3D Obstacle Avoidance Using Modified 2D Potential Flow results

Hrishav Das

Indian Institute of Technology Madras, Indian Institute of Science

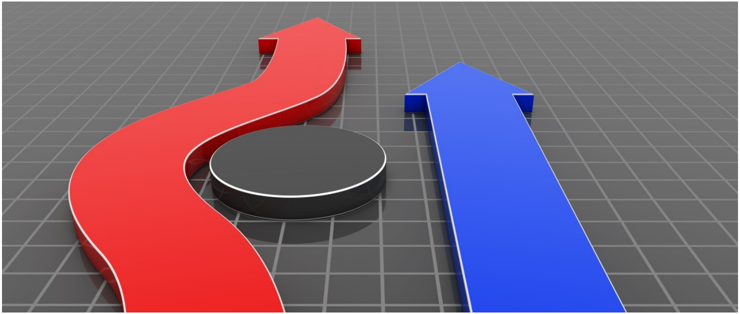
26th June, 2024

Outline

- Laying out the General Problem.
- Literature Review.
- Building on the Fluid Potential Method.
- Optimality.
- To-Do List.

Laying out the General Problem 1

- In any unmanned system, we require the plant to be able to adapt to the environment.
- The most fundamental aspect of such a system is Obstacle Avoidance.



Obstacle Avoidance in UAVs, in general, can be divided into 2 sub-tasks.

- Sensing and Detection
- Flight Path Modification

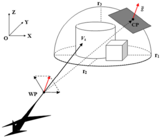


Figure 1: 3D Obstacle Problem

Sensing and Detection involve localizing the obstacle to be avoided, and the flight path modification that occurs after detection ensures that the UAV doesn't collide with the obstacle.

Laying out the General Problem 3

Problem Statement: Given a UAV intended to fly through a given set of waypoints, devise a feasible path that avoids any obstacles present in between the waypoints.
Let's look at existing solutions and methodologies that look to solve similar problems.

Literature Survey (Overview) 1

Avoidance and Path Planning methods/concepts researched in the existing literature:

Fuzzy Logic
Biologically Inspired
Neural Network
Gap Vector
Personal Space
Genetic Algorithms
Rolling Window
Velocity Obstacles
Simulated Annealing
Directive Circle
Fitting Circle
Collision Cone
Artificial Bee Colony
Potential Fields
Bayesian Occupancy Filter
RRT*
Escaping Algorithm
Ant Colony
Bacterial Foraging
A*
Voronoi Diagrams
Particle Swarm Optimization
Honey Bee Mating
Q-Learning
Sub goal-Guided Force Field
Vector Field Histogram
Generalized Complete Coverage

and MANY more....

Every single obstacle avoidance method boils down to a simple if condition applied to the path-planning problem without obstacles:

```
if (About_To_Collide)
    Do_Something_To_Avoid_Collision;
end
```

The methods differ in how the 'Do_Something_To_Avoid_Collision' is undertaken. For the sake of contextualizing my method, let's broadly divide these methods into 2 categories.

- General Obstacle Avoidance
- Fluid Inspired Obstacle Avoidance

General Obstacle Avoidance (Distance Based)

- A more rudimentary approach to obstacle avoidance. Involves ensuring distance from the UAV is kept above a certain tolerance limit (measured visually or from ultrasonic sensors).
- This method although avoids collision, does not ensure efficient path tracking.
- Usually used as a sub-part of other avoidance methods. (Ex, ensuring corridor tracking)

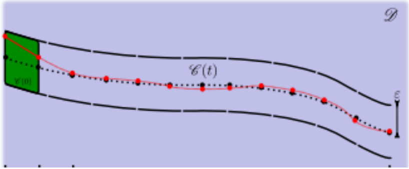
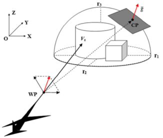


Figure 2: Corridoor Tracking

General Obstacle Avoidance (Geometry Based)

- Similar to what was discussed before. We reduce obstacles into desirable enclosing shapes and add alternate waypoints (Contact Point here) not passing through the shape.
- There are various methods involved in both aspects of this method: Deciding and building up a suitable shape, and the way of choosing the alternate point.
- We could use multiple spheres having a radius equal to the minimum turn radius of the UAV. Or make triangulation meshes of obstacles (If we want a finer picture of the obstacles).



General Obstacle Avoidance (Optimizations Based)

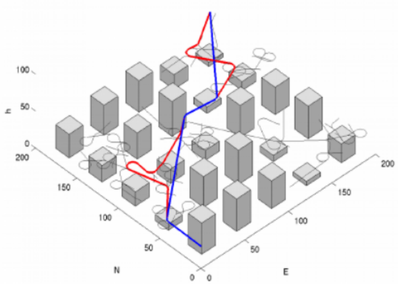
- Most avoidance/guidance algorithms will look to optimize a certain cost/reward function. Let's take an example:

$$J = \int_0^{t_f} \left(W_1 c_t^2 + W_2 h^2 + W_3 f_{TA} \right) dt \quad (1)$$

- Here W_i refers to certain weights which allow us to impart relative importance to each parameter.
- Ct denotes deviation from a desired trajectory, h denotes the height and f_{TA} denotes the probability of collision.
- If we were to minimise this function J, then we'd be looking for a flight path that closely follows some predefined trajectory, with an affliction towards flying low (maybe for reconnaissance) and a path that avoids obstacles.

General Obstacle Avoidance (Random Tree, A*, Ant Colony; Search Based)

- Usually used for limited decision space path planning. Could be extended to obstacle avoidance as well.
- One example can be considering the decision space to be turning by x degrees to the left, right or pitch up and down.
- One could iterate over a wide variety of probable paths (N steps into the future) that do not involve collision and choose an appropriate path.



General Obstacle Avoidance (Some comments)

- A distance-based method is best suited for plants with limited sensors and cannot be scaled to operate more complex objectives.
- Geometry-based method requires the obstacles to be "reasonable" and reduced to a simpler convex shape. Ideally, prior knowledge is required.
- Optimizations-based method is good for multi-objective scenarios. The price of optimality is paid in computational time.
- Search-based methods have an excellent ability to explore and solve higher dimensional spaces. However, such methods have low real-time performances, and the paths may not be perfectly trackable.

Fluid Inspired (Artificial Potential Field (APF) based) 1

- Involves specifying a virtual velocity vector field for the UAV to follow.
- Obstacles can be treated as "things to orbit around" (Lyapunov or Tangential Vector Field Guidance) or be treated as fluid sources (flow moves away from sources).

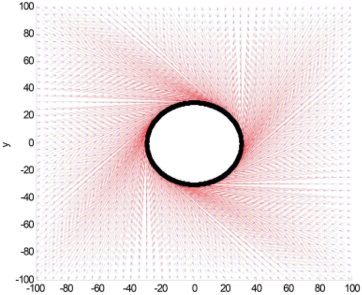


Figure 4: Tangential Vector Field

Fluid Inspired (APF based) 2

Some comments:

- APF Solutions have a simple structure.
- Good Real-Time Performance.
- Sometimes undesirable oscillations can occur near obstacles.
- Velocity Potential could have a local minimum, causing the plant to come to a halt at these points.
- There have been workarounds proposed for these drawbacks: extra virtual force, multi-point potential field, extra virtual obstacles, etc.

Fluid Inspired (Interfered Fluid Dynamic System, IFDS) 1

- Can be considered a type of APF that treats target points as fluid sinks, and each obstacle is reduced to a super ellipse.

$$\Gamma(x, y, z) = \left(\frac{x - x_0}{a}\right)^{2p} + \left(\frac{y - y_0}{b}\right)^{2q} + \left(\frac{z - z_0}{c}\right)^{2r} = 1 \tag{2}$$

- The baseline trajectory is modified to avoid obstacles through a modulation matrix M.

$$\boldsymbol{M}(x, y, z) = \boldsymbol{I} + \frac{1}{\Gamma^{\frac{1}{\rho}} \boldsymbol{n}^{\mathrm{T}} \boldsymbol{n}} \left(\boldsymbol{n}^{\mathrm{T}} \boldsymbol{n} \boldsymbol{I} - 2\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}\right) \tag{3}$$

where I is the identity matrix, $\rho \geq 0$ is a constant, $n = [\frac{\partial \Gamma}{\partial x}, \frac{\partial \Gamma}{\partial y}, \frac{\partial \Gamma}{\partial z}]^T$
- The interfered flow velocity is defined as:

$$\bar{\boldsymbol{u}}(x, y, z) = \boldsymbol{M}(x, y, z)\boldsymbol{u}(x, y, z) \tag{4}$$

Fluid Inspired (IFDS) 2

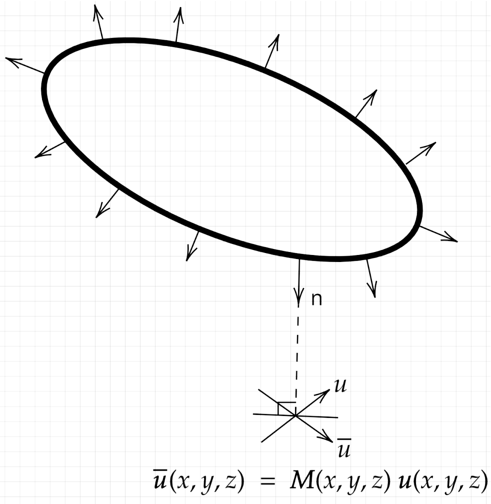
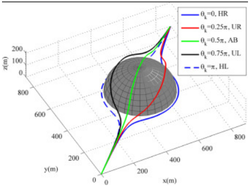


Figure 5: Visualizing the working principle of IFDS

Fluid Inspired (IFDS) 3

Many papers are present in the literature that build on the IFDS framework shown above.

- Notion of deciding an avoidance direction by adding terms to the M matrix has been discussed.



- Extensions to multi-obstacle scenarios: $M = \sum \omega_k M_k$.
- Optimizations on the direction and ρ value to ensure turning loads are within limits.

Fluid Inspired (IFDS) 4

Some comments:

- To the best of my knowledge, such methods have been used for scenarios with complete knowledge of the positions and sizes of the obstacles.
- The Optimizations preferred in existing works tend toward data-driven methods like Deep Reinforcement Learning.
- Idea of streamline hopping (i.e., dynamically varying ρ during the avoidance maneuver) has not been discussed.

Fluid Inspired (Potential Flows) 1

- This method takes motivation from Potential Flow results. Any irrotational, non-compressible, and non-viscous flow satisfies the Laplace Equation.
- Functions satisfying the Laplace equations do not suffer from the local minima problem.
- Model obstacles as sources and desired waypoints as sinks.

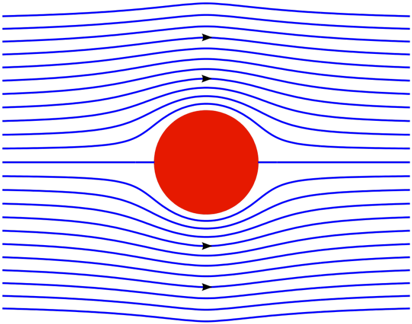


Figure 6: Potential Flow Across a cylinder

Fluid Inspired (Potential Flows) 2

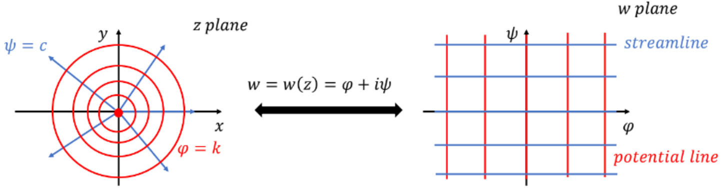


Figure 7: $\phi + i\psi$

2D Potential Flow can be represented as complex numbers in the format
$Z = \phi + i\psi(PotentialFunction + iStreamFunction)$
The locus of points satisfying a constant value of the stream function is called a streamline. The stream function gives us a family of streamlines.
One important property of streamlines of such flows is that it is a smooth, continuously differentiable curve. (Desirable for path following).

Fluid Inspired (Potential Flows) 3

Let's find the streamlines for a flow across a cylinder.

- Such a flow can be considered a superposition of 2 flows.
- One flow is a uniform velocity flow. Let us say flow from left to right.
- The other flow is due to a doublet of appropriate strength. (Analogous to a dipole in electrostatics)

Fluid Inspired (Potential Flows) 4

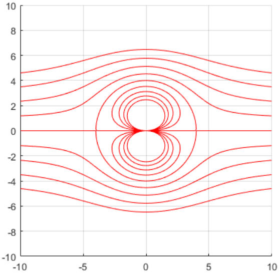
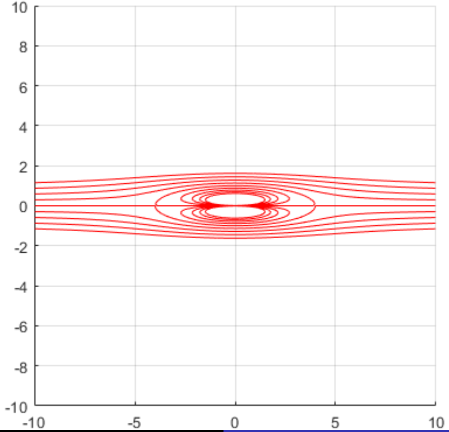


Figure 8: Flow Across a Cylinder

Fluid Inspired (Potential Flows) 5

We can warp the plane (stretch and squeeze) along any direction to get a flow around a transformed shape. For example, if we want to find out streamlines across an ellipse, all we need to do is scale the axes.



We can also find out streamlines for a flow directed at an angle. (Multiply the complex function by $\exp^{i\theta}$)

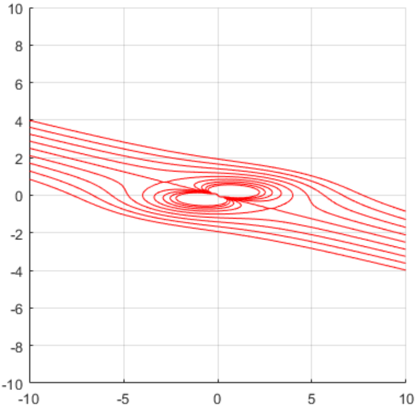


Figure 10: Flow With an angle of attack

Some comments:

- Streamlines from flows discussed here have been used to tackle 2D obstacle avoidance problems.
- This method falls short when we want to apply it in 3 dimensions. The theory of superposition is the same (Uniform Flow + Doublet Flow), but finding out streamlines in 3 dimensions is a much more involved procedure.

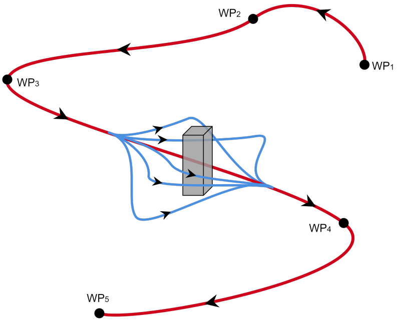
End of Literature Review (for now........)

Some gaps/novelties the following document will look to fill.

- Using 2D solutions to the Potential Flow for a 3D Obstacle Avoidance.
- Streamline Hopping, does it aid in reducing turning loads?
- Optimality of paths in non-deterministic environments.

Problem Formulation:

A UAV is tasked with going through a set of predetermined ordered way-points given by $[WP_1, WP_2, WP_3, ...]$. The minimum snap trajectory can be utilized for a completely obstacle-less environment (baseline case shown in red (11)). However, if there exist finitely sized obstacle/s in the path of the baseline trajectory, an alternate solution that ensures obstacle avoidance while not straying away from the higher level objective of goal seeking is needed (blue trajectory in (11)).



Proposition

- Upon detection of the obstacle, reduce it to a Minimum Volume Enclosing Ellipsoid (E).
- Utilize the 2D Potential Flow solution of a streamline and apply that on a plane (P) intersecting the ellipsoid.
- The intersection yields us an ellipse (E*). Streamlines across an ellipse are easily solvable.
- The cutting plane must contain the position of the UAV.
- Once the streamline equation is obtained, devise a scheduled hopping between streamlines that ensures lesser turning loads.

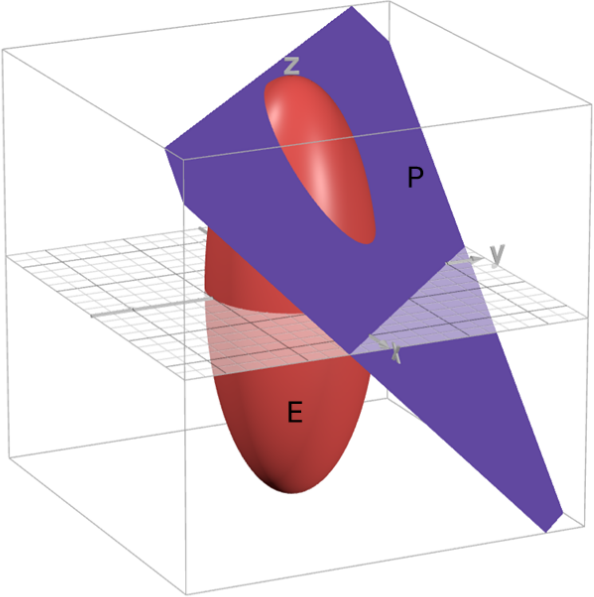
The complex potential of a uniform flow plus a doublet (for flow across a unit circle) is represented as:

$$\phi + i\psi = z + \frac{1}{z}, \text{where } z = (x + iy) \times [cos(\theta) + isin(\theta)] \tag{5}$$

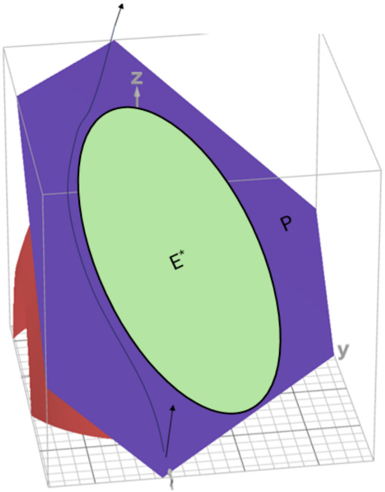
for an ellipse of semi-major and semi-minor axes r_x and r_y respectively, the stream function(ψ) is given by:

$$\left[\frac{y\cos\theta}{r_y} + \frac{x\sin\theta}{r_x}\right]\left[1 - \frac{1}{\left(\frac{x\cos\theta}{r_x} - \frac{y\sin\theta}{r_y}\right)^2 + \left(\frac{y\cos\theta}{r_y} + \frac{x\sin\theta}{r_x}\right)^2}\right] \tag{6}$$

The (PFM) in 3D (2)



The (PFM) in 3D (3)



The (PFM) in 3D (4)

Let's break down the algorithm involved here:

- Check for proximity and Collision Heading. If True, proceed to the next step; otherwise, follow the original path.
- Decide a plane to cut the ellipsoid.
- Find the center and size of the ellipse cross-section. From this, find out the orientation of the semi-major and semi-minor axes of the ellipse. A separate coordinate system is defined centered at this ellipse center, and the axes are aligned accordingly.

The (PFM) in 3D (5)

Algorithm Continued:

- From the orientation and velocity information, we obtain the angle of attack for the imagined flow in the new coordinate system.
- From the above information computed, we use the equation of the family of streamlines to choose a streamline containing the point where the UAV is located.
- We convert each streamline from the new coordinate system to the world coordinate system.
- We set this streamline as the path to be followed until the condition in point 1 becomes false.

The (PFM) in 3D (6)

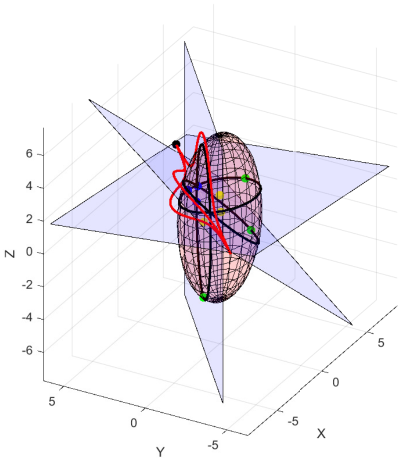
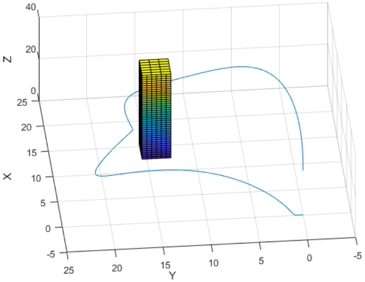
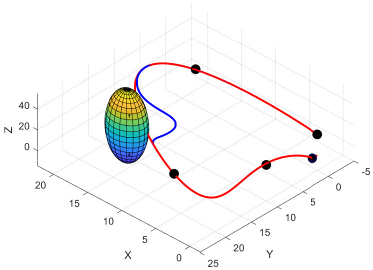


Figure 12: Desired Streamlines on various planes



(a) Sim1



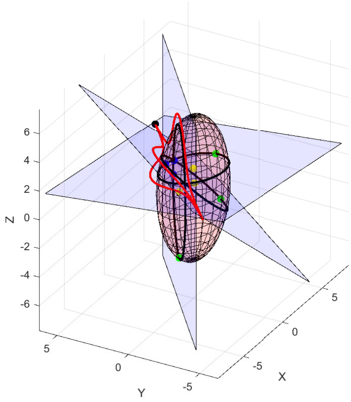
(b) Sim2

Figure 13: Prelimnary Simulation Results

The (PFM) in 3D: Optimality (8)

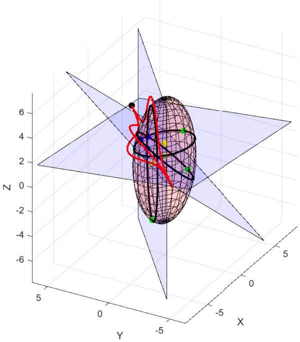
Key questions:

- What do we want to optimize? (Path length, Max Turn Rate, or some combination of these 2).
- What are the variables at play in this optimization?



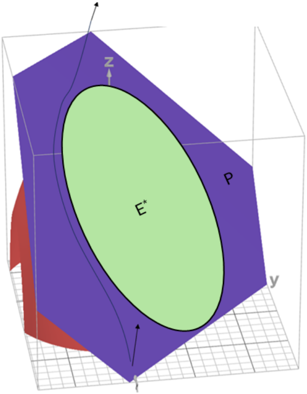
The (PFM) in 3D: Optimality (9)

Upon careful observation, it can be concluded that the path length is largely (if not solely) dictated by the choice of the plane.



The greater the path length, the more the UAV is required to deviate from the initial path, leading to higher turn loads.

The (PFM) in 3D: Optimality (10)



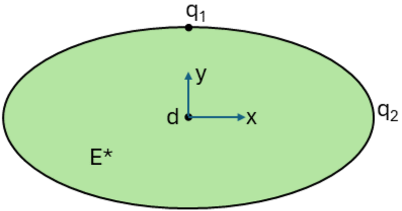
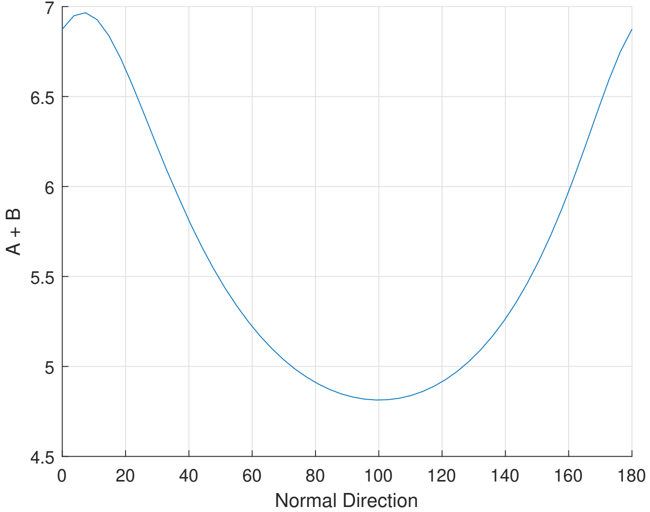


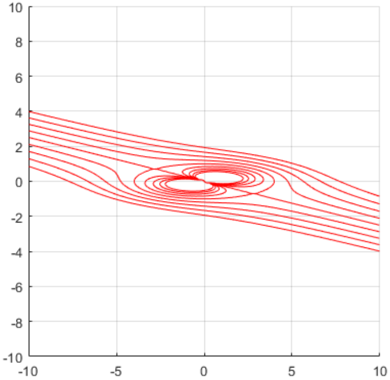
Figure 14: Visualizing E*

Sum of axes length across different plane intersections



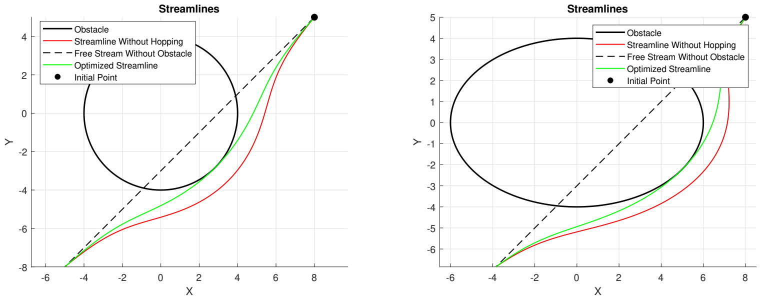
The (PFM) in 3D: Optimality (12)

Let plane P* be the plane that intersects with the ellipsoid E, which leads to the minimum path length. Let us now look at hopping between streamlines to improve turning performance further. We saw before that the streamline corresponding to $\psi = 0$ is the stagnation streamline. This streamline hugs the ellipse the tighest at its boundary. (Limiting case of the fluid particle coming to a stop at the stagnation point).



The (PFM) in 3D: Optimality (13)

As a test to see that hopping between streamlines is beneficial, we consider a scaled version of the streamline from potential flow results (Red), such that this scaled deviation (Green) is tangent to the obstacle at one point.

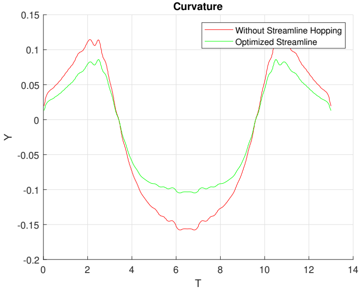


(a) Cylinder Case

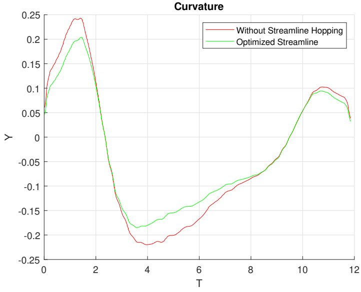
(b) Ellipse Case

Figure 16: Scaled Deviation

$$R = \frac{\left(1 + \left(\frac{dy}{dx}\right)^2\right)^{3/2}}{\left|\frac{d^2y}{dx^2}\right|}$$



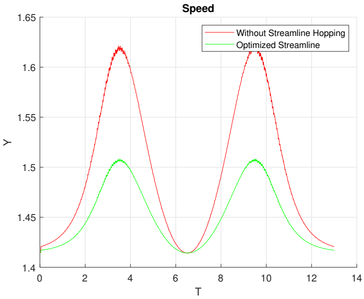
(a) Cylinder Case



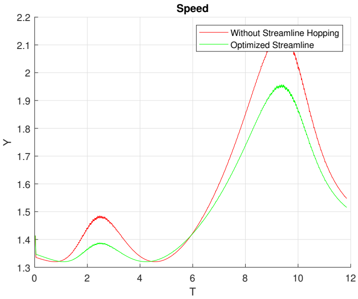
(b) Ellipse Case

Figure 17: Curvature $(\frac{1}{R})$

Assuming the UAV needs to avoid the obstacle and reach the next point at the same time as in the non-obstacle scenario.



(a) Cylinder Case

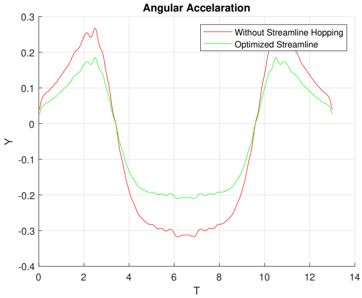


(b) Ellipse Case

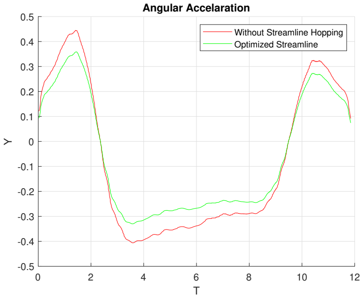
Figure 18: Speed

The (PFM) in 3D: Optimality (16)

$$\text{Angular Acceleration} = \frac{V^2}{R}$$



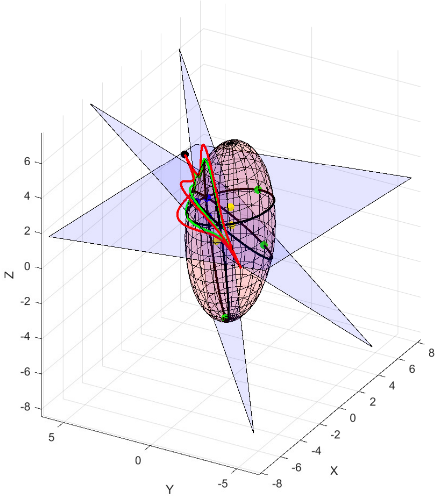
(a) Cylinder Case



(b) Ellipse Case

Figure 19: Angular Acceleration

The (PFM) in 3D: Optimality (17)

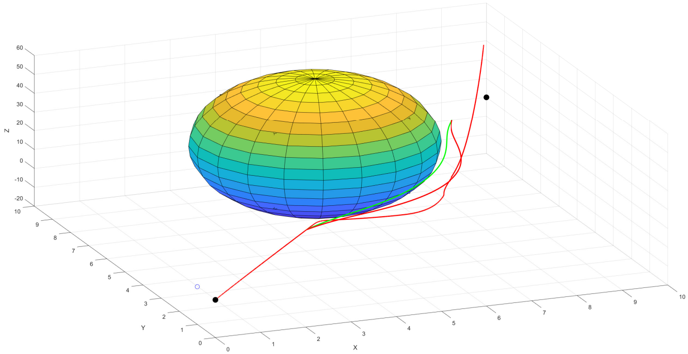


To-Do List:

- Find out an analytical formula for the best plane P*.
- Devise a streamline hopping schedule that may or may not be better than the scaled deviation.
- Simulate.

Issue

Simulating with a multirotor model was giving awry results. The issue (to the best of my knowledge) lies in the way I tuned the outer loop tracking controller.



Issue

- In the interest of time, I started building a simpler point object simulation environment to test out the trajectory design for Hopping.
- For the hopping results in the following section of the document, the simulation is that of a point object.

Hopping: Motivation (1)

Can we include the influence of the next waypoint in aiding streamline-inspired obstacle avoidance?

Hopping: Motivation (2)

- At point X, proximity conditions are met, and the path shifts to an avoidance streamline (shown in blue).
- The proposed method would generate a streamline with a virtual flow in the direction of the dotted line.

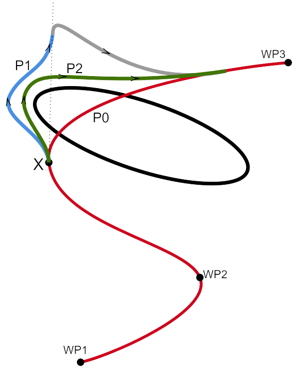
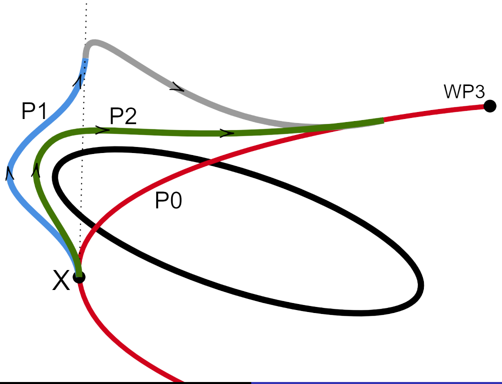


Figure 20: Motivation to use an altered version of the streamline

Hopping: Motivation (3)

- The high-level goal is to go to WP3; once the obstacle has been avoided, the blue path shifts to the grey path to revert back to the baseline trajectory.
- There could be a better way of avoiding the obstacle by using information about baseline trajectory. The green path is more desirable than the blue and grey path.



Hopping: Designing P2 (1)

Let P_2 be a linear combination of P_1 and P_0, as shown below

$$P_2 = K_0 P_0 + K_1 P_1 \tag{7}$$

Before we design the coefficients K_0, K_1, recall how the following equation defines an ellipsoid at the origin:

$$\Gamma(X) = X^{'} A X = 1$$

For any arbitrary position X:

- If X lies outside the ellipsoid, $\Gamma(X) - 1 > 0$.
- If X lies inside the ellipsoid, $\Gamma(X) - 1 < 0$.
- If X lies on the ellipsoid boundary, $\Gamma(X) - 1 = 0$.

Hopping: Designing P2 (2)

$$P_2 = K_0 P_0 + K_1 P_1$$

Desired Properties:

- When close to the obstacle, the influence of streamline trajectory P_1 should be greater. P_0 should have minimal effect here.
- When the plant is far away from the obstacle, the influence of baseline trajectory P_0 should be greater. P_1 should have minimal effect here.

Hopping: Designing P2 (3)

$$P_2 = K_0 P_0 + K_1 P_1$$

Mathematically, the desirables can be written as:

- As $\Gamma \to 1$, $K_0 \to 0$ and $K_1 \to 1$
- As Γ increases in magnitude, $K_0 \to 1$ and $K_1 \to 0$
- Trivial condition: $\Gamma \geq 1$. (Obstacle Avoidance)

Additionally, let $K_1 = 1 - K_0$

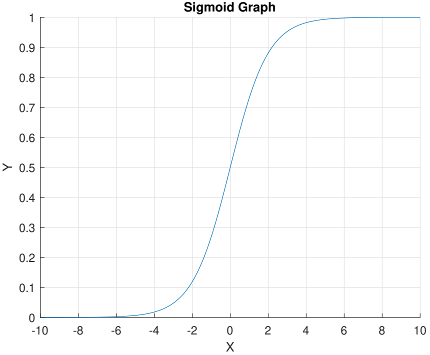
$$P_2 = K_0 P_0 + (1 - K_0) P_1 \tag{8}$$

Hopping: Designing P2 (4)

$$P_2 = K_0 P_0 + (1 - K_0) P_1$$

Probable Function of K_0: Sigmoid

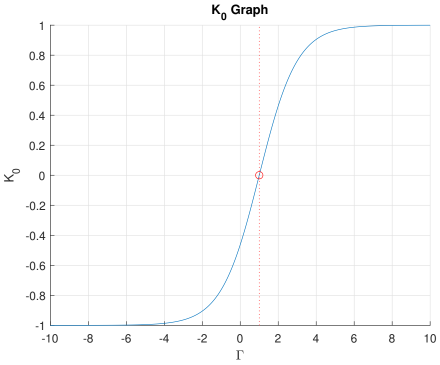
$$f(x) = \frac{1}{1 + e^{-x}}$$



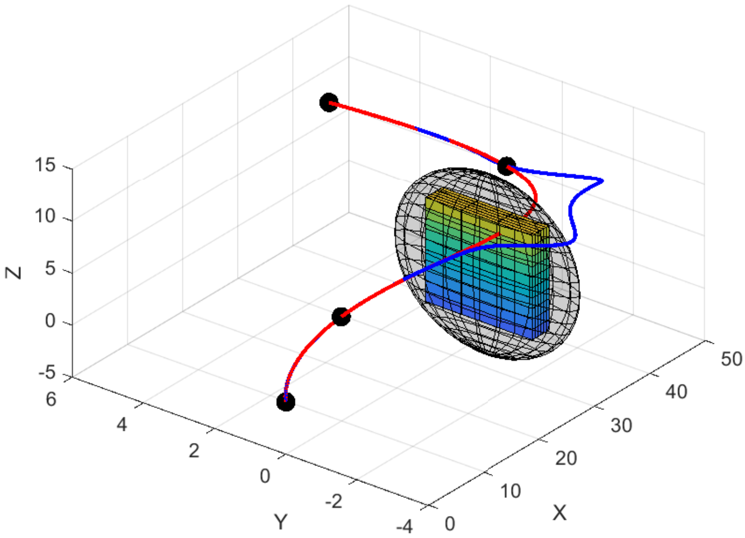
Hopping: Designing P2 (5)

$$P_2 = K_0 P_0 + (1 - K_0) P_1$$

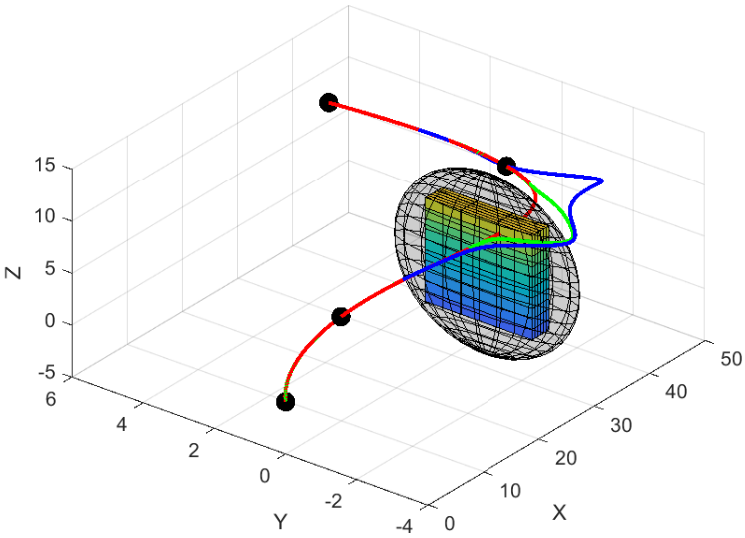
$$K_0(\Gamma) = \frac{2}{1 + e^{-(\Gamma - 1)}} - 1$$



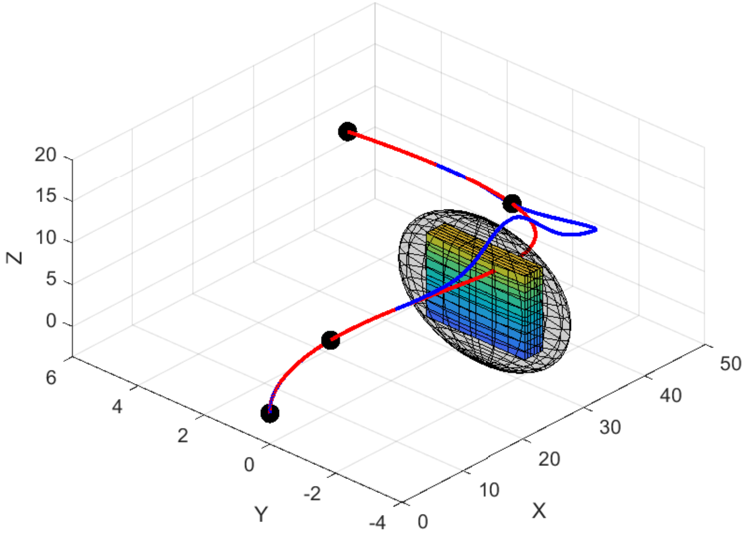
Hopping Results (1)



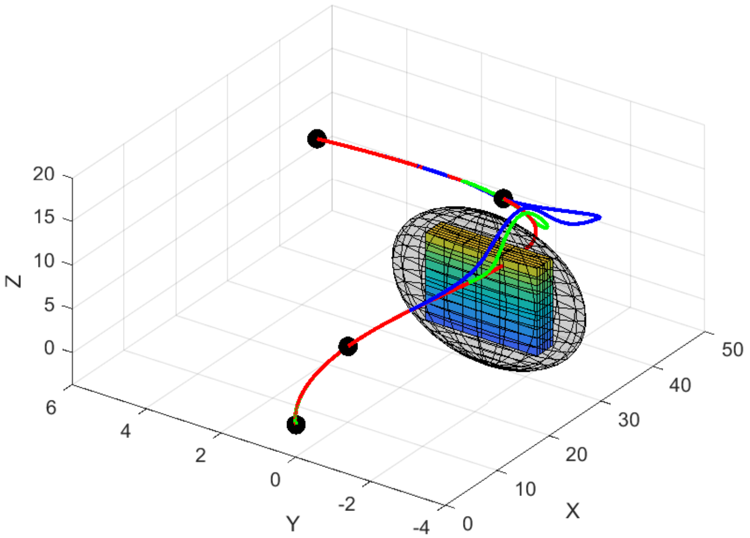
Hopping Results (2)



Hopping Results (3)



Hopping Results (4)

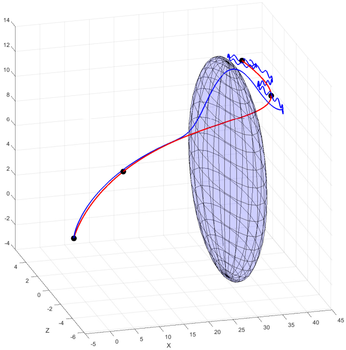


Hopping: Some Comments

- From the results presented, it is evident that the higher objective of waypoint tracking is better met with the hopping scheme proposed than without it.
- However, it can also be seen that due to the way K_0 is designed, the change in direction to avoid the obstacle occurs later compared to a pure streamline.
- This could lead to higher turning loads.
- A small modification to the hop scheduling is proposed to tackle this.

Hopping: Some Comments

When attempted with a multirotor:



It can be seen that the trajectory is trackable; the issue stated before is to do with the modeling and tuning.

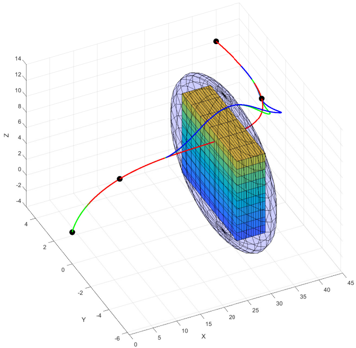
Hopping: Modification (1)

$$P_2 = K_0 P_0 + (1 - K_0) P_1$$

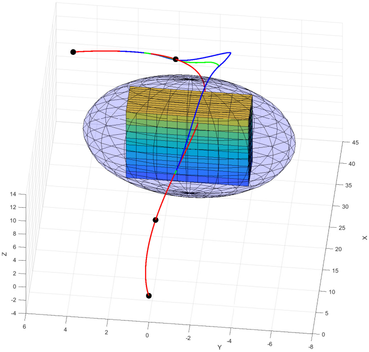
- It is proposed that K_0 is to be fixed at 0 till the time Γ doesn't start rising.
- This is another way of saying, follow the streamline till you have "half-cleared" the obstacle. (Related to how Γ varies).
- While Γ is rising:

$$K_0(\Gamma) = \frac{2}{1 + e^{-(\Gamma - 1)}} - 1$$

Hopping: Modification (2)



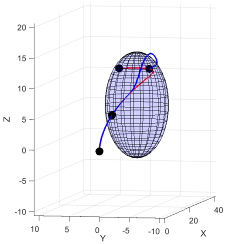
(a) View 1



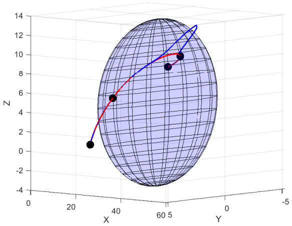
(b) View 2

Figure 21: With Modification

After lots of effort.....



(a) Vertical Avoidance



(b) Lateral Avoidance

Figure 22: Multirotor Simulations

Looking back at conclusions from Literature Review

Going back to when a few gaps in existing literature were identified:
Some gaps/novelties the following document will look to fill.

- Using 2D solutions to the Potential Flow for a 3D Obstacle Avoidance.
- Streamline Hopping, does it aid in reducing turning loads?
- Optimality of paths in non-deterministic environments.

Comments:

- The first objective was cleared a while back when the plane of avoidance idea was formulated and simulated for simple scenarios.
- Moreover, the method proposed has been simulated for a scenario where the obstacle isn't known to the drone initially. Hence, setting itself apart from previous works of IFDS and other related papers.
- As shown in the latest simulations, adding the gain scheduled hopping provides the drone with a smoother transition out of and back into the baseline trajectory. The turning angles required can be met and tracked.

Future Prospects:

- Since the drone is not aware of the obstacle until it comes close enough to it, extending the scenario to multi-objective scenarios will not be too much of a jump.
- The hopping function K0 can be an interesting avenue to explore. For example, we can also use a sinusoidal function (smooth jump from 0 to 1). We can also use different functions at different stages of avoidance (a sort of hysteresis function).
- More robust simulations where the detection model is more involved (like a real-world data collection scenario from Lidar or camera images), incorporating effects of noisy sensors/feedback.
- Real-world testing.

Thank you

A big thank you to everyone involved at AVL for hosting me for 2 Internships