**State Space Model For Spring-Mass-Damper**

**k**

**m**

**x**

**c**





use states  to compute derivative of states 





**assumed solution** 



  is an eigenvalue, NOT  from EOM



**for c = 0**, 

**critical damping**  

**underdamped** 

**State Space Model For Double Spring-Mass-Damper**

**m1**

**c2**

**c1**

**x1**

**x2**

**k2**

**k3**

**k1**

**m2**

**c3**







use states  to compute derivative of states 





**assumed solution** 

**for c1 = c2 = c3 = 0,**

a = (m1k2+ m1k3+ m2k1+ m2k2) / (m1m2), b = (k1k2+ k1k3+ k2k3) / (m1m2)



**State Space Model for Cylindrical Coordinate Manipulator**

**T**

**X**

**Y**

****

**X**

**Y**

****

**X**

**Y**

****

**r**

**r3**

**a**

**Link 2**

**Link 3**

**Both links**

**F**

Main body link 2 - Shaft and end-effector link 3

Mass centers at a and r3 from waist rotation axis, a=constant, r3 = variable

Masses m2 and m3 - centroidal mass moments of inertia J2 and J3

 CCW from positive x axis – a and r3 radial from rotation axis

T is rotary actuator torque of ground on body 2 about waist measured CCW positive

F is radial actuator force of body 2 on body 3 measured positive outward

Gravity g acts along negative y axis





use states  to compute derivative of states 



**State Space Model for Two Link Anthropomorphic Manipulator (Double Pendulum)**

3

2

a3

d3

d2

a2

Y

A

B

X

C

**T2**

**T3**

m2, J2

m3, J3

Two solid rigid bars with revolute joints A and B

Lengths d2 and d3 - mass centers at a2 and a3 from proximal ends

Masses m2 and m3 - centroidal mass moments of inertia J2 and J3

2 CCW from positive x axis T2 is torque of ground on bar 2 about pin A, CCW positive

3 CCW from centerline of bar 2 T3 is torque of bar 2 on bar 3 about pin B, CCW positive

Gravity g acts along negative y axis

















use states  to compute derivative of states 



**Numerically Evaluate Linear State Matrix**

assume  using ns number of states and n number of time samples





Note that samples need not be computed using fixed h, and do not need to be continuous.

for n = ns



for n ≥ ns



extract  from    prefer 