Homework 6, Problem 1

Exercise 6.10 (b)

David Lim

A16398479

02/25/25

```
clear
addpath('/Users/davidlim/Documents/ModernRobotics/packages/MATLAB/
mr');
```

Desired end-effector frame:

```
Rsd = [1/sqrt(2) -1/sqrt(2) 0; 1/sqrt(2) 1/sqrt(2) 0; 0 0 1]
```

```
Rsd = 3×3
    0.7071   -0.7071         0
    0.7071    0.7071         0
         0         0    1.0000
```

Rotation (screw) axes of zero position in space/body frame:

```
omega = [[0 0 1]' [0 1 0]' [0 0 1]']
```

```
omega = 3×3
     0     0     0
     0     1     0
     1     0     1
```

Initial guess for joint angles:

```
theta = [0 pi/6 0]'
```

```
theta = 3×1
         0
    0.5236
         0
```

Forward kinematics for initial guess:

```
% Rsb =
MatrixExp3(VecToso3(omega(:,1)*theta(1)))*MatrixExp3(VecToso3(omeg
a(:,2)*theta(2)))*MatrixExp3(VecToso3(omega(:,3)*theta(3)))
```

```
Rsb = [cos(theta(1))*cos(theta(2))*cos(theta(3))-
sin(theta(1))*sin(theta(3))
-cos(theta(1))*cos(theta(2))*sin(theta(3))-
sin(theta(1))*cos(theta(3)) cos(theta(1))*sin(theta(2)); ...
       sin(theta(1))*cos(theta(2))*cos(theta(3))
+cos(theta(1))*sin(theta(3))
-sin(theta(1))*cos(theta(2))*sin(theta(3))
+cos(theta(1))*cos(theta(3)) sin(theta(1))*sin(theta(2)); ...
       -sin(theta(2))*cos(theta(3)) sin(theta(2))*sin(theta(3))
cos(theta(2))]
```

```
Rsb = 3×3
    0.8660         0    0.5000
         0    1.0000         0
   -0.5000         0    0.8660
```

Error rotation (transforation) matrix:

```
Rbd = Rsb'*Rsd
```

```
Rbd = 3×3
    0.6124   -0.6124   -0.5000
    0.7071    0.7071         0
    0.3536   -0.3536    0.8660
```

Matrix log to obtain error rotation (twist) in so(3):

```
omegab_so3 = MatrixLog3(Rbd)
```

```
omegab_so3 = 3×3
         0   -0.7670   -0.4962
    0.7670         0    0.2055
    0.4962   -0.2055         0
```

Extract error rotation vector:

```
omegab = so3ToVec(omegab_so3)
```

```
omegab = 3×1
   -0.2055
   -0.4962
    0.7670
```

Compute body Jacobian for initial guess:

```
% JacobianBody([omega; zeros(3)], [0 pi/6 0]')
Jb = [-sin(theta(2))*cos(theta(3)) sin(theta(3)) 0; ...
      sin(theta(2))*sin(theta(3)) cos(theta(3)) 0; ...
      cos(theta(2)) 0 1]
```

```
Jb = 3×3
   -0.5000         0         0
         0    1.0000         0
    0.8660         0    1.0000
```

Update joint angles:

```
theta = theta + pinv(Jb)*omegab
```

```
theta = 3×1
    0.4110
    0.0274
    0.4110
```

Forward kinematics for new guess:

```
Rsb = [cos(theta(1))*cos(theta(2))*cos(theta(3))-
sin(theta(1))*sin(theta(3))
-cos(theta(1))*cos(theta(2))*sin(theta(3))-
sin(theta(1))*cos(theta(3)) cos(theta(1))*sin(theta(2)); ...
       sin(theta(1))*cos(theta(2))*cos(theta(3))
+cos(theta(1))*sin(theta(3))
-sin(theta(1))*cos(theta(2))*sin(theta(3))
+cos(theta(1))*cos(theta(3)) sin(theta(1))*sin(theta(2)); ...
       -sin(theta(2))*cos(theta(3)) sin(theta(2))*sin(theta(3))
cos(theta(2))]
```

```
Rsb = 3×3
    0.6804   -0.7324    0.0251
    0.7324    0.6808    0.0110
   -0.0251    0.0110    0.9996
```

New error rotation (transforation) matrix:

```
Rbd = Rsb'*Rsd
```

```
Rbd = 3×3
    0.9990    0.0368   -0.0251
   -0.0365    0.9993    0.0110
    0.0255   -0.0100    0.9996
```

Matrix log to obtain new error rotation (twist) in so(3):

```
omegab_so3 = MatrixLog3(Rbd)
```

```
omegab_so3 = 3×3
         0    0.0367   -0.0253
```

```
   -0.0367         0    0.0105
    0.0253   -0.0105         0
```

Extract new error rotation vector:

```
omegab = so3ToVec(omegab_so3)
```

```
omegab = 3×1
   -0.0105
   -0.0253
   -0.0367
```