Alexander Dowell

ME 401-0007 HW2

Homework 2 - Aircraft Flight Dynamics

1. Model the aircraft lateral and longitudinal dynamics using the state-space equations and stability derivatives provided in the reference section
2. Write functions that can be used to recall a specific set of dynamics, including:
    1. First-order roll dynamics (p)
    2. Second-order short period mode dynamics (q, alpha)
    3. Third-order Dutch roll mode dynamics (beta, p, r)
    4. Fourth-order longitudinal dynamics (V, alpha, q, theta)
    5. Fourth-order lateral dynamics (beta, p, r, phi)
3. Choose an aircraft and flight condition then show the dynamic modes for each of the five sets of dynamics

NOTE: There are many ways to structure your files and functions. One way of doing this is to combine the function arguments into a structure. For instance:

Define the aircraft stability derivatives as fields of the structure, such as:
air_data.C_m_q = C_m_q

Define the aircraft geometric parameters also as fields of that structure, such as:
air_data.b = b

You can put a practically unlimited number of fields into a structure, and you can also logically organize them into sub-fields, such as:
air_data.stability_derivatives.C_m_q = C_m_q

or

air_data.aircraft_geometry.b = b

If you setup your function like this:

output = compute_dynamics(air_data)

Then you can put whatever you want into air_data structure, and assign whatever you want into the output structure, such as:

output.sys_roll_mode or

output.roll_mode.A_matrix = [A]

In the next homework, we will look at using these models for digital control.

Alexander Dowell

HW2 script for Navion:

```
v = 84; % (ft/s^2)

[sys_f,sys_n] = ME_401_0007_roll_dynamics(v);
[sys_f_sp,sys_n_sp] = ME_401_0007_second_period(v);
[sys_f_dr,sys_n_dr] = ME_401_0007_dutch_roll(v);
[sys_f_long,sys_n_long] = ME_401_0007_longitudinal(v);
[sys_f_lat,sys_n_lat] = ME_401_0007_lateral(v);
roll_dynamics_pole = pole(sys_n)
sys_n
second_period_poles = pole(sys_n_sp)
sys_n_sp
dutch_roll_poles = pole(sys_n_dr)
sys_n_dr
longitudinal_poles = pole(sys_n_long)
sys_n_long
lateral_poles = pole(sys_n_lat)
sys_n_lat
```

HW2 output:

```
ME_401_0007_HW2_final

roll_dynamics_pole =

   -4.0083



sys_n =
```

Alexander Dowell

```
  A =
          x1
   x1  -4.008

  B =
          u1
   x1  -6.589

  C =
       x1
   y1   1

  D =
       u1
   y1   0

Continuous-time state-space model.

second_period_poles =

  -0.9724 + 1.4149i
  -0.9724 - 1.4149i

sys_n_sp =

  A =
```

Alexander Dowell

	x1	x2
x1	-0.9542	1
x2	-2.002	-0.9907

B =

	u1
x1	-6.409
x2	-7.746

C =

	x1	x2
y1	1	1
y2	1	1

D =

	u1
y1	0
y2	0

Continuous-time state-space model.

dutch_roll_poles =

-3.9848

0.2274

-1.5941

Alexander Dowell

sys_n_dr =

A =

	x1	x2	x3
x1	-0.1212	0	-1
x2	-3.639	-4.008	1.046
x3	-1.037	-0.5621	-1.222

B =

	u1	u2
x1	0	0
x2	-6.589	5.262
x3	-0.0511	-1.051

C =

	x1	x2	x3
y1	1	1	1
y2	1	1	1
y3	1	1	1

D =

	u1	u2
y1	0	0
y2	0	0
y3	0	0

Continuous-time state-space model.

Alexander Dowell

longitudinal_poles =

```
   0.0000 + 0.0000i
   0.0000 + 0.0000i
  -0.9724 + 1.4149i
  -0.9724 - 1.4149i
```

sys_n_long =

```
  A = 
            x1       x2       x3       x4
   x1  -0.9542        1        0        0
   x2   -2.002  -0.9907        0        0
   x3   -5.957        0        0   -32.17
   x4        0        1        0        0

  B = 
           u1
   x1  -6.409
   x2  -7.746
   x3       0
   x4       0

  C = 
       x1  x2  x3  x4
   y1   1   1   1   1
   y2   1   1   1   1
   y3   1   1   1   1
```

Alexander Dowell

```
  y4   1   1   1   1


 D =

      u1

  y1   0

  y2   0

  y3   0

  y4   0
```

Continuous-time state-space model.

```
lateral_poles =


 -4.0670 + 0.0000i

  0.1485 + 0.5543i

  0.1485 - 0.5543i

 -1.5816 + 0.0000i



sys_n_lat =


 A =

          x1       x2       x3       x4

  x1  -0.1212        0       -1    0.383

  x2   -3.639   -4.008    1.046        0

  x3   -1.037  -0.5621   -1.222        0

  x4        0        1        0        0
```

Alexander Dowell

B =

	u1	u2
x1	0	0.03374
x2	-6.589	5.262
x3	-0.0511	-1.051
x4	0	0

C =

	x1	x2	x3	x4
y1	1	1	1	1
y2	1	1	1	1
y3	1	1	1	1
y4	1	1	1	1

D =

	u1	u2
y1	0	0
y2	0	0
y3	0	0
y4	0	0

Continuous-time state-space model.