Table of Contents


```
% ARO 4090 - Space Vehicle Dyn. & Cntrl. | Dr. Maggia | Justin Millsap %
clc; clear; close all;
```

Problem 1

```
% Vectors V and W in "A" Reference Frame
v_a = [1 ; 2 ; 3] ;
w_a = [-1 ; 2 ; 1] ;

% Vectors V and W in "B" Reference Frame
v_b = [3.56186 ; 1.13448 ; 0.16150]  ;
w_b = [1.33712 ; 1.31501 ; -1.57572] ;

% Part A) Calculate the inner product in both reference frames

% ~~~~~~~~~~~~~~~~~~~~~~~ A-RF ~~~~~~~~~~~~~~~~~~~~~~~%
A_rf_inner_product = dot(v_a,w_a)

% ~~~~~~~~~~~~~~~~~~~~~~~ B-RF ~~~~~~~~~~~~~~~~~~~~~~~%
B_rf_inner_product = dot(v_b,w_b)


% Part B) Calculate the outter product in both reference frames

% ~~~~~~~~~~~~~~~~~~~~~~~ A-RF ~~~~~~~~~~~~~~~~~~~~~~~%
A_rf_outter_product = cross(v_a,w_a)

% ~~~~~~~~~~~~~~~~~~~~~~~ B-RF ~~~~~~~~~~~~~~~~~~~~~~~%
B_rf_outter_product = cross(v_b,w_b)

% Part c) Find the Coordiante Transformation Matrix R_BA
A = [1 2 3; -1 2 1; -4 -4 4];

% Right-hand side vectors b1, b2, and b3 for each system
b1 = [3.56; 1.33712; -2];
b2 = [1.13448; 1.31501; 5.82];
b3 = [0.16150; -1.57572; 3.16];

% Solving each system
r1 = A\b1; % Solves for r11, r12, r13
r2 = A\b2; % Solves for r21, r22, r23
r3 = A\b3; % Solves for r31, r32, r33
```

```
% Displaying the results
disp('r11, r12, r13:');
disp(r1);
disp('r21, r22, r23:');
disp(r2);
disp('r31, r32, r33:');
disp(r3);
```

```
A_rf_inner_product =

     6


B_rf_inner_product =

    6.0000


A_rf_outter_product =

    -4
    -4
     4


B_rf_outter_product =

   -2.0000
    5.8284
    3.1669

r11, r12, r13:
    0.4995
    0.6124
    0.6119

r21, r22, r23:
   -0.7493
   -0.0467
    0.6590

r31, r32, r33:
    0.4336
   -0.7886
    0.4350
```

Problem 2

```
clc; clear; close all

% INPUTS TO RIB matrix
```

```
r11 = -0.17101007;  r12 = 0.46984631 ;   r13 = -0.86602540;
r21 = 0.98432795 ;  r22 = 0.04305861 ;   r23 = -0.17101007;
r31 = -0.04305861;  r32 = -0.88169745;   r33 = -0.46984631;
R_BI = [r11 r12 r13 ; r21 r22 r23 ;  r31 r32 r33];

% 3-2-1 Rotation Matrix
theta = asind( abs(R_BI(1,3)) );
alpha = atan2d(R_BI(1,2) , R_BI(1,1));
gamma = atan2d(R_BI(2,3) , R_BI(3,3) ) + 360;
R1 = [ 1 0 0  ; 0 cosd(gamma) sind(gamma) ; 0 -sind(gamma) cosd(gamma)];
R2 = [cosd(theta) 0 -sind(theta); 0 1 0; sind(theta) 0 cosd(theta)];
R3 = [ cosd(alpha) sind(alpha) 0 ; -sind(alpha) cosd(alpha) 0 ; 0 0 1];
Rot_321 = R1*R2*R3

% 2-3-2 Rotation Matrx
beta = acosd(R_BI(2,2))
alpha = atan2d(R_BI(3,2) , R_BI(1,2)) + 360
gamma = atan2d(R_BI(2,3) , -R_BI(2,1)) + 360
R1 = [ 1 0 0  ; 0 cosd(alpha) sind(alpha) ; 0 -sind(alpha) cosd(alpha)];
R2 = [cosd(beta) 0 -sind(beta); 0 1 0; sind(beta) 0 cosd(beta)];
R3 = [ cosd(gamma) sind(gamma) 0 ; -sind(gamma) cosd(gamma) 0 ; 0 0 1];


Rot_321 =

   -0.1710    0.4698   -0.8660
    0.9843    0.0431   -0.1710
   -0.0431   -0.8817   -0.4698


beta =

   87.5322


alpha =

  298.0526


gamma =

  189.8558
```

Problem 3

```
clc; clear; close all

% Part A) Take MATLAB Onramp online course ~~ FINISHED

% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~Part B~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ %
disp('Part B')
```

```
% Write a MATLAB function with the following inputs/outputs:
    % ~Inputs: Eurler Angles (gamma, beta, alpha) in degrees and any
        %feasible rotation sequence (e.g. 3-2-1)
    % ~Outputs: Coordiate Transformation Matrix R_BI



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%      EDIT       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Input Eurler Angles [degrees]

gamma = 200 ;
beta = 60   ;
alpha = 110 ;

% Define Rotation Sequence (e.g. [3-2-1] use 321)

Rot_Seq = 321 ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  DO NOT EDIT    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Define the elementary rotation matrices
R1 = [ 1 0 0  ; 0 cosd(gamma) sind(gamma) ; 0 -sind(gamma) cosd(gamma)];
R2 = [cosd(beta) 0 -sind(beta); 0 1 0; sind(beta) 0 cosd(beta)];
R3 = [ cosd(alpha) sind(alpha) 0 ; -sind(alpha) cosd(alpha) 0 ; 0 0 1];



% Determine and compute the rotation matrix based on Rot_Seq
if Rot_Seq == 321
    R_BI = R1*R2*R3; % [3-2-1]
elseif Rot_Seq == 231
    R_BI = R1R3*R2; % [2-3-1]
elseif Rot_Seq == 121
    R_BI = R1*R2*R1; % [1-2-1]
elseif Rot_Seq == 131
    R_BI = R1*R3*R1; % [1-3-1]
elseif Rot_Seq == 132
    R_BI = R2*R3*R1; % [1-3-2]
elseif Rot_Seq == 312
    R_BI = R2*R1*R3; % [3-1-2]
elseif Rot_Seq == 212
    R_BI = R2*R1*R2; % [2-1-2]
elseif Rot_Seq == 232
    R_BI = R2*R3*R2; % [2-3-2]
elseif Rot_Seq == 213
    R_BI = R3*R1*R2; % [2-1-3]
elseif Rot_Seq == 123
    R_BI = R3*R2*R1; % [1-2-3]
elseif Rot_Seq == 313
    R_BI = R3*R1*R3; % [3-1-3]
elseif Rot_Seq == 323
    R_BI = R3*R2*R3; % [3-2-3]
else
```

```
    error('Invalid rotation sequence');
end

disp('The Coodinate Transformation Matrix R_BI = ')
disp(R_BI)



% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~Part C~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ %
disp(' Part C')
% Write a MATLAB function with the following inputs/outputs:
    % ~Inputs: Coordiante Tranformation Matrix R_BI and the 3-2-1
    %          Rotation Sequence.
    % ~Outputs: Eurler angles ( psi , theta , and phi ) in degrees.


% INPUTS TO R_BI matrix Inputs
r11 = -0.17101007;  r12 = 0.46984631 ;   r13 = -0.86602540;
r21 = 0.98432795 ;  r22 = 0.04305861 ;   r23 = -0.17101007;
r31 = -0.04305861;  r32 = -0.88169745;   r33 = -0.46984631;

% R_BI Matrix
R_BI = [r11 r12 r13 ; r21 r22 r23 ;  r31 r32 r33];


Rot_Seq = R1*R2*R3;          % [3-2-1] Rotation Sequence

% Solve for Eurler Angles by using most opitmal positions to compare between
R_BI & Rotation Sequence Matrix


theta = asind(-R_BI(1,3));
phi   = atan2d(R_BI(2,3) , R_BI(3,3)) + 360 ;
psi   = atan2d(R_BI(1,2) , R_BI(1,1)) ;

% Check if Eurler angles work

Rot_Seq = R1*R2*R3;
disp(' The Eurler Angles for the given R_BI Matrix for a given Rotation
Sequence are')
disp(theta) ; disp(phi) ; disp(psi)
```

```
Part B
The Coodinate Transformation Matrix R_BI =
   -0.1710    0.4698   -0.8660
    0.9843    0.0431   -0.1710
   -0.0431   -0.8817   -0.4698

 Part C
 The Eurler Angles for the given R_BI Matrix for a given Rotation Sequence
are
   60.0000
```

```
   200.0000

   110.0000
```

Published with MATLAB® R2023b