ATHENS Hackathon #2 Flight Path and Scoring

James Walker, Michael Heim, Pablo Bueno,
Bapi Surampudi, Alex Carpenter, Sidney Chocron,
Jon Cutshall, Richard Lammons, Brian Swenson,
Sydney Whittington, Joel Allardyce, Ted Bapty, Joseph Hite

11/7/22

Overall Summary of Hackathon #2

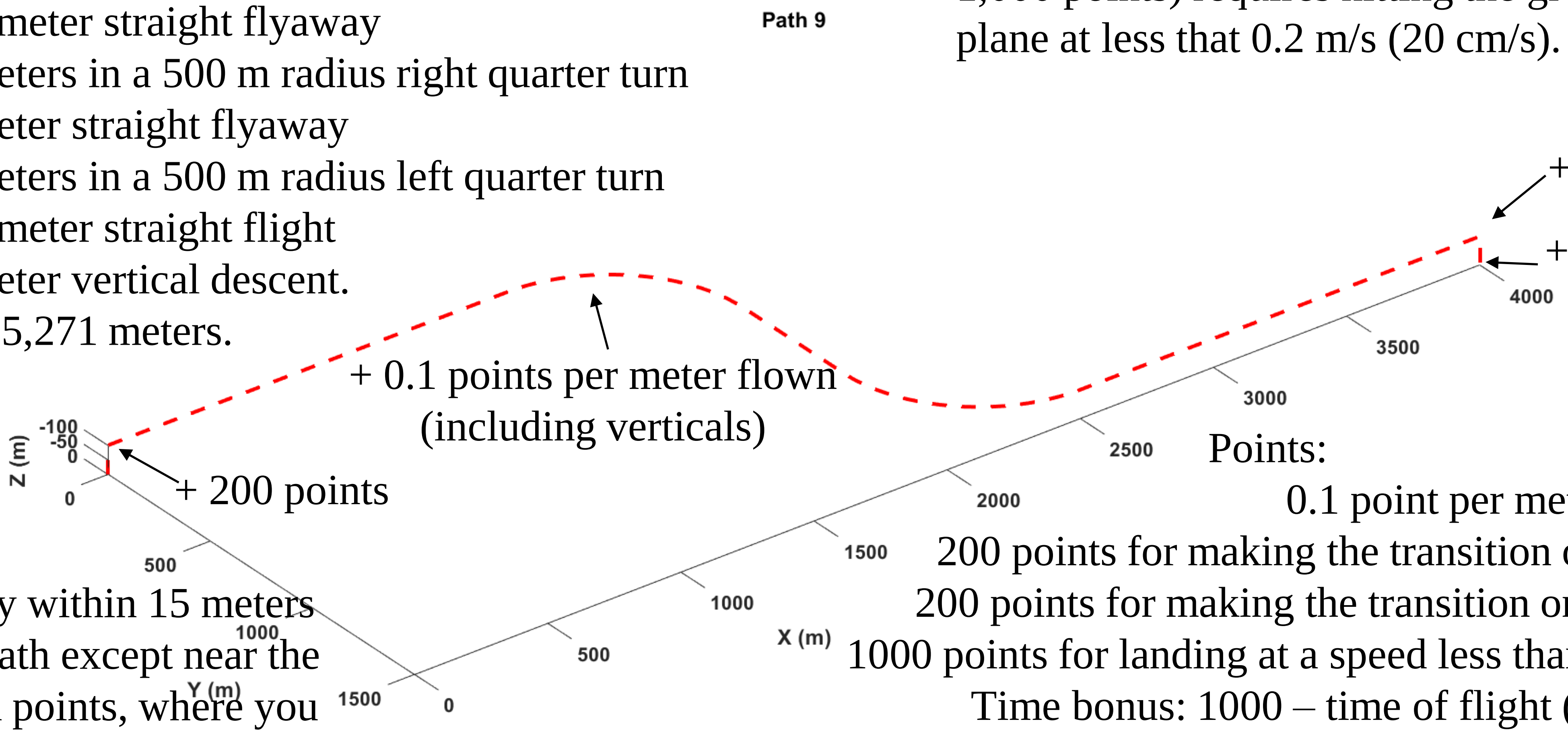
- For this Hackaton, there is only one scored flight path (9), though it needs to be traversed twice:
  - Once with the design carrying the cargo.
  - Once with the same geometric design not carrying the cargo.
    - Though the geometry cannot change, the controls can be changed for each flight.
  - The name of the flight path is Path 9 (still labeled Hackathon 1)
- As before, we are providing an autopilot.
  - Our autopilot utilizes lateral trim states, and computes them from 0 m/s to 50 m/s.
    - If there is a trim state, it also tries to find a set of controls for that trim state.
  - Based on user input, a time-parameterized flight path is created.
  - Our autopilot uses the closest-to-the-parameterized-flight-path trim state with controls to fly the vehicle.
  - It is NECESSARY to have a 1 m/s trim state. This trim state is used as the initial state and is used during the vertical ascent and decent parts of the path. Without a 1 m/s trim state, it cannot start or do these aspects of the course.
- You can provide your own autopilot, if you wish, through a provided interface.
  - If you are providing your own autopilot, you can also provide your own path parameterization.
    - Scoring is purely geometric. Timing does not matter, save for the total time to traverse the path.

Path 9 Geometry and Points from FDM (in score.out)

Path 9 has 7 segments:

1. 100 meter vertical ascent
2. 1,500 meter straight flyaway
3. 785 meters in a 500 m radius right quarter turn
4. 500 meter straight flyaway
5. 785 meters in a 500 m radius left quarter turn
6. 1,500 meter straight flight
7. 100 meter vertical descent.
8. Total: 5,271 meters.

Sticking the landing (and getting the 1,000 points) requires hitting the ground plane at less that 0.2 m/s (20 cm/s).



+ 200 points

+ 1,000 points for landing

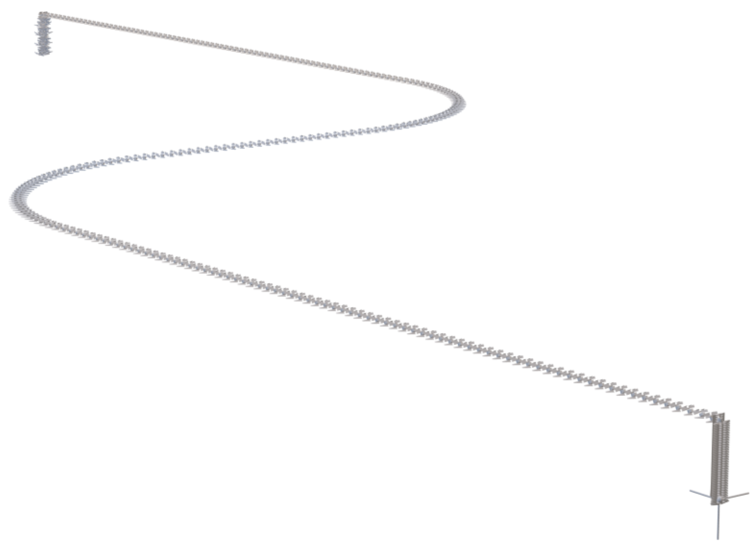
+ 0.1 points per meter flown (including verticals)

+ 200 points

You must stay within 15 meters of the flight path except near the two transition points, where you must stay within 30 meters of the flight path.

Points:
0.1 point per meter flown.
200 points for making the transition on ascent.
200 points for making the transition on descent.
1000 points for landing at a speed less than 0.2 m/s.
Time bonus: 1000 – time of flight (s) points.
Total points: 527 + 200 + 200 + 1000 + time bonus
= 1927 + time bonus

Example of Flight Path 9



Some Comments on Scoring

- You must stay within 15 meters of the flight path except near the two transition points (vertical to/from horizontal), where you must stay within 30 meters of the flight path.
  - Think of it as a bigger error sphere for the hard parts.
- The scoring algorithm, except for the bonus points based on final course time, is entirely time agnostic and only depends on position.
- You get points for any flight, even if you only fly 10 meters up (but you have to be able to this at least). This is being done so that you can improve performance based on partial success.
  - In a similar fashion you get 200 points bonus for transitioning from vertical to horizontal flight on ascent and another 200 points bonus for transition from horizontal to vertical flight at descent, to encourage your techniques to make these important transitions.
- In theory, the maximum score if you flew at the speed of light is 2927 points for each traverse of Path 9.
- We now also provide in output and Metrics.out the battery usage. This is used to assign bonus points when the two paths (with cargo and without) are combined, asking the question of whether you could go, deliver, and return with 80% battery.

Time Parameterization of Path 9 for Our Autopilot

To parameterize the time requires providing cruise speeds and some accelerations. The parameterization is either constant speed or constant acceleration. In particular, the following are inputs with examples and sign specifications:

```
control%i_flight_path = 9  (specifies Path 9)
control%requested_lateral_speed = 10 > 0 (this is 10 m/s of lateral or horizontal flight)
control%requested_vertical_speed = -1 < 0 (this is UP at 1 m/s; its negative because +Z is down)
control%requested_vertical_down_speed = 1 > 0 (this is DOWN at 1 m/s)
control%requested_lateral_acceleration = 2 > 0 (this is lateral/horizontal acceleration at 2 m/s^2)
control%requested_lateral_deceleration = -5 < 0 (this is lateral/horizontal deceleration at 5 m/s^2)
control%requested_vertical_acceleration = -5 < 0 (this is UP acceleration at 5 m/s^2)
control%requested_vertical_deceleration = 5 > 0 (this is DOWN acceleration at 5 m/s^2)
```

The following can be played with, but I probably would not. They define the final approach to the ground.

```
control%vertical_landing_approach_speed = 0.5 > 0 (target speed at ground height)
control%landing_approach_height  = 3 > 0 (this is a distance above the ground for reaching the previous line target speed; this number is positive, but it represents Z = -3 m in this case, somewhat inconsistent with other signs)
control%vertical_landing_speed_at_ground  = 0.1 > 0 (attempted landing speed at the ground plane, positive is down)
```

Output File Echoes Speeds/Accelerations and Also States Distances under Acceleration

```
Hackathon                    1
Path performance, flight path                 9
Details of path 9
   Up speed                             =    -1.000000000000000
   Down speed                           =     1.000000000000000
   Horizontal speed                     =     10.00000000000000
   Landing approach speed               =    0.5000000000000000
   Landing speed at ground              =    0.1000000000000001

   Up acceleration                      =    -5.000000000000000
   Up acceleration distance             =    0.1000000000000001
   Up deceleration                      =     5.000000000000000
   Up deceleration distance             =    0.1000000000000001

   Down acceleration distance           =    0.1000000000000001
   Down deceleration distance 1         =     7.4999999999999997E-002
   Down deceleration distance 2         =     3.000000000000000

   Horizontal acceleration distance =     25.00000000000000
   Horizontal deceleration distance =     10.00000000000000
```

Output File Includes the Time Parameterization at Various Points

Location	i	X	Y	Z	Time	Delta time	Vx	Vy	Vz	Delta v	Acce
Start	1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
End of upward acceleration	2	0.000	0.000	-0.100	0.200	0.200	0.000	0.000	-1.000	-1.000	-
End of upward cruise	3	0.000	0.000	-99.900	100.000	99.800	0.000	0.000	-1.000	0.000	
End of upward deceleration	4	0.000	0.000	-100.000	100.200	0.200	0.000	0.000	0.000	1.000	
End of lateral acceleration	5	25.000	0.000	-100.000	105.200	5.000	10.000	0.000	0.000	10.000	
Arrival at right turn	6	1500.000	0.000	-100.000	252.700	147.500	10.000	0.000	0.000	0.000	
Finish right turn	7	2000.000	500.000	-100.000	331.240	78.540	0.000	10.000	0.000	0.000	
Arrival at left turn	8	2000.000	1000.000	-100.000	381.240	50.000	0.000	10.000	0.000	0.000	
Finish left turn	9	2500.000	1500.000	-100.000	459.780	78.540	10.000	0.000	0.000	0.000	
End of cruise	10	3990.000	1500.000	-100.000	608.780	149.000	10.000	0.000	0.000	0.000	
End of lateral deceleration	11	4000.000	1500.000	-100.000	610.780	2.000	0.000	0.000	0.000	-10.000	-
End of downward acceleration	12	4000.000	1500.000	-99.900	610.980	0.200	0.000	0.000	1.000	1.000	
End of downward cruise	13	4000.000	1500.000	-3.075	707.805	96.825	0.000	0.000	1.000	0.000	
Arrival at landing approach	14	4000.000	1500.000	-3.000	707.905	0.100	0.000	0.000	0.500	-0.500	-
Landing	15	4000.000	1500.000	0.000	717.905	10.000	0.000	0.000	0.100	-0.400	-
Downward drift if needed	16	4000.000	1500.000	2.000	737.905	20.000	0.000	0.000	0.100	0.000	

- If a Delta time is negative, something is wrong, such as a sign is wrong on your input speed and/or acceleration .
  - Possibly you do not have enough distance for your accelerations, though you should get a warning printed out in that situation in the previous page output.
- FDM will try to fly anyway (and in some cases, successfully).
- Also, there is an additional column at the end of the flight information (a line is printed every second) called "itrim" which identifies the trim state the autopilot is using at that time for reference.

Additional Flight Paths, Paths 6 - 8.

- We have three additional paths in fdm that are “new.”
  - (All the old paths from the first Hackathon should still work, if you want to play with them.)
- These are not scored (for the Hackathon) but are there to potentially assist you in your design process.

Path 8 = Previous Path 9

- Path 8 is simply the previous Path 9, in case you would like to use it.
- For example, if you had a design working on the previously released Path 9, then you can run it again on Path 8 which is that same flight path.
- Path 8 (old Path 9) only differs from the Hackathon Path 9 in the straightaway distances, where in each case Path 8 (old 9) is shorter than new Path 9:
  - 100 meter vertical ascent
  - 700 meter straight flyaway * different
  - 785 meters in a 500 m radius right quarter turn
  - 300 meter straight flyaway * different
  - 785 meters in a 500 m radius left quarter turn
  - 700 meter straight flight * different
  - 100 meter vertical descent.
  - Total: 3,471 meters.

Path 7 (= Landing Path 8)

- Our experience in designing vehicles is that the hardest part is at the end, making the transition from horizontal flight to vertical flight and then landing.
- Thus, to save computational time, Path 7 inserts you onto Flight Path 8 (old Path 9) at the beginning of the lateral deceleration, and you fly the rest of the course.
  - If successful, you end up with the same score as if you had flown Path 8.
- You MUST have a trim state that corresponds to the cruise speed.
  - Notice that this is not actually required for flying Path 8 or 9, but it is for Path 7 initialization.
  - You MUST initialize control%start_trim_state = 5 (for the 5 m/s trim state, for example) in your input deck (for Paths 8 and 9 this input is not used). Otherwise input is identical to Path 8.
- Typically, for us, if you can make this final transition, then the initial ascent and transition is not a problem (except you may run out of battery for the extended flight).
  - Thus, this path allows you to focus on the final, hardest part and perhaps save three or four seconds of computational time.
  - The inputs for Path 7, 8 and 9 are all the same with the above start_trim_state exception.
    - There is only one issue: your deceleration distance must be less than 700 meters (while on Path 9 it can be 1,500 meters), but you probably do not want it to be that long anyway.

Path 6

- Path 6 is for exploring horizontal acceleration, in case your aircraft seems to be having a problem with that.
- Path 6 uses the following inputs
- <   control%i_flight_path = 6
- <   control%requested_lateral_speed = 20.d0
- <   control%requested_vertical_speed = 0.d0
- <   control%requested_lateral_acceleration = 0.1d0
- <   control%requested_lateral_deceleration = -0.1d0
- <   control%start_trim_state = 10
- The requested_lateral_speed is the final horizontal speed.
- The start_trim_state is the initial horizontal speed.
- If the final speed is greater than the initial speed, then the requested_lateral_acceleration is used as the constant acceleration.
- If the final speed is less than the initial speed, then the requested_lateral_deceleration is used as the constant acceleration.

Path 6 Example Outputs

```
Hackathon                1
Path performance, flight path            6

Maximum distance error (measured perpendicularly) from flight path was    3.03383613      m
Acceleration test path where input initial speed, final speed, and requested acceleration or deceleration determine distance and time.
   Requested initial speed (from start_trim_state):    10.0000000
   Requested final speed (from requested_lateral_speed):    20.0000000       Actual final speed =    19.9352417
   Requested acceleration (+) or deceleration (-):    0.100000001
   Expected distance:    1500.00000       Distance flown =    1499.24207
   Expected time:    100.000000       Actual time =    100.000000
   This is not a scored path, but a score of path distance/10  - 10 * error >= 0 is provided for analysis:    119.58584553951806
Final score (rounded) =    120.000000
```

```
Hackathon                1
Path performance, flight path            6

Maximum distance error (measured perpendicularly) from flight path was    22.1214542      m
Acceleration test path where input initial speed, final speed, and requested acceleration or deceleration determine distance and time.
   Requested initial speed (from start_trim_state):    20.0000000
   Requested final speed (from requested_lateral_speed):    10.0000000       Actual final speed =    14.2942257
   Requested acceleration (+) or deceleration (-):  -0.100000001
   Expected distance:    1500.00000       Distance flown =    1533.64355
   Expected time:    100.000000       Actual time =    100.000000
   This is not a scored path, but a score of path distance/10  - 10 * error >= 0 is provided for analysis:    0.0000000000000000
Final score (rounded) =    0.00000000
```

This bottom deceleration example had some erratic behavior, hence the large final speed (it was jumping around).

End of Slides