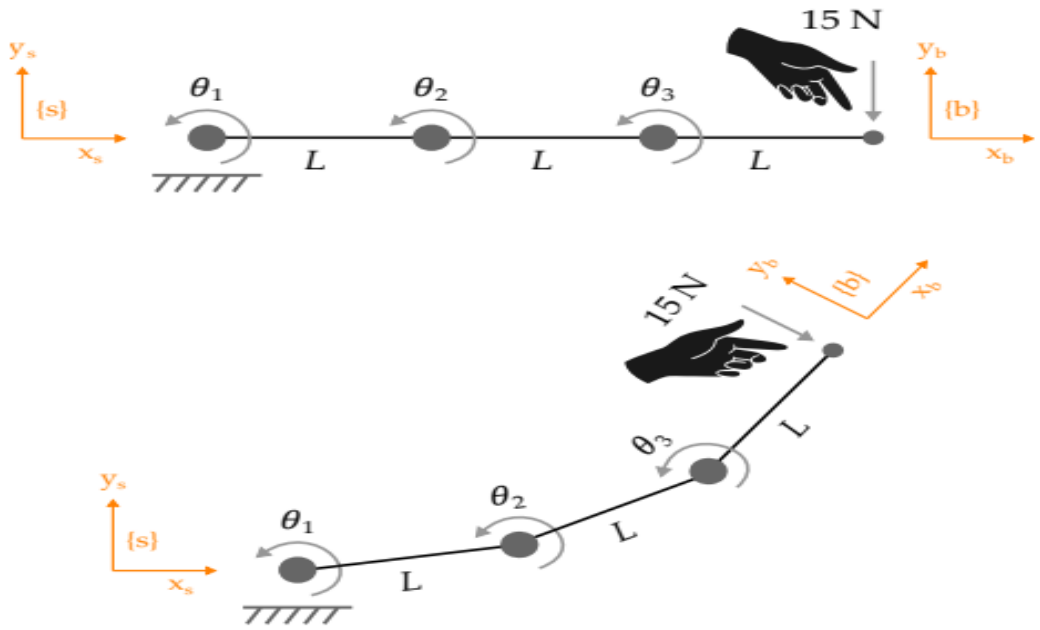

A human is pushing on the end-effector of the 3-DoF robot shown above. (Top Drawing) The human applies a 15 N force along the negative y_b axis. (Bottom Drawing) The human keeps this force aligned with the negative y_b axis as the robot moves.

What wrench does the robot need to apply at the end-effector to maintain static equilibrium?

Solution: Opposite force of $+15N$ over positive y-axis.

Thus the wrench required is provided below with the formulae: $F_b = \begin{bmatrix} m_b \\ f_b \end{bmatrix}$ and moment m_b is zero as the force is directly applied over the body frame.

```
clc
clear

syms theta1 theta2 theta3 real


% Opposite force of +15 N over positive y-axis
% Thus the wrench required is provided below with the formulae: Fb = [mb;fb] and
moment is zero as the force is directly applied over the body frame

Fb = [0;0;0;0;15;0];
```

Case 1: Let = 1 and let = [0, /4, /4] ' . Find the joint torques needed to balance out the force applied by the human.

Case 2: Let = 1 and let = [0, /8, 0] ' . Find the joint torques needed to balance out the force applied by the human

```
theta = [theta1;theta2;theta3];


omega = [0;0;1];

q1 = [0;0;0];
q2 = [1;0;0];
q3 = [2;0;0];

S1 = [omega; -cross(omega, q1)];
S2 = [omega; -cross(omega, q2)];
S3 = [omega; -cross(omega, q3)];

S_eq = [S1, S2, S3];
M = [eye(3), [3;0;0]; 0 0 0 1];

% T with initial joint positions
T_0 = simplify(expand(fk(M, S_eq, theta)))
```

T_0 =

$$\begin{pmatrix} \cos(\theta_1+\theta_2+\theta_3) & -\sin(\theta_1+\theta_2+\theta_3) & 0 & \cos(\theta_1+\theta_2+\theta_3)+\cos(\theta_1+\theta_2)+\cos(\theta_1) \\ \sin(\theta_1+\theta_2+\theta_3) & \cos(\theta_1+\theta_2+\theta_3) & 0 & \sin(\theta_1+\theta_2+\theta_3)+\sin(\theta_1+\theta_2)+\sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
R_0 = T_0(1:3, 1:3);
JS = simplify(expand(JacS(S_eq, theta))) %Space Jacobian
```

JS =

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & \sin(\theta_1) & \sin(\theta_1+\theta_2)+\sin(\theta_1) \\ 0 & -\cos(\theta_1) & -\cos(\theta_1+\theta_2)-\cos(\theta_1) \\ 0 & 0 & 0 \end{pmatrix}$$

```
Jb = simplify(expand(adjointM(inv(T_0))*JS)) %Body Jacobian
```

Jb =

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ \sin(\theta_2+\theta_3)+\sin(\theta_3) & \sin(\theta_3) & 0 \\ \cos(\theta_2+\theta_3)+\cos(\theta_3)+1 & \cos(\theta_3)+1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

```
J_geometric = simplify(expand([R_0, zeros(3); zeros(3), R_0] * Jb)) %Geometric
Jacobian
```

J_geometric =

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ -\sigma_1 - \sin(\theta_1+\theta_2) - \sin(\theta_1) & -\sigma_1 - \sin(\theta_1+\theta_2) & -\sigma_1 \\ \sigma_2 + \cos(\theta_1+\theta_2) + \cos(\theta_1) & \sigma_2 + \cos(\theta_1+\theta_2) & \sigma_2 \\ 0 & 0 & 0 \end{pmatrix}$$

where

$$\sigma_1 = \sin(\theta_1+\theta_2+\theta_3)$$

$$\sigma_2 = \cos(\theta_1+\theta_2+\theta_3)$$

```
% Fs calculation:
Fs = simplify(expand(adjointM(inv(T_0)))'*Fb)
```

Fs =

$$\begin{pmatrix} 0 \\ 0 \\ 15\cos(\theta_2+\theta_3) + 15\cos(\theta_3) + 15 \\ -15\sin(\theta_1+\theta_2+\theta_3) \\ 15\cos(\theta_1+\theta_2+\theta_3) \\ 0 \end{pmatrix}$$

```
% symbolic tau calculation
tau = simplify(expand(JS'*Fs))
```

tau =

$$\begin{pmatrix} 15\cos(\theta_2+\theta_3) + 15\cos(\theta_3) + 15 \\ 15\cos(\theta_3) + 15 \\ 15 \end{pmatrix}$$

```
% CASE 1
Case1_tau = double(subs(tau,[theta1,theta2,theta3],[0,pi/4,pi/4]))
```

```
Case1_tau = 3×1
   25.6066
   25.6066
   15.0000
```

```
% CASE 2
Case2_tau = double(subs(tau,[theta1,theta2,theta3],[0,pi/8,0]))
```

```
Case2_tau = 3×1
   43.8582
   30.0000
   15.0000
```

Let ||𝜏|| be the magnitude (i.e., the length) of the joint torque vector.

• Find a joint position 𝜃 that maximizes ||𝜏||

• Find a joint position 𝜃 that minimizes ||𝜏||

```
% ||tau||
magnitude_tau = simplify(expand(norm(tau)))
```

```
magnitude_tau =
```

$$\frac{15\sqrt{2}\,\sqrt{\cos(2\theta_3)+4\cos(\theta_3)+2\,(\cos(\theta_2+\theta_3)+\cos(\theta_3)+1)^2+5}}{2}$$

```
% Maximum ||tau|| with theta1 = theta2 = theta3 = 0
% or theta1 = 90 degrees, theta2 = 0, theta3 = 0
Max__mag_tau = double(subs(magnitude_tau,[theta1,theta2,theta3],[0,0,0]))
```

```
Max__mag_tau = 56.1249
```

```
% f = (15*sqrt(sym(2))*sqrt(cos(2*theta3) + 4*cos(theta3) + 2*(cos(theta2 + theta3)
+ cos(theta3) + 1)^2 + 5))/2

% Minimum ||tau|| with theta1 = 90 degrees, theta2 = 90 degrees, theta3 = 180
degrees
Min_mag_tau = double(subs(magnitude_tau,[theta1,theta2,theta3],[-pi/2,pi/2,pi]))
```

```
Min_mag_tau = 15
```

```
% Verifying the optimal results of all theta values for maximum and minimum
% evaluation of the magnitude of joint torque values:

fprintf(['Verifying results for maximum and minimum magnitude of tau with' ...
    'optimal theta values:\n']);
```

```
Verifying results for maximum and minimum magnitude of tau withoptimal theta values:
```

```
% Define the objective function to maximize Magnitude_tau
```

```
objectiveFunction_Max = @(theta) -double(norm(subs(magnitude_tau, [theta1, theta2,
theta3], double(theta))));
% Define the objective function to mainimize Magnitude_tau
objectiveFunction_Min = @(theta) double(norm(subs(magnitude_tau, [theta1, theta2,
theta3], double(theta))));

% Define initial guess for thetas
x0 = [pi/4, pi/4, pi/4];

% Define bounds on thetas
lb = [0, 0, 0];
ub = [pi, pi, pi];

% Set up the optimization options
options = optimoptions('fmincon', 'Display', 'iter'); % Display optimization process

% Solve the optimization problem to find maximum magnitude_tau
[xMax, fMax] = fmincon(objectiveFunction_Max, x0, [], [], [], [], lb, ub, [],
options);
```

```
                                          First-order      Norm of
 Iter F-count            f(x)  Feasibility   optimality         step
    0       4   -3.919689e+01    0.000e+00    9.966e+00
    1       8   -5.609177e+01    0.000e+00    1.272e+01    1.060e+00
    2      12   -5.608346e+01    0.000e+00    9.967e-02    6.658e-02
    3      16   -5.611481e+01    0.000e+00    2.462e-01    1.043e-01
    4      20   -5.612178e+01    0.000e+00    3.808e-02    4.584e-02
    5      24   -5.612351e+01    0.000e+00    3.991e-02    9.528e-03
    6      28   -5.612379e+01    0.000e+00    8.705e-03    8.547e-03
    7      32   -5.612387e+01    0.000e+00    2.934e-03    8.270e-03
    8      36   -5.612392e+01    0.000e+00    7.059e-03    2.686e-02
    9      40   -5.612394e+01    0.000e+00    8.432e-03    1.066e-01
   10      44   -5.612391e+01    0.000e+00    5.702e-03    2.502e-01
   11      48   -5.612387e+01    0.000e+00    2.014e-03    3.047e-01
   12      52   -5.612386e+01    0.000e+00    1.000e-03    1.057e-01
   13      56   -5.612436e+01    0.000e+00    3.966e-02    3.296e-02
   14      60   -5.612466e+01    0.000e+00    4.157e-03    2.772e-02
   15      64   -5.612466e+01    0.000e+00    2.304e-03    9.228e-03
   16      68   -5.612466e+01    0.000e+00    2.000e-04    1.521e-03
   17      72   -5.612479e+01    0.000e+00    7.200e-05    1.730e-03
   18      76   -5.612482e+01    0.000e+00    4.355e-05    8.456e-04
   19      80   -5.612485e+01    0.000e+00    1.109e-05    8.913e-04
   20      84   -5.612486e+01    0.000e+00    2.976e-06    4.371e-04
   21      88   -5.612486e+01    0.000e+00    9.574e-07    2.019e-04

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>
```

```
% Display the results (Maximum)
fprintf('Maximum Magnitude_tau: %f\n', abs(-fMax)); % Negate the value back to the
original form
```

```
Maximum Magnitude_tau: 56.124860
```

```
fprintf('Optimal values for theta1, theta2, and theta3: %f, %f, %f\n',
rad2deg(xMax(1)), rad2deg(xMax(2)), rad2deg(xMax(3)));
```

```
Optimal values for theta1, theta2, and theta3: 89.992386, 0.012731, 0.007796
```

```
% Solve the optimization problem to find minimum magnitude_tau
[xMin, fMin] = fmincon(objectiveFunction_Min, x0, [], [], [], [], lb, ub, [],
options);
```

```
                                         First-order      Norm of
 Iter F-count            f(x)  Feasibility   optimality         step
    0       4    3.919689e+01    0.000e+00    2.214e+01
    1       8    1.626950e+01    0.000e+00    6.725e+00    2.648e+00
    2      13    1.531488e+01    0.000e+00    2.331e+00    6.207e-01
    3      17    1.500300e+01    0.000e+00    4.903e-01    2.868e-01
    4      21    1.501394e+01    0.000e+00    4.960e-01    2.316e-01
    5      25    1.501695e+01    0.000e+00    9.723e-02    9.207e-02
    6      29    1.500593e+01    0.000e+00    1.378e-01    1.109e-01
    7      33    1.500462e+01    0.000e+00    2.021e-02    8.398e-03
    8      37    1.500207e+01    0.000e+00    2.610e-02    5.367e-02
    9      41    1.500075e+01    0.000e+00    2.970e-02    5.741e-02
   10      45    1.500030e+01    0.000e+00    1.618e-02    3.999e-02
   11      49    1.500013e+01    0.000e+00    7.487e-03    2.953e-02
   12      53    1.500007e+01    0.000e+00    2.871e-03    1.762e-02
   13      57    1.500005e+01    0.000e+00    5.473e-04    7.422e-03
   14      61    1.500005e+01    0.000e+00    2.000e-04    1.676e-03
   15      65    1.500002e+01    0.000e+00    2.684e-03    1.976e-02
   16      69    1.500001e+01    0.000e+00    4.277e-04    8.775e-03
   17      73    1.500001e+01    0.000e+00    2.442e-04    3.851e-03
   18      77    1.500001e+01    0.000e+00    4.907e-05    6.269e-04
   19      81    1.500001e+01    0.000e+00    4.000e-05    8.088e-05
   20      85    1.500000e+01    0.000e+00    1.377e-04    1.197e-02
   21      89    1.500000e+01    0.000e+00    5.174e-04    7.289e-03
   22      93    1.500000e+01    0.000e+00    9.668e-05    2.785e-03
   23      97    1.500000e+01    0.000e+00    8.002e-06    5.607e-04
   24     101    1.500000e+01    0.000e+00    8.000e-06    9.221e-05
   25     105    1.500000e+01    0.000e+00    5.457e-04    1.075e-02
   26     109    1.500000e+01    0.000e+00    2.207e-04    5.955e-03
   27     113    1.500000e+01    0.000e+00    3.215e-05    2.612e-03
   28     117    1.500000e+01    0.000e+00    5.963e-06    5.719e-04
   29     121    1.500000e+01    0.000e+00    1.135e-05    1.139e-04
   30     125    1.500000e+01    0.000e+00    2.641e-05    5.544e-04

                                         First-order      Norm of
 Iter F-count            f(x)  Feasibility   optimality         step
   31     129    1.500000e+01    0.000e+00    8.033e-05    3.022e-03
   32     133    1.500000e+01    0.000e+00    2.478e-04    1.476e-02
   33     137    1.500000e+01    0.000e+00    4.596e-04    3.045e-02
   34     141    1.500000e+01    0.000e+00    8.102e-04    8.421e-02
   35     145    1.500000e+01    0.000e+00    1.127e-03    1.524e-01
   36     149    1.500000e+01    0.000e+00    9.945e-04    1.782e-01
   37     153    1.500000e+01    0.000e+00    2.145e-04    1.382e-01
   38     157    1.500000e+01    0.000e+00    4.519e-06    1.195e-02
   39     161    1.500000e+01    0.000e+00    1.825e-06    1.939e-03
   40     165    1.500000e+01    0.000e+00    1.600e-06    5.409e-04
   41     169    1.500000e+01    0.000e+00    2.055e-04    4.460e-02
   42     173    1.500000e+01    0.000e+00    1.798e-05    3.395e-02
   43     177    1.500000e+01    0.000e+00    1.452e-05    9.646e-03
   44     181    1.500000e+01    0.000e+00    5.795e-06    1.337e-04
```

```
   45     185    1.500000e+01    0.000e+00    3.196e-07    2.981e-05

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>
```

```
% Display the results (Minimum)
fprintf('Minimum Magnitude_tau: %f\n', abs(-fMin)); % Negate the value back to the
original form
```

```
Minimum Magnitude_tau: 15.000000
```

```
fprintf('Optimal values for theta1, theta2, and theta3: %f, %f, %f\n',
rad2deg(xMin(1)), rad2deg(xMin(2)), rad2deg(xMin(3)));
```

```
Optimal values for theta1, theta2, and theta3: 90.001144, 90.817724, 179.176356
```