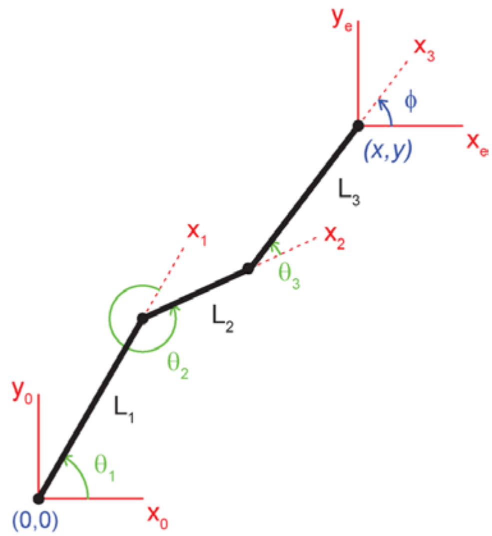
MAE 263C Homework #2

(Due online by 5pm on Friday, 4/21)

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:

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

<u>Velocity analysis:</u>

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.

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.

   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. 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. **If you do not print your plots in color, make sure you differentiate between the velocity and force manipulability ellipsoids with labels or color-coding by hand.**

   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. The more intermediate results and comments you provide, the greater the opportunity for partial credit. To receive full credit, you must **submit a hard copy of your original MATLAB code and plots** with your assignment.

```
%***************************************************************************
% VERONICA J. SANTOS
%
% HW2.m
%
% This script file performs singular value decomposition on the Jacobian
% matrix of a planar 3R manipulator and plots the velocity manipulability
% ellipses for multiple manipulator configurations.  It also performs
% singular value decomposition on the Jacobian inverse transpose matrix
% and plots the force manipulability ellipses.
%
% FUNCTIONS CALLED: FUNC_Jacobian_Planar3R
%***************************************************************************
clear all; close all;

%-------------------------------------
% DEFINE CONSTANTS
%-------------------------------------
% Link lengths (arbitrary units)
L1 = 2;
L2 = 1;
L3 = 0.75;

L_vec = [L1; L2; L3];

% Joint angles (rad)
th1 = (pi/180)*[0; -22.5; -45];
th2 = (pi/180)*[-0.05; -22.5; -67.5];
th3 = (pi/180)*[0; -45; -67.5];

%-------------------------------------
% CALCULATE THE JACOBIAN
%-------------------------------------
for i=1:length(th1)
    J(:,:,i) = FUNC_Jacobian_Planar3R(L_vec, [th1(i); th2(i); th3(i)]);
    JinvT(:,:,i) = inv(J(:,:,i)');
    if (sum(isinf(JinvT(:,:,i))) > 0)    % Need to calculate pseudoinverse
        JinvT(:,:,i) = pinv(J(:,:,i)');
        disp('Using pseudoinverse for')
        i
    end
end

%-------------------------------------
% PERFORM SINGULAR VALUE DECOMPOSITION
%-------------------------------------
for i=1:length(th1)
    % FOR velocity analysis
    [U_vel(:,:,i), S_vel(:,:,i), V_vel(:,:,i)] = svd(J(:,:,i));
    SingVals_vel(:,i) = svd(J(:,:,i));

    % FOR force analysis
    [U_force(:,:,i), S_force(:,:,i), V_force(:,:,i)] = svd(JinvT(:,:,i));
    SingVals_force(:,i) = svd(JinvT(:,:,i));
end

disp('HW #2, Prob. #1')
```

```
SingVals_vel

disp('HW #2, Prob. #3')
SingVals_force

%------------------------------------
% CALCULATE endpoints of links for plotting
%------------------------------------
L1_x = L1.*cos(th1);
L1_y = L1.*sin(th1);

L2_x = L1_x + L2.*cos(th1+th2);
L2_y = L1_y + L2.*sin(th1+th2);

L3_x = L2_x + L3.*cos(th1+th2+th3);
L3_y = L2_y + L3.*sin(th1+th2+th3);

%------------------------------------
% CREATE UNIT JOINT 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);

% Creating a unit radius sphere for force analysis (arbitrary units)
[tau1, tau2, tau3] = sphere(N);

%------------------------------------
% ROTATE AND SCALE JOINT INPUTS
%------------------------------------
% FOR velocity analysis
th1dot_vec = reshape(th1dot, numel(th1dot),1);
th2dot_vec = reshape(th2dot, numel(th2dot),1);
th3dot_vec = reshape(th3dot, numel(th3dot),1);

thdot_mat = [th1dot_vec'; th2dot_vec'; th3dot_vec'];

% FOR force analysis
tau1_vec = reshape(tau1, numel(tau1),1);
tau2_vec = reshape(tau2, numel(tau2),1);
tau3_vec = reshape(tau3, numel(tau3),1);

tau_mat = [tau1_vec'; tau2_vec'; tau3_vec'];

for i=1:length(th1)
    % FOR velocity analysis
    xdot_mat(:,:,i) = U_vel(:,:,i)*S_vel(:,:,i)*V_vel(:,:,i)'*thdot_mat;

    % FOR force analysis
    f_mat(:,:,i) = U_force(:,:,i)*S_force(:,:,i)*V_force(:,:,i)'*tau_mat;
end

%------------------------------------
% PLOT RESULTS
%------------------------------------
% FOR joint angle velocity inputs
```

```
tempcmd = sprintf('h = figure(''name'',''Joint angle velocity inputs (unit
sphere)'', ''NumberTitle'',''off'');');
eval(tempcmd);
hSurface = surf(th1dot, th2dot, th3dot);
set(hSurface,'FaceColor',[1 0 0],'FaceAlpha',0.35);
grid on;
axis equal;
xlabel('\theta_1-dot');
ylabel('\theta_2-dot');
zlabel('\theta_3-dot');

% Plot options
Lwidth = 3; % Linewidth
Msize = 20;  % Marker size
PlotCol = ['r', 'g', 'b'];
axisvec_2D = [-5 8 -5 5];

% FOR velocity manipulability ellipses
tempcmd = sprintf('h = figure(''name'',''Prob. #2:  3R planar manip. w/
velocity manipulability ellipses'', ''NumberTitle'',''off'');');
eval(tempcmd);
plot(0,0,'k.', 'MarkerSize', Msize);
hold on;
for i=1:length(th1)
    % Plot velocity manipulability ellipses
    tempcmd = sprintf('plot(xdot_mat(1,:,i) + L3_x(i), xdot_mat(2,:,i) +
L3_y(i), ''%s-'');', PlotCol(i));
    eval(tempcmd);

    % Plot links
    plot([0, L1_x(i)], [0, L1_y(i)], 'k', 'LineWidth', Lwidth);
    plot(L1_x(i), L1_y(i), 'k.', 'MarkerSize', Msize);
    plot([L1_x(i), L2_x(i)], [L1_y(i), L2_y(i)], 'k', 'LineWidth', Lwidth);
    plot(L2_x(i), L2_y(i), 'k.', 'MarkerSize', Msize);
    plot([L2_x(i), L3_x(i)], [L2_y(i), L3_y(i)], 'k', 'LineWidth', Lwidth);
    plot(L3_x(i), L3_y(i), 'k.', 'MarkerSize', Msize);
end
axis(axisvec_2D);
xlabel('x or x-dot');
ylabel('y or y-dot');

% FOR joint torque inputs
tempcmd = sprintf('h = figure(''name'',''Joint torque inputs (unit sphere)'',
''NumberTitle'',''off'');');
eval(tempcmd);
hSurface = surf(tau1, tau2, tau3);
set(hSurface,'FaceColor',[0 0 1],'FaceAlpha',0.35);
grid on;
axis equal;
xlabel('\tau_1');
ylabel('\tau_2');
zlabel('\tau_3');

% FOR force manipulability ellipses
tempcmd = sprintf('h = figure(''name'',''Prob. #4:  3R planar manip. w/ force
manipulability ellipses'', ''NumberTitle'',''off'');');
eval(tempcmd);
```

```
plot(0,0,'k.', 'MarkerSize', Msize);
hold on;
for i=1:length(th1)
    % Plot force manipulability ellipses
    tempcmd = sprintf('plot(f_mat(1,:,i) + L3_x(i), f_mat(2,:,i) + L3_y(i),
''%s-'');', PlotCol(i));
    eval(tempcmd);

    % Plot links
    plot([0, L1_x(i)], [0, L1_y(i)], 'k', 'LineWidth', Lwidth);
    plot(L1_x(i), L1_y(i), 'k.', 'MarkerSize', Msize);
    plot([L1_x(i), L2_x(i)], [L1_y(i), L2_y(i)], 'k', 'LineWidth', Lwidth);
    plot(L2_x(i), L2_y(i), 'k.', 'MarkerSize', Msize);
    plot([L2_x(i), L3_x(i)], [L2_y(i), L3_y(i)], 'k', 'LineWidth', Lwidth);
    plot(L3_x(i), L3_y(i), 'k.', 'MarkerSize', Msize);
end
axis(axisvec_2D);
xlabel('x or f_x');
ylabel('y or f_y');

for i=1:length(th1)
    % FOR velocity analysis
    xdot(:,:,i) = reshape(xdot_mat(1,:,i), size(th1dot,1), size(th1dot,2));
    ydot(:,:,i) = reshape(xdot_mat(2,:,i), size(th1dot,1), size(th1dot,2));
    phidot(:,:,i) = reshape(xdot_mat(3,:,i), size(th1dot,1), size(th1dot,2));

    % FOR force analysis
    fxdot(:,:,i) = reshape(f_mat(1,:,i), size(th1dot,1), size(th1dot,2));
    fydot(:,:,i) = reshape(f_mat(2,:,i), size(th1dot,1), size(th1dot,2));
    Mzdot(:,:,i) = reshape(f_mat(3,:,i), size(th1dot,1), size(th1dot,2));
end

for i=1:length(th1)
    tempcmd = sprintf('h = figure(''name'',''Prob. #5:  Velocity (red) and
force (blue) manipulability ellipsoids'', ''NumberTitle'',''off'');');
    eval(tempcmd);

    % 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

    tempcmd = sprintf('title(''Configuration i=%d'');', i);
    eval(tempcmd);
    xlabel('x-dot or f_x-dot');
    ylabel('y-dot or f_y-dot');
    zlabel('\phi-dot or M_z-dot');
    view(-37, 30);
end
```

+3

```
%**************************************************************************
% VERONICA J. SANTOS
%
% FUNC_Jacobian_Planar3R.m
%
% This function takes in a 3x1 vector of link lengths [L1; L2; L3] and a
% 3x1 vector of joint angles [th1; th2; th3] in radians and returns the
% 3x3 Jacobian matrix for a planar 3R manipulator.
%**************************************************************************
function J = FUNC_Jacobian_Planar3R (L_vec, th_vec)

%-------------------------------------
% UNPACK VARIABLES
%-------------------------------------
L1 = L_vec(1);
L2 = L_vec(2);
L3 = L_vec(3);

th1 = th_vec(1);
th2 = th_vec(2);
th3 = th_vec(3);

%-------------------------------------
% SHORTHAND
%-------------------------------------
s1 = sin(th1);
s12 = sin(th1+th2);
s123 = sin(th1+th2+th3);

c1 = cos(th1);
c12 = cos(th1+th2);
c123 = cos(th1+th2+th3);

%-------------------------------------
% BUILD THE JACOBIAN MATRIX
%-------------------------------------
% ROW 1
J(1,1) = -(L1*s1 + L2*s12 + L3*s123);
J(1,2) = -(L2*s12 + L3*s123);
J(1,3) = -L3*s123;

% ROW 2
J(2,1) = (L1*c1 + L2*c12 + L3*c123);
J(2,2) = (L2*c12 + L3*c123);
J(2,3) = L3*c123;

% ROW 3
J(3,:) = [1, 1, 1];
```

COMMAND WINDOW OUTPUT:

+6

(+2 per config.)

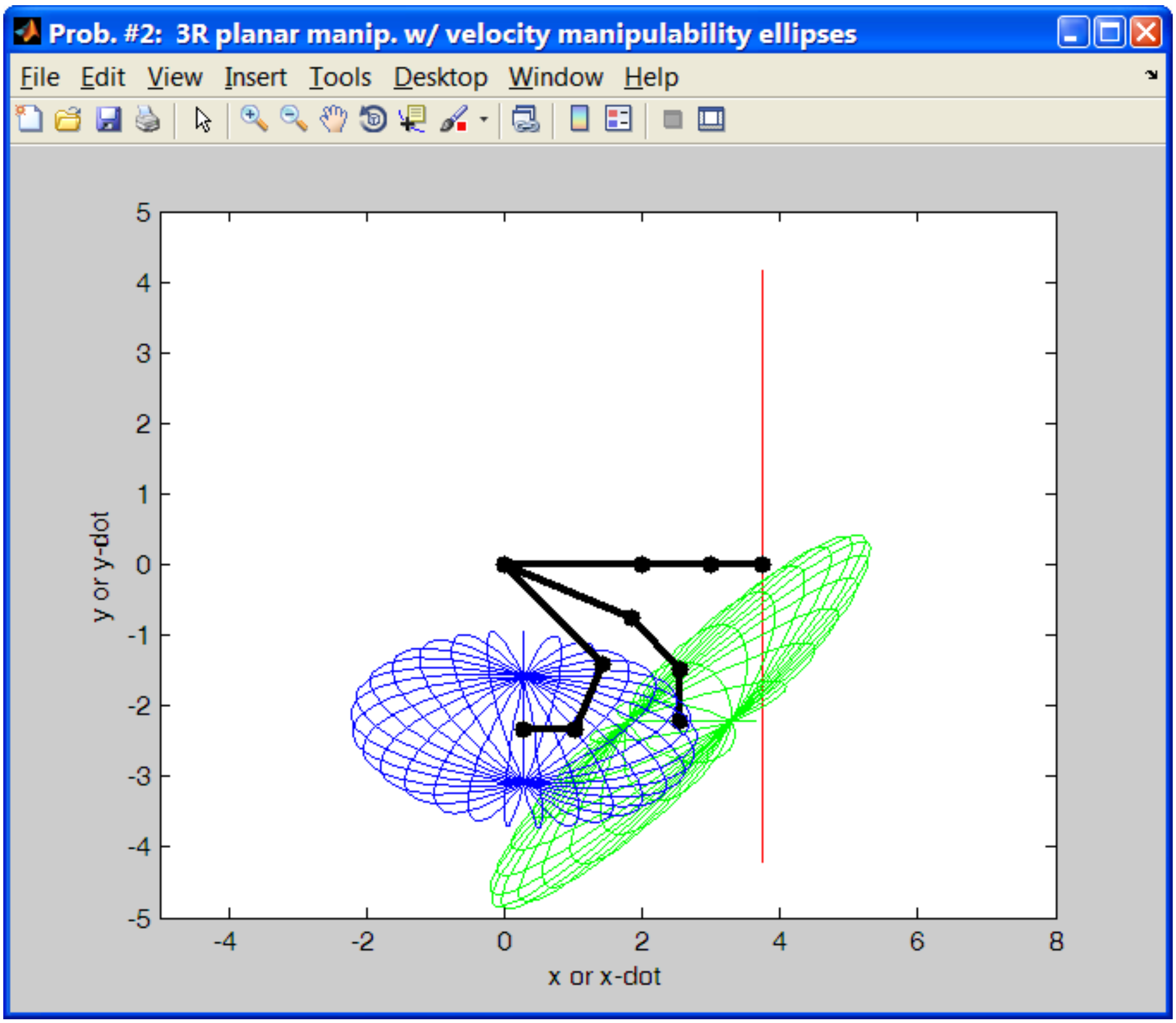
HW #2, Prob. #1 (Cols 1-3 = Configurations 1-3):

```
SingVals_vel =

    4.4707    4.0357    2.9210
    0.8369    1.1570    1.5988
    0.0005    0.1639    0.3957
```

+9

(+3 per config.)



+6

(+2 per config.)

HW #2, Prob. #3 (Cols 1-3 = Configurations 1-3):

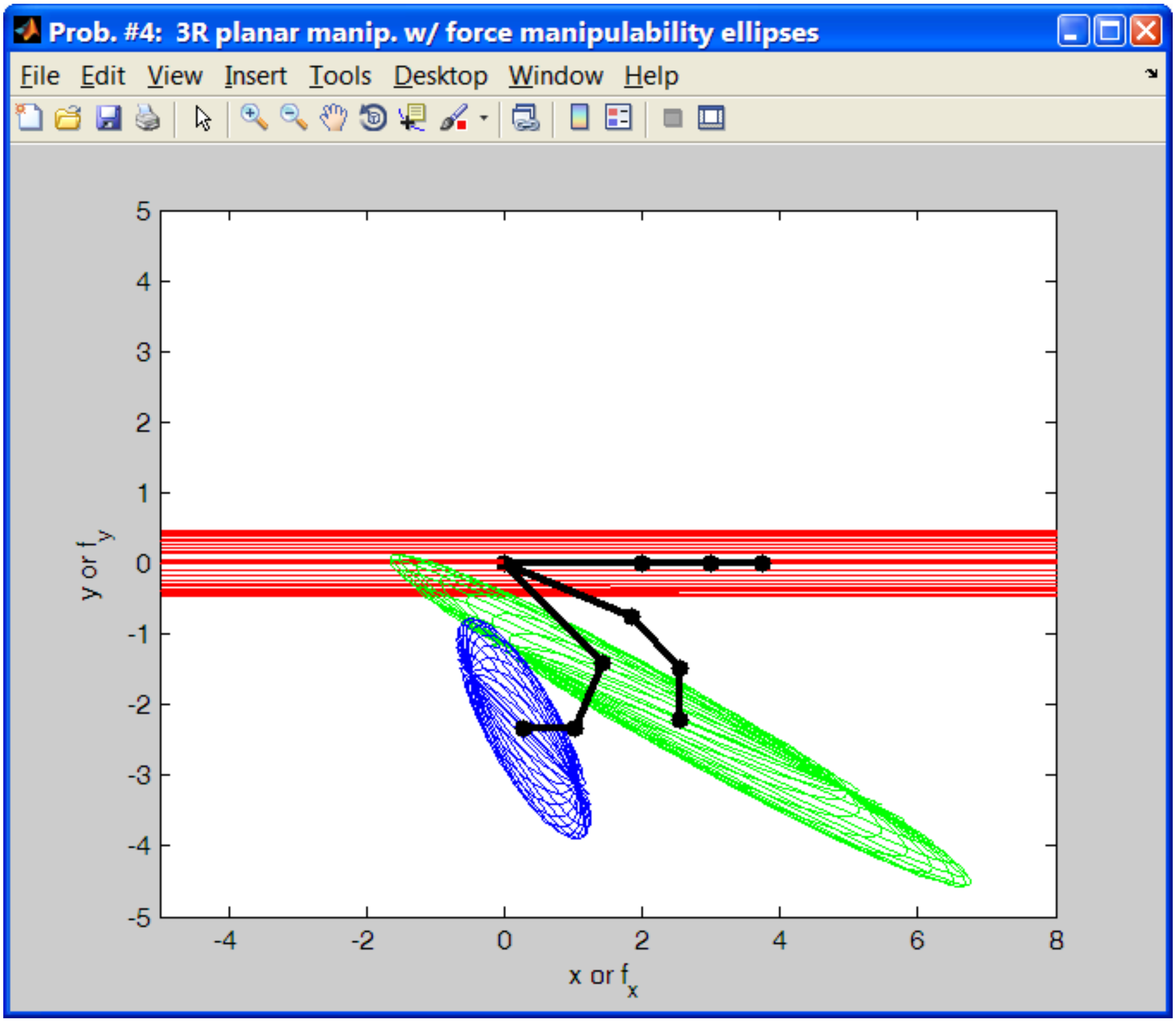
```
SingVals_force =

  1.0e+003 *

    2.1438    0.0061    0.0025
    0.0012    0.0009    0.0006
    0.0002    0.0002    0.0003
```

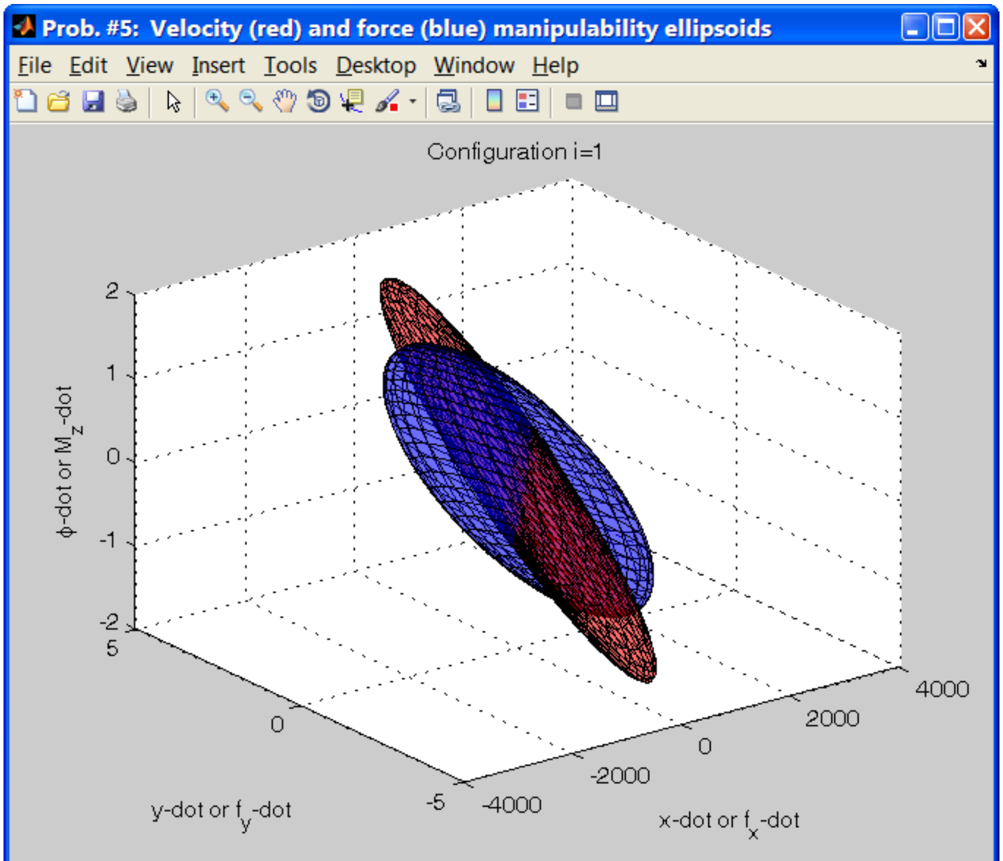
+9

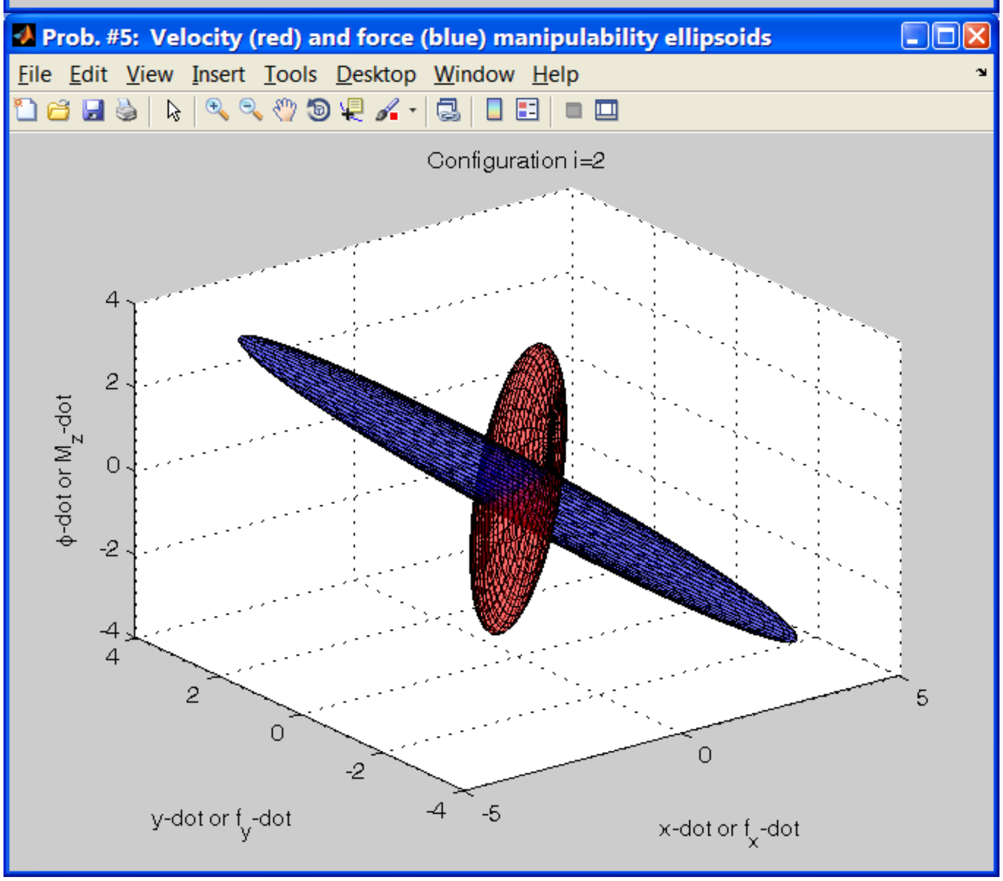
(+3 per config.)

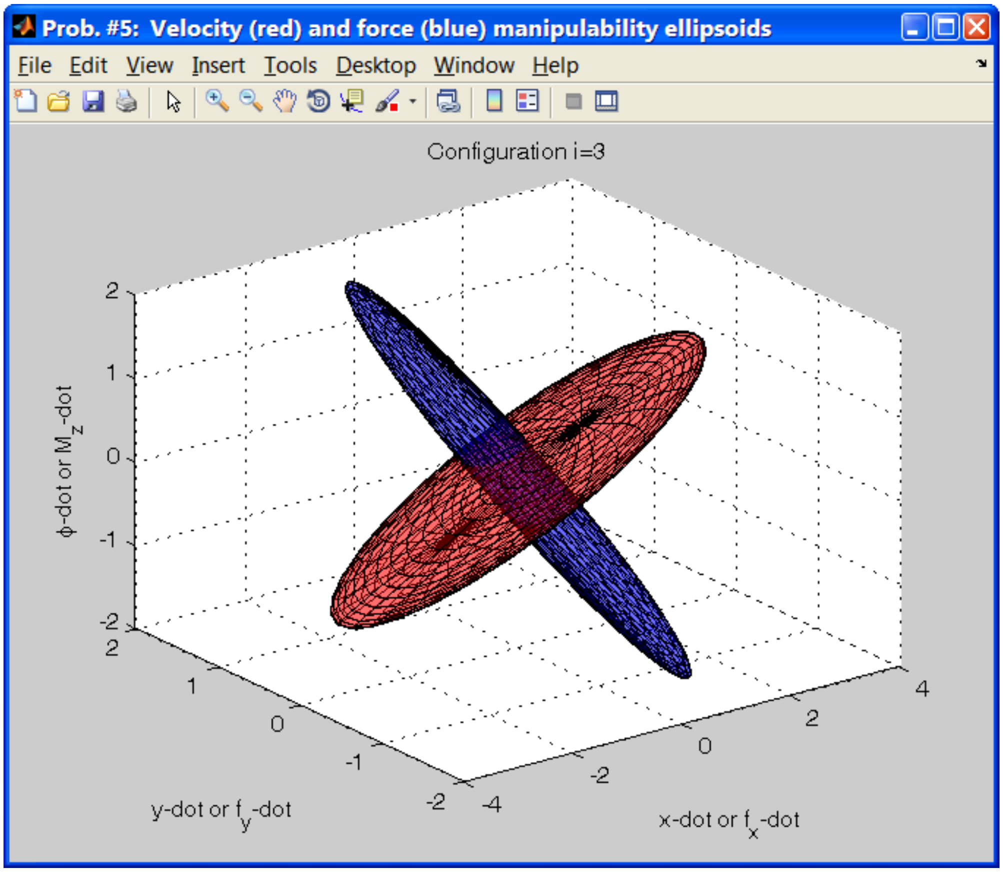


+15

(+5 per config.)






Prob. #5: Velocity (red) and force (blue) manipulability ellipsoids
File Edit View Insert Tools Desktop Window Help
Configuration i=3
ϕ-dot or M_z-dot
y-dot or f_y-dot
x-dot or f_x-dot
2
1
0
-1
-2
2
1
0
-1
-2
-4
-2
0
2
4