A3.Motion Planning and Control of Unmanned Aerial Vehicles

Vilhelm Dinevik and Paula Carbó

***Abstract*—The core of this project focuses on how to make aerial vehicles fly autonomously from an initial position to a goal. This is done by making a mathematical model for the UAV, a brief study of the sensors needed to estimate the UAVs state, then designing an LQR controller for the trajectory tracking and finally using an artificial potential field function for the navigation. The mathematical model is done by studying the kinematics and dynamics for a single UAV, it is then linearised and the systems observability and controllability are checked to develop the LQR. We conduct computer simulation to test the theoretical findings and evalue the proposed methods. Finally, we conclude the paper with some discussion and provide future directions and ideas to do further research on the topic.**

I. Introduction

UNMANNED aerial vehicles, also known as UAVs, are becoming more and more popular nowadays because they are small, cheap to produce, have low operating and maintenance cost, great maneuverability, can perform steady flight operations and are able to enter high-risk areas without having to compromise human safety. Most applications that involve UAVs have been used in open areas without any obstacles and with a human in control of the UAV. But in recent years people have come up with more modern applications of UAVs that will need them to fly autonomously in densely populated areas, with a lot of other autonomous vehicles around, e.g. Amazon Prime Air delivery system, AltiGator drones services for inspection and data adquisition, or multi-UAVs used to deploy an aerial communications network [1]. This places high demands on UAVs obstacle avoidance capabilities for both moving and static obstacles.

There are many different UAV manufacturers and a vast amount of different models, all with different motors, weights, sensors and lift-to-weight ratio. To make a standard autonomous flight applicable to all these kinds of UAVs, a simple and easy-to-implement multi-UAV mathematical model, that will still be able to avoid obstacles with as few sensors as possible, is needed.

There is plenty of previous work that has focused on modelling and controlling a single UAV in great detail [2], on simulating high-fidelity multi-UAV scenarios [3] and on finding the most precise sensing techniques to create indoor and outdoor capability [4]. Based on all this previous research, the core of this project is to develop a control method that can make multiple quadrotor UAVs fly safely to their goals in an environment filled with obstacles. To develop this simple but complete and comprehensive solution, this report aims to study the mathematical model for a single vehicle, the sensors placed in the UAV that provide all the information of its state, a linearisation-based tracking controler and a navigation controler based on potential fields.

In section II the development of the mathematical model for the quad is explained. Section III analyses different sensors on board the UAV and how they could be used to measure the state of the UAV accurately. In section IV, the tracking controller of the UAV is studied. In section V, the navigation method is explained and described. The simulations performed and the results obteined are explained in section VI. Finally, in the last section of this paper, the results are explained and discussed. Future works and ideas derived from the work done on this project are also mentioned.

II. Quadcopter modelling

A. Overview

The UAV is a rigid body quad copter, with a cross-shaped body and four electrical propellers. Front and rear rotors rotate in a clockwise direction, while right and left rotors rotate in a counter clockwise direction (see Fig.1). Its motion has 6 degrees of freedom but there are only 4 propellers, therefore the system is under actuated.

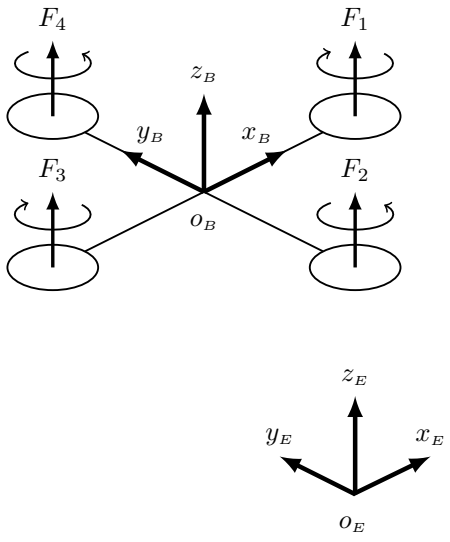

Fig. 1. Quadrotor with propellers and the two reference frames.

B. Kinematics

In order to describe the motion of the UAV, a kinematic model for was developed. Two right-hand reference frames

are defined: the Earth frame and the body frame, as can be seen in Fig. 1.

The Earth frame is static, while the body frame is attached to the UAV. In this case, the axis origin o_B coincides with the quadrotor's center of mass.

The generalized position $\boldsymbol{\xi}$ contains the inertial position and Euler-angle orientation, and is described in the Earth frame, as in (1). The linear position $\boldsymbol{x}^E$ of the UAV is the vector between the origin of the Earth frame o_E and the origin of the body frame o_B, and the Euler angles $\boldsymbol{\eta}^E$ are defined as stated in Fig. 2.

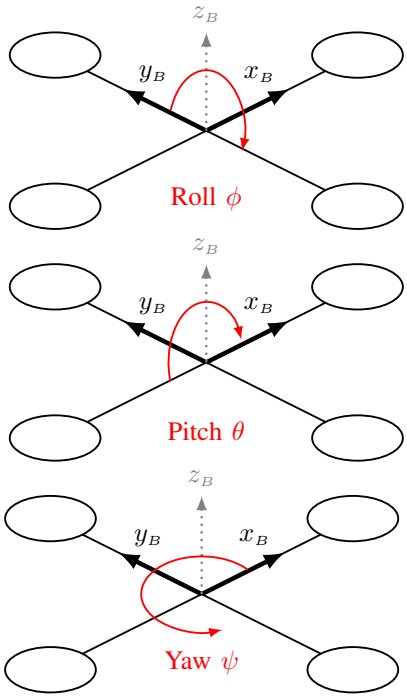

Fig. 2. Euler angles ϕ, θ, ψ.

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{x}^E & \boldsymbol{\eta}^E \end{bmatrix}^T = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T \quad (1)$$

The generalized velocity $\boldsymbol{\nu}$ (2) contains the linear and angular velocity, and it is expressed in the body frame.

$$\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{v}^B & \boldsymbol{\omega}^B \end{bmatrix}^T = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \quad (2)$$

Three rotation matrixes around each of the x, y, z axes, that correspond to the Euler-angle orientation, can be defined according to (3, 4, 5) respectively.

$$\boldsymbol{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (3)$$

$$\boldsymbol{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (4)$$

$$\boldsymbol{R}_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

The complete rotation matrix $\boldsymbol{R}_\Theta$, that expresses the orientation from the body frame to the Earth frame, can be obtained by multiplying these three matrices, as in (6).

$$\boldsymbol{R}_\Theta(\phi, \theta, \psi) = \boldsymbol{R}_x(\phi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_z(\psi) \quad (6)$$

The transfer matrix $\boldsymbol{T}_\Theta$ that allows to change between the angular velocity in the body frame $\boldsymbol{\omega}^B$ and the Euler rates in the Earth frame $\dot{\boldsymbol{\eta}}^E$ can be determined and is as shown in (7).

$$\boldsymbol{T}_\Theta(\phi, \theta) = \begin{bmatrix} 1 & \sin(\phi)\cdot\tan(\theta) & \cos(\phi)\cdot\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \quad (7)$$

A generalized matrix $\boldsymbol{J}_\Theta$ can be built joining the rotation and the transfer matrix (6, 7), as shown in (8).

$$\boldsymbol{J}_\Theta(\phi, \theta, \psi) = \begin{bmatrix} \boldsymbol{R}_\Theta & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{T}_\Theta \end{bmatrix} \quad (8)$$

Where the notation $\boldsymbol{0}_{3\times3}$ means a matrix filled with zeros with a 3×3 dimension.

In order to relate the derivative of the generalized position in the Earth frame with the generalized velocity on the body frame, the generalized matrix (8) can be used as seen in (9), and that is the final model of the quadrotor's kinematics [5], [6].

$$\dot{\boldsymbol{\xi}} = \boldsymbol{J}_\Theta\ \boldsymbol{\nu} \quad (9)$$

C. Dynamics

The dynamic model for the UAV relates the acceleration of the vehicle with the forces and torques acting on the quadrotor. The Newton-Euler formulation allows to express these variables in the body frame, as in equations (10) and (11), as clearly stated by Bresciani in [6].

$$\boldsymbol{F}^B = m(\dot{\boldsymbol{v}}^B + \boldsymbol{\omega}^B \times \boldsymbol{v}^B) \quad (10)$$

$$\boldsymbol{\tau}^B = \boldsymbol{I}\ \dot{\boldsymbol{\omega}}^B + \boldsymbol{\omega}^B \times (\boldsymbol{I}\ \boldsymbol{\omega}^B) \quad (11)$$

III. Sensor Modelling and Feedback

The core of this project focuses on how to make aerial vehicles fly autonomously from an initial position to a goal. Therefore, apart from the main algorithm that makes this possible, it is important that the vehicle can acquire accurate information about its condition and its surroundings. Sensors do not only have to provide information about the state of the UAV as to close the loop for the controller, but also provide information about the objects the vehicle may encounter throughout its path, as to make the navigation safe and prevent collisions. The sensors analised that make this possible are Inertial Measurement Units (IMU), GPS devices, infrared and ultrasonic sensors.

A. Inertial Measurement Unit

This module is in charge of measuring almost all the variables related to the motion of the vehicle. Usually inside this module a 3-axis accelerometer, gyroscope and magnetometer can be found. The most precise IMUs integrate specially designed sensors and sometimes include a GPS, a RS232 transceiver and a processor, that runs a real-time Kalman filter in order to provide the most accurate data directly to the CPU.

1) Triple axis accelerometer: This sensor measures proper acceleration along the three axes on the body frame. It has to be placed so its three axes match the axes established for the body frame. It can measure dynamic acceleration as a result of the motion of the drone. As shown in (12), the rotation matrix is used to change from acceleration provided by the IMU to the acceleration in the Earth frame [7].

$$\boldsymbol{a}_{\mathrm{IMU}} = \boldsymbol{R}_{\Theta}^{T}\ (\ddot{\boldsymbol{x}}^{E} - g\ \vec{\boldsymbol{z}}_{\boldsymbol{E}}) \tag{12}$$

2) Triple axis gyroscope: This device can measure angular rates in its three axes. Therefore, it gives the angular velocity of the body frame relative to the Earth frame, expressed in the body frame (13).

$$\boldsymbol{\omega}_{\mathrm{IMU}} = \boldsymbol{\omega}^{B} \tag{13}$$

3) Triple axis magnetometer: This kind of sensors are able to measure the ambient magnetic field. Ideally this corresponds to the Earth's magnetic field, therefore the orientation of the vehicle can be measured (14).

$$\boldsymbol{m}_{\mathrm{IMU}} = \boldsymbol{R}_{\Theta}^{T}\ \boldsymbol{m}_{\mathrm{Earth}} \tag{14}$$

Where $\boldsymbol{m}_{\mathrm{Earth}}$ corresponds to the Earth's magnetic field expressed in the Earth frame. This measurement can be accurate if the bias caused by the local magnetic disturbance $\boldsymbol{b}_{\mathrm{m}}$ is taken into account (15) and the sensor is placed as far as possible from the elements that may cause this disturbance onboard the UAV, such as the wires that power the rotors [7].

$$\boldsymbol{m}_{\mathrm{IMU}} = \boldsymbol{R}_{\Theta}^{T}\ \boldsymbol{m}_{\mathrm{Earth}} + \boldsymbol{b}_{\mathrm{m}} \tag{15}$$

In all the specified sensors, bias and noise are also present. Gyroscopes are usually robust against this noise. But once placed in an UAV, accelerometers are affected by the vibration, and need filtering for its measurements to be considered reliable.

B. GPS receiver

This device is basically a receiver that makes use of the satellite-based Global Positioning System to calculate the vehicle's geographical position (longitude and latitude) thanks to a 24 satellite constellation around Earth and with trilateration. Casual and inexpensive GPS devices have some meters of accuracy, therefore either better GPS devices or suplementary information from other sensors are needed in order to estimate the position of the vehicle as accurately as possible. For example, motion tracking via smart cameras together with Simultaneous Localization and Mapping solver techniques [7]. GPS may also not function indoors, so its usefulness is limited.

C. Infrared sensors

In order to sense the UAV's immediate surroundings, a device that is able to know if there is any obstacle around and its relative position to the drone is needed. An array of active infrared sensors correctly placed on the quadrotor is a good solution for this application. An IR sensor consists basically of a LED acting as an emitter and a photo detector acting as a receiver. Both need to have a peak in the same wavelength for optimal power radiation in the emitter and sensitivity in the receiver. The LED emits a light beam in the infrared range ($700\,\mathrm{nm}$ to $1\,\mathrm{mm}$ wavelength), and when the beam finds an obstacle, it is reflected. The receptor is a Position-Sensitive Device that is able to detect the angle of the received beam, and therefore the device is able to detect the distance to the obstacle thanks to triangulation, as can be seen in Fig. 3.

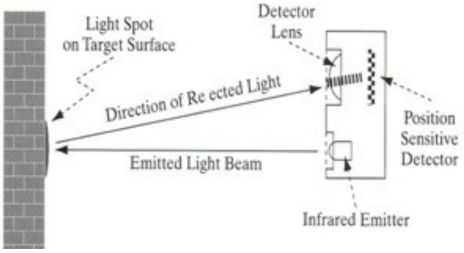

Fig. 3. Infrared obstacle detection diagram.

Since the light beam needs to be reflected by an object, its reflectance is an important factor to take into account, since poor reflective objects could not be detected on time. Also, some other natural or artificial sources of radiation such as the Sun may cause interferences. To improve the circuit's response to interferences the signal must be properly conditioned and modulated [6], [8].

D. Ultrasonic sensors

This device, together with the IR sensors, allow measuring the distance from the vehicle to an obstacle. An ultrasonic sensor consists of a high-frequency sound emitter and a receiver. Both are electrical signals – sound wave transducers, and their operation is similar to the IR sensors: the emitted wave is reflected by the obstacle, and when received, the distance to the obstacle can be calculated based on the time of flight (TOF) of the signal in the air, as can be seen in Fig. 4. Since the velocity of the sound in the air is known, just by knowing the time that passed between emission and reception, the distance to the obstacle can be known, according to (16).

$$d_{\mathrm{obstacle}} = \mathrm{v}_{\mathrm{sound,air}}\ \frac{TOF}{2} \tag{16}$$

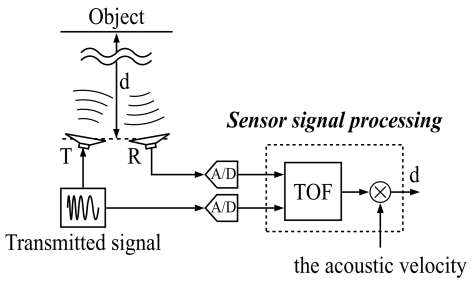

Fig. 4. Ultrasonic obstacle detection diagram, obtained from [9].

These sensors may also be used to measure the altitude of the UAV, that can be combined with a barometer to know both the relative and absolute altitude. According to Adarsh in [10], both IR and ultrasonic sensors usually have high correlation between the measured values, except for some specific materials. Both types have proven to be accurate when performing further processing techniques of the acquired data.

IV. Trajectory tracking

In order for us to be able to make the UAV follow a given path we have to implement a controller into the UAV system controll. The individual UAV systems in this project could be described with a block diagram as the one pictured in Fig.5. Where $\boldsymbol{r}$ is our reference signal and $\boldsymbol{u}$ is the output from the

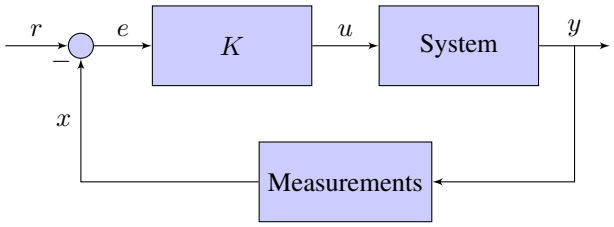

Fig. 5. block diagram for individual quadrotor

controller $\boldsymbol{K}$ defined as

$$\boldsymbol{u} = \begin{bmatrix} f_{\mathrm{t}} & \tau_{\mathrm{x}} & \tau_{\mathrm{y}} & \tau_{\mathrm{z}} \end{bmatrix}^T \tag{17}$$

We linearise the mathematical modell and write it on the linear state space form according to [5] seen in equations (18) and (19), to do optimal control with LQR.

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{18}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} \tag{19}$$

In order to linearise the mathematical model we define a $\hat{x}(t)$ and $\hat{u}(t)$ to be the desired trajectory, we then define $\Delta x(t)$ and a $\Delta u(t)$ as

$$\begin{cases} \Delta x(t) = x(t) - \hat{x}(t) \\ \Delta u(t) = u(t) - \hat{u}(t) \end{cases} \tag{20}$$

This gives us the linearised state-space model form [5]

$$\Delta \dot{x}(t) = A(t)\Delta x(t) + B(t)\Delta u(t) \tag{21}$$

$$\Delta y = C\Delta x \tag{22}$$

We then estimate $\Delta x(t)$ and $\Delta u(t)$ using a first order taylor approximation around an equilibrium point. With the model rewritten according to 18 and 19 we verify that all the states of the system are both observable and controllable. In order to check the observability and controllability of the system we have to design two matrices, $\mathcal{O}$ and $\mathcal{C}$ defined as following.

$$\mathcal{O} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{C}\boldsymbol{A} \\ \boldsymbol{C}\boldsymbol{A}^2 \\ \boldsymbol{C}\boldsymbol{A}^3 \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^{11} \end{bmatrix} \in \mathbb{R} \tag{23}$$

$$\mathcal{C} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{B}\boldsymbol{A} & \boldsymbol{B}\boldsymbol{A}^2 & \boldsymbol{B}\boldsymbol{A}^3 & \cdots & \boldsymbol{B}\boldsymbol{A}^{11} \end{bmatrix} \in \mathbb{R} \tag{24}$$

The system is controllable and observable if $\mathcal{C}$ and $\mathcal{O}$ has full rank.

The controller used in this section is as mentioned above an LQR, An LQR utilizes a cost function and minimises said cost function to optimise the controller [11]. To design the LQR we use some of the constants in the state-space equations to express the Q matrix in the following way.

$$\boldsymbol{Q} = \boldsymbol{C}^T\boldsymbol{C}\alpha_1 \tag{25}$$

where α_1 is a scaling factor for our $\boldsymbol{Q}$ matrix. the $\boldsymbol{Q}$ matrix and the identity matrix $\boldsymbol{R}$ is then implemented into the following cost function J.

$$J = \int_0^\infty (\boldsymbol{x}^T(t)\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}\boldsymbol{u}(t))dt \tag{26}$$

J is then minimised to obtain the optimal LQR controller $\boldsymbol{K}$ with the given α_1. This gives us the linear state feedback

$$\boldsymbol{u(t)} = -\boldsymbol{K}[\boldsymbol{x}(t) - \boldsymbol{r}(t)] \tag{27}$$

where $\boldsymbol{K}$ can be written as

$$\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S} \tag{28}$$

$\boldsymbol{S}$ is as stated in [6] the solution of the Riccati's algebraic equation:

$$\boldsymbol{S}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{S} - \boldsymbol{S}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S} + \boldsymbol{C}^T\boldsymbol{Q}\boldsymbol{C} = 0 \tag{29}$$

The motivation behind using a LQR is that it usually has small steady-state errors[12]. It would be reasonable to focus on getting the deviations from the optimal track to be as small as possible, hence minimising the error.

V. Multi-UAV Navigation with Collision Avoidance

This section studies aims to study one method to find the optimal path to follow from an initial position to a goal. In the case presented in this paper, we consider multiple UAVs flying towards their goals, in an environment filled with obstacles. Some proposed graph-search methods to solve this problem are Rapidly-exploring Random Tree (RRT) algorithms or A* search algorithms and other Dijkstra

extensions. Nonetheless, these techniques do not perform very well when the test environment is constantly changing. These methods excel at finding the appropriate path to follow in a labyrinth-like environment. However, if the environment is different in every iteration, these algorithms are inefficient since the UAV may end up following an inefficient path [13], [14].

Another way of dealing with this specific problem, which is considered in this work, is based on artificial potential fields. We choose to use this method because it is a simple, movement-efficient algorithm that can navigate the UAV to its goal using a short path. It provides a closed form solution, therefore it is computationally efficient. With this technique a high computational speed and an optimised path can be achieved, with simple and elegant calculations[15]. The main objective is to provide an efficient solution for multi-UAV navigation to points of interest while avoiding collisions with each other and the obstacles in the environment.

Artificial potential fields consist of the sum of an attractive and a repulsive potential. In this case the goal generates the attractive potential and the obstacles generate repulsive potentials. This will determine the motion of the UAV by following the negative gradient of the potential field.

In our case we have a random number of vehicles $v_{\mathrm{i}} \ \forall \ i \ \in \ \{1, ...M\}$ (each with its respective goal $g_{\mathrm{i}} \ \forall \ i \in \{1, ...M\}$) and also a random number of obstacles $o_{\mathrm{j}} \ \forall \ j \in \{1, ...N\}$, all of them considered as spheres. Each UAV has a radius of $r_i = r$, since we consider all the UAVs to have the same size, and each obstacle of R_j. We can define the position of one UAV in space by $\boldsymbol{q}_i$, of a goal by $\boldsymbol{q}_{goal,i}$ and of an obstacle by $\boldsymbol{q}_{obst,j}$.

The distances from the center of a UAV to a goal or obstacle are calculated with the Euclidean norm, according to (30) and (31) respectively. For the distance to an obstacle, the radius of the object is taken into account, either if it is an obstacle or another UAV ()

$$\rho_{\mathrm{goal,i}}(\boldsymbol{q}_{\mathrm{i}}) = \| \ \boldsymbol{q}_{\mathrm{i}} - \boldsymbol{q}_{\mathrm{goal,i}} \ \| \tag{30}$$

$$\rho_{\mathrm{obst,i,j}}(\boldsymbol{q}_{\mathrm{i}}) = \| \ \boldsymbol{q}_{\mathrm{i}} - \boldsymbol{q}_{\mathrm{obst,j}} \ \| - a \tag{31}$$

$$a \in \{R_j, \ r \mid \forall j\} \tag{32}$$

The repulsive and the attractive potentials are calculated according to [16], and are described by equations (33) and (34) respectively.

$$U_{\mathrm{att,i}}(\boldsymbol{q}_{\mathrm{i}}) = \begin{cases} \frac{1}{2}\xi_1 \rho^2_{\mathrm{goal,i}}(\boldsymbol{q}_{\mathrm{i}}) & \text{if } \rho_{\mathrm{goal,i}}(\boldsymbol{q}_{\mathrm{i}}) \leq d \\ d\xi_2 \rho_{\mathrm{goal,i}}(\boldsymbol{q}_{\mathrm{i}}) & \text{if } \rho_{\mathrm{goal,i}}(\boldsymbol{q}_{\mathrm{i}}) > d \end{cases} \tag{33}$$

$$U_{\mathrm{rep,i,j}}(\boldsymbol{q}_{\mathrm{i}}) = \begin{cases} \frac{1}{2}\eta(\frac{1}{\rho_{\mathrm{obst,i,j}}(\boldsymbol{q}_{\mathrm{i}})} - \frac{1}{\rho_0}) & \text{if } \rho_{\mathrm{obst,i,j}}(\boldsymbol{q}_{\mathrm{i}}) \leq \rho_0 \\ 0 & \text{if } \rho_{\mathrm{obst,i,j}}(\boldsymbol{q}_{\mathrm{i}}) > \rho_0 \end{cases} \tag{34}$$

The d parameter is the distance from the goal where the attractive function changes from a conic to a parabolic well configuration. The constant ρ_0 is the sensing radius of the UAV. Finally, ξ_1, ξ_2 and η are positive scaling factors that allow potential adjustment.

The potential fields' desired force on the UAV is calculated by (35), since the negative gradient of a potential field $-\boldsymbol{\nabla} U$ is a vector that points in the direction of steepest descent.

$$\vec{F}(\boldsymbol{q}) = -\boldsymbol{\nabla} U(\boldsymbol{q}) \tag{35}$$

This leads to (36) and (37), that show how these repulsive and attractive forces are calculated.

$$\vec{F}_{\mathrm{att,i}}(\boldsymbol{q}_{\mathrm{i}}) = \begin{cases} -\xi_1(\boldsymbol{q}_{\mathrm{i}} - \boldsymbol{q}_{\mathrm{goal,i}}) & \text{if } \rho_{\mathrm{goal,i}}(\boldsymbol{q}_{\mathrm{i}}) \leq d \\ -d\xi_2 \frac{(\boldsymbol{q}_{\mathrm{i}} - \boldsymbol{q}_{\mathrm{goal,i}})}{\|\boldsymbol{q}_{\mathrm{i}} - \boldsymbol{q}_{\mathrm{goal,i}}\|} & \text{if } \rho_{\mathrm{goal,i}}(\boldsymbol{q}_{\mathrm{i}}) > d \end{cases} \tag{36}$$

$$\vec{F}_{\mathrm{rep,i,j}}(\boldsymbol{q}_{\mathrm{i}}) = \begin{cases} \eta(\frac{1}{\rho_{\mathrm{obst,i,j}}(\boldsymbol{q}_{\mathrm{i}})} - \frac{1}{\rho_0})(\frac{1}{\rho^2_{\mathrm{obst,i,j}}(\boldsymbol{q}_{\mathrm{i}})}) \frac{(\boldsymbol{q}_{\mathrm{i}} - \boldsymbol{q}_{\mathrm{obst,i,j}})}{\|\boldsymbol{q}_{\mathrm{i}} - \boldsymbol{q}_{\mathrm{obst,i,j}}\|} & \text{if } \rho_{\mathrm{obst,i,j}}(\boldsymbol{q}_{\mathrm{i}}) \leq \rho_0 \\ 0 & \text{if } \rho_{\mathrm{obst,i,j}}(\boldsymbol{q}_{\mathrm{i}}) > \rho_0 \end{cases} \tag{37}$$

As mentioned before, there are two reasons for splitting the attractive potential and force functions into two parts. In the case of a parabolic configuration for all the space, there is a linear dependence of the force with the distance between UAV and goal, hence the force grows indefinitely when far from the goal. At the same time, in the case of a conic configuration for all the space, there is a singular point in the goal that could make the UAV oscilate around it. The solution is to combine both configurations to solve this problems, by making the force constant when far from the goal with a conic configuration, and then using a parabolic function when reaching the goal to avoid the singularity.

When an UAV is moving, it also considers other vehicles as obstacles. We can define another set of obstacles $v'_{\mathrm{k}} \ \forall \ k \in \{1, ...M\}$, that correspond to all the vehicles but considered as obstacles.

Finally, the repulsive and attractive potentials and forces are added, according to (38) and (39), in order to obtain the total value.

$$U_{\mathrm{total,i}}(\boldsymbol{q}_{\mathrm{i}}) = U_{\mathrm{att,i}}(\boldsymbol{q}_{\mathrm{i}}) + \sum_{\forall j} U_{\mathrm{rep,i,j}}(\boldsymbol{q}_{\mathrm{i}}) + \sum_{\forall k \neq i} U_{\mathrm{rep,i,k}}(\boldsymbol{q}_{\mathrm{i}}) \tag{38}$$

$$\vec{F}_{\mathrm{total,i}}(\boldsymbol{q}_{\mathrm{i}}) = \vec{F}_{\mathrm{att,i}}(\boldsymbol{q}_{\mathrm{i}}) + \sum_{\forall j} \vec{F}_{\mathrm{rep,i,j}}(\boldsymbol{q}_{\mathrm{i}}) + \sum_{\forall k \neq i} \vec{F}_{\mathrm{rep,i,k}}(\boldsymbol{q}_{\mathrm{i}}) \tag{39}$$

To simplify the simulations, first order dynamics are assumed for the quadcopter model. This means that the control signal $\boldsymbol{u}(t)$ can be estimated as the velocity obtained from the negative gradient of the potential, as can be seen in (40).

$$\boldsymbol{u}_{\mathrm{i}}(t) = \dot{\boldsymbol{p}}_{\mathrm{i}}(t) = -\boldsymbol{\nabla} U_{\mathrm{total,i}}(\boldsymbol{q}_{\mathrm{i}}(t)) \tag{40}$$

The potential fields method, nevertheless, has some clear limitations. The most important problem to solve is the local

minima situation, where a vehicle can get stuck and therefore never arriving to its goal. To exemplify this case, we consider a case with a vehicle v_1 perceiving its goal g_1 and only one obstacle o_1. We can rewrite (39) for this example as (41).

$$\vec{F}_{\text{total},1}(\boldsymbol{q}_1) = \vec{F}_{\text{att},1}(\boldsymbol{q}_1) + \vec{F}_{\text{rep},1,1}(\boldsymbol{q}_1) \tag{41}$$

Since the attractive and repulsive forces, determined by (36) and (37), have different signs, there is a certain configuration where they cancel each other. This means that the gradient of the potential field is zero (42) and that there is a local minimum in the potential function. Therefore the vehicle will not move from such a point since its force vector will be zero.

$$\begin{aligned} \vec{F}_{\text{att},1}(\boldsymbol{q}_1) &= -\vec{F}_{\text{rep},1,1}(\boldsymbol{q}_1) \\ \vec{F}_{\text{total},1}(\boldsymbol{q}_1) &= 0 \\ \boldsymbol{\nabla} U_{\text{total},1}(\boldsymbol{q}_1) &= 0 \end{aligned} \tag{42}$$

There are some other problems, like the difficulty of passing between closely-spaced obstacles or inherent oscilations in the trajectory when near obstacles, as stated by Koren and Borenstein in [17].

VI. Simulation and Results

All the necessary formulas described in this report have been implemented in MatLab to test their validity in the context. We have tested both the LQR controller developed in section IV and the multi-UAV navigation controller studied in section V.

For the trajectory tracking we constructed the linear state-space equations of the mathematical model in a MatLab script using the initial starting position as the equilibrium point. The equations described in section IV for controllability and observability was then used to check that the system was observable and controllable. We then used MatLab tools to design an LQR with R as an identity matrix and the scaling factor for Q, α set to 50. We then analysed the step responses of the system which produced the results shown in figures Fig.6, Fig.7, Fig.8 and Fig.9 where one can see that the steady state error is zero for all step responses.

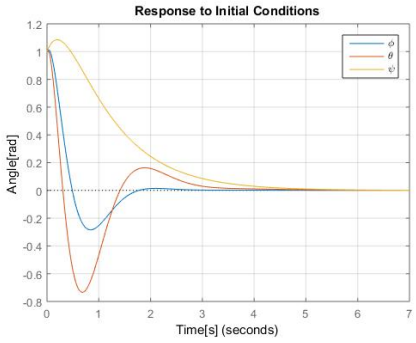

Fig. 6. Step response for the Euler angles 1 rad from desired position.

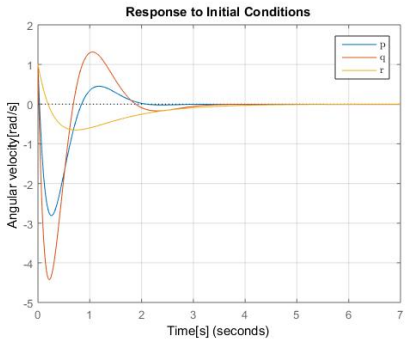

Fig. 7. Step response for angular velocities 1 rad/s from desired angular velocity.

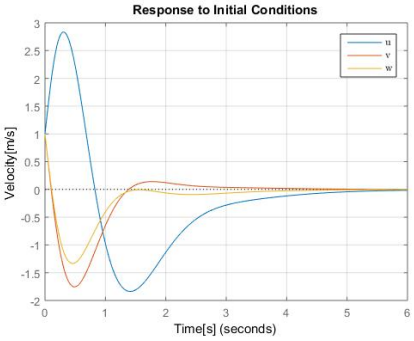

Fig. 8. Step response for velocities 1 m/s from desired velocity.

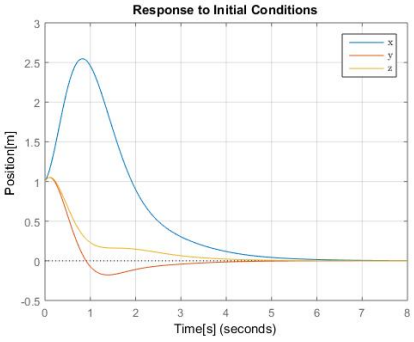

Fig. 9. Step response for positions 1 meter from desired position.

For the navigation part we developed a script to test the guidance of our potential field function by simulating a random number of UAVs in the range $[5, 7]$ and obstacles in the range $[50, 100]$. The initial and goal configurations for the UAVs, and the center of the obstacles are random within a $30 \times 30 \times 30\ \text{m}^3$ cube. Obstacles can not appear too close to the quadcopters or their goals.

Obstacles have a random radius between $0.3\,\text{m}$ and $2\,\text{m}$. Each UAV has the same radius of $25\,\text{cm}$ and the same radius of the sphere of influence of $2\,\text{m}$. These values are inspired by real-life situations, in the case we considered a relatively large UAV. For the sphere of influence, the most

common IR and ultrasonic sensors, placed in the adequate places, allow sensing up to $2\,\mathrm{m}$ without any problem. For the scaling factors, we used $\xi_1 = 1$, $\xi_2 = 0.3$ and $\eta = 1$. The value of ξ_1 is assigned so the UAVs, when far from the goal, have a desired velocity of around $0.5\,\frac{\mathrm{m}}{\mathrm{s}}$. Both the attractive and the repulsive potentials have the same scaling by establishing η with the same value, as to prevent crashes. When closer to the goal, the scaling factor is lower to avoid oscillation around the goal. The distance to change from conic to parabolic well is set to $d = 1\,\mathrm{m}$ so the velocity of the UAV is restricted most part of its flight, to guarantee safety.

The algorithm works by calculating, for each vehicle, the distance to the goal and if an obstacle is close enough to be detected. The potential function is calculated consequently. Then as stated in (40), the drone advances in space with a set discrete time step of $0.1\,\mathrm{s}$, to assure that the UAVs do not move for a long period of time without checking again the state and the environment.

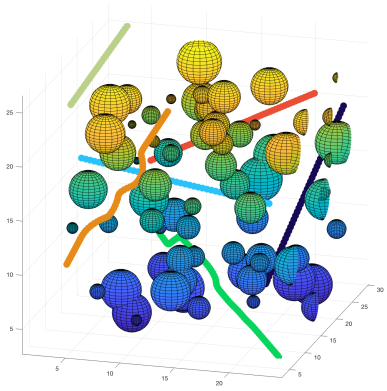

Fig. 10. Random case environment from first perspective for the potential fields simulation.

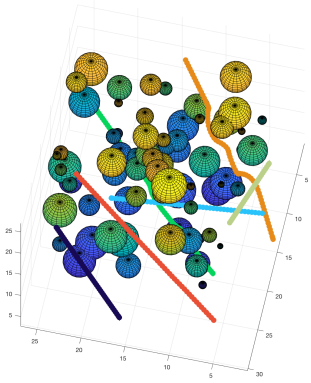

Fig. 11. Random case environment from second perspective.

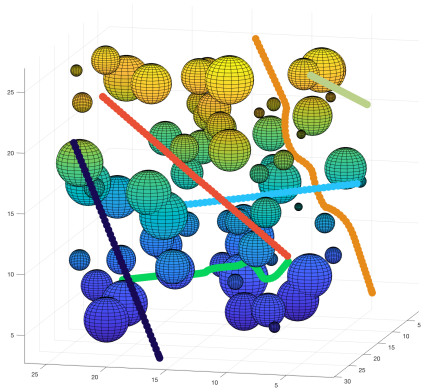

Fig. 12. Random case environment from third perspective.

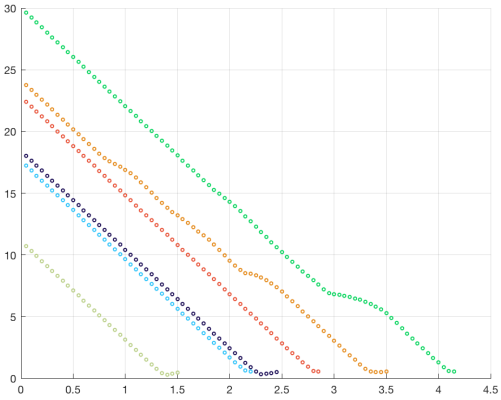

Fig. 13. Random case environment - Distance from vehicles to their respective goal.

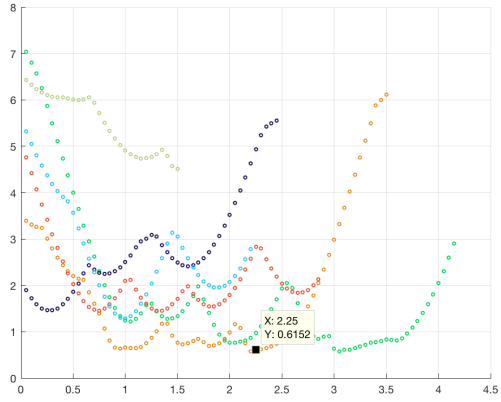

Fig. 14. Random case environment - Distance from vehicles' surface to any other obstacle's surface.

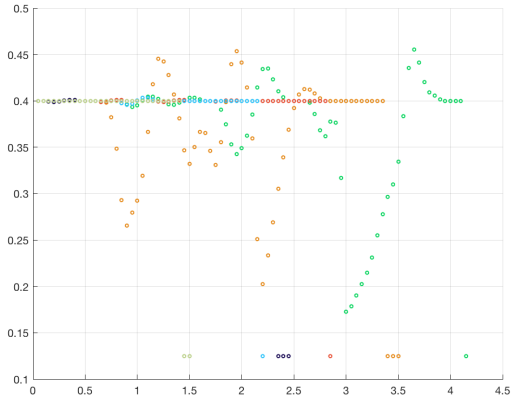

Fig. 15. Random case environment - Velocities for each vehicle.

As can be seen in Fig. 10, Fig. 11 and Fig. 12, each UAV (represented by a colour) reaches its goal (represented by a ♦ of the same colour) while avoiding the obstacles that it encounters throughout its path. In Fig. 13 we can see the distance from the vehicle to its goal at each time step. As can be seen in Fig. 14, the distance between a the surface of a UAV and its closest obstacle's surface (static or dynamic) its always above zero. In Fig. 15, we can see the velocity for all UAVs in each time step.

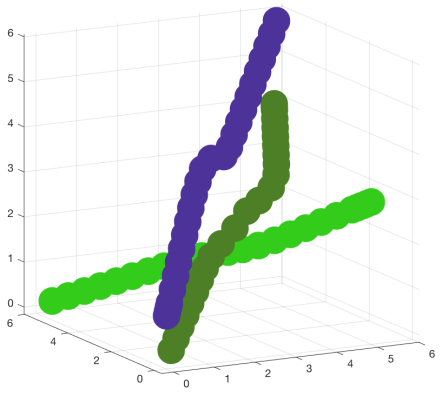

Fig. 16. Multi UAV avoidance case.

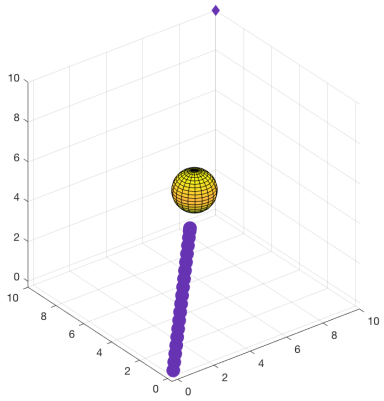

Fig. 17. Local minima problem for the symmetric case.

VII. Discussion, conclusion and Future development

As mentioned in the Simulation and results section Fig. 6, Fig. 7, Fig. 8 and Fig. 9 all show a steady state error of zero, which is the case for all 12 outputs from the feedback system. This implies that the LQR is working the way it was intended to and that we actually have full control of the UAV. The parameters Q and R were heuristically chosen, the rise, settling and overshoot times can be further improved by changing the values of Q and R, this could be a subject to study in a future projects within the field.

The potential field method has been proved to be reliable by not allowing UAVs to crash with each other or other obstacles, as can be seen in the multi-drone-avoidance case in Fig. 16. It also provides good enough safe margins of more than $0.5\,\mathrm{m}$ of minimum distance between a moving UAV and any other type of static or dynamic obstacle, as can be seen in Fig. 14. The velocities of the different UAVs, as seen in Fig. 15, are appropriate for this case study, in terms of real life UAV capabilities. The vehicles should not be slow but neither too fast to assure obstacle detection with enough margin to avoid them in time. This is due to the discrete nature of the computer unit that implements the controller and the sensor's measurements. Therefore, an 'open environment' velocity of around $1\,\frac{\mathrm{m}}{\mathrm{s}}$ in our case is considered to be safe. Since we can see in Fig. 14 that the safe margin is large, the velocity could also be increased.

However, as explained in the last sections, potential fields can be problematic in some specific cases, this should be solved before this method can be used to test how UAVs would move in more realistic environments, e.g a city-live environment. As can be seen in Fig. 17, the geometries where a UAV can get stuck in a local minima of a plain potential field function are not complex, but also not likely to happen.

Future projects could try to control the simulated drones with the full dynamic model designed, to see if it would actually provide a safe navigation for the drones and test it out in a real quad copter. Another future project could

be to focus on the difference between different navigation algorithms, i.e. Djikstra compared to RRT, potential fields and so on, and develop more tools and tests to compare them. Even further down in the topic, a project could be developed to explain how to analyse the images resulting from a camera a UAV could carry, and how this information could be useful to help the drone navigate.

ACKNOWLEDGMENT

REFERENCES

[1] A. Guillen-Perez, R. Sanchez-Iborra, M. D. Cano, J. C. Sanchez-Aarnoutse, and J. Garcia-Haro, "Wifi networks on drones," in *2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT)*, Nov 2016, pp. 1–8.

[2] M. Costandin, P. Dobra, and B. Costandin, "Nonlinear model and control of a quadcopter," in *2017 21st International Conference on System Theory, Control and Computing (ICSTCC)*, Oct 2017, pp. 687–692.

[3] W. Meng, Y. Hu, J. Lin, F. Lin, and R. Teo, "An efficient high-fidelity 3d multi-uav navigation and control simulator in gps-denied environments," in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, Nov 2015, pp. 002 562–002 567.

[4] E. Chirtel, R. Knoll, C. Le, B. Mason, N. Peck, J. Robarge, and G. C. Lewin, "Designing a spatially aware, autonomous quadcopter using the android control sensor system," in *2015 Systems and Information Engineering Design Symposium*, April 2015, pp. 35–40.

[5] F. Sabatino and K. H. Johansson, "Quadrotor control: modeling, nonlinearcontrol design, and simulation," KTH, Skolan fr elektro- och systemteknik (EES), Reglerteknik, 2015.

[6] T. Bresciani, "Modelling, identification and control of a quadrotor helicopter," 2008, student Paper.

[7] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 20–32, Sept 2012.

[8] J. A. Chavez and S. Silvestre, "Infrared remote control systems," University lecture, UPC, Electronic Engineering Department, 2017.

[9] S. Hirata, M. K. Kurosawa, and T. Katagiri, "Cross-correlation by single-bit signal processing for ultrasonic distance measurement," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 91, no. 4, pp. 1031–1037, 2008.

[10] S. Adarsh, S. M. Kaleemuddin, D. Bose, and K. I. Ramachandran, "Performance comparison of infrared and ultrasonic sensors for obstacles of different materials in vehicle/ robot navigation applications," *IOP Conference Series: Materials Science and Engineering*, vol. 149, no. 1, September 2016.

[11] T. Glad, *Reglerteknik : grundläggande teori*, 4th ed. Lund: Studentlitteratur, 2006, pp. 187–188.

[12] L. M. Argentim, W. C. Rezende, P. E. Santos, and R. A. Aguiar, "Pid, lqr and lqr-pid on a quadcopter platform," in *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, May 2013, pp. 1–6.

[13] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," 05 1999.

[14] D. S. Yershov and S. M. LaValle, "Simplicial dijkstra and a* algorithms for optimal feedback planning," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2011, pp. 3862–3867.

[15] S. Ge and Y. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous Robots*, vol. 13, no. 3, pp. 207–222, Nov 2002. [Online]. Available: https://doi.org/10.1023/A:1020564024509

[16] N. Amato, "Potential field methods," University lecture, Università degli Studi di Padova, 2004.

[17] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, Apr 1991, pp. 1398–1404 vol.2.