USE OF A DJI TELLO DRONE AS AN EDUCATIONAL PLATFORM IN THE FIELD OF CONTROL ENGINEERING

Georges Ghazi, and Julien Voyer
École de technologie supérieure (ÉTS), Montreal, Quebec, H3C 1K3, Canada
georges.ghazi@etsmtl.ca

Abstract – *This paper presents a hands-on pedagogical approach using a DJI Tello drone as an interactive teaching platform in the field of automatic control engineering. The DJI Tello is a small commercial quadcopter drone and includes a software development kit (SDK) that allows developers to control the Tello using various programming languages, including Python. The drone is also equipped with a large number of sensors that can be used in real-time to collect data and analyze how changes in control inputs such as thrust, pitch, roll, and yaw affect its flight path and stability. These features make the Tello a good teaching tool for demonstrating control concepts in an attractive and practical way. Two examples of pedagogical applications are presented in this paper. The first example aims to illustrate in practice how system identification can be used to create a mathematical model of the DJI Tello drone using transfer functions. The second example aims to illustrate how to design a Proportional-Integral (PI) controller and validate it after its implementation on the DJI Tello drone. Through these teaching demonstrations, it was possible to enhance cognitive learning while providing students with a better understanding of the fundamental concepts of modeling and control. It was also observed that even though the students had no background in aeronautics, the use of an atypical system such as a drone aroused their curiosity, encouraging them to participate, thus making the in-class demonstrations more dynamic.*

Keywords: Educational platform, Demonstrator, Drone, Visualization, Student perception, Control theory.

1. INTRODUCTION

"Teaching" is an art in which teachers and researchers strive to illustrate fundamental concepts, sometimes abstract, to a diverse audience. However, in many fields, it is often difficult to find practical ways to illustrate the concepts needed to train students. In control theory, for example, there are many concepts that students have difficulty grasping because they have no support to picture them. One such example is the concept of stability, which is fundamental in control theory. Stability refers to the ability of a system to return to a stable state after a disturbance. It is an essential property that must be considered when designing control systems, as an unstable system can result in unpredictable and potentially dangerous behavior. Without a concrete example, it can be difficult for students to understand the importance of stability and how it can be achieved through control theory techniques.

Although there are several tools for modeling systems and performing numerical simulations, the results that students must interpret generally remain numerical and theoretical. Unfortunately, they do not always have the imagination to translate the results obtained by the simulations into practical aspects. This lack of concrete application widens the gap between the theoretical and practical training of students, which increases the deeply rooted lack of trust in academics in the industry [1,2]. As a result, many students graduate without a practical overview of how the fundamentals of control theory are applied in industry [3].

One solution to facilitate the teaching of control theory concepts would be to use more experimental material to reinforce the demonstration of fundamental concepts taught in a course. Today, there are many companies specializing in the development of educational equipment to support teaching or to be used in laboratory courses. These companies provide equipment ranging from a simple inverted pendulum to more complex systems such as robotic arms [4]. For example, the Qube Servo 2, manufactured by Quanser Inc., can be a very good alternative to illustrate concepts in the field of automatic control [5]. The Qube Servo 2 is a small and compact inverted pendulum that can be interfaced with several software such as Matlab/Simulink© or Labview to teach students how to design controllers to stabilize the pendulum position [6]. However, this type of educational equipment is usually very expensive or requires expensive software licenses. Another constraint is the maintenance of these systems, which can be a problem both from a practical and a cost point of view.

An alternative to illustrate most concretely how theoretical fundamentals can be applied to physical systems consist in using simulations and 3D animation models. Indeed, the use of a 3D model allows students to have a visual representation of the system they are working on, and thus to have a more realistic perception of the results they obtain. Tejado *et al.* in [7], or instance, designed a 3D model of the

Qube Servo 2 using SimMechanics to create a simulator to support automatic control teaching. Within the same context, OpenAI Gym is a free Python library developed by OpenAI that can be used to teach critical concepts about control using reinforcement learning techniques [8]. From a robotic perspective, the Robot Operating System (ROS) has become a standard in industry, research, and education [9]. ROS is a set of software libraries that can be used with mobile robot platforms for educational purposes [10].

Using a simulation model for teaching has the benefit of creating a visual context to help students to analyze their results better. However, they do not have a true picture of the industrial hardware used in process control or know how to implement their solution on the real system. In addition, numerical simulations do not always allow students to experience the constraints and limitations present in a real physical system. It is therefore important to incorporate demonstrations during class to help students understand the theoretical concepts and more importantly understand how to apply these concepts to real systems.

Developing a demonstrator for teaching purposes remains a challenge for teachers and researchers. The objective of this paper is, therefore, to present a teaching practice that was developed by the authors at the École de technologie supérieure (ÉTS) to use a DJI Tello quadcopter drone as an educational demonstrator to illustrate basic concepts related to the fields of automatic control. The choice of a drone was made for several reasons. Indeed, in addition to being increasingly popular with the general public, drones are versatile systems. A typical drone uses a combination of sensors, feedback control, and estimation algorithms to perform complex tasks such as hovering, following a path, or avoiding obstacles. These tasks involve many of the key concepts in control theory, including feedback control, stability, estimation, and optimization. Thus, by using a drone as a demonstrator system, students can see firsthand how these concepts are applied in a real-world application. Another advantage of a drone is that it is a very versatile system that can be considered as a SISO (Single Input - Single Output) or MIMO (Multiple Inputs - Multiple Outputs) system. Finally, drones are relatively affordable and easy to use, making them an accessible tool for teaching control theory to students of different educational levels.

The use of a drone as a research and educational platform has been explored by Giernacki *et al.* [11]. Their study focused on using a commercial drone, similar to the one used in this study, to demonstrate control concepts using artificial intelligence and computer vision. This paper, however, focuses on concepts of control theory that are typically covered in undergraduate or entry-level graduate courses. The objective of this paper is therefore to present how a drone can be used as a demonstrator to illustrate the fundamental concepts of control theory.

The rest of this paper is organized as follows: **Section 2** introduces the DJI Tello drone, as well as the modes of communication using a stand-alone computer (i.e., PC). **Section 3** illustrates typical application examples used with students to demonstrate fundamental control engineering techniques. **Section 4** discusses students' feedback and appreciation. Finally, the paper concludes with several remarks and a conclusion.

2. DJI Tello Quadrotor Drone

The quadrotor drone used in this document to develop the educational demonstrators is the DJI Tello. This drone was chosen among many others for its affordable price, as well as for its ability to be interfaced with different programming languages, such as Python. The purpose of this section is to provide the reader with an overview of the Tello drone, as well as a brief review of existing Python packages for programming the drone. It should be noted that, although the application examples presented in this paper were carried out with the Tello drone, they can be easily adapted to other types of drones.

2.1. DJI Tello Quadrotor

The Tello drone is a small-size (98 x 92.5 x 41 mm), lightweight (87 g, including battery and propeller guards) quadcopter designed for beginners and experienced pilots. This low-cost drone is manufactured by Shenzhen Ryze Technology. A schematic illustration of the Tello drone is shown in Figure 1.

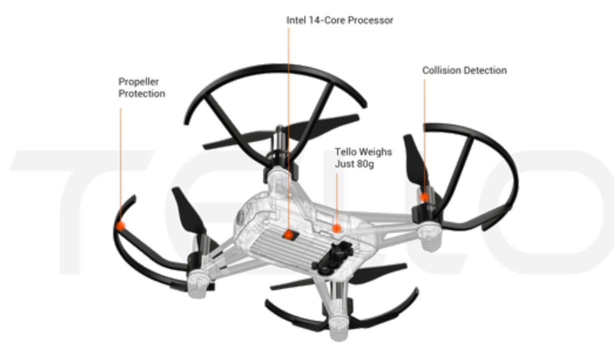

Fig. 1. **DJI Tello Drone [https://www.ryzerobotics.com/tello]**

The drone is powered by four brushless motors capable of providing stable flight, even in challenging weather conditions. With its 3.8V (1100mAh) Li-Po battery and well-designed 3-inch propellers, the drone can fly for up to 13 minutes under normal operating conditions (i.e., no wind and a rated speed of 15 km/h). In addition to having good performance, the drone is equipped with a series of onboard sensors, including:

- A barometer;
- An optical flow sensor;
- A laser Time of Flight (ToF) sensor;
- Three accelerometers, and three gyroscopes.

The optical flow sensor (i.e., a low-resolution camera), located at the bottom of the drone and facing downwards, is used to analyze the drone's longitudinal and lateral movements over time. The images taken by this camera are combined with IMU data to stabilize the drone when it is hovering or to prevent it from drifting. When the drone is moving, the optical flow sensor is used to determine the velocity of the drone. The laser ToF sensor, also located at the bottom of the drone, is used to determine the relative height of the drone, while the barometer gives the pressure altitude. Both of these sensors can be used to determine the height of the drone, but for indoor conditions, the ToF remains the only alternative. Finally, the DJI Tello is also equipped with an onboard nose-mounted highly-quality camera capable of recording 720p HD video (at 30 FPS), and capturing 5 MP photos with a field of view of 82.6º.

With its diverse sensors, the Tello drone is an ideal and flexible tool for both research and educational purposes.

2.2. Communication with the Drone

One of the advantages of the DJI Tello drone is that it can be programmed using the Python programming language. Indeed, DJI has provided an official software development kit (Tello SDK 2.0) with a variety of functions and libraries to interact with the drone's various sensors and inner flight control systems. The drone communicates with an external PC workstation via WiFi, while the programming interface relies on text messages transmitted through the UDP port.

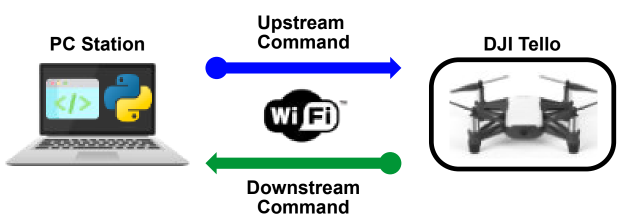

Fig. 2. Communication with the Tello Drone

As illustrated in Figure 2, communication with the Tello drone involves two data streams – one for the upstream command, and one for the downstream command:

- *The upstream command* allows to send a command to control the movement of the drone. The SKD 2.0 librairie includes a wide range of commands ranging from simple radio control (RC) commands to more complex maneuvers such as a flip.
- *The downstream command* allows access to the status of the drone and, more specifically, to the measurement of specific sensors. The essential data that can be acquired are accelerations, velocities, attitudes (pitch, roll, and yaw), battery level, and altitude.

The front camera image can also be acquired using another independent downstream command.

It should be noted that although there are different controls to control the movement of the drone, the most practical ones are the RC controls. Four standard RC commands allow users to control the drone around its four axes: forward/backward, right/left, up/down and yaw. Each RC command can range from -100 to 100; with 0 meaning zero speed, and 100 meaning maximum speed. The sign indicates the direction, e.g., forward (+) or backward (-).

2.3. Python Librairies for the Tello

There are various packages available to operate the DJI Tello drone using Python. Although different in terms of applications, these libraries use the Tello SDK and are structured according to the command protocol provided by the manufacturer. One of the most popular packages is the *Tello-Python*[1] library, developed by the DJI company. *Tello-Python* library includes a wide range of functions that allow users to control the drone's movements in a very simple way, using different pre-commands. This package also includes many example applications ranging from controlling the drone's movements using the keyboard keys to using the drone's camera to make a panorama.

Given the potential applications in scientific research, a community of enthusiasts has also developed other Python libraries to extend the possibilities of the Tello. Among them, the *DJITelloPy*[2] package provides a simplified interface for controlling the drone's movements, capturing video streams, and receiving telemetry data. This package also includes extended features enabling to control the drone in a swarm configuration. The package *TelloPy*[3], meanwhile, offers a more advanced interface for controlling the drone, including the ability to operate multiple drones simultaneously and access the sensors embedded in the drones. Another example is *easyTello*[4], a Python library created to provide users with a simple way to interface and send commands to the DJI Tello drone. This library, although more straightforward than the previous ones, has been developed to facilitate the learning of control techniques applied to drones.

In this paper, the *DJITelloPy* library was used as the basic interface to communicate with the DJI Tello drone. However, it should be noted that all other libraries can be used and adapted to reproduce the methodology presented in the following section.

3. DEMONSTRATORS FOR TEACHING FUNDAMENTAL CONCEPTS IN CONTROL

This section presents two application examples of how

[1] https://github.com/dji-sdk/Tello-Python
[2] https://github.com/damiafuentes/DJITelloPy
[3] https://github.com/hanyazou/TelloPy
[4] https://github.com/ezrafielding/easyTello

the DJI Tello drone has been used at the École de technologie supérieure (ÉTS) to illustrate fundamental concepts of control theory to undergraduate and graduate students. These applications aim to show students, through classroom demonstrations, all the steps in the design of a controller, from the development of a mathematical model using identification techniques, to the implementation of the controller on the Tello drone to verify its effectiveness.

3.1. General Approach

All demonstrations are performed live with students in class. The main idea is to provide the student with concrete applications of the theoretical elements they learn during the course. For this purpose, the first half of the class begins with the introduction of theoretical concepts, including essential mathematical equations and engineering principles. Then, in the second half, a demonstration of the material that has been taught is done with the students.

At each key step of the demonstrations, students are invited to perform the calculations simultaneously with the teacher and share their results with other students. In this way, students can put into practice the theoretical notions acquired previously. In addition to arousing students' curiosity and capturing their attention, such an approach promotes active learning by encouraging discussions and exchanges between students and the teacher.

Also, performing an in-class demonstration provide immediate feedback and allows students to see the results of their actions. This can help them better understand cause-and-effect relationships and improve their ability to analyze and adjust their approach.

To better illustrate how demonstrations are conducted in the classroom, two typical examples are provided in the following subsections.

3.2. Identification of a Mathematical Model of the drone

One of the fundamental concepts in control theory is system identification. Indeed, system identification is the process of determining a mathematical model that describes the behavior of a system based on experimental data. In most industrial applications, this approach is essential in order to obtain an accurate mathematical model of a process in order to subsequently design a controller to improve its performance. Regardless of the system to be modeled, system identification involves collecting data from the system, analyzing it, and using it to develop a model that can predict the system's behavior under different conditions. Within this context, the DJI Tello becomes a valuable tool to familiarize students with the basics of system identification and show them a concrete application through demonstrations.

In the following, a step-by-step application is provided to develop a demonstrator using the DJI Tello drone (or equivalent system) to teach the basic principles of system identification. Note that for the sake of simplicity, only the vertical movement of the Tello drone is considered, the other movements being assumed "blocked". Furthermore, it is assumed that the drone is controlled using the RC command corresponding to the vertical motion, and that the collected flight parameters are the vertical velocity $V_z(t)$, and the altitude $h(t)$ (estimated from the ToF sensor).

In this case, the drone can be represented schematically by the following block diagram:

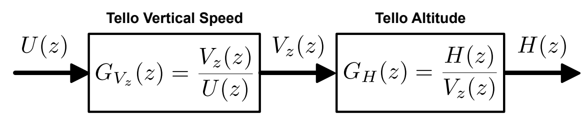

Fig. 3. Model of the Tello Drone using Two Transfer Functions

where $U(z)$ represents the input (i.e., the vertical RC command), while $G_{V_z}(z)$ and $G_H(z)$ are unknown discrete transfer functions used to model the vertical velocity and altitude, respectively.

To analyze the behavior of the Tello, and then propose a model structure for $G_{V_z}(z)$ and $G_H(z)$, it is necessary to send an input (i.e., RC command) to the drone and observe its reaction. Different types of input can be used: one-step, two-step, sine wave, etc. Thus, at this step of the demonstration, the students are asked to propose a type of input in order to test it in class and see how the drone behaves accordingly. This allows the students to understand better the specificity and complexity of each type of input.

Figure 4 shows an example of responses of the Tello drone to a two-step input. As can be seen, this maneuver aimed to send a first RC command of 50, to make the drone climb from an initial altitude of 75 cm to about 160 cm with a speed of about 42 cm/s. Then, the drone was held in hover at about 160 cm for two seconds by sending a zero RC command. Finally, a last RC command of -50 was sent to the Tello to bring it down to about 85 cm.

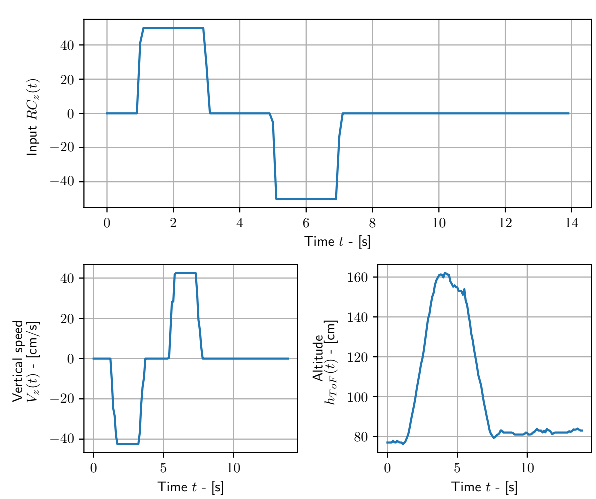

Fig. 4. Open Loop Restults for the Tello with a Two-Step Input

Based on the results in Fig. 4, students are invited to analyze the curves and draw a parallel with the movement of the drone they observed. The idea behind this exercise is to allow students to perceive the typical response characteristics that can be observed in any dynamic system, such as overshoot, time response, and steady-state error. Once the data is analyzed, students must then postulate a transfer function model capable of reproducing the observations. In this case, it can be seen that the vertical velocity can be approximated by a first-order transfer function with the following representation in the continuous domain:

$$G_{V_z}(s) = \frac{K_1}{1+\tau_1 s} e^{-\tau_d s} \quad (1)$$

where K_1 is the static gain of the transfer fuction, τ_1 is the time constante, and τ_d represents the time delay due to the communication between the Python script, and the drone.

Using a zero-order blocker technique, it can be then shown that the equivalent of the transfer function in Eq. (1) in the discrete domain can be obtained as follows:

$$G_{V_z}(z) = \mathcal{Z}\{B_0(s) G_{V_z}(s)\} \quad (2)$$

where $B_0(s)$ is the zero-order blocker transfer function, and $\mathcal{Z}\{F(s)\}$ denotes the *z-transform* of $F(s)$.

By arranging the expression of Eq. (2) and replacing the expression of the zero-order blocker, the following equation can be obtained:

$$G_{V_z}(z) = \frac{z-1}{z}\frac{1}{z^{\alpha}}\mathcal{Z}\left\{\frac{K}{s(1+\tau_1 s)}\right\} \quad (3)$$

where $\alpha = \tau_d / T_s$, and T_s is the sampling period.

Finally, by noting that the expression in brackets in Eq. (3) is a standard form, the discrete expression of $G_{V_z}(z)$ can be derived as follows :

$$G_{V_z}(z) = \frac{1}{z^{\alpha}}\frac{K\left(1-e^{-T_s/\tau_1}\right)}{z - e^{-T_s/\tau_1}} \quad (4)$$

where the sampling periode is assumed to be $T_s = 0.1$ s.

Starting from the expression in Eq. (4), students are asked to use a graphical technique to determine the unknown parameters of the model, i.e., K, τ_1, and τ_d. In addition, in order to challenge their programming skills, each student needs to test their model, and use the experimental data to validate their solution graphically.

An example of the solution obtained for the following transfer function:

$$G_{V_z}(z) = \frac{-0.4136}{z^3(z-0.5134)} \quad (7)$$

is given in Fig. 5.

This approach allows students to test several parameter combinations and thus understand the influence of each parameter on the response of the mathematical model. Similarly, it is possible to do the same work assuming a transfer function of order two or higher. The advantage of doing the calculations with Python is that it becomes easy to postulate a model with a high order in the continuous domain, and then use Python functions to obtain the equivalent form in the discrete domain, and simulate it to compare theoretical and experimental results.

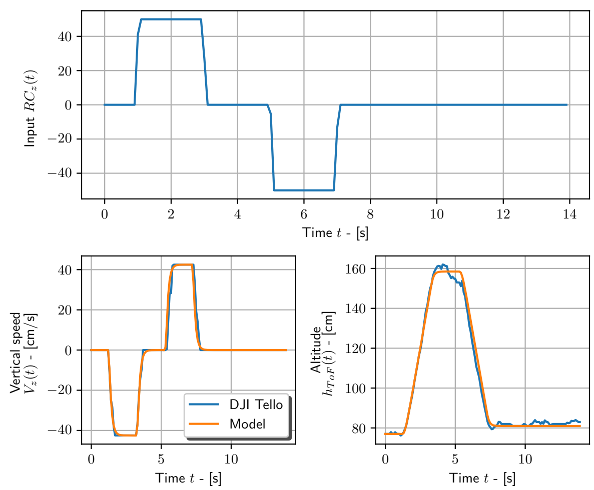

Fig. 5. Identification Restults for the Vertical Velocity of the Tello Drone

Using the same procedure as for vertical velocity, it can be shown that the following transfer function for the altitude can be obtained:

$$G_H(z) = \frac{H(z)}{V_z(z)} = \frac{-0.09412}{z-1} \quad (8)$$

which is mainly a discrete integrator with a gain of 0.94.

3.3. Designing a Controller for the Altitude

Once a model for the drone is identified, it becomes possible to use the resulting transfer functions to design a controller for the altitude (or the vertical velocity). For that purpose, a second demonstrator was developed to show students how to apply a specific technique learned in class to design a controller and verify its effectiveness through experimentation. The main advantage of this demonstrator is that it can be reused as needed to illustrate a different design technique or controller architecture.

Several control architectures can be considered depending on the class level or course being taught. For example, for undergraduate courses, classical type controllers such as Proportional-Integral-Derivative (PID) can be used, while for more advanced graduate courses, it would be more interesting to move towards robust type controllers. In any case, the application example presented in this section can be adapted as needed.

Without loss of generality, an architecture based on a Proportional-Integral (PI) controller has been considered as an example in this paper, as illustrated in Fig. 7,

Peer reviewed

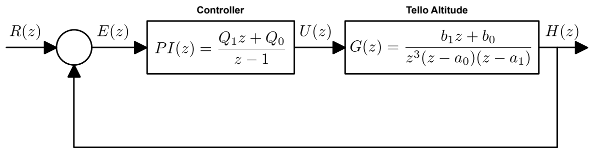

Fig. 7. Proposed Control Architecture for the Altitude of the Tello

where $\{Q_0, Q_1\}$ are the parameters of the controller to be tuned, and $\{b_0, b_1, a_0, a_1\}$ are constant coefficients determined based on the results obtained in Eqs. (7) and (8).

As with controller type or architecture, there are different design techniques for tuning the gains of a controller. This paper presents an example of an application using a method based on pole-zero cancellation and poles-placement. Nevertheless, the methodology presented can easily be adapted to other design techniques.

The following mathematical development is carried out in class with students' participation. However, in order to create a learning dynamic, students are encouraged to perform the calculation steps in parallel, or to use another design technique as long as they keep the same PI architecture for the controller. This approach favors exchanges between the teacher and the students, especially during the validation phase when it is interesting to compare and discuss the different solutions.

For the design of the controller, the expression of the PI is arranged as follows:

$$PI(z) = \frac{Q_1(z + Q_0/Q_1)}{z - 1} \tag{9}$$

With the new expression in Eq. (9), the zero of the controller can be used to cancel one of the two poles of the Tello. For instance, assuming that the zero of the PI is used to cancel the pole a_1, this implies therefore that:

$$\frac{Q_0}{Q_1} = -a_0 \Rightarrow Q_0 = -a_1 Q_1 \tag{10}$$

and the new architecture control simplifies to the one shown in Fig. 8:

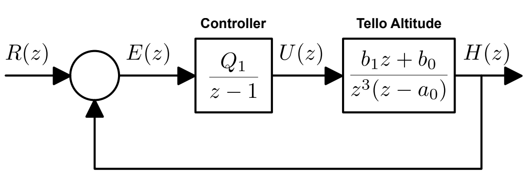

Fig. 8. New Control Architecture after Pole-Zeros Cancellation

By assuming that the delay represented by the term z^3 in the denominator of the altitude transfer function can be neglected, it can therefore be shown that the closed loop of the system can be obtained as follows:

$$\frac{H(z)}{R(z)} = \frac{Q_1(b_1 z + b_0)}{z^2 - (1 + a_0 - Q_1 b_1)z + Q_1 b_0 + a_0} \tag{11}$$

The result in Eq. (11) indicates that the closed-loop system behaves as a second-order transfer function whose characteristic polynomial is:

$$\Delta(z) = z^2 - (1 + a_0 - Q_1 b_1)z + Q_1 b_0 + a_0 \tag{12}$$

Assuming that the closed loop should have zero overshoots, and a time response of about 0.5 seconds, this implies, therefore, that the desired characteristic polynomial in closed loop should be:

$$\begin{aligned} \Delta_d(z) &= z^2 - 2e^{-\omega_n T} z + e^{-2\omega_n T} \\ &= z^2 - 2\beta z + \beta^2 \end{aligned} \tag{13}$$

where $\beta = e^{\omega_n T}$, and $\omega_n = 9.45$ rad/s (according to the 0.5 seconds time response requirement).

Equating the coefficients of the polynomial in Eq. (12) and Eq. (13), the following system of equations is obtained:

$$\begin{aligned} 1 + a_0 - Q_1 b_1 &= 2\beta \\ Q_1 b_0 + a_0 &= \beta^2 \end{aligned} \tag{14}$$

Finally, solving Eq. (14) for Q_1 and β gives:

$$\begin{aligned} \beta &= \frac{-b_0 + \sqrt{b_0^2 + b_1^2 a_0 + b_0 b_1 (1 + a_0)}}{b_1} \\ Q_1 &= \frac{\beta^2 - a_0}{b_0} \end{aligned} \tag{15}$$

Applying the results in Eqs. (10) and (15), the following parameters for the PI controller can be obtained: $\beta = 0.9791$, $Q_0 = 1.91$, and $Q_1 = -1.91$.

The results obtained after implementation on the drone using the *DJITelloPy* package are shown in Fig. 9.

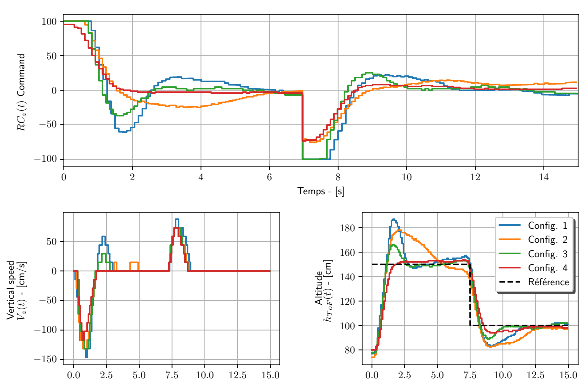

Fig. 9. DJI Tello Altitude Control Results for Different Controller Configurations

Figure 9 shows the results for several controller configurations. Configuration #4 (in red) was based on the parameters obtained from Eq. (15), while the others were designed based on different approaches proposed by several students. It is interesting to note that the controller for configuration #4 was able to reach the reference altitude of 150 cm with almost no overshoot and a time response of about

1 second instead of the expected 0.5 seconds. This difference can be attributed to the time delay that was neglected in the design of the controller. This observation is very interesting because it allows students to to understand better the effect of a delay in a control loop. Considering a delay in the design of a controller increases the order of the system and thus makes the design task more complex. Conversely, neglecting the delay reduces the order of the system and thus simplifies the controller design but has a price on the performance of the closed-loop system that results in either increased response time or, in the worst case, instability. Nevertheless, the results remain very satisfactory, and it can be concluded that the controller performs well on the DJI Tello Drone.

Regarding the other configurations, it may be interesting to observe the different behavior of the drone. This analysis of the results, combined with the visualization of the drone's behavior during the experimental tests, clearly allows the students to understand better the effect of the controller parameters on the control energy and the response of the drone. Configure #1 has, for example, a very large overshoot (about 25%), which can be explained by a very high value of the controller integral gain. Indeed, a high integrator gain reduces the steady-state error but introduces oscillations in the transient dynamics. This aggressiveness can also be seen very well on the RC command outputted by the controller, which saturates very quickly at 100 and requires a lot of energy. The other configurations (i.e., #2 and #3) were selected by gradually decreasing the contribution of the integral gain of the controller. As expected, it can be noticed that there is less overshoot, and the controller becomes less aggressive.

4. STUDENTS' FEEDBACK AND APPRECIATION OF THE DEMONSTRATORS

The demonstrators presented in this paper were first used in an undergraduate course on digital control. They were then used in a graduate course on industrial process modeling and control. In both courses, the number of students enrolled was approximately 35.

Because the demonstrators are still in the testing and improvement phase, no written survey has yet been conducted with students to gather their feedback. This survey will be conducted once all the demonstrators have been used in enough courses in order to obtain a large sample of students to conclude their effectiveness. Nevertheless, in the meantime, many comments were collected at the end of the courses in which the demonstrators were used. Overall, all students greatly appreciated having this type of demonstration and, in the majority of cases, the demonstrators helped them better understand the theoretical concepts seen in class. Although none of the students enrolled in these courses had an aeronautical background, they all found using a drone to illustrate some of the general concepts of control theory to be a very innovative approach. Another comment that came up quite often was about the benefit of live demonstrations, as students can see the results of their actions and get immediate feedback.

Finally, it is important to remember that one of the reasons to choose a drone like the DJI Tello is its price, around 99$ (USD). This price is very affordable compared to other types of equipment for education that cost a few hundred to a few thousand dollars. Some students have even expressed interest in getting one Tello drone to learn how to program it and try to apply different control techniques to improve their knowledge.

CONCLUSION AND FUTURE WORK

In this paper, a pedagogical approach using a DJI Tello drone as an interactive teaching platform in the field of automatic control engineering was presented.

The main objective of this paper was to highlight the fact that it is possible to use an inexpensive drone to allow teachers to illustrate fundamental concepts related to the field of control theory. In addition to the affordability of the Tello, one of its great assets is the ability to control it using a programming language such as Python.

Two demonstrators have been presented in this paper as application examples. The first demonstrator aims to show how the Tello drone can be used to train students in system identification. The second demonstrator, on the other hand, illustrates a method of designing a controller from theoretical design with a technique taught in class to implementation and analysis of the resulting solution on the drone.

In addition to the practical aspect of these demonstrations, it is important to emphasize the cognitive aspect. Indeed, carrying out the demonstrations in class and encouraging the students to participate allows for creating a dynamic that favors the exchange between the teacher and the students. Moreover, the "entertaining" and "surprising" aspect of the drone arouses the curiosity of the students.

Other demonstrators are being developed to encourage the use of the drone for educational purposes and to enable a wider range of applications. The main objective of these new demonstrators will be to show students, through dynamic examples, how to use the drone's camera to perform image processing, but also vision-based navigation. In addition, Python is a programming environment that offers a large number of libraries adapted to artificial intelligence. Thus, several graduate students are working on a project to develop demonstrators to illustrate how the drone's movement can be controlled using artificial intelligence-based techniques such as neural networks or fuzzy logic.

Acknowledgements

This study was performed at the École de technologie supérieure (ÉTS) in Montreal (QC), Canada. The authors would like to thank the ÉTS funding program PSIRE-ENS for contributing to the development of the demonstrators. Thanks are also due to the ÉTS teaching support office

(Bureau de Soutien à l'Enseignement, BSE) for their support.

References

[1] Blevins, T. L., and McMillan, G. K., "Bridging the Gap between Universities and Industry," *2009 American Control Conference*, St. Louis, MO, USA, 2009. DOI:10.1109/ACC.2009.5160750.

[2] Wallin, J., Isaksson, O., Larsson, A., and Elfström, B.-O., "Bridging the Gap between University and Industry: Three Mechanisms for Innovation Efficiency," *International Journal of Innovation and Technology Management*, Vol. 11, No. 01, 2014, p. 1440005. DOI:10.1142/S0219877014400057.

[3] Shinskey, F. G., "Process Control: As Taught vs as Practiced," *Industrial & Engineering Chemistry Research*, Vol. 41, No. 16, 2002, pp. 3745–3750. DOI:10.1021/ie010645n.

[4] Bordeneuve-Guibé, J., and Duvernois, P., "A Global System Approach as a Tool for Control Education," *IFAC Proceedings Volumes*, Vol. 30, No. 12, 1997, pp. 133–137. DOI:10.1016/S1474-6670(17)42777-7.

[5] Govind, A., and Selva Kumar, S., "A Comparative Study of Controllers for QUANSER QUBE Servo 2 Rotary Inverted Pendulum System," *Advances in Electrical and Computer Technologies*, Springer Singapore, Singapore, 2020, Chap. 672.

[6] Kathpal, A., and Singla, A., "SimMechanics™ Based Modeling, Simulation and Real-Time Control of Rotary Inverted Pendulum," *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, India, 2017. DOI:10.1109/ISCO.2017.7855975.

[7] Tejado, I., Torres, D., Pérez, E., and Vinagre, B. M., "Physical Modeling Based Simulators to Support Teaching in Automatic Control: The Rotatory Pendulum," *International Federation of Automatic Control (IFAC)*, Vol. 49, No. 6, 2016, pp. 75–80. DOI:10.1016/j.ifacol.2016.07.156.

[8] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W., "OpenAI Gym," 2016. DOI:10.48550/ARXIV.1606.01540.

[9] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y., "Quigley, Morgan, et al. "ROS: An Open-Source Robot Operating System," *ICRA workshop on open source software*, Vol. 3, No. 3.2, 2009, p. 5. .

[10] Karalekas, G., Vologiannidis, S., and Kalomiros, J., "EUROPA: A Case Study for Teaching Sensors, Data Acquisition and Robotics via a ROS-Based Educational Robot," *Sensors*, Vol. 20, No. 9, 2020, p. 2469. DOI:10.3390/s20092469.

[11] Giernacki, W., Rao, J., Sladic, S., Bondyra, A., Retinger, M., and Espinoza-Fraire, T., "DJI Tello Quadrotor as a Platform for Research and Education in Mobile Robotics and Control Engineering," *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, Dubrovnik, Croatia, 2022. DOI:10.1109/ICUAS54217.2022.9836168.