Laboratory E: Vector field navigation and obstacle avoidance

Assistants: Julien Lecoeur and Ludovic Daler

11.11.2014

INTRODUCTION

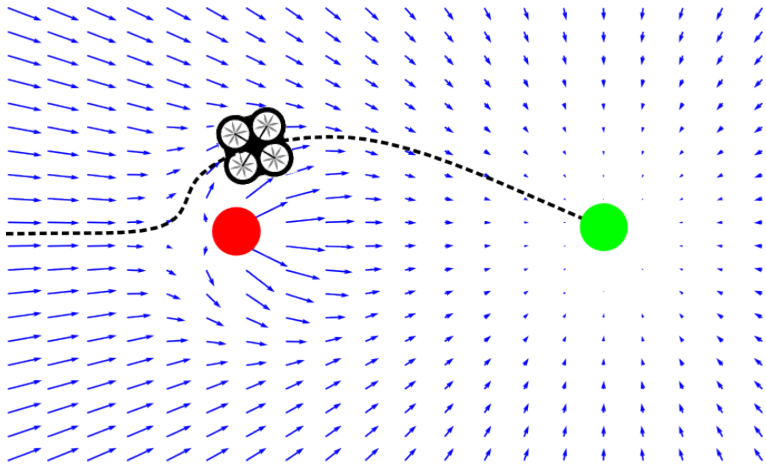
The goal of the laboratory is to implement and experiment with a method of path planning and obstacle avoidance: vector fields. Assuming an 3D environment with known geometry, vector fields will be used to simultaneously guide the quadcopter to a goal and avoid collisions with obstacles. The content of the laboratory will address the following aspects:

- Implement a vector field to control the altitude of the quadcopter.
- Derive vector fields to fly the quadcopter towards a point, away from obstacles and along a circular 3D path.
- Experiment with different potential functions and observe their efficiency for collision avoidance.

PRINCIPLE

During this laboratory, different objects (attractors, repulsors or circular paths) will be created using the waypoint interface of QGroundcontrol (cf. laboratory D), and sent to the quadcopter. In a real scenario, attractors could be the destination indicated by the human operator, and repulsors could be obstacles detected by the robot itself.
Each object creates a field of vectors that can be interpreted as velocity commands by the quadcopter. For example, for every location of the quadcopter, an attractor object will create a 3D velocity vector pointing towards it, with a norm depending on the distance between the quadcopter and the object.

If more than one object is present in the environment, the quadcopter will follow the sum of all the corresponding velocity vectors. This strategy allows the quadcopter to plan a collision-free path towards a goal without requiring the operator to design manually the flight path.



Implementation-wise, we will use the velocity controller developped during the previous sessions. The goal will be to fill the functions computing the vector fields for each kind of object.

Exercise 1: Floor avoidance

Description

As a starting point, we will create a vector field pushing the quadcopter away from the ground.

For a position (x, y, z) of the quadrotor, this field is defined as follow:

- It acts only on the altitude. The vectors of this field will all be vertical and pointing towards the sky (remember that we use the NED coordinate system):

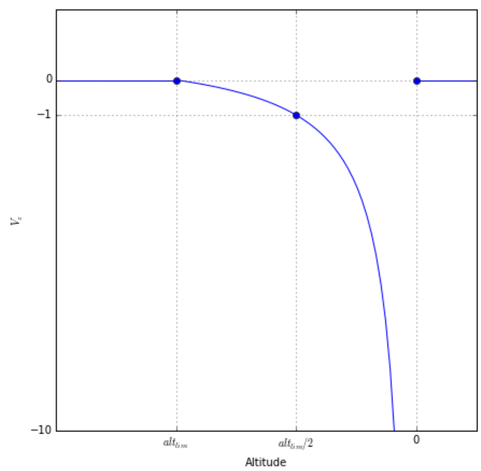
$$\overrightarrow{V}(x, y, z) = [0, 0, V_z(z)]$$

- It is active only when the quad is close to the ground:

$$\overrightarrow{V}(x, y, z \leq alt_{lim}) = [0, 0, 0]$$

- It makes it impossible for the quadcopter to reach the ground:

$$||V_z(z)|| \xrightarrow[z \to 0]{} +\infty$$



QUESTIONS

1. Design a function that fits the requirements above.
2. Open the file **vector_field_waypoint.c** and complete the function **vector_field_floor**
3. Compile and flash the autopilot

To test your code, set the quadcopter in **AUTO** mode (cf. previous labs and reminder at the end of this document). You should see the quadcopter stabilizing its altitude at 20 meters above ground level (ie. absolute altitude of 420 meters).

EXERCISE 2: ATTRACTIVE VECTOR FIELD

DESCRIPTION

In this exercise, we will implement a vector field for attractive objects. An attractive objects is described by its 3D position and an attractiveness coefficient.
With $A(x, y, z)$ the position of the MAV, $B(x_B, y_b, z_B)$ the position of the object, and k the attractiveness coefficient. The vector field is defined as follow:

$$\overrightarrow{V}(x, y, z) = f(k, ||\overrightarrow{AB}||).\frac{\overrightarrow{AB}}{||\overrightarrow{AB}||}$$

Where $\frac{\overrightarrow{AB}}{||\overrightarrow{AB}||}$ is the unit vector pointing towards the attractive object, and $f(k, ||\overrightarrow{AB}||)$ is the desired speed. The simplest definition of f is $f(k, d) = k * d$.

QUESTIONS

1. Open the file **vector_field_waypoint.c** and complete the function **vector_field_attractor**
   a) Compute $\frac{\overrightarrow{AB}}{||\overrightarrow{AB}||}$ and store it in the variable "direction"
   b) Compute $f(k, ||\overrightarrow{AB}||)$ and store it in the variable "speed"
2. Compile and flash the autopilot
3. Add a waypoint on Qgroundcontrol
   a) Set the MAV in **AUTO** mode, it takes off to an altitude of 20m because of what you implemented in Exercise 1.
   b) Go to the tab **Mission** of Qgroundcontrol
   c) In the subwindow called **Mission plan** (lower part of the window), select the first tab (Edit Waypoints). We will use this interface to send the location of the attractor object to the quadcopter.
   d) Delete any existing Waypoint ("-" button on the right)

e) Double-click on the map to add a waypoint. Use drag&drop to change its position.

4. Set the parameters of the waypoint. In the **Mission plan** subwindow, select:
   - The coordinate frame: **Global/Abs Alt.**
   - The type of waypoint: **Other**
   - The associated command: **CMD 22**
   - The attractiveness: **P1** (0.1 is a good start)
   - The altitude: **P7** (430m is a good start)
5. Send the waypoint to the MAV (button **Set** in the lower right corner)
6. Observe how the MAV flies towards the attractor object.

   a) Try to increase the attractiveness, you should observe oscillations. How could you prevent it from happening?

   b) Create one or two additional waypoints with **CMD 22**, what do you observe? Try to set a lower attractiveness to one of the waypoints and observe where the quadcopter stabilizes.

EXERCISE 3: REPULSIVE VECTOR FIELD

DESCRIPTION

In this exercise, we will implement a vector field for repulsive objects with two shapes: cylinders and spheres. Like attractive objects, a repulsive object is described by its 3D position and repulsiveness coefficient. We also add a safety radius (which is the minimum distance the quadcopter should keep from the object), and a range of action (which is the maximum distance at which the object influences the quadcopter).
With $A(x, y, z)$ the position of the MAV, $B(x_B, y_B, z_B)$ the position of the object, k the repulsiveness coefficient, d_{min} the minimum radius, and d_{max} the range of action. The vector field is defined as follow:

$$\overrightarrow{V}(x, y, z) = f(k, d_{min}, d_{max}, d).\frac{\overrightarrow{BA}}{||\overrightarrow{BA}||}$$

Where $\frac{\overrightarrow{BA}}{||\overrightarrow{BA}||}$ is the unit vector pointing away from the repulsive object, d is the distance between the quadcopter and the center of the object, and $f(k, d_{min}, dmax, d)$ is the desired speed.
For the definition of f we will use:

$$f(k, d_{min}, d_{max}, d) = \begin{cases} \frac{k.(d_{max}-d)}{d-d_{min}}, & \text{if } d_{min} < d < d_{max} \\ 0, & \text{if } d \leq d_{min} \text{ or } d \geq d_{max} \end{cases}$$

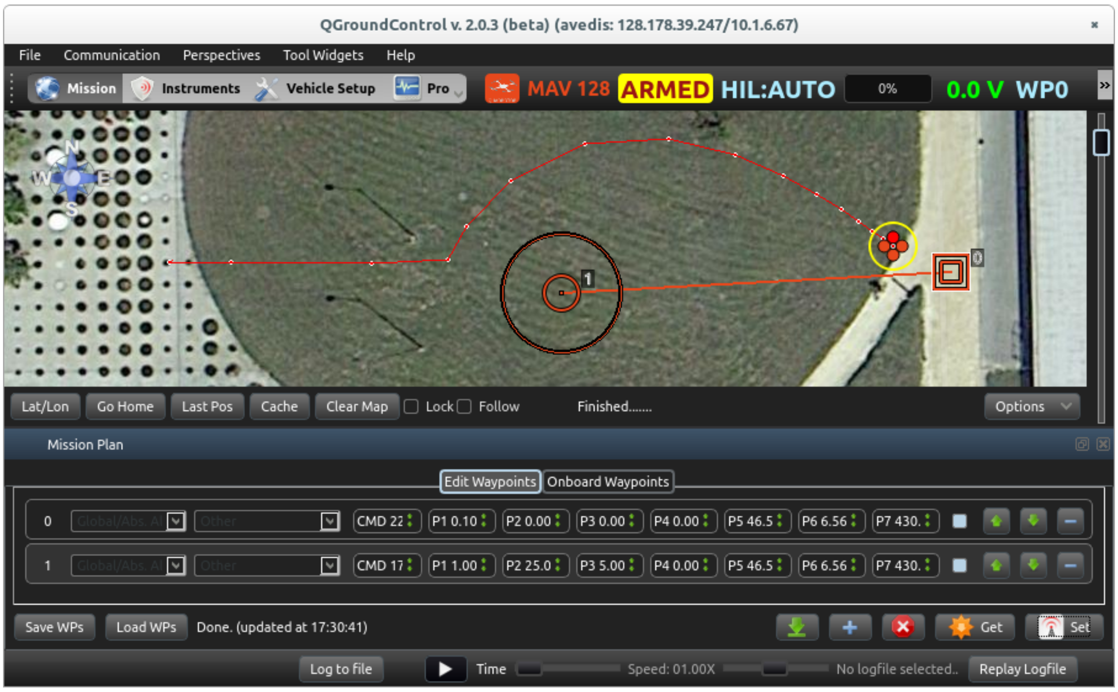
Questions

1. What is the distance d between a MAV flying on the point $A(x, y, z)$ and
   a) the center of a sphere $B(x_B, y_B, z_B)$?
   b) the main axis of a vertical cylinder passing by the point $B(x_B, y_B, z_B)$?

2. In **vector_field_waypoint.c**, complete the functions **vector_field_repulsor_cylinder** and **vector_field_repulsor_sphere**

3. In Qgroundcontrol, proceed as in exercise 2 in order to add new waypoints. In the **Mission plan** subwindow, select:
   - The coordinate frame: **Global/Abs Alt.**
   - The type of waypoint: **Other**
   - The associated command: **CMD 17** for a cylinder, **CMD 18** for a sphere,
   - The repulsiveness: **P1** (1.0 is a good start)
   - The range of action d_{max}: **P2** (25m is a good start)
   - The safety radius d_{min}: **P3** (5m is a good start)
   - The altitude: **P7** (430m is a good start)

4. Reproduce what is shown in the figure bellow with one attractive object and one repulsive cylinder. Observe how the quadcopter modifies its trajectory to avoid a collision with the repulsive object.



5. Modify the radii, attractiveness and repulsiveness of the objects. Observe their relative effect.

6. Replace the cylinder (**CMD 17**) by a sphere (**CMD 18**), how does the trajectory change?

7. **BONUS QUESTION:**

   a) Using 1 attractor and 3 repulsive cylinders, design a world in which the quadcopter is blocked in a local minimum and cannot reach the attractor.

   b) Local minimum is a major issue with vector field navigation, cite two path planning strategies seen during the course that would not fail in the previous world.

Exercise 4 (optional): Circular vector field

Description

In this bonus exercise, we will implement a vector field for circle following. Like attractive objects, this field will drive the quadcopter towards a 3D location according to an attractiveness coefficient.
Contrary to the previous types of objects, the quadcopter will not be attracted towards the center of the circle, but rather towards the closest point of the circle.
Finally, a tangential component will drive the quadcopter around the circle at a defined cruise speed.

Questions

1. What is the distance between the quadcopter flying on the point $A(x, y, z)$ and a circle of radius r and centered on $B(x_B, y_B, z_B)$?

2. In **vector_field_waypoint.c**, complete the function **vector_field_circular_waypoint**

3. In QGroundcontrol, add a circular waypoint with the following parameters:
   - The coordinate frame: **Global/Abs Alt.**
   - The type of waypoint: **Other**
   - The associated command: **CMD 16**
   - The attractiveness: **P1** (0.1 is a good start)
   - The cruise speed: **P2** (1.5 is a good start)
   - The radius r: **P3** (30m is a good start)

4. Observe the trajectory of the quadcopter

5. Add repulsive objects along the circular path.

Reminder

To switch on the joystick:

1. Close QGroundControl,
2. Connect the joystick to an USB slot of your computer,
3. Open QGroundControl,
4. Click on “File”, “Settings”,
5. Click on the “Controllers” tab,
6. Click on “Enable controllers”,
7. Map the joystick axis with the respective control axis:
   a) Moving left and right is the roll axis (right is positive),
   b) Moving front and back is the pitch axis (back is positive),
   c) Rotating the joystick along a vertical axis is the yaw axis (right is positive),
   d) The rotating stick is the throttle (up is positive).
8. For the “Mad Catz V.1 Stick”, the order is the folowing:
   a) Axis 0: Roll,
   b) Axis 1: Pitch,
   c) Axis 2: Throttle (inverted),
   d) Axis 3: Yaw,
9. Click on “Close”.