Table of Contents

Clean 1
Problem 4 1
Problem 5 1
Problem 6 1
Problem 7 1
Problem 8 2
Functions 2

Clean

```
close all; clear; clc;
```

Problem 4

```
q = [0.1;0.1;0.2;0.06];
xdot = [5;10];
J = jacob(q);

% Least norm solution
jInt = inv(J*J');
qdotLN = J'*inv(J*J')*xdot;

% Add from null space
nullJ = null(J);
q1 = qdotLN + nullJ(:,1);
q2 = qdotLN + nullJ(:,2);
```

Problem 5

```
q = [-0.03; 0; 0.2; 0.05];
J = jacob(q);
xdot = [0; 8];

qdotLN = J'*inv(J*J')*xdot;
```

Problem 6

```
J = [J(:,1:2) J(:,4)];
xdot = [0; 8];
qdotLN = J'*inv(J*J')*xdot;
qTest = [qdotLN(1:2);0;qdotLN(3)];
J = jacob(q);
```

Problem 7

```
q = [0.07;0.024;-0.15;0.06;0;0.04;-0.02];
xDot = [4;3];
```

```
J = jacob(q);

A1 = [J(:,1) xDot];
A2 = [J(:,2) xDot];
A3 = [J(:,3) xDot];
A4 = [J(:,4) xDot];
A5 = [J(:,5) xDot];
A6 = [J(:,6) xDot];
A7 = [J(:,7) xDot];

J = J(:,1:2);

qDot = inv(J)*xDot;
```

Problem 8

```
q = [0.04;-0.15;0.06;0.09;-0.04;0.01];
J = jacob(q);
xDot = [-7;1];

% Determine which combinations of motors has a solution for xDot
for i = 1:length(J(1,:))-1
    for j = i+1:length(J(1,:))
        A = [J(:,i) J(:,j) xDot];
        if det(A'*A) <= 1e-6
            fprintf("%d,%d\n", i, j);
        end
    end
end

% Calculate the optimal solution that reduces the total power
qDotBest = zeros(length(q),1);
for i = 1:length(J(1,:))-1
    for j = i+1:length(J(1,:))
        A = [J(:,i) J(:,j)];
        qDot = A \ xDot;
        if norm(qDot) < norm(qDotBest) || (i==1 && j==2)
            qDotBest = zeros(length(q),1);
            qDotBest(i) = qDot(1);
            qDotBest(j) = qDot(2);
        end
    end
end
```

Functions

```
function J = jacob(q)

    n = length(q);
    J = zeros(2, n);

    % X pos
    for i = 1:n
```

```
        totSum = 0;
        for j = i:n
            thetaSum = 0;
            for k = 1:j
                thetaSum = thetaSum + q(k);
            end
            totSum = totSum - sin(thetaSum);
        end
        J(1,i) = totSum;
    end

    % Y pos
    for i = 1:n
        totSum = 0;
        for j = i:n
            thetaSum = 0;
            for k = 1:j
                thetaSum = thetaSum + q(k);
            end
            totSum = totSum + cos(thetaSum);
        end
        J(2,i) = totSum;
    end

end
```

Published with MATLAB® R2023b