IDC OS Functions

function Hu = fcn(u, phi)

```
a1 = 0.15;  % link 1 length
a2 = 0.15;  % link 2 length
m1 = 0.092;  % link 1 mass
m2 = 0.077; % link 2 mas
r01 = 0.062; % link 1 center of mass
r12 = 0.036; % link 2 COM
I1 = 0.64e-3;  % link 1 inertia
I2 = 0.30e-3;  % link 2 inertia
Jm1 = 0.65e-6; % motor inertias
Jm2 = 0.65e-6;
b1 = 3.1e-6; % viscous damping constants
b2 = 3.1e-6;
c1 = 0.0001; % coulomb friction constants
c2 = 0.0001;
g = 9.8; % gravitational constant
N1 = 70; % gear ratios
N2 = 70;

H11 = N1^2*Jm1 + I1 + m2*a1^2;
H12 = a1*r12*m2*cos(phi(2)-phi(1));
H21 = H12;
H22 = N2^2*Jm2 + I2;

H = [H11 H12; H21 H22]; % inertia matrix

Hu = H*u;
```

function Nhat = fcn(phi, phidot) Feedback Comp

```
a1 = 0.15;  % link 1 length
a2 = 0.15;  % link 2 length
m1 = 0.092;  % link 1 mass
m2 = 0.077; % link 2 mas
r01 = 0.062; % link 1 center of mass
r12 = 0.036; % link 2 COM
I1 = 0.64e-3;  % link 1 inertia
I2 = 0.30e-3;  % link 2 inertia
Jm1 = 0.65e-6; % motor inertias
Jm2 = 0.65e-6;
b1 = 3.1e-6; % viscous damping constants
b2 = 3.1e-6;
c1 = 0.0001; % coulomb friction constants
c2 = 0.0001;
g = 9.8; % gravitational constant
N1 = 70; % gear ratios
N2 = 70;

h = a1*r12*m2*sin(phi(2)-phi(1));
```

```
G1 = (r01*m1+a1*m2)*g*cos(phi(1));
G2 = r12*m2*g*cos(phi(2));
F1 = N1^2*b1*phidot(1) + N1*c1*sign(phidot(1));
F2 = N2^2*b2*phidot(2) + N2*c2*sign(phidot(2));

V = [0 -h ;h 0]*[phidot(1)^2;phidot(2)^2]; % centr torques
G = [G1;G2]; % gravity torques
F = [F1;F2]; % frictional torques

Nhat = V + G + F;
```

Published with MATLAB® R2022b

Jacobian Transpose and Inverse Function Write-out

Table of Contents

Jacobian Transpose

function tau = fcn(phi, F)

```
a1=0.15;
a2=0.15;

J =[-a1*sin(phi(1)) - a2*sin(phi(2)), -a2*sin(phi(2)) ; a1*cos(phi(1)) +
 a2*cos(phi(2)), a2*cos(phi(2))] *[1, 0 ; -1, 1];

Jtrans = J';

tau = Jtrans * F;
```

Jacobian Inverse

function phi_err = fcn(phi, x_err)

```
a1=0.15;
a2=0.15;

J =[-a1*sin(phi(1)) - a2*sin(phi(2)), -a2*sin(phi(2)) ; a1*cos(phi(1)) +
 a2*cos(phi(2)), a2*cos(phi(2))] *[1, 0 ; -1, 1];

Jinv = inv(J);

phi_err = Jinv * x_err;
```

Published with MATLAB® R2022b

Table of Contents


```
% IDC_OS_RC Function List
```

Hx Matrix

function Hxu = fcn(u, phi) Inertia Matrix.

```
a1 = 0.15;  % link 1 length
a2 = 0.15;  % link 2 length
m1 = 0.092;  % link 1 mass
m2 = 0.077; % link 2 mas
r01 = 0.062; % link 1 center of mass
r12 = 0.036; % link 2 COM
I1 = 0.64e-3;  % link 1 inertia
I2 = 0.30e-3;  % link 2 inertia
Jm1 = 0.65e-6; % motor inertias
Jm2 = 0.65e-6;
b1 = 3.1e-6; % viscous damping constants
b2 = 3.1e-6;
c1 = 0.0001; % coulomb friction constants
c2 = 0.0001;
g = 9.8; % gravitational constant
N1 = 70; % gear ratios
N2 = 70;

H11 = N1^2*Jm1 + I1 + m2*a1^2;
H12 = a1*r12*m2*cos(phi(2)-phi(1));
H21 = H12;
H22 = N2^2*Jm2 + I2;

H = [H11 H12; H21 H22]; % inertia matrix


J =[-a1*sin(phi(1)) - a2*sin(phi(2)), -a2*sin(phi(2)) ; a1*cos(phi(1)) +
 a2*cos(phi(2)), a2*cos(phi(2))] *[1, 0 ; -1, 1];

Jtrans = J';
Jinv = inv(J);



Hxu = inv(Jtrans)*H*Jinv*u;
```

Jacobian Transpose

function tau = fcn(F, phi)

```
%Inverse jacobian
a1=0.15;
a2=0.15;

J =[-a1*sin(phi(1)) - a2*sin(phi(2)), -a2*sin(phi(2)) ; a1*cos(phi(1)) +
 a2*cos(phi(2)), a2*cos(phi(2))] *[1, 0 ; -1, 1];

Jtrans = J';

tau = Jtrans * F;
```

Jacobian Dot

function y = fcn(phi, phidot)

```
%Inverse jacobian
a1=0.15;
a2=0.15;

Jdot =[-a1*cos(phi(1)), -a2*cos(phi(2)) ; -a1*sin(phi(1)), -a2*sin(phi(2))];

%take derrivate of J

y = Jdot * phidot;
```

Feedback Comp

function N = fcn(phi, phidot) Feedback Comp

```
a1 = 0.15;  % link 1 length
a2 = 0.15;  % link 2 length
m1 = 0.092;  % link 1 mass
m2 = 0.077; % link 2 mas
r01 = 0.062; % link 1 center of mass
r12 = 0.036; % link 2 COM
I1 = 0.64e-3;  % link 1 inertia
I2 = 0.30e-3;  % link 2 inertia
Jm1 = 0.65e-6; % motor inertias
Jm2 = 0.65e-6;
b1 = 3.1e-6; % viscous damping constants
b2 = 3.1e-6;
c1 = 0.0001; % coulomb friction constants
c2 = 0.0001;
g = 9.8; % gravitational constant
N1 = 70; % gear ratios
N2 = 70;

h = a1*r12*m2*sin(phi(2)-phi(1));
```

```
G1 = (r01*m1+a1*m2)*g*cos(phi(1));
G2 = r12*m2*g*cos(phi(2));
F1 = N1^2*b1*phidot(1) + N1*c1*sign(phidot(1));
F2 = N2^2*b2*phidot(2) + N2*c2*sign(phidot(2));

V = [0 -h ;h 0]*[phidot(1)^2;phidot(2)^2]; % centr torques
G = [G1;G2]; % gravity torques
F = [F1;F2]; % frictional torques

N = V + G + F;
```

Published with MATLAB® R2022b