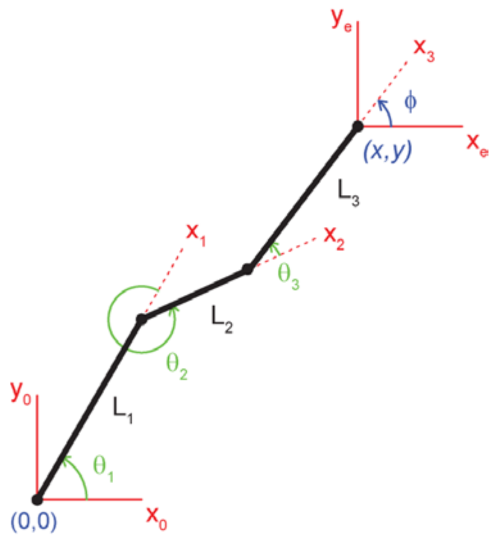
MAE 263C Homework #2

(Due online by 5pm on Friday, 4/20)

Consider the following planar 3R manipulator with joint coordinates θ_1, θ_2, θ_3 and end-effector coordinates x, y, and ϕ:



The Jacobian for the planar 3R manipulator is shown in the velocity equation below,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -(L_1 s_1 + L_2 s_{12} + L_3 s_{123}) & -(L_2 s_{12} + L_3 s_{123}) & -L_3 s_{123} \\ (L_1 c_1 + L_2 c_{12} + L_3 c_{123}) & (L_2 c_{12} + L_3 c_{123}) & L_3 c_{123} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

Assume the following link lengths (arbitrary units):

```
L1 = 2;
L2 = 1;
L3 = 0.75;
```

Use the following three manipulator configurations (joint angles in radians) for the subsequent analyses. Note that `th1` is NOT by itself a single manipulator configuration. Rather, you would take the first elements in each of `th1`, `th2`, and `th3` to form configuration 1. The second elements in each of the `th1`, `th2`, and `th3` vectors would form configuration 2, etc.

```
th1 = (pi/180)*[0; -22.5; -45];
th2 = (pi/180)*[-0.05; -22.5; -67.5];
th3 = (pi/180)*[0; -45; -67.5];
```

Velocity analysis:

1. Write a function to calculate the Jacobian matrix for each of the three manipulator configurations. Use singular value decomposition (MATLAB function `svd`) on the Jacobian matrix (J) and report the three singular values for each configuration.

2. Transform a unit sphere joint angle velocity input into the velocity manipulability ellipsoid. On a single x-y plot, plot the manipulator in the three configurations and overlay the velocity manipulability ellipses (x-dot and y-dot values only) such that each is centered at the distal end of its respective link 3. **To facilitate grading, use the axis limits indicated on the next page.**

For joint angle velocity inputs:

```
% Creating a unit radius sphere for velocity analysis (rad/s)
N = 29; % For generating three (N+1)x(N+1) matrices of coordinates
[th1dot, th2dot, th3dot] = sphere(N);
```

For x-y plot limits:

```
axis ([-5 8 -5 5]);
```

When plotting the ellipses, use solid lines (e.g., `‘r-‘`).

Force analysis:

3. Use singular value decomposition on the Jacobian inverse transpose matrix (J^{-T}) and report the three singular values for each configuration.

4. Transform a unit sphere joint torque input into the force manipulability ellipsoid. On a single x-y plot, plot the manipulator in the three configurations and overlay the force manipulability ellipses (f_x and f_y values only) such that each is centered at the distal end of its respective link 3. **To facilitate grading, use the axis limits indicated below.**

   For joint torque inputs:

   ```
   % Creating a unit radius sphere for force analysis (arbitrary units)
   N = 29; % For generating three (N+1)x(N+1) matrices of coordinates
   [tau1, tau2, tau3] = sphere(N);
   ```

   For x-y plot limits:

   ```
   axis ([-5 8 -5 5]);
   ```

   When plotting the ellipses, use solid lines (e.g., `‘r-‘`).

5. For each of the three configurations, plot the velocity and force manipulability ellipsoids on a single x-y-z plot. You should have three separate plots (one for each configuration). The velocity ellipsoid coordinates will be x-dot, y-dot, and phi-dot. The force ellipsoid coordinates will be f_x, f_y, and M_z. **To facilitate grading, use the default axis limits for each configuration plot.** Note that you can check your answers for problems 2 and 4 above by viewing the x-y plane of each ellipsoid plot. **Differentiate between the velocity and force manipulability ellipsoids with labels or color-coding, as suggested below.**

   For each of i=1:3 configurations,

   ```
   % FOR velocity analysis
   hSurface1 = surf(xdot(:,:,i), ydot(:,:,i), phidot(:,:,i));
   set(hSurface1,'FaceColor',[1 0 0],'FaceAlpha',0.35);   % Red
   hold on;

   % FOR force analysis
   hSurface2 = surf(fxdot(:,:,i), fydot(:,:,i), Mzdot(:,:,i));
   set(hSurface2,'FaceColor',[0 0 1],'FaceAlpha',0.35);   % Blue
   view(-37, 30);
   ```

NOTE: Each student must submit his/her own independent work. For all MATLAB-related problems, you must provide all code (main scripts, function files), figures, and the final results from your command window, as appropriate. You may save this content to PDF or take photographs for electronic submission via Gradescope. The more intermediate results and comments you provide, the greater the opportunity for partial credit.