Programming Assignment 2 Report

Author: Curran Flanders

Course: ENGR-211

Problem statement

There is a need for a computer program that simulates a system in which a part is held in place by a 3D joint and three cables attached to a wall. The user inputs any number of external forces that act at up to five locations on the part (points A, B, C, D, and E), as well as the locations of the anchors at which the cables are connected to the wall. The program outputs the tension forces in the cables and the reaction forces from the 3D joint necessary to keep the part in static equilibrium.

Variable list

A, B, C, D, F, G, H, I: locations of points A, B, C, D, E, F, and I respectively.

Ff: keeps a running total of the applied forces.

Mf: keeps a running total of the moments of the applied forces.

Fi, Fh, Fg: tensions of cables I, H, and G.

CoefSys: coefficients of the matrix equation.

EqnSys: the matrix equation that when solves gives the tension and reaction forces.

answer: the solution matrix containing the tension and reaction forces.

number of forces: the total number of forces that the user wishes to apply to the part

pos: the position of an individual force entered by the user

magnitude: the magnitude of a force entered in magnitude-angle form

form: contains 1 if the user chose component form and 2 for magnitude-angle form

alpha, beta, gamma: the direction angles of the entered force

x, y, z: the components of the entered force

Physical Faults of Program Output

The program will sometimes output a negative value for a tension force, meaning that the cable is in compression. The simplest fix would be to adjust the direction of the net force so that it points in the positive x and negative y directions.

What if all anchors are in the same quadrant?

In this case, the resultant force must point in the quadrant opposite that of the anchors to avoid compression.

Program outputs

```
StaticsProject2
enter the number of additional forces applied on the part1
enter the position of the force (capital A, B, C, D, E, or FD
enter 1 for component form or 2 for polar form2
enter the magnitude1000
enter angle alpha in degrees60
enter angle beta in degrees140
enter angle gamma in degrees80.6
   1.0e+03 *

   -0.3007
    2.4595
    0.0000
    1.4288
   -0.3171
    0.0000
```

```
StaticsProject2
enter the number of additional forces applied on the part1
enter the position of the force (capital A, B, C, D, E, or FD
enter 1 for component form or 2 for polar form1
enter the x component220
enter the y component-300
enter the z component150
   1.0e+03 *

   -0.2761
    1.1604
         0
    0.5823
   -0.1779
    0.0000


enter the number of additional forces applied on the part1
enter the position of the force (capital A, B, C, D, E, or FD
enter 1 for component form or 2 for polar form2
enter the magnitude500
enter angle alpha in degrees110
enter angle beta in degrees22.6
enter angle gamma in degrees100
   1.0e+03 *

    0.1598
   -1.3655
         0
   -0.9056
    0.1415
   -0.0000
```

```
>> StaticsProject2
enter the number of additional forces applied on the part1
enter the position of the force (capital A, B, C, D, E, or FE
enter 1 for component form or 2 for polar form1
enter the x component220
enter the y component-300
enter the z component150
 -124.2614
  641.8788
   76.2500
  302.8977
    3.6212
  -37.5000
>> StaticsProject2
enter the number of additional forces applied on the part1
enter the position of the force (capital A, B, C, D, E, or FF
enter 1 for component form or 2 for polar form2
enter the magnitude500
enter angle alpha in degrees110
enter angle beta in degrees22.6
enter angle gamma in degrees100
 -187.4265
 -327.8036
 -372.8075
 -543.6543
 -138.8345
  -31.3815
```

```
>> StaticsProject2
enter the number of additional forces applied on the part6
enter the position of the force (capital A, B, C, D, E, or FA
enter 1 for component form or 2 for polar form1
enter the x component150
enter the y component300
enter the z component-100
enter the position of the force (capital A, B, C, D, E, or FB
enter 1 for component form or 2 for polar form2
enter the magnitude200
enter angle alpha in degrees60
enter angle beta in degrees150
enter angle gamma in degrees90
enter the position of the force (capital A, B, C, D, E, or FC
enter 1 for component form or 2 for polar form1
enter the x component-40
enter the y component-240
enter the z component110
enter the position of the force (capital A, B, C, D, E, or FD
enter 1 for component form or 2 for polar form2
enter the magnitude150
```

```
enter angle alpha in degrees115
enter angle beta in degrees40
enter angle gamma in degrees103.6
enter the position of the force (capital A, B, C, D, E, or
enter 1 for component form or 2 for polar form2
enter the magnitude330
enter angle alpha in degrees30
enter angle beta in degrees110
enter angle gamma in degrees97.6
enter the position of the force (capital A, B, C, D, E, or
enter 1 for component form or 2 for polar form1
enter the x component65
enter the y component-200
enter the z component-170
enter the position of G[0,900,600]
enter the position of H[0,1000,0]
enter the position of I[0,-700,800]
   1.0e+03 *

    0.3375
    1.2503

  0.3375
  1.2503
 -0.3770
  0.0713
 -0.3080
  0.2484
```

Appendix: Source Code

```
A = [0,0,0];
B = [450,0,0];
C = [650,0,0];
D = [650,200,0];
E = [450,80,0];
F = [450,0,-60];
Ff = [0,0,0];
Mf = [0,0,0];
```

```
number_of_forces = input("enter the number of additional forces applied on the
part");
for i = 1:number_of_forces
    pos = input("enter the position of the force (capital A, B, C, D, E, or F");
    form=input("enter 1 for component form or 2 for polar form");

    if (form==1)
        xyz = [input("enter the x component"),input("enter the y
component"),input("enter the z component")];
    elseif (form == 2)
        magnitude = input("enter the magnitude");
        alpha = input("enter angle alpha in degrees");
        beta = input("enter angle beta in degrees");
        gamma = input("enter angle gamma in degrees");
        xyz = [magnitude*cosd(alpha), magnitude*cosd(beta), magnitude*cosd(gamma)];
    end
    Ff = Ff + xyz;
    Mf = Mf + cross(pos,xyz);
end
```

```
G = input("enter the position of G");
H = input("enter the position of H");
I = input("enter the position of I");
```

```
Fi = [I-D]/norm([I-D]);
Fh = [H-E]/norm([H-E]);
Fg = [G-F]/norm([G-F]);
Mi = cross(D-A,Fi);
Mh = cross(E-A,Fh);
Mg = cross(F-A,Fg);
EqnSys = [
    Fi(1),Fh(1),Fg(1),1,0,0;
    Fi(2),Fh(2),Fg(2),0,1,0;
    Fi(3),Fh(3),Fg(3),0,0,1;
    Mi(1),Mh(1),Mg(1),0,0,0;
    Mi(2),Mh(2),Mg(2),0,0,0;
    Mi(3),Mh(3),Mg(3),0,0,0];
CoefSys = [-Ff(1);-Ff(2);-Ff(3);-Mf(1);-Mf(2);-Mf(3)];
answer = EqnSys\CoefSys;
disp(answer);
```