



RobotMe: A drone platform for control education

Ahmad Hably, Rattena Tang, Jonathan Dumon, Aurelien Carriquiry

► To cite this version:

Ahmad Hably, Rattena Tang, Jonathan Dumon, Aurelien Carriquiry. RobotMe: A drone platform for control education. IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society, Oct 2019, Lisbonne, Portugal. 10.1109/IECON.2019.8927551 . hal-02398822

HAL Id: hal-02398822

<https://hal.science/hal-02398822>

Submitted on 8 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

RobotMe: A drone platform for control education

Ahmad Hably, Rattena Tang, Jonathan Dumon, Aurelien Carriquiry
Univ. Grenoble Alpes, CNRS, Grenoble INP*, GIPSA-Lab,
arcSepA= F-38000 Grenoble, France.

*Institute of Engineering Univ. Grenoble Alpes
ahmad.hably@grenoble-inp.fr

Abstract—In this paper, RobotMe, a new educational robotic platform is presented. It is dedicated to drone position control using various control designs. Different material and software elements employed in the platform are detailed. Some improvement guidelines are also proposed.

I. INTRODUCTION

Robotics is considered as one of the future key technologies with an estimated European market of 100 billion euros in 2020. During the last decade, we have seen major developments of numerous robotic platforms such as drones. These robots are now widespread because of their design and agility. Drones are used to carry out diverse civil and military missions. They are used, for instance, for inspection tasks in open areas like dams, communication and electrical networks and in confined environments like tunnels. They are also used for monitoring infrastructure and road traffic. The agricultural drone became a valuable ally for farmers who want to monitor their fields.

RobotMe is an educational project that aims to boost the teaching activities of the University of Grenoble Alpes (UGA). Thanks to its platform using the 3D motion capture technique and different robots, engineering students have the possibility to develop their creativity, and acquire different skills in modeling, control, signal and image processing, embedded programming, and industrial computing. RobotMe platform will undoubtedly allow a practical learning of robotics, prototyping and rapid developments.

In this paper, technical parts used in the platform are presented. Several control techniques are also shown. The paper is organized as follows. After a technical presentation of the platform in Section II, system modeling is given in section III. Controller design is addressed in Section IV. Some technical issues that has to be addressed in building a similar platform are detailed in Section V. The paper ends by conclusions and future work in Section VI.

II. PLATFORMS TECHNICAL PRESENTATION

RobotMe platform is composed of the following elements:

- A portable PC with a set of software to be detailed in the next paragraph.
- Logitech Gamepad F310 joysticks.
- A movement capture system Optitrack composed of 8 infra-red cameras FLEX 13 with a dongle [1].
- Several small commercial drones of the **BLADE Inductrix FPV+** [2]. These drones can be safely used by students in practical sessions. Fluorescent beads are glued on drones for localization in the area covered by the Optitrack system. Each geometrical figure formed by the beads allows to detect each robot in a unique way.
- A 4-block radio transmitter/receiver system for sending commands from PC to drones.
- A batch of LiPo E-Flite 1S 3.7V 500mAh EFLB5001S25UM batteries serving as source of energy to drones.

- A Hitec X4 microbattery charger.

A set of software makes it possible to carry out the signal acquisition, real-time data processing and commands transmission to drones.

- The Motive software detects the position and speed of moving objects in real time.
- An Optitrack Drone Interactive software that allows Matlab/Simulink to recover the coordinate values of the moving objects from the Motive software.
- Matlab/Simulink software that contains the control algorithms to be presented in Section IV.

The Simulink diagram of Fig.4 allows sending control signals to one drone thanks to a joystick. These signals can be either pre-programmed trajectories or manual movement. For simplicity, we will focus on the diagram for a single drone. These diagrams are duplicated in case of a second drone. The diagram can be separated into 5 main blocks.

- Inputs:
 - The GAMEPAD controller block manages data acquisition of signal generated by the joystick.
 - The Optitrack block manages data acquisition of all objects coordinates.
- Outputs:
 - The IRC block generates commands sent to the drone. They are thrust, roll and pitch angles, and yaw rate.
 - The Workspace block saves a set of data for post-processing in Matlab.
- The Control block contains all the control algorithms to be detailed in the Section IV.

III. SYSTEM MODELING

In our application, the drone is a quadrotor whose 4 rotors are spaced to form the 4 corners of a square. In order to avoid the reaction torque of quadrotors [3], the rotors rotate in opposite directions. The rotational direction of the rotors is indicated to the left of Fig.??.

A. Rotations

Rotors rotation generates a thrust force that allows the drone to take-off. By varying the rotational speeds of each rotor, it is possible to move the drone into three dimensional space. Four elementary displacements are possible to move the drone, vertical, roll, pitch, and yaw movements. To describe these elementary movements, we will consider that the drone is in an initial horizontal position in space.



Fig. 1. Drone Inductrix FPV+.

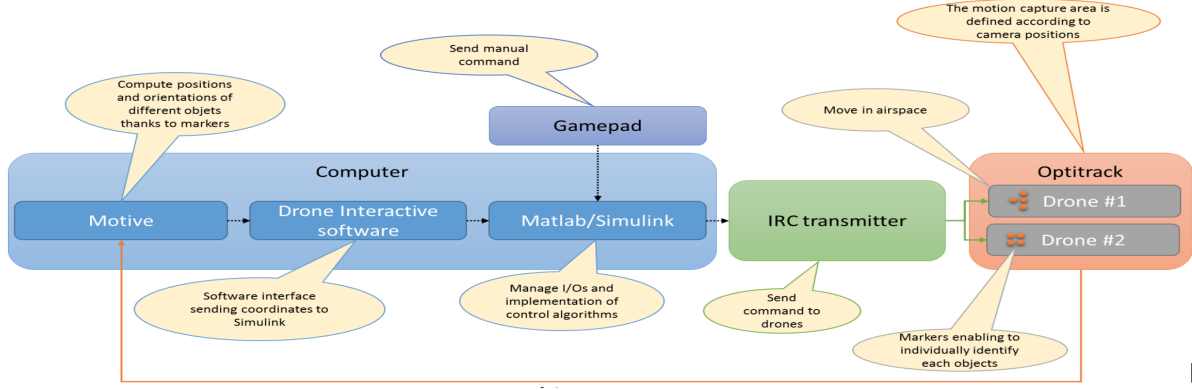


Fig. 2. The set of software used in the RobotMe platform.

1) *Vertical movements*: To move the drone vertically, the rotational speed of the 4 rotors has to be increased or decreased simultaneously.

2) *Roll movements*: To create the roll rotation, a left-right dissymmetry is needed. For example, increasing the rotation speed of the rotors on the left and decreasing the speed of the rotors on the right leans the drone to the right. Decreasing the rotation speed of the opposite side keeps the drone at a constant altitude. This change of orientation is accompanied by a movement of the drone to the right. So we have a coupling between roll and horizontal displacement.

3) *Pitch movements*: In a similar way to the roll rotation, a front-to-back dissymmetry is used to perform a pitch rotation. Increasing the rotational speed of the rotors at the rear and decreasing the speed of the rotors at the front leans the drone forward. Decreasing the rotation speed of the opposite side keeps the drone at a constant altitude. This change of orientation is accompanied by a movement of the drone forward. One has a coupling between the pitch rotation and the horizontal movement.

4) *Yaw movements*: To create the yaw movement, it is necessary to create an unbalance clockwise rotation/counterclockwise rotation. Thus, increasing the speed of the rotors rotating clockwise and decreasing the speed of the rotors rotating counter-clockwise rotates the drone counter-clockwise at the same altitude.

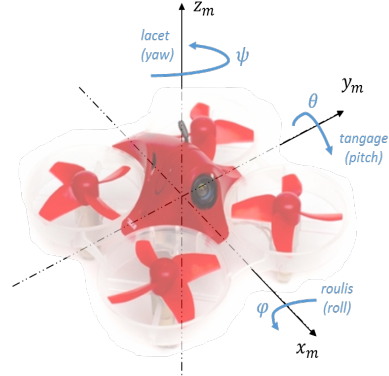


Fig. 3. The configuration of roll, pitch, and yaw angles.

$$R_\varphi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\varphi & -s_\varphi \\ 0 & s_\varphi & c_\varphi \end{pmatrix}; R_\theta = \begin{pmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{pmatrix};$$

$$R_\psi = \begin{pmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

B. Coordinate systems

Since the drone is a mobile system in a three-dimensional space, it is convenient to define two different coordinate frames in which one will describe on one hand the position of the drone and on the other hand the efforts applied on the system. The rotation matrix converts the coordinates from one frame to another by simple matrix product. The first coordinate frame is a fixed frame $R(\vec{x}, \vec{y}, \vec{z})$ related to the earth-fixed coordinate frame. It is a direct orthogonal frame with \vec{z} upwards. The second coordinate frame is a mobile (body) frame $R_m(\vec{x}_m, \vec{y}_m, \vec{z}_m)$ related to the drone whose origin coincides with its center of gravity G . It is a direct orthogonal frame with \vec{x}_m forward (camera orientation).

C. Frames

The rotation matrix of the mobile coordinate frame R_m to the fixed coordinate frame R is defined using the Euler angles. These angles (φ, θ, ψ) represent respectively the roll, pitch and yaw angles. They are defined in Fig.3. We denote by R_φ (respectively R_θ and R_ψ) the rotation matrices corresponding to the rotations of angle φ (respectively θ, ψ). The following rotation matrices are then obtained:

using the following notations $c_\alpha = \cos\alpha$ and $s_\alpha = \sin\alpha$. Since the transformation of coordinates from one coordinate frame to another involves only the successive rotations φ, θ, ψ (chosen in this order), the rotation matrix becomes $M_P = R_\psi R_\theta R_\varphi$.

$$M_P = \begin{pmatrix} c_\psi c_\theta & c_\psi s_\theta s_\varphi - s_\psi c_\varphi & c_\psi s_\theta c_\varphi + s_\psi s_\varphi \\ s_\psi c_\theta & s_\psi s_\theta s_\varphi + c_\psi c_\varphi & s_\psi s_\theta c_\varphi - c_\psi s_\varphi \\ -s_\theta & c_\theta s_\varphi & c_\theta c_\varphi \end{pmatrix} \quad (1)$$

D. Dynamic modeling

The fundamental principle of dynamics can be applied in order to find the drone dynamic model. Forces acting on the drone are:

- P : the weight of the whole system (brought back to the center of gravity).
- T_i : the lift produced by the rotor i in the mobile coordinate frame reference R_m .
- The resultant of the torques produced by the rotors i compared to the center of gravity:

$$T = [0 \ 0 \ T]^T_{/R_m}$$

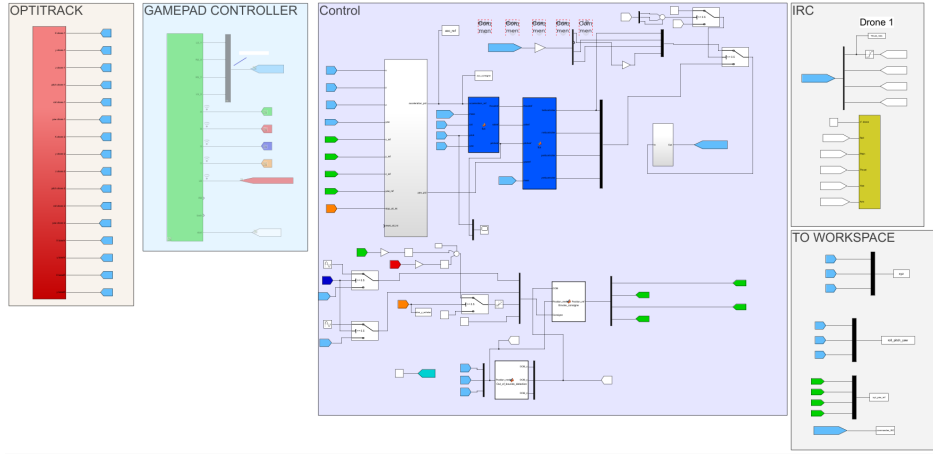


Fig. 4. Simulink diagram used in real time experiments.

We note Ω the angular velocity of the drone as: $\Omega = [\varphi \ \theta \ \psi]^T$. Taking $\ddot{r} = [\ddot{x} \ \ddot{y} \ \ddot{z}]^T$ the linear acceleration of the drone center of mass, one can obtain:

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + Mp \begin{bmatrix} 0 \\ 0 \\ \sum T_i \end{bmatrix} \quad (2)$$

IV. CONTROL STRATEGY

In this section, some control strategies that can be employed in RobotMe platform are presented. Since the angle regulation loop is already implemented in the drone as an internal loop using PID controllers, in this present paper we are mainly interested in the outer control loop for the drone's position control, mainly in the x-y plane. Students have the choice to test different control strategies studied and explained in the theoretical control and system courses. Different control strategies have been studied: PD controller (section IV-A), loop shaping design (section IV-B), linear quadratic controller (LQR) (section IV-C), and robust controller design (H_∞) (section IV-D). The control block diagram is shown in Fig.5. For each experiment,

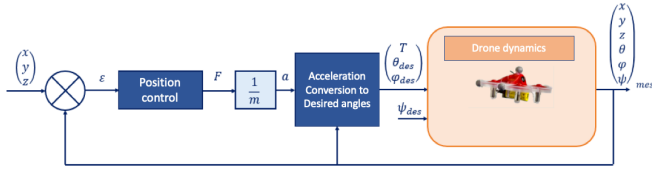


Fig. 5. The control block diagram with the inner and outer control loops.

a controller is designed and validated in simulation than on the real platform for the position control along the \vec{x} axis. Then the controller is tested to the position control along a circular trajectory in the x-y plane.

A. PD controller

Firstly, students can study a proportional derivative (PD) controller. The closed-loop transfer function with a PD controller is a second order transfer function:

$$\frac{x(s)}{F_x(s)} = \frac{K_p K_1}{s^2 + K_v K_1 s + K_p K_1} = \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2} \quad (3)$$

One wants to design a controller that satisfies the following performance criteria:

- Minimum overshoot: $\%D = 5\%$ so $\zeta = 0.7$.

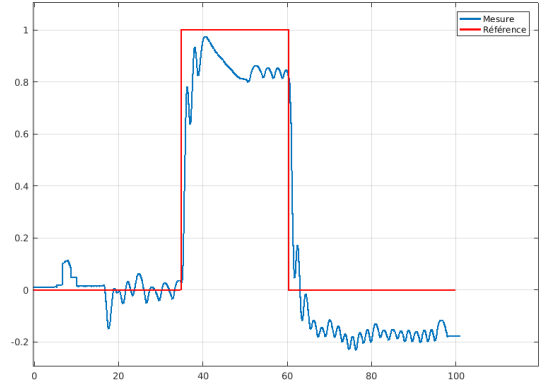


Fig. 6. The time response of the position x controlled by the PD controller.

- Fast time response: $T_r = 2.3$ s with $w_n = 1.4$ rad/s.
- No saturation is considered to stay in the linear case.

By identification, one can get the control parameters:

$$K_p = \frac{w_n^2}{K_1} ; \quad K_v = \frac{2\zeta w_n}{K_1} \quad (4)$$

For $k_1 = 15$, the response of the position control along the x-axis is given on Fig.6. The response to the circular trajectory centered at (0,0) with a radius of 1m in the x-y plane is presented on Fig.7.

B. Loop shaping Design

The loop shaping design can also be applied [4]. The design specifications will be the largest possible phase margin and cutoff frequency. The controller will have the following form:

$$C_x = \frac{L_4}{G_x} \quad (5)$$

where G_x is the dynamics model of \vec{x} axis movement assumed here as scaled transfer function of double integrator with a low pass filter. L_4 is designed as explained in the following steps:

- 1) First, L_1 is proposed in order to eliminate disturbances.
- 2) Since the obtained system is unstable, a "Lead network" is added with

$$L_2 = G_d \frac{1 + \tau_1 s}{1 + \alpha_1 \tau_1 s}$$

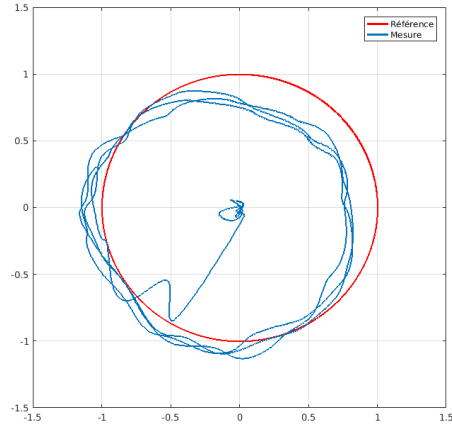


Fig. 7. Drone response to a circular trajectory using a PD controller.

and $\alpha_1 = 0.02$ (corresponds to 74 deg) and $w_c = 2.2$ so $\tau_1 = \frac{1}{w_c \sqrt{\alpha_1}}$. The maximum of the phase margin corresponds to a frequency lower than the design frequency $w_c = 1.02 < 2.2 \text{ rad/s}$ since the low-pass filter decreases the phase.

- 3) To eliminate the static error an integrator is added such that $L_3 = L_2 \frac{1 + \tau_i s}{s}$ with $\tau_i = \tau_1$. This value is the maximum frequency that can be used for fast error elimination. The maximum of the phase margin corresponds to a frequency other than the design frequency: $w_c = 1.51 \neq 2.2 \text{ rad/s}$ since the low-pass filter and the integrator decrease the phase margin. One can choose to use this new frequency which corresponds to the maximum of the phase margin $w_c = 1.51$, the gain corresponding to this frequency is 6.55 dB
- 4) A gain of -6.55 dB is added to have a cutoff frequency of 1.51 rad/s such that $L_4 = L_3 K_1$. At the end one has a phase margin of 43 deg (see Figure 8).

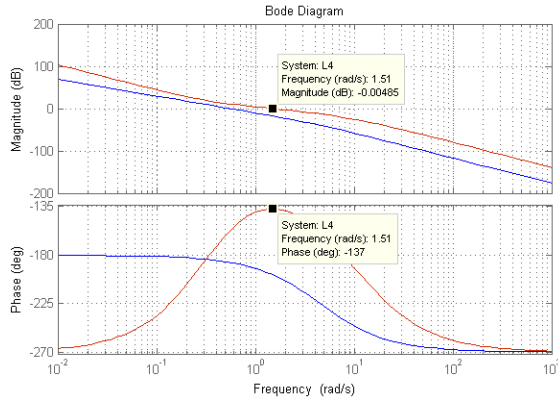


Fig. 8. Bode diagram of L_4

The obtained controller is tested experimentally using step signal reference on the drone's \vec{x} axis, Fig.9, and the \vec{y} axis. One can notice that at for the first direct step signal of Fig.9, the system responds in 22 sec with a null static error. On the other hand, for the step reference in the indirect sense, the response has an overshoot. This can be explained the need to calibrate the pitch angle of the drone. However, the static error is always null. The system's response for a circular trajectory in the x-y plane is shown on Fig.10.

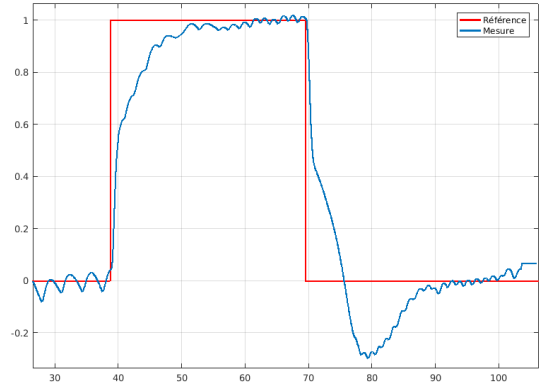


Fig. 9. The time response of the position x controlled by the loop shaping controller.

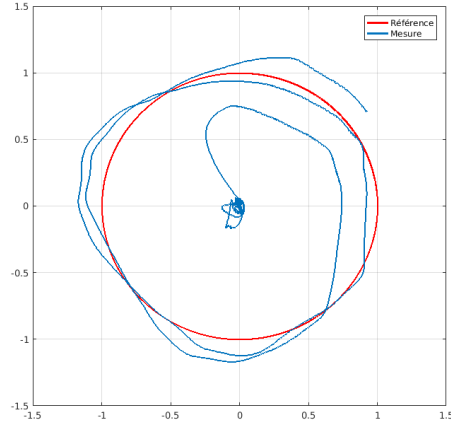


Fig. 10. Drone response for a circular trajectory.

C. Linear quadratic regulator

As for the previous section, the model dynamics of the drone along the x-axis is considered as double integrator : $m\ddot{x} = u_x$ with m is the mass of the drone in the x-axis. For this part, the specification to be respected are: Response time < 10 s and with an overshoot $< 5\%$. First, An observer is designed in order to estimated the drone speed. The observer's gain is given by a dual LQ problem. To improve the rejection of disturbances, a linear quadratic controller with an integral action is designed. A step reference of 3m is tested experimentally (Fig.11). An overshoot of about 17% is observed with a response time at 95% is about 5s. Circular trajectories at different speeds have been tested. One observes that the lower the speed, more oscillations appear (Fig.12).

D. Robust control design

The robust control is designed using the previous model and only the measured output. The specifications and the desired performances are translated in frequency domain by templates the system's sensitivity functions (W_e , W_u and W_d). The system is put in its standard form ("generalized plant") to be able to define the H_∞ problem and obtain the associated corrector [5]. As it is shown on Fig.13, P is the generalized plant (contains the plant, the weights, the uncertainties if any) and K is the controller. The closed-loop transfer matrix from w

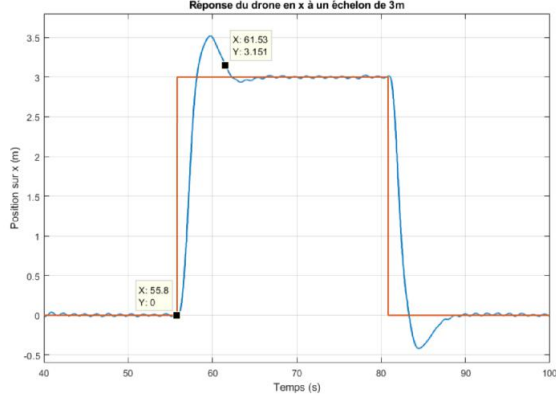


Fig. 11. The step response of drone in the x axis while applying the LQR controller.

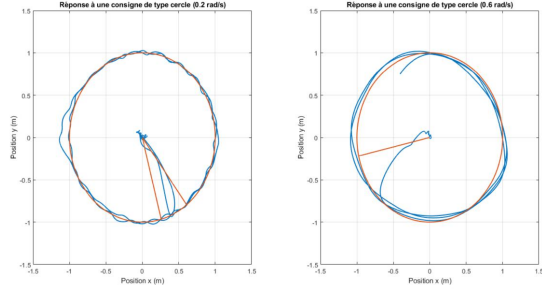


Fig. 12. Circular trajectories obtained while applying the LQR controller.

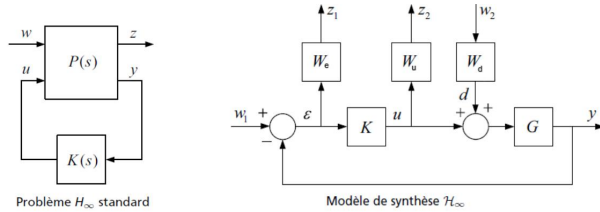


Fig. 13. The closed-loop transfer matrix from disturbance w to the controlled output z , $T_{zw}(s)$ with the controller K .

to z is given by:

$$T_{zw}(s) = F_l(P, K) = P_{11} + P_{12}K(IP_{22}K)^{-1}P_{21}$$

where $F_l(P, K)$ is referred to as a lower Linear Fractional Transformation. The overall control objective is to minimize some norm of the transfer function from disturbance and reference w to the controlled output z , the H_∞ norm. Three frequency templates have been created: W_e , for closed-loop performance, W_u for actuator constraints and W_d , for input disturbances:

- The W_e template is a low-pass filter.

$$W_e = \frac{\frac{1}{M_s}s + w_b}{s + w_b\epsilon}$$

where $\frac{1}{M_s}$ represents the module margin, w_b the cutoff frequency and ϵ is the static error.

- The template W_u is a high-pass filter.

$$W_u = \frac{s + \frac{w_{bc}}{Mu}}{\epsilon_1 s + w_{bc}}$$

with $\frac{1}{Mu}$ represents the amplitude relation between reference and command, w_{bc} the cutoff frequency and ϵ_1 is the attenuation at high frequencies.

- The template W_d (disturbance attenuation) is considered constant.

Different combinations of coefficients are analyzed. The controller can be found using the `hinfsyn` function of Matlab Robust Control Toolbox. The obtained controller is a dynamic system of four states, with an input and an output. We first tested the templates of Fig.14. It can be noted that the templates are not always satisfied, but the

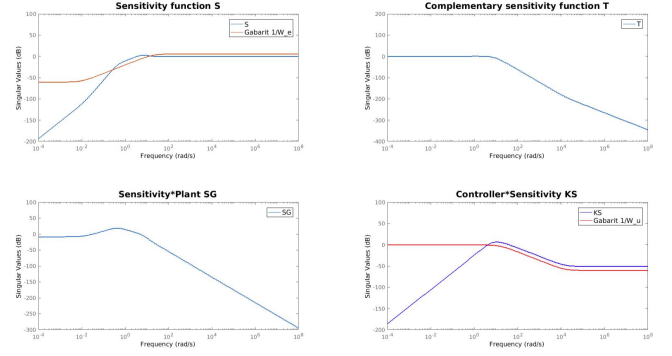


Fig. 14. Sensibility functions of the robust control design.

frequency response and system sensitivity functions are acceptable, with good cut-off and attenuation frequencies (-50 dB for high-frequency disturbances, such as seen in KS). A 3m step response is tested experimentally (Fig.15). It can be noticed that there is an overshoot of about 33%. The response time at 95% is about 14 s. Permanent steady state oscillations are observed of an amplitude that is less than 3cm. The response to circular trajectories is also shown on Fig.15. The obtained results of the robust control can be improved

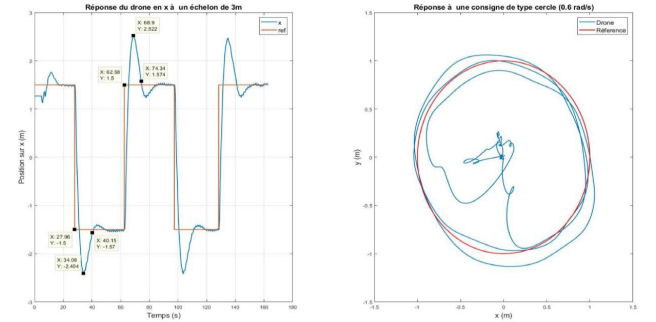


Fig. 15. Experimental application of the robust control.

by modification of the frequency templates. The cutoff frequency of W_e and W_u have to be reduced, as well as the tolerance of the static error. W_d is no longer constant, instead a low-pass filter is used to have a more important rejection of disturbances of low frequency. A new template W_T has to be introduced for the T sensitivity function to increase the robustness of the system to parametric variations (see for example Fig.16). Students can try to use new templates and sensitivities functions of Fig.17.

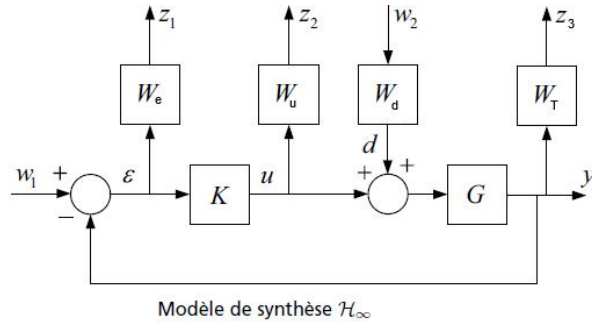


Fig. 16. New H_∞ control design to increase the robustness of the system to parametric variations.

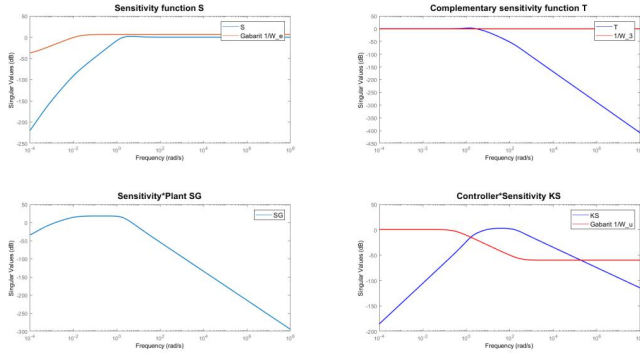


Fig. 17. New templates and sensitivities functions.

V. SOME IMPORTANT ISSUES

In this section, we will list some important issues to be addressed before using the platform with the students.

A. Modeling issue

For the mathematical modeling of the drone system, a convention must be chosen for angle orientation. However, this convention may differ from the one selected by the manufacturer and a certain attention must be drawn to angle signs. For instance for the Inductrix FPV+, the positive roll corresponds to the negative roll of the convention of our model while pitch and yaw angles have the same orientation.

B. Technology issues

1) *Mobile coordinate system:* In Motive software, one could create an object, called "body", according to markers placed on this object. Creating this "body" generates a center of gravity and a mobile coordinate system. In our convention, the axis \vec{x} is pointing the front of the drone. Thus, the operator must ensure that the front of the drone points to the positive direction \vec{x}_+ of the reference frame before creating the "body" in the software. Otherwise the drone coordinate system will not correspond to the convention and unexpected movement will be observed.

2) *Drone binding:* Binding a drone with its transmitter may be a delicate operation, especially using multiple drones at the same time. To avoid to bind a drone to a wrong transmitter, a reliable and robust procedure must be emphasized.

3) *Loss of IRC communication:* For some reason, the communication between a drone and its transmitter can sometimes interrupted. In this case, the drone receives a set of zeros as command. The origin of this problem has to be identified in a future work.

4) *Loss of position detection:* Once an object leaves the motion capture area, it is not detected anymore. However, its last coordinates are maintained and sent to Simulink. Thus, wrong position information is used for the control algorithm and the drone movement are unpredictable. This issue occurs if a calibration of the motion capture system is required or if drones fly too close to the capture area. To remedy the last situation, a software limitation has been developed to avoid all drones to move too close to the capture area limits. This limitation consists in creating virtual walls that cannot be crossed by drones (Fig.18). The designed controller compares the current drone position $[x_{mes} \ y_{mes} \ z_{mes}]^T$ to the reference $[x_{ref}, y_{ref}, z_{ref}]^T$ to generate the desired acceleration $[\ddot{x}_{ctr} \ \ddot{y}_{ctr} \ \ddot{z}_{ctr}]^T$ to reach the reference position. Then a function converts the desired acceleration into thrust, roll, pitch and yaw rate commands. The virtual wall algorithm consists in adding a function that detects if the current drone position $[x_{mes} \ y_{mes} \ z_{mes}]^T$ is out of a cube defined by $[x_{lim}, y_{lim}, z_{lim}]^T$. If the drone leaves the cube, the boolean $[b_{out_x} \ b_{out_y} \ b_{out_z}]^T$ indicate which border has been reached (respectively $x_{lim}, y_{lim}, z_{lim}$). These boolean and the current position are used into the control algorithm to adapt the reference position $[x_{ref}, y_{ref}, z_{ref}]^T$ according to which limit has been crossed.

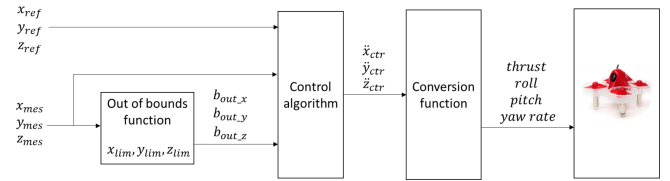


Fig. 18. Algorithm of the virtual wall.

VI. CONCLUSIONS

In this paper, the RobotMe platform is presented. In addition to the technical material used. Different control design methods have been tested for the position control of a drone. System non-linearities, neglected in this present paper, has to be presented in a future work. In this sense, sliding mode controllers can be explored. The effect of the battery's state-of-charge (SOC) on the drone's movement, especially in the vertical axis has also to be studied.

ACKNOWLEDGMENTS

The authors would like to thank IDEX Universite Grenoble Alpes for their financial support. In addition, they would thank the students (Bilal, Hamza, Esteban, Pedro, Isaac, Douglas, and Hiacheng) who participated in developing and validating the different controllers presented in the paper.

REFERENCES

- [1] (2019) <https://optitrack.com/>.
- [2] (2019) Flash rc: <https://www.flashrc.com>.
- [3] 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, 2012.
- [4] K. J. Aström and R. M. Murray, *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.
- [5] K. Zhou and J. C. Doyle, *Essentials of robust control*. Prentice hall Upper Saddle River, NJ, 1998, vol. 104.