

---

```
% HW3 Part 2
% Matt Boler
clc; clear all; close all;
```

## Ginsberg 3.18

```
% Constants:
BC = .12;

% Initial conditions:
L_1 = .25;
theta_1 = 0;
beta_1 = 90;
gamma_1 = 0;

% Final conditions:
L_2 = .5;
theta_2 = 120;
beta_2 = 120;
gamma_2 = -90;

% Find the displacement of point C between the two poses.
% Note: Translating A down to the surface b/c won't affect
% displacement.
% eg find P_c_in_w = vector from 0 to c in world frame

% Define an intermediate frame for beta rotation
i_w = [1 0 0]';
j_w = [0 1 0]';
k_w = [0 0 1]';

i_theta_1 = cosd(theta_1)*i_w + sind(theta_1)*j_w;
j_theta_1 = cosd(theta_1)*j_w - sind(theta_1)*i_w;
k_theta_1 = k_w;
R_theta_1 = [i_theta_1, j_theta_1, k_theta_1];

i_beta_1 = sind(beta_1)*i_theta_1 + cosd(beta_1)*k_theta_1;
j_beta_1 = j_theta_1;
k_beta_1 = -cosd(beta_1)*i_theta_1 + sind(beta_1)*k_theta_1;
R_beta_1 = [i_beta_1, j_beta_1, k_beta_1];

i_gamma_1 = i_beta_1;
j_gamma_1 = cosd(gamma_1)*j_beta_1 + sind(gamma_1)*k_beta_1;
k_gamma_1 = cosd(gamma_1)*k_beta_1 - sind(gamma_1)*j_beta_1;
R_gamma_1 = [i_gamma_1, j_gamma_1, k_gamma_1];

B_world_1 = L_1 * i_beta_1;

C_world_1 = B_world_1 + BC * k_gamma_1

i_theta_2 = cosd(theta_2)*i_w + sind(theta_2)*j_w;
j_theta_2 = cosd(theta_2)*j_w - sind(theta_2)*i_w;
```

---

```

k_theta_2 = k_w;
R_theta_2 = [i_theta_2, j_theta_2, k_theta_2];

i_beta_2 = sind(beta_2)*i_theta_2 + cosd(beta_2)*k_theta_2;
j_beta_2 = j_theta_2;
k_beta_2 = -cosd(beta_2)*i_theta_2 + sind(beta_2)*k_theta_2;
R_beta_2 = [i_beta_2, j_beta_2, k_beta_2];

i_gamma_2 = i_beta_2;
j_gamma_2 = cosd(gamma_2)*j_beta_2 + sind(gamma_2)*k_beta_2;
k_gamma_2 = cosd(gamma_2)*k_beta_2 - sind(gamma_2)*j_beta_2;
R_gamma_2 = [i_gamma_2, j_gamma_2, k_gamma_2];

B_world_2 = L_2 * i_beta_2;

C_world_2 = B_world_2 + BC * k_gamma_2

displacement = C_world_2 - C_world_1

C_world_1 =

    0.2500
         0
    0.1200

C_world_2 =

   -0.3204
    0.3150
   -0.2500

displacement =

   -0.5704
    0.3150
   -0.3700

```

*Published with MATLAB® R2018b*