2dd+(4*(lta)^(2)*mt*q2dd+(4*lta*ltb*mt*q2dd+(2*(ltb)^(2)*mt
  *q2dd+(lb*lta*mb*cos((q2+-1*q3))*q3dd+(lb*ltb*mb*cos((q2+-1*q3))*q3dd+(-2*(lta)^(2)*
  ms*cos((q2+-1*q4))*q4dd+(-4*lta*ltb*ms*cos((q2+-1*q4))*q4dd+(-2*(ltb)^(2)*ms*cos((q2
  +-1*q4))*q4dd+(-2*lta*ltb*mt*cos((q2+-1*q4))*q4dd+(-2*(ltb)^(2)*mt*cos((q2+-1*q4))*
  q4dd+(-2*lsb*lta*ms*cos((q2+-1*q5))*q5dd+-2*lsb*ltb*ms*cos((q2+-1*q5))*q5dd)))))))))
  ))))))))))))))))))))))))))))))))))))))
```

```
dyn3 = HurGlobalELEquation[[3]]
HurToJulia[dyn3]
```

```
1/4*(-4*tau3+(4*tau4+(-2*g*lb*mb*sin(q3)+(-2*lb*(lsa+lsb)*mb*sin((q1+-1*q3))*(q1d)^(2)
  +(-2*lb*(lta+ltb)*mb*sin((q2+-1*q3))*(q2d)^(2)+(2*lb*lsa*mb*cos((q1+-1*q3))*q1dd+(2*
  lb*lsb*mb*cos((q1+-1*q3))*q1dd+(2*lb*lta*mb*cos((q2+-1*q3))*q2dd+(2*lb*ltb*mb*cos((
  q2+-1*q3))*q2dd+(4*Ib*q3dd+(lb)^(2)*mb*q3dd)))))))))))
```

```
dyn4 = HurGlobalELEquation[[4]]
HurToJulia[dyn4]
```

```
(-1*tau4+(tau5+(g*lta*ms*sin(q4)+(g*ltb*ms*sin(q4)+(g*ltb*mt*sin(q4)+((lsa+lsb)*(lta*
  ms+ltb*(ms+mt))*sin((q1+-1*q4))*(q1d)^(2)+((lta+ltb)*(lta*ms+ltb*(ms+mt))*sin((q2+-1
  *q4))*(q2d)^(2)+(lsb*lta*ms*sin((q4+-1*q5))*(q5d)^(2)+(lsb*ltb*ms*sin((q4+-1*q5))*(
  q5d)^(2)+(-1*lsa*lta*ms*cos((q1+-1*q4))*q1dd+(-1*lsb*lta*ms*cos((q1+-1*q4))*q1dd+(-1
  *lsa*ltb*ms*cos((q1+-1*q4))*q1dd+(-1*lsb*ltb*ms*cos((q1+-1*q4))*q1dd+(-1*lsa*ltb*mt*
  cos((q1+-1*q4))*q1dd+(-1*lsb*ltb*mt*cos((q1+-1*q4))*q1dd+(-1*(lta)^(2)*ms*cos((q2+-1
  *q4))*q2dd+(-2*lta*ltb*ms*cos((q2+-1*q4))*q2dd+(-1*(ltb)^(2)*ms*cos((q2+-1*q4))*q2dd
  +(-1*lta*ltb*mt*cos((q2+-1*q4))*q2dd+(-1*(ltb)^(2)*mt*cos((q2+-1*q4))*q2dd+(It*q4dd+
  ((lta)^(2)*ms*q4dd+(2*lta*ltb*ms*q4dd+((ltb)^(2)*ms*q4dd+((ltb)^(2)*mt*q4dd+(lsb*lta
  *ms*cos((q4+-1*q5))*q5dd+lsb*ltb*ms*cos((q4+-1*q5))*q5dd)))))))))))))))))))))))))))
```

```
dyn5 = HurGlobalELEquation[[5]]
HurToJulia[dyn5]
```

$$-\text{tau5} + \text{g lsb ms Sin}[\text{q5}[\text{t}]] + \text{lsb}\left(\text{lsa} + \text{lsb}\right)\text{ ms Sin}[\text{q1}[\text{t}] - \text{q5}[\text{t}]]\,\text{q1}'[\text{t}]^2 + \text{lsb}\left(\text{lta} + \text{ltb}\right)\text{ ms Sin}[\text{q2}[\text{t}] - \text{q5}[\text{t}]]\,\text{q2}'[\text{t}]^2 - \text{lsb lta ms Sin}[\text{q4}[\text{t}] - \text{q5}[\text{t}]]\,\text{q4}'[\text{t}]^2 - \text{lsb ltb ms Sin}[\text{q4}[\text{t}] - \text{q5}[\text{t}]]\,\text{q4}'[\text{t}]^2 - \text{lsa lsb ms Cos}[\text{q1}[\text{t}] - \text{q5}[\text{t}]]\,\text{q1}''[\text{t}] - \text{lsb}^2\,\text{ms Cos}[\text{q1}[\text{t}] - \text{q5}[\text{t}]]\,\text{q1}''[\text{t}] - \text{lsb lta ms Cos}[\text{q2}[\text{t}] - \text{q5}[\text{t}]]\,\text{q2}''[\text{t}] - \text{lsb ltb ms Cos}[\text{q2}[\text{t}] - \text{q5}[\text{t}]]\,\text{q2}''[\text{t}] + \text{lsb lta ms Cos}[\text{q4}[\text{t}] - \text{q5}[\text{t}]]\,\text{q4}''[\text{t}] + \text{lsb ltb ms Cos}[\text{q4}[\text{t}] - \text{q5}[\text{t}]]\,\text{q4}''[\text{t}] + \text{Is q5}''[\text{t}] + \text{lsb}^2\,\text{ms q5}''[\text{t}]$$

```
(-1*tau5+(g*lsb*ms*sin(q5)+(lsb*(lsa+lsb)*ms*sin((q1+-1*q5))*(q1d)^(2)+(lsb*(lta+ltb)*
  ms*sin((q2+-1*q5))*(q2d)^(2)+(-1*lsb*lta*ms*sin((q4+-1*q5))*(q4d)^(2)+(-1*lsb*ltb*ms
  *sin((q4+-1*q5))*(q4d)^(2)+(-1*lsa*lsb*ms*cos((q1+-1*q5))*q1dd+(-1*(lsb)^(2)*ms*cos(
  (q1+-1*q5))*q1dd+(-1*lsb*lta*ms*cos((q2+-1*q5))*q2dd+(-1*lsb*ltb*ms*cos((q2+-1*q5))*
  q2dd+(lsb*lta*ms*cos((q4+-1*q5))*q4dd+(lsb*ltb*ms*cos((q4+-1*q5))*q4dd+(Is*q5dd+(lsb
  )^(2)*ms*q5dd)))))))))))))
```

```
HurUnifyTriadsCoord[FootSW, n] // MatrixForm
```

$$\begin{pmatrix} -\left(\text{lsa} + \text{lsb}\right)\text{ Sin}[\text{q1}[\text{t}]] - \left(\text{lta} + \text{ltb}\right)\text{ Sin}[\text{q2}[\text{t}]] + \left(\text{lta} + \text{ltb}\right)\text{ Sin}[\text{q4}[\text{t}]] + \left(\text{lsa} + \text{lsb}\right)\text{ Sin}[\text{q5}[\text{t}] \\ \left(\text{lsa} + \text{lsb}\right)\text{ Cos}[\text{q1}[\text{t}]] + \left(\text{lta} + \text{ltb}\right)\text{ Cos}[\text{q2}[\text{t}]] - \left(\text{lta} + \text{ltb}\right)\text{ Cos}[\text{q4}[\text{t}]] - \left(\text{lsa} + \text{lsb}\right)\text{ Cos}[\text{q5}[\text{t}]] \\ 0 \\ \text{n} \end{pmatrix}$$

```
stepLength = HurUnifyTriadsCoord[FootSW, n][[1]]
HurToJulia[stepLength]
```

$$-\left(\text{lsa} + \text{lsb}\right)\text{ Sin}[\text{q1}[\text{t}]] - \left(\text{lta} + \text{ltb}\right)\text{ Sin}[\text{q2}[\text{t}]] + \left(\text{lta} + \text{ltb}\right)\text{ Sin}[\text{q4}[\text{t}]] + \left(\text{lsa} + \text{lsb}\right)\text{ Sin}[\text{q5}[\text{t}]]$$

```
(-1*(lsa+lsb)*sin(q1)+(-1*(lta+ltb)*sin(q2)+((lta+ltb)*sin(q4)+(lsa+lsb)*sin(q5))))
```

```
stepHeight = HurUnifyTriadsCoord[FootSW, n][[2]]
HurToJulia[stepHeight]
```

$$(lsa + lsb)\,\text{Cos}[q1[t]] + (lta + ltb)\,\text{Cos}[q2[t]] - (lta + ltb)\,\text{Cos}[q4[t]] - (lsa + lsb)\,\text{Cos}[q5[t]]$$

```
((lsa+lsb)*cos(q1)+((lta+ltb)*cos(q2)+(-1*(lta+ltb)*cos(q4)+-1*(lsa+lsb)*cos(q5))))
```

```
HurGlobalMMatrix
HurToJulia[%]
```

```
[(Is+(2*lsa*lsb*(mb+(ms+2*mt))+((lsb)^(2)*(mb+(ms+2*mt))+(lsa)^(2)*(mb+2*(ms+mt))))))
  (lsa+lsb)*(ltb*(mb+(ms+mt))+lta*(mb+(ms+2*mt)))*cos((q1+-1*q2))
  1/2*lb*(lsa+lsb)*mb*cos((q1+-1*q3))
  -1*(lsa+lsb)*(lta*ms+ltb*(ms+mt))*cos((q1+-1*q4))
  -1*lsb*(lsa+lsb)*ms*cos((q1+-1*q5));(lsa+lsb)*(ltb*(mb+(ms+mt))+lta*(mb+(ms+2*mt)))*
  cos((q1+-1*q2))
  (It+(2*lta*ltb*(mb+(ms+mt))+((ltb)^(2)*(mb+(ms+mt))+(lta)^(2)*(mb+(ms+2*mt))))))
  1/2*lb*(lta+ltb)*mb*cos((q2+-1*q3))
  -1*(lta+ltb)*(lta*ms+ltb*(ms+mt))*cos((q2+-1*q4))
  -1*lsb*(lta+ltb)*ms*cos((q2+-1*q5));1/2*lb*(lsa+lsb)*mb*cos((q1+-1*q3))
  1/2*lb*(lta+ltb)*mb*cos((q2+-1*q3)) (Ib+1/4*(lb)^(2)*mb)
  0 0;-1*(lsa+lsb)*(lta*ms+ltb*(ms+mt))*cos((q1+-1*q4))
  -1*(lta+ltb)*(lta*ms+ltb*(ms+mt))*cos((q2+-1*q4)) 0
  (It+((lta)^(2)*ms+(2*lta*ltb*ms+(ltb)^(2)*(ms+mt))))
  lsb*(lta+ltb)*ms*cos((q4+-1*q5));-1*lsb*(lsa+lsb)*ms*cos((q1+-1*q5))
  -1*lsb*(lta+ltb)*ms*cos((q2+-1*q5)) 0
  lsb*(lta+ltb)*ms*cos((q4+-1*q5)) (Is+(lsb)^(2)*ms)]
```

```
HurGlobalCMatrix
HurToJulia[%]
```

```
[0 (lsa+lsb)*(ltb*(mb+(ms+mt))+lta*(mb+(ms+2*mt)))*sin((q1+-1*q2))*q2d
  1/2*lb*(lsa+lsb)*mb*sin((q1+-1*q3))*q3d
  -1*(lsa+lsb)*(lta*ms+ltb*(ms+mt))*sin((q1+-1*q4))*q4d
  -1*lsb*(lsa+lsb)*ms*sin((q1+-1*q5))*q5d;-1*(lsa+lsb)*(ltb*(mb+(ms+mt))+lta*(mb+(ms+2
  *mt)))*sin((q1+-1*q2))*q1d 0 1/2*lb*(lta+ltb)*mb*sin((q2+-1*q3))*q3d
  -1*(lta+ltb)*(lta*ms+ltb*(ms+mt))*sin((q2+-1*q4))*q4d
  -1*lsb*(lta+ltb)*ms*sin((q2+-1*q5))*q5d;-1/2*lb*(lsa+lsb)*mb*sin((q1+-1*q3))*q1d
  -1/2*lb*(lta+ltb)*mb*sin((q2+-1*q3))*q2d 0 0
  0;(lsa+lsb)*(lta*ms+ltb*(ms+mt))*sin((q1+-1*q4))*q1d
  (lta+ltb)*(lta*ms+ltb*(ms+mt))*sin((q2+-1*q4))*q2d 0 0
  lsb*(lta+ltb)*ms*sin((q4+-1*q5))*q5d;lsb*(lsa+lsb)*ms*sin((q1+-1*q5))*q1d
  lsb*(lta+ltb)*ms*sin((q2+-1*q5))*q2d 0 -1*lsb*(lta+ltb)*ms*sin((q4+-1*q5))*q4d 0]
```

```
HurGlobalGVector
HurToJulia[HurList2Column[%]]
```

```
[-1*g*(lsb*(mb+(ms+2*mt))+lsa*(mb+2*(ms+mt)))*sin(q1);-1*g*(ltb*(mb+(ms+mt))+lta*(mb+(
  ms+2*mt)))*sin(q2);-1/2*g*lb*mb*sin(q3);g*(lta*ms+ltb*(ms+mt))*sin(q4);g*lsb*ms*sin(
  q5)]
```

```
JacFootSW = HurGetJacobian[FootSW, e, n];
JacFootSW.HurList2Column[{q1'[t], q2'[t], q3'[t], q4'[t], q5'[t]}]
verticalVel = %[[2, 1]]
HurToJulia[verticalVel]
```

```
(-1*(lsa+lsb)*sin(q1)*q1d+(-1*(lta+ltb)*sin(q2)*q2d+((lta+ltb)*sin(q4)*q4d+(lsa+lsb)*
  sin(q5)*q5d)))
```

```
JacFootSW.HurList2Column[{q1'[t], q2'[t], q3'[t], q4'[t], q5'[t]}]
horizontalVel = %[[1, 1]]
HurToJulia[horizontalVel]
```

```
(-1*(lsa+lsb)*cos(q1)*q1d+(-1*(lta+ltb)*cos(q2)*q2d+((lta+ltb)*cos(q4)*q4d+(lsa+lsb)*
  cos(q5)*q5d)))
```

```
HurSaveData["data1.m", "FootST", "ShankSTCOM", "KneeST", "ThighSTCOM", "Hip", "TorsoCOM",
 "ThighSWCOM", "KneeSW", "ShankSWCOM", "FootSW", "dyn1", "dyn2", "dyn3", "dyn4", "dyn5",
 "stepLength", "stepHeight", "verticalVel", "horizontalVel", "JacFootSW", "invans"]
```

```
HurUnifyTriadsCoord[FootSW, n];
HurToJulia[%[[1]]]
```

```
(-1*(lsa+lsb)*sin(q1)+(-1*(lta+ltb)*sin(q2)+((lta+ltb)*sin(q4)+(lsa+lsb)*sin(q5))))
```

```
HurUnifyTriadsCoord[FootSW, n];
HurToJulia[%[[2]]]
```

```
((lsa+lsb)*cos(q1)+((lta+ltb)*cos(q2)+(-1*(lta+ltb)*cos(q4)+-1*(lsa+lsb)*cos(q5))))
```

```
HurUnifyTriadsCoord[KneeSW, n];
HurToJulia[%[[1]]]
```

```
(-1*(lsa+lsb)*sin(q1)+-1*(lta+ltb)*(sin(q2)+-1*sin(q4)))
```

```
HurUnifyTriadsCoord[KneeSW, n];
HurToJulia[%[[2]]]
```

```
((lsa+lsb)*cos(q1)+(lta+ltb)*(cos(q2)+-1*cos(q4)))
```

```
HurUnifyTriadsCoord[Hip, n];
HurToJulia[%[[1]]]
```

```
(-1*(lsa+lsb)*sin(q1)+-1*(lta+ltb)*sin(q2))
```

```
HurUnifyTriadsCoord[Hip, n];
HurToJulia[%[[2]]]
```

```
((lsa+lsb)*cos(q1)+(lta+ltb)*cos(q2))
```

```
HurUnifyTriadsCoord[TorsoCOM, n];
HurToJulia[%[[1]]]
```

```
(-1*(lsa+lsb)*sin(q1)+(-1*(lta+ltb)*sin(q2)+-1/2*lb*sin(q3)))
```

```
HurUnifyTriadsCoord[TorsoCOM, n];
HurToJulia[%[[2]]]
```

```
((lsa+lsb)*cos(q1)+((lta+ltb)*cos(q2)+1/2*lb*cos(q3)))
```

```
HurUnifyTriadsCoord[KneeST, n];
HurToJulia[%[[1]]]
```

```
-1*(lsa+lsb)*sin(q1)
```

```
HurUnifyTriadsCoord[KneeST, n];
HurToJulia[%[[2]]]
```

```
(lsa+lsb)*cos(q1)
```

```
Transpose[HurGetJacobian[Hip, c, n]].
  HurList2Column[{forcex, forcey, 0, 0, 0, 0}] // MatrixForm
```

```
HurToJulia[%[[2, 1]] ]
```

```
HurGetJacobian[Hip, c, n] // MatrixForm
```

```
HurTurnOffSimplify[]
```

```
False
```

```
invans = HurELInverse[]
```

$$\{\{q1''[t] \to -\left(\left(\left(-\left(\left(Is + lsb^2\, ms\right)\left(\cdots 1 \cdots\right) - \left(\cdots 1 \cdots\right)\left(\cdots 1 \cdots + \cdots 1 \cdots\right)\right)\left(\cdots 1 \cdots\right) + \cdots 1 \cdots\right)\left(\cdots 1 \cdots\right) - \cdots 1 \cdots\right)\Big/ \left(-\left(-\left(\left(Ib + \frac{lb^2\, mb}{4}\right)\left(Is + lsb^2\, ms\right)\left(\cdots 1 \cdots\right) - \cdots 1 \cdots\right)\left(\cdots 1 \cdots\right) + \left(\cdots 1 \cdots\right)\cdots 1 \cdots\right)\left(\cdots 1 \cdots\right) + \cdots 1 \cdots\right)\right), \cdots 3 \cdots, q5''[t] \to \cdots 1 \cdots\}\}$$

large output | show less | show more | show all | set size limit...

```
q5''[t] /. invans[[1]];
HurToJulia[%]
```

Whole body COM

```
COMWhole =
 Total[Table[HurGlobalCOMPos[[ i ]] * HurGlobalMass[[ i ]], {i, 2, Length[HurGlobalRF]}]] /
  Total[HurGlobalMass]
```

Whole body COM linear momentum

```
LinearMomentumWholeBody =
  D[HurUnifyTriadsCoord[COMWhole, n], t] * Total[HurGlobalMass] // Simplify;
LinearMomentumWholeBody // MatrixForm
```

Whole body COM linear momentum rate change

```
LinearMomentumRateWholeBody =
  D[LinearMomentumWholeBody, t] * Total[HurGlobalMass] // Simplify;
LinearMomentumRateWholeBody // MatrixForm
```

Whole body Angular Momentum

```
Table[HurCross[HurGlobalCOMPos[[ i ]] - COMWhole, HurGlobalLinearMomentum[[i]], n ] +
  HurGlobalAngularMomentum[[i]], {i, 2, Length[HurGlobalRF]}]
HurUnifyTriadsCoord[Total[%], n] // Simplify
```

$$\left\{a3\ Is\ q1'[t] + n3\left(\cdots 102 \cdots + \frac{\cdots 1 \cdots}{\cdots 1 \cdots} + \frac{2\ lsa\ \cdots 4 \cdots\ q1'[t]}{\left(\cdots 1 \cdots\right)^2} + \frac{2\ lsa\ lsb\ ms^2\ mt\ Sin[q1[t]]\ Sin[q5[t]]\ q1'[t]}{(mb+2\ ms+2\ mt)^2}\right),\right.$$

$$\left.\cdots 3 \cdots, e3\ Is\ q5'[t] + n3\left(\frac{lsa\ lsb\ \cdots 4 \cdots\ Cos[q5[t]]\ q1'[t]}{\left(\cdots 1 \cdots\right)^2} + \cdots 1040 \cdots + \frac{\cdots 1 \cdots}{\cdots 1 \cdots^2}\right)\right\}$$

large output | show less | show more | show all | set size limit...

```
JacWhole = HurGetJacobian[COMWhole, n, n][[1 ;; 2, ;;]] // Simplify;
JacWhole // MatrixForm
```

```
HurToJulia[JacWhole]
```

```
[-1*((mb+2*(ms+mt)))^(-1)*(lsb*(mb+(ms+2*mt))+lsa*(mb+2*(ms+mt)))*cos(q1)
  -1*((mb+2*(ms+mt)))^(-1)*(ltb*(mb+(ms+mt))+lta*(mb+(ms+2*mt)))*cos(q2)
  -1/2*lb*mb*((mb+2*(ms+mt)))^(-1)*cos(q3)
  ((mb+2*(ms+mt)))^(-1)*(lta*ms+ltb*(ms+mt))*cos(q4)
  lsb*ms*((mb+2*(ms+mt)))^(-1)*cos(q5);-1*((mb+2*(ms+mt)))^(-1)*(lsb*(mb+(ms+2*mt))+
  lsa*(mb+2*(ms+mt)))*sin(q1)
  -1*((mb+2*(ms+mt)))^(-1)*(ltb*(mb+(ms+mt))+lta*(mb+(ms+2*mt)))*sin(q2)
  -1/2*lb*mb*((mb+2*(ms+mt)))^(-1)*sin(q3)
  ((mb+2*(ms+mt)))^(-1)*(lta*ms+ltb*(ms+mt))*sin(q4)
  lsb*ms*((mb+2*(ms+mt)))^(-1)*sin(q5)]
```

```
NJacWhole = NullSpace[JacWhole] // Simplify;
NJacWhole // MatrixForm (*I should expect 3 5D column vectors. It'
  s weird that Mathematica gives 3x5 matrix, not 5x3 matrix. Anyway,
 a vector with 5 element is the basis vector for the null space.*)
```

```
JacWhole.HurList2Column[NJacWhole[[1, ;;]]] // Simplify
```

```
{{0}, {0}}
```

```
Nmat = Transpose[NJacWhole];
Nmat // MatrixForm
```

$$\begin{pmatrix} -\frac{lsb\ ms\ Csc[q1[t]-q2[t]]\ Sin[q2[t]-q5[t]]}{lsb\ (mb+ms+2\ mt)+lsa\ (mb+2\ (ms+mt))} & -\frac{(lta\ ms+ltb\ (ms+mt))\ Csc[q1[t]-q2[t]]\ Sin[q2[t]-q4[t]]}{lsb\ (mb+ms+2\ mt)+lsa\ (mb+2\ (ms+mt))} & \frac{lb\ mb\ Csc[q1[t]-q2[t]]}{2\ (lsb\ (mb+ms+2\ mt)+lsa} \\ \frac{lsb\ ms\ Csc[q1[t]-q2[t]]\ Sin[q1[t]-q5[t]]}{ltb\ (mb+ms+mt)+lta\ (mb+ms+2\ mt)} & \frac{(lta\ ms+ltb\ (ms+mt))\ Csc[q1[t]-q2[t]]\ Sin[q1[t]-q4[t]]}{ltb\ (mb+ms+mt)+lta\ (mb+ms+2\ mt)} & -\frac{lb\ mb\ Csc[q1[t]-q2[t]]}{2\ (ltb\ (mb+ms+mt)+l} \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

```
HurToJulia[Nmat]
```

```
[-1*lsb*ms*((lsb*(mb+(ms+2*mt))+lsa*(mb+2*(ms+mt)))^(-1)*csc((q1+-1*q2))*sin((q2+-1*
  q5))
  -1*(lta*ms+ltb*(ms+mt))*((lsb*(mb+(ms+2*mt))+lsa*(mb+2*(ms+mt)))^(-1)*csc((q1+-1*
  q2))*sin((q2+-1*q4))
  1/2*lb*mb*((lsb*(mb+(ms+2*mt))+lsa*(mb+2*(ms+mt)))^(-1)*csc((q1+-1*q2))*sin((q2+-1
  *q3));lsb*ms*((ltb*(mb+(ms+mt))+lta*(mb+(ms+2*mt)))^(-1)*csc((q1+-1*q2))*sin((q1+-
  1*q5))
  (lta*ms+ltb*(ms+mt))*((ltb*(mb+(ms+mt))+lta*(mb+(ms+2*mt)))^(-1)*csc((q1+-1*q2))*
  sin((q1+-1*q4))
  -1/2*lb*mb*((ltb*(mb+(ms+mt))+lta*(mb+(ms+2*mt)))^(-1)*csc((q1+-1*q2))*sin((q1+-1*
  q3));0 0 1;0 1 0;1 0 0]
```

```
Projector = Nmat.Inverse[Transpose[Nmat].Nmat].Transpose[Nmat];
```

```
Projector[[1, 1]] // Simplify
```

```
$Aborted
```