Modular ESC Motors Drone

Alessando Assirelli, *231685*
Marco Perini, *229203*
Erik Mischiatti, *233242*
Yahia El Ourrak, *224835*

***Abstract*—In recent years, the use of drones has increased in a wide range of applications, including delivery, inspection, and mapping. However, the limited flight time of drones and the need for rapid battery replacement or recharging has become a major challenge for their extended use.**

In this paper, we present an innovative method for creating a communication link between a flight controller (FC) and an electronic speed controller (ESC) in a drone using ultra-wideband (UWB) technology. Our approach utilizes a UWB module to transmit control signals between the flight controller and the ESC.

We present a solution to the realization of a modular drone using UWB technology along with an an analysis of the communication system. We then report the results demonstrating the effectiveness of our method.

We believe that our proposed method, utilizing UWB technology, will be a key step forward in the development of efficient and reliable drones that doesn't have any geometric constraint.

I. Introduction

DRONES, also known as unmanned aerial vehicles (UAVs), have become increasingly popular in recent years for a variety of applications. The development of micro-electro-mechanical systems (MEMS), sensors, fabrication, navigation methods, remote control capabilities, and power storage systems have enabled to design and manufacture of a wide range of UAVs which can be used in many circumstances and tasks. Therefore, UAVs vary widely in their sizes, configurations, and performances.
Drones are typically controlled by a flight controller, which manages the drone's flight by adjusting the speed of its motors.

The main components of a UAV are:

- UAV Airframe: it refers to the physical structure of the UAV, which provides the framework for attaching various components. It also gives the UAV its shape, size, weight, and aerodynamic properties.
- Flight Controller: it uses sensors such as accelerometers, gyroscopes, magnetometers and GPS to estimate the UAV's attitude and position. It is responsible for controlling the UAV motion and stability by controlling the motors speed.
- Payload: it is any equipment or devices that are carried by the UAV. This can include cameras, external sensors, delivery packages or other specialized equipment depending on the specific application of the UAV.
- Propulsion system: it provides the necessary thrust to enable the UAV to fly and perform its intended functions. It is made up by the Electronic Speed Controllers (ESC), the motors and the propellers. The choice of the propulsion system depends on the desired characteristics and mission of the drone.

As we said, nowadays, drones are used in a variety of applications and this represents a great challenge on designing a drone. In fact when designing a drone we choose its dimension, the number of rotors, the size of the propellers and motors, the thrust required and the battery to use. These drone's specification are selected depending on the constumer's application, considering its flight time, the payload, the maneouvrability and the safety needed. It comes by itself the necessity of a large number of different drones able to perform the specific applications effectively and with an adeguate performance.

Our proposal to this challenge is to take an alternative approach on designing the drone: instead of designing several drones that depends on the application, we design a drone that is able to meet the requirements of the different applications. This is possible by designing a modular drone that easily permits changes on its specification, such as number of rotors or the airframe. While a traditional drone has a fixed frame and its components are hardly interchangable, a modular one can be easily assembled and disassembled. This design is characterized by switchable parts and components, allowing greater flexibility and adaptability in the types of tasks that the drone can perform. A modular design also enables the possibility to quickly replace or upgrade components as needed. Our idea of a modular drone is to have several separate modules that may be attached to an arbitrary object that would become the UAV airframe.

This approach may be implemented in different ways. One possible way is to design the drone that can have an arbitrary number of rotors and is able to select a subset of them depending on the flight mission and the payload. Another possible design is to have an adaptable drone: namely a drone where the position of the rotors or the length of their arms are adjustable, always depending on the drone's desired application. Finally another possible way, that is also in the line with our idea of modular drone, is to design a drone made up off different separate modules attached to the payload or to an airframe. These modules may change in number depending on the application and their communication may be wired or wireless.

Our solution is based on the idea to have a master module which contains the flight algorithms, that communicates wirelessly with an arbitrary number of modules containing the propulsion system. Our work concentrates on the communication link between the master and the slaves exploiting the UWB technology. In fact there have been recent innovations in UWB technology that show potential for providing reliable and robust communication links in a variety of applications. Our project aims to utilize this technology in order to create a communication system for drones that can transmit control signals between the flight controller and the ESC, with the goal of having the most suitable solution for each application. For example, if we consider a delivery application were the drone need to deliver packages, its dimension and propulsion thrust condition the type of packages that is able to transport. But with our solution the number of modules will be proportional to the weight of the package, making it possible to transport a variety of packages. Furthermore, given that the modules are connected wirelessly the dimension of the package or its shape are not a problem as the modules can be attached directy to it or to an adjustable airframe, without worrying about any cables that connect them.

The main steps involved for the completion of this research projects are:

- Assembled the drone;
- Installed the firmware and calibrated ESCs and sensors;
- Characterized the PWM signals generated by the flight controller;
- Created the communication scripts and installed the receivers and transmitter on the drone;
- Simulated the new communication system on the Gazebo virtual environment;
- Performed real flight tests and gathered diagnostic data;
- Validated solution and conceptualize future work.

This paper is organised as follows:

Related Work is re-viewed in Section II. In Sections III we present our work, where we show the technology used, the hardware and we outline the research and development process of our solution.

On the HW side, we devoted sections to the UWB, master and slave modules. Then we present a possible housing solution for the slave modules.

On the SW side we focused on the simulation of the wireless connection using Gazebo, we described the key points in the communication script, the battery consumption as a crucial point on the actual efficiency of the design, and everything concerning the use and analysis of the PWM signal used to control the motors.

In Section IV, we show a characterization of our communication system. We evaluated its performances and we included it in the simulation using Gazebo. Finally we present and discuss the results collected in both the wired and our wireless solutions in a real flight test.

The last section, V, concludes the paper and presents possible future develpments highlighting pros and cons of our implementation.

II. Related Work

There has been different research papers published on this subject, describing different approaches and techniques for implementing modular drones or adjustable drones, where physical reconfiguration on the drone are permitted. These papers cover topics such as the design of the drone, the theory behind those designs, their implementation and the experimental results obtained.

Some papers compare modular and traditional drones operating performances and suggests that a modular approach can save time and energy during flight missions [1]. There are many different approaches to the design of the drone in order to make it modular or adjustable on the present state -of-the-art works, but they can be grouped in few main strategies. The first one is to make the drone easily reconfigurable, so that it is possible to move the arms of the rotors or simply adjust their length [2] [3] [4]. Another approach is to combine small drones, that can fly alone or together to make a complex structure able to fly. The different combination of shapes of the structure are determined by units positions and those shapes impose the performances of the structure [5]. Finally there's the modular drone approach, where different modules are combined to form a drone capable of flying. This solution is possible by having a main body where it is possible to attach an arbitrary number of actuators in different positions [4] [6] or by having different modules that are able to fly only when they are connected [7]. One example is the Distributed Flight Array [8], in which all modules have a hexagonal structure equipped with a propeller, motor, battery, omnidirectional wheels, sensors, and control board.

Our solution suggest a similar approach to the modular drone with a main body where the different actuators can be attached to, but in our case the main body is a custom airframe, or even the payload, depending on the application. On the main body we place another module hosting the flight controller which commands the actuators. This solution is enabled by a wireless communication between the master module and the slaves.

In fact some works explore the feasibility to control the speed of DC Motor wirelessly through PWM technique [9] [10]. However these works limit themeselves on the feasibility of this approach of control but don't examine its performances on a real application. So our works not only follows this approach of controlling the actuators wirelessly, but also its performances on the drone by analysing the experimental results.

III. Description of the project

Our project consists on developing a wireless communication between a flight controller of a generic UAV and its propulsion system. The control signals generated by the flight controller are sent to the motors using a unidirectional wireless connection, thus removing the cables constraint. The architecture we propose to realize a modular drone relies on two types of modules. The first one is the master module which consists of a transmitter module which reads the signals that a flight controller generates and send the information via UWB. The second one is the slave module which consists of a receiver module, a battery, an ESC and a motor with its propeller. The receiver reads the messages sent by the master

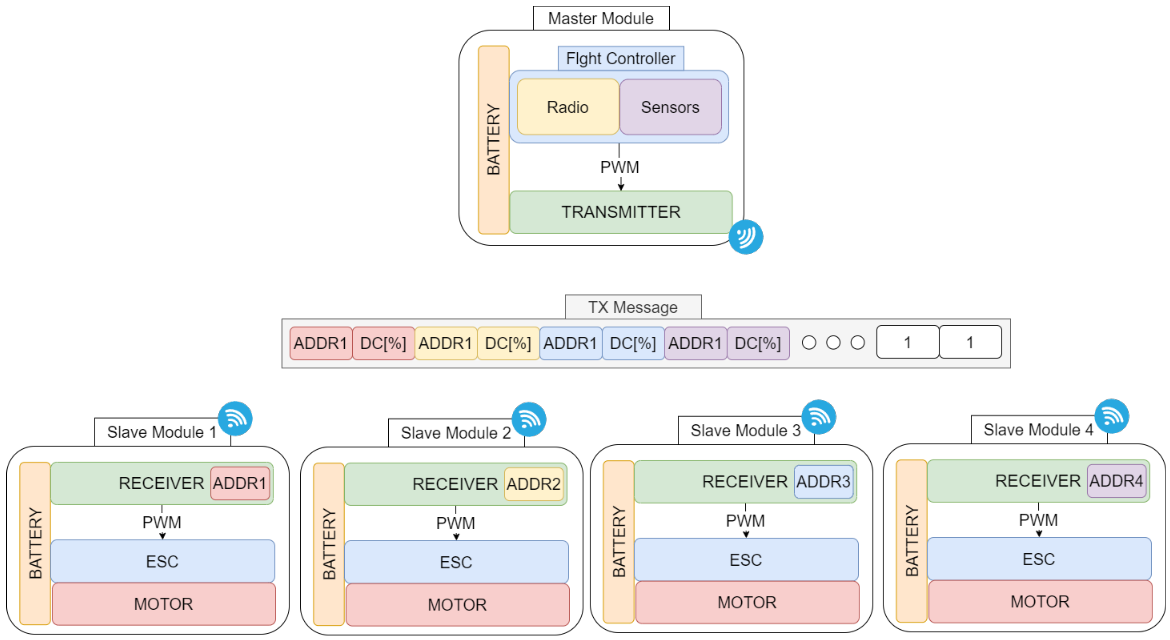


Fig. 1: Modular drone architecture proposal

and generates the PWM for the ESC. Figure 1 outlines the architecture we propose in order to realize a modular drone. Each block is then discussed in detail in the following sections. With respect to the architecture of Figure 1 in our work the battery is actually common to every module, but the concept remains the same.

The main focus of the project is to undestand if this design is feasible, how much is the delay introduced by the communication system and how it affects the performance of the UAV. To do so we decided for a centralized architecture which allowed us to use an off-the-shelf flight controller and focus just on the communication. The flight controller, thanks to the data of various sensors such as accelerometers, gyroscopes, magnetometers, barometers, and GPS performs different actions: stabilization, navigation to a specific location or by following a predetermined flight path through waypoints or navigation by following commands received from the operator via a radio controller. Generally the flight controller uses as control algorithm a PID control which adjusts the motor speeds by sending PWM signals to the ESCs. In our project the signal is not sent directly to the ESCs, instead the information about the duty cycle is collected by the MCU of the transmitter module and then sent via UWB to the receiver modules. The message contains the address of every receiver followed by the duty cycle it has to be reproduced, as shown in Figure 1.

A. *UWB Technology*

Ultra-Wideband (UWB) technology is a wireless communication technology that uses extremely short-duration, low-power radio pulses to transmit data over a wide frequency range. It is characterized by its ability to transmit data at very high speeds over short distances, while consuming minimal power. UWB is used in a variety of applications including high-speed data transfer, indoor positioning, radar imaging, and wireless sensor networks. It is also commonly used in applications that require low power consumption and high levels of security [11][12].

UWB technology revolutionized wireless communications with unparalleled convenience and mobility in devices for home and office. Ideal for short-range WPANs, UWB is the go-to solution for transmitting high-bandwidth data like video and audio wirelessly across multiple devices.

UWB offers major advantages, including efficient use of existing radio service spectrum without interference. Its large transmission bandwidth offers immunity to interference effects and improved multipath fading robustness, making it reliable and less prone to signal degradation.

Related to our project, we can highlight the pros and cons of this technology:

- Very high data rate, which allows for the transmission of large amounts of data over short distances quickly;
- Low power consumption, which means that it consumes less power than other wireless technologies, making it ideal for battery-powered devices;
- It can operate using spectrum already occupied by existing radio services without causing interference, which makes it an efficient use of scarce spectrum resources;
- UWB technology can be susceptible to interference from other wireless devices operating in the same frequency range.

B. *Master Module*

The modular drone is composed by an arbitrary number of slave modules and one master module. The master module consists of three components:

- **Flight Controller (PixHawk4)**: it generates the control signals for the motors. It generates one independent PWM signal per channel, so in the case of a quadcopter four

PWMs. The PWM signals have a fixed frequency (400 [Hz] in our case) and they are synchronized so the rising edges of every channel occur at the same time. The falling edges are of course dependent on the duty cycle (DC) of the specific channel. The minimum DC, which correspond to 0 [rpm] is 40% (i.e. minimum duration of 1[ms]) and a maximum of 80% (i.e. minimum duration of 2[ms]) which corresponds to full throttle as shown in Figure 2.

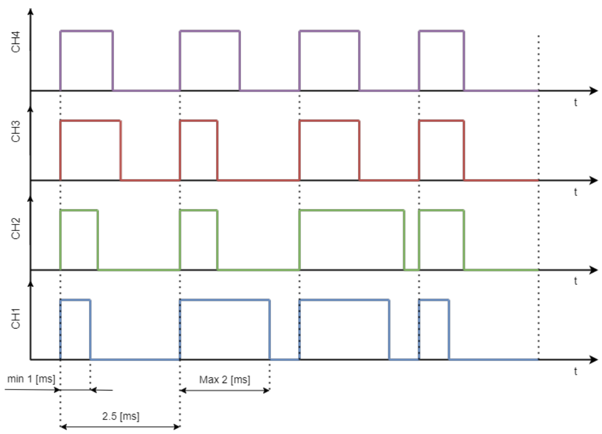


Fig. 2: Quadcopter FC PWMs example

- **Radio receiver**: it receives the commands from the remote control and tranfers them to the flight controller.
- **Transmitter Module (DWM1001 module)**: it's wired to the flight controller and it's made up by a Nordic nrf52832 MCU and a DWM1000 UWB transceiver. Its purpose is to acquire the PWM signals from the FC and send the duty cycle information to the receiver modules.

We implemented the PWM acquisition through Programmable Peripheral Interconnect (PPI). PPI allows to trigger a task in one peripheral as result of an event occuring in another peripheral while excluding the CPU from these operations. The synchronization clock of PPI is 16 [MHz]. We dedicated one timer for each PWM channel. The interconnections are between an event on each pin connected to the FC and the Capture Compare (CC[0]) register of the associated timer. In this way when a rising or falling edge is detected, the timer counter is automatically and almost immediately saved in the register. In order to retrieve the information we need, which is the duty cycle, we have to distinguish between the time the rising and falling edges occur. To do so when an edge is detected the counter value is saved in the register without the intervention of the CPU, then the CPU enters in an interrupt routine which is designed to distinguish the edge and to save the counter value in memory. The duty cycle of the signals are then computed by the simple relation:

$$\text{DC} = \frac{T_{fall} - T_{rise}}{T} \quad (1)$$

where T_{fall} and T_{rise} are the times at which the falling or rising edge occur and T is the period of the signal which is 2.5 [ms] in our setting. In order to transfer the PWM informations to the receivers we create a message made up by the adress of every receiver followed by the duty cycle that must be replicated. The message is sent at the end of every input PWM period, which is every 2.5[ms].

C. Slave Module

The slave modules are the ones that can be added in arbitrary number to the airframe in order to increase the maximum payload or flight distance. They are made up by a battery, an ESC, a motor with the propeller and by the receiver module. Also in this case the communication module is the DWM1001. In the slave module the PWM generated by the FC must be replicated. To do so, the output GPIO of the MCU is connected to its TIMER[0] which is configured in order to provide a fixed frequency square wave of 400 [Hz]. The timer gets started when an initialization signal is sent by the transmitter. That event is a sync signal and when it is received every transmitter starts its timer.
When a command is sent by the transmitter (i.e. every 2.5[ms]), every receiver looks for its address in the message and reads the desired DC. To apply the new DC, the CPU changes the value of the CC[0] register which causes the duty cycle to change.

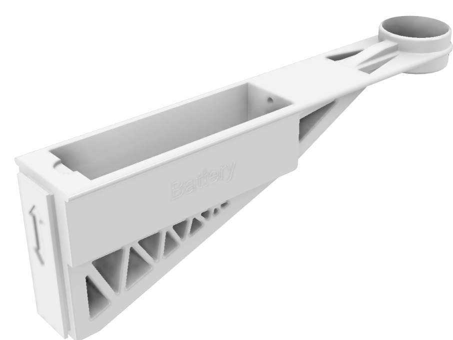


Fig. 3: Concept module rendering

D. Simulation

Hardware-in-the-Loop (HITL) simulation is a valuable tool for testing and validating flight control software in a virtual environment, which emulates the behaviour of a real-world system. In HITL, sensors readings are generated by the simulated environment while the calculations to control the drone are made using a real flight controller connected through serial communication. This method allows developers to test flight algorithms under realistic conditions, without the risks and costs associated with testing on existing hardware.
To simulate the wireless connection between the flight controller and ESCs using UWB modules, we created a Gazebo plugin to introduce a configurable delay in the drone PWM transmission. The delay was modelled as a FIFO buffer (Figure 4) as it was necessary to simulate the latency introduced by this new system and to evaluate the performance of the flight control software.

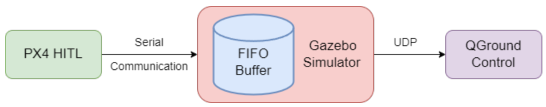


Fig. 4: HITL simulation stack with FIFO buffer

E. Power Consumption

In general, UAVs rely on batteries to power their electronic components, including the propulsion system, flight controller, sensors, and communication system. This is possible by using the Power Distribution Board (PDB), a printed circuit board that distributes the power from the battery to all the different components.
An important aspect, when designing a UAV and choosing the onboard battery, is the power consumption since it affects the flight time. The main factors to deal with are: the payload, the components that need to be supplied and the propulsion system. Considering these factors we can decide the battery capacity needed to be used.
By choosing the modular drone design, the consideration to be done regarding the power consumption may seems to change. This because not only we need more components that need to be supplied, such as the trasmitter and the receivers, but also we need to consider the different ways the PWM signals can be sent to the ESCs. However those changes can be considered negligible as the power consumed by the electronic components of the drone are on the order of a few watts, meanwhile the power needed for the propulsion system to make the drone fly are in the hundreds of watts [13]. So regarding the power consumption our solution practically consume the same amount as the other drones. The only consideration to make is that the modular drone is composed of an arbitrary number of separate modules, so each of them need its own battery.

IV. Experimental results

A. PWM replay scheme

In order to verify the effectiveness of our solution, wich have been discussed in Subsection III-B and III-C, we designed two tests. The first one is intended to assess the maximum allowable number of modules to reliably capture the PWMs, which depends on our acquisition technique, while the second is to characterize the overall delay introduced by this system. The measurements have been collected using a PicoScope 5444D MSO oscilloscope and the plots have been realized by importing the raw data on Matlab without any preprocessing.

Δt	Value	Description
$t_2 - t_1$	16[us]	time needed to enter in the ISR
$t_3 - t_2$	7[us]	time needed to get on which channel and edge occuredand to store its CCR[0] value in memory
$t_3 - t_1$	23[us]	total time to store CCR[0] value of Channel 0 in memory
$t_4 - t_3$	15[us]	total time to store CCR[0] value of the other channels in memory
$t_5 - t_1$	68[us]	total time to store four channels

TABLE I: Delays summary of Figure 5

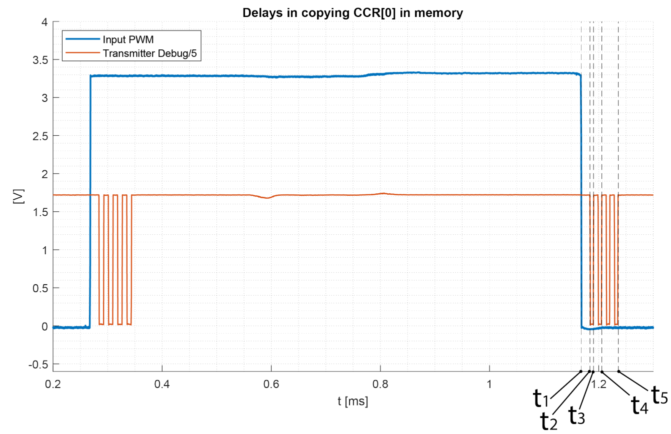


Fig. 5: Delays in interrupt

1) Evaluation of the maximum number of modules: As previously discussed, when an event on a GPIO occurs, the timer counter is automatically saved on the CC[0] register. This happens almost instantaneously since the PPI clock is 16[MHz]. The register must be read and saved in memory before the opposite edge occurs, since it is overwritten in both the rising and falling case.
The FC has a frequency of 400[Hz] with a period of 2.5 [ms]. The minimum duty cycle of the pwm generated by the FC is 40% and the maximum is 80%. The measurements must be completed by reading every CC[0] register of every channel within 20% of the period, which is the smallest delta and corresponds to 500[us].

We set up an experiment with the transmitter reading four PWM signals from the FC. We measured the time needed to store the counter register of every channel in memory. To do so we lower the value of a debug pin when the CPU enters in the corresponding ISR and raise it when the reading of the channel is complete. The measure will thus contain also the time needed to toggle a pin that we consider negligible. Figure 5 shows the results of this experiment. Only one input PWM signal is shown as the other three are synchronized. Table I reports the values of the delays.
The amount of time needed to store the counter in memory is made up by two terms. The first one is the time needed to enter in the ISR which is denoted as $\Delta t = t_2 - t_1$. The second is the time needed to figure out which channel generated the interrupt and to copy the counter value in memory which is $\Delta t = t_3 - t_2$. The time $t_3 - t_2$ remains constant whichever is the number of channels used, while the time needed for the CPU to start executing the ISRs changes. In particular, if the signals are synchronized, as in the FC case, the CPU will receive many interrupts at the same time. It can execute only one ISR at a time, so the requests will be accumulated. To start executing the first ISR it takes $t_2 - t_1$, while to serve the others it will take less. In conclusion to calculate how much time is needed to store in memory all the CCR[0] values we should consider: 23[us] for the first channel, plus 15[us] for each additional signal.
Since we cannot exceed 500 [us] in order to not lose any

edge of the input signal, we found out that the maximum allowed number of modules is:

$$\text{max n. channels} = \frac{500[\text{us}] - 23[\text{us}]}{15[\text{us/channel}]} + 1 > 31 \qquad (2)$$

which is more than enough for standard applications. For four channels the total time needed to store the counter values in memory is 68 [us].

This description refers to the collection of the falling edges. From Figure 5 it might be noted that the process of storing the CCR[0] values is slower in the rising edge case. The reason why the analysis has been carried out on the falling edge case is that the time constraints are so tight that this has to be consider the worst case.

2) Total delay introduced: Another important information is the total delay, which has to be measured from the time that the FC sends a signal to the time at which it is replayed on the receiver side. To evaluate it we set up another experiment in which the four channels of the FC are read by the transmitter, sent and then replicated on the receiver side. Figure 6 shows the results of this experiment.

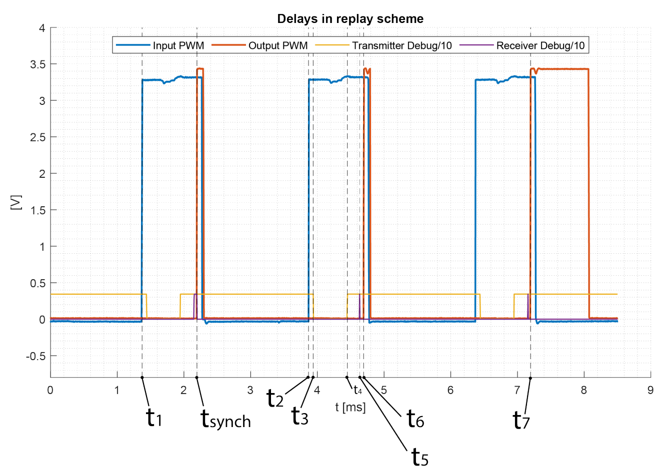


Fig. 6: Delays in the replay scheme

As can be seen, the overall delay contains different contributions that are summarized in Table II. The figure shows one PWM channel of the FC which has to be replicated, the replicated PWM on the receiver side and one auxiliary signal per module which are used for debug and here discussed. Note that the debugging signals are scaled down by a factor of ten for better clarity in the plot.
By design our acquisition scheme needs to wait the end of one period of the PWM signal, which is 2.5 [ms].
When the period ends the duty cycle informations have been already collected. The starting of the transmission has a delay $\Delta t = t_3 - t_2$ due to the fact that the transmission has to be started just after the period ends, but at the same time the MCU has to serve the interrupt service routines that read the CC[0] registers of the timers and store them in memory. This delay is captured by lowering the debug pin on the transmitter when the preparation of the message starts.
After that the CPU prepares the message containing the addresses of the receivers followed by the DC informations and sends it to the receivers, once it has done it raises its debug pin. The amount of time needed to complete these operations is $\Delta t = t_4 - t_3$.
The message arrives to the receiver with a delay that depends on the distance from the transmitter. In this case the delay is collected by rising the value of a pin on the receiver side when the reception starts and by lowering it when it ends. The time needed to receive the message and parse it is $\Delta t = t_5 - t_4$.

The Figure shows the initialization stage of the scheme, the PWM on the receiver side starts when it receives the synch signal sent by the transmitter (this time is denoted as t_{synch} in the Figure). The first period of the output PWM thus starts from an arbitrary value different from the one produced by the FC. When the rising edge of the second period of the input PWM comes (which is denoted as t_2 in Figure 6), the transmitter sends the message containing the information about DC_{I1}. Even if the message is received and parsed before the receiver starts the second period (T_{O2}), DC_{O2} is not updated. To see the duty cycle collected initially (DC_{I1}) on the receiver side, we need to wait untill the third period T_{O3}. This means that the duty cycle is replicated with more than two periods of delay.

Δt	Value	Description
$t_3 - t_2$	72[us]	delay in starting the communication
$t_4 - t_3$	509[us]	time needed to prepare the message and complete the transmission
$t_5 - t_4$	189[us]	time needed to receive the message and parse it
$t_6 - t_5$	59[us]	time margin on the next pwm period
$t_7 - t_1$	5831[us]	total delay from of the system

TABLE II: Delays summary of Figure 6

B. Simulation results

Before actually testing our transmission system on a real UAV, we opted to conduct some simulated flights in the Gazebo environment, varying the possible latency introduced by the wireless communications. By adjusting the delay in the Gazebo plugin, we were able to test the software's ability to stabilize and control the drone under different latency conditions. The graphs in Figure 7, show the pitch, roll and yaw angle values both with zero and 6 [ms] latency; the norm is plotted as well to emphasize the differences which are not so evident due to the small delay. Our results showed that the flight control software was able to maintain stable flight performance even with significant delays in the PWM transmission. We therefore assessed that a latency of 6 [ms] should not be a problem for the stability of the UAV.

C. Real flight test

After the preliminary experimental results obtained, we analyzed the behaviour of our drone. The flight test conducted had the aim to highlight the general behaviour of our drone in two different setup: the traditional one and the one with ESC

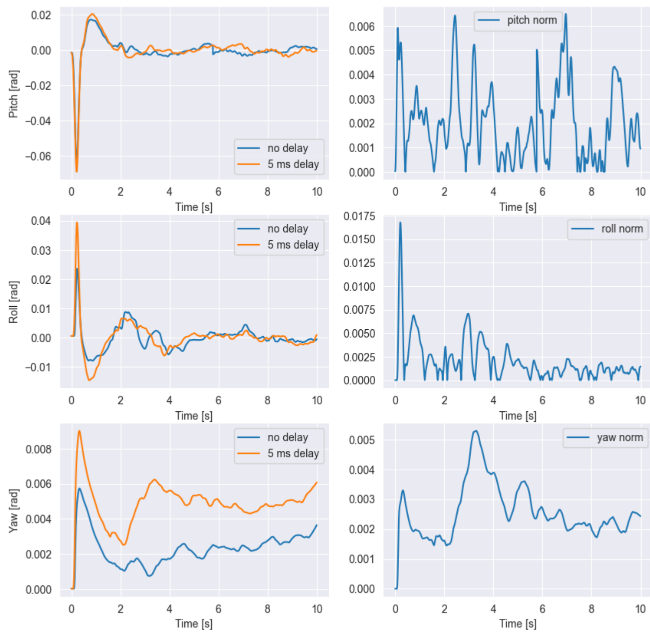
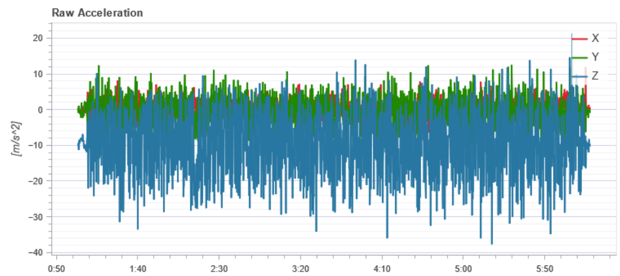
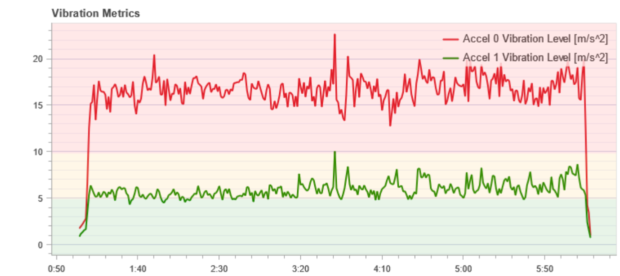


Fig. 7: Simulation pitch, roll and yaw angles



(a) Drone's raw acceleration

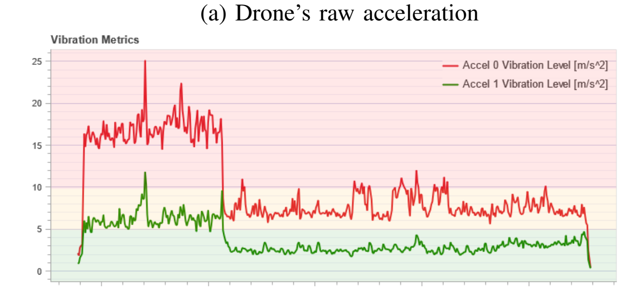


(b) Vibration level

Fig. 9: Flight test with wireless ESC connection setup

modules connected wirelessly. This is done in order to see if the drone works properly in both setup and to compare them. The general results that emerges from the flight test is that even thought both the setups work properly, with the wireless ESC connection, visually the drone vibrates a lot compared to the other setup. This can be seen also from the flight data. In fact analysing the log of the flights using the online tool Flight Review we obtain the following plots in Figure 8 and 9.

(a) Drone's raw acceleration



(b) Vibration level

Fig. 8: Flight test with traditional setup

To interpret those results we need to know that in general as a rule of thumb if, in the raw acceleration's plot, the z-axis graph is touching the x/y-axis graph during hover or slow flight, the vibration levels are too high. In our case this happens at the start with the traditional setup but it's more evident in the wireless setup where the vibration level is high during the whole flighting time.
Some vibrations are to be expected given the nature of our drone: a prototype designed to analyze the feasibility of a modular drone and not to be efficient like a commercial one. However in the wireless ESC connection the vibrations level is too high and needs to be investigated. High vibration levels can lead to different kind of problems: a less efficient flight and a reduced flight time, position estimation failures that potentially results in fly-aways or an inability to tune the vehicle tightly, resulting in degraded flight performance.

In order to understand the nature of these vibrations and why they occur we examined the behavior of the drone in the two setups when it's hovering. From the flight tests conducted we select a time interval where the drone, in both the setups, is hovering: to select this time interval we search in the log data when approximately all the local velocities of the drone are zero(Figure 10).

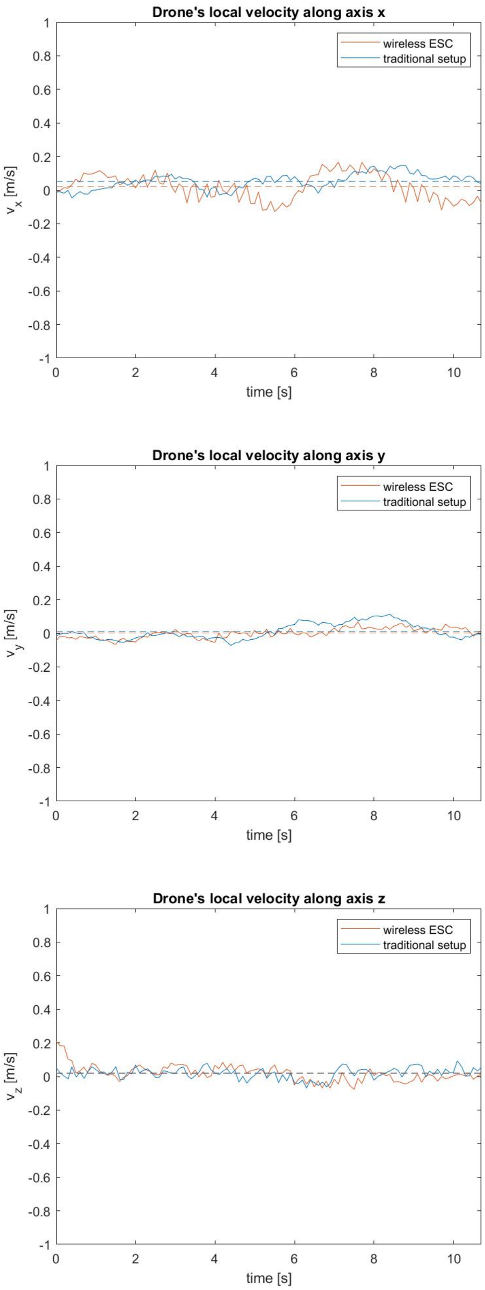


Fig. 10: Drone's local velocities

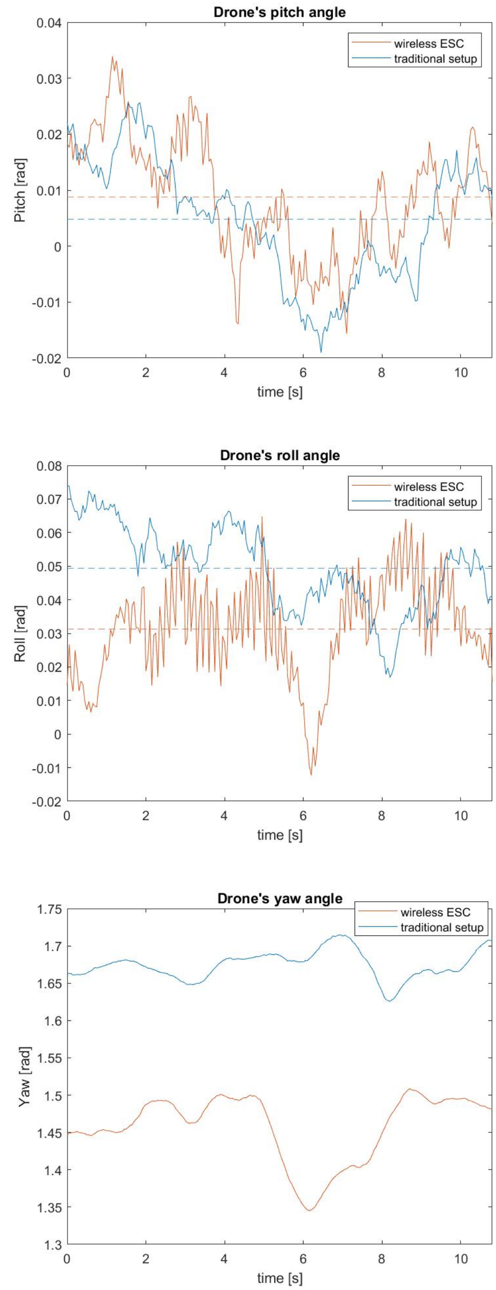


Fig. 11: Drone's attitude

After establishing the time interval we analyze the drone's attitude, seen in the Figure 11. From the figure we see that in the wireless connected ESCs we have more oscillations than with the traditional setup along the pitch and roll angles, showing off how less stable this setup is than the traditional one.

Those oscillations are not only greater in amplitude but also in frequency as shown in the Figure 12, where the Fast Fourier transform of the angles are calculated. This highlights how the drone in the wireless ESC connection vibrates at higher frequencies. This low level of stability is not present along the yaw angle, as it can be seen in Figure 11.

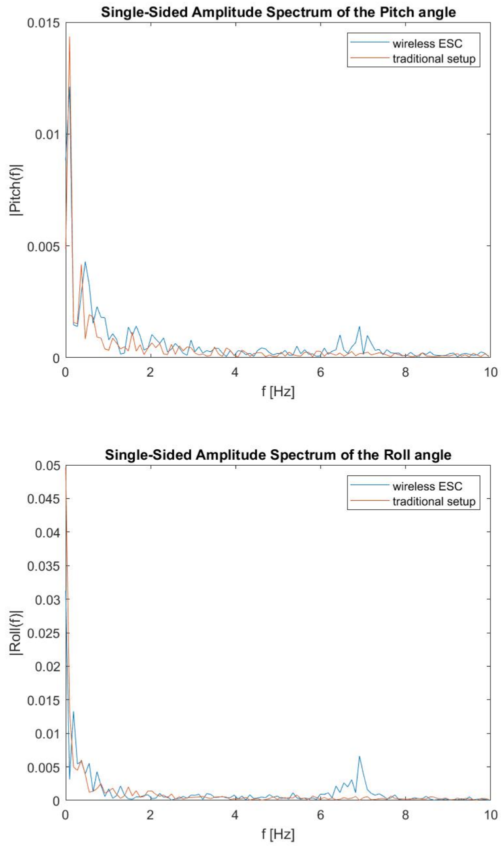


Fig. 12: Fast Fourier transform of the pitch and roll angles

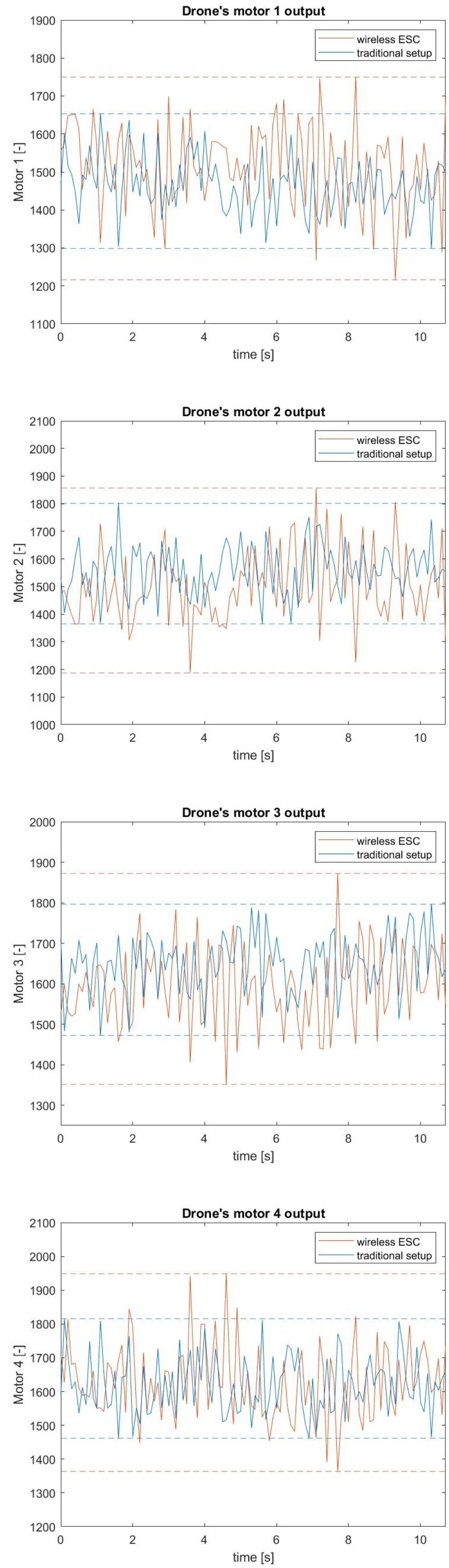


Fig. 13: Drone's actuators output

Overall those results show us that the drone when the ESC modules are connected wirelessly still manage to work properly and follow the desired commands, but with a certain level of instability. The cause of this instability may be due to the delay of the communication between the flight controller and the ESC module or due to possible asynchronization of the ESC modules. Both factors at certain levels introduce instabilities which can also be seen in the output signal of the actuators. In fact, in Figure 13 we see how the output signals have greater oscillations in the case of the wireless ESCs connection setup.

V. Conclusion

After designing and testing a modular drone prototype, we have demonstrated that it is feasible to create a UAV with interchangeable modules that can be quickly and easily replaced. Our design offers a flexible and customizable solution that allows users to easily adapt it to different applications and environments.

The testing of the drone showed that it can perform well in different configurations, both with the traditional and wireless connected ESCs setup. While the vibrations in the latter could have been introduced by the delays and asynchronization in communication.

The delay can be reduced by removing the PWM reading from the loop, by using a FC that communicates digital information for example with CAN bus. Otherwise a straightforward way to reduce the delay is to use a higher PWM frequency update in the FC. By doing so the period reduces, and so the time we need to wait to collect all the information about one period. This has the drawback of reducing the maximum allowed number of modules discussed previously. We think the main problem is asynchronization. It is due to the fact that each receiver has to generate its own PWM. To do so they start a timer, when they receive the synch command. That command is received in the same instant from all the receivers. The problem is that each receiver has its own clock and they can count with a different rate. Even if the difference is very small, after some time the PWMs, that have the clock as a source, go out of phase and they are no more synchronized. This means that when the FC create a new signal, it is not applied with the same delay to all the ESCs, but with a delay that depends on how much the receivers are out of phase. In order to avoid this we need to include a constant synchronization between the master and the slaves' clocks and further investigation is needed to understand how to do it.

Overall, our design provides a promising solution for a more efficient and adaptable drone. Future work would involve refining the communication system between the modules to improve stability, testing the drone in more challenging environments and exploring potential applications for the modular design.

While in this work we modyfied an existing UAV to validate our idea, the next step would be the actual creation of the wireless modules that communicate to each other without the need of a central flight controller, therefore studying the possible architectures and infrastructures for such a system.

References

[1] J. Lee, "Optimization of a modular drone delivery system," in *2017 Annual IEEE International Systems Conference (SysCon)*, 2017, pp. 1–8.

[2] M. Erbil, S. Prior, M. Karamanoglu, S. Odedra, C. Barlow, and D. Lewis, "Reconfigurable unmanned aerial vehicles," *International Conference on Manufacturing and Engineering Systems. Proceedings*, pp. 392–396, 12 2009.

[3] D. Falanga, K. Kleber, S. Mintchev, D. Floreano, and D. Scaramuzza, "The foldable drone: A morphing quadrotor that can squeeze and fly," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 209–216, 2019.

[4] S. Brischetto, A. Ciano, and C. Ferro, "A multipurpose modular drone with adjustable arms produced via the fdm additive manufacturing process," *Curved and Layered Structures*, vol. 3, 07 2016.

[5] G. Li, B. Gabrich, D. Saldaña, J. Das, V. Kumar, and M. Yim, "Modquad-vi: A vision-based self-assembling modular quadrotor," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 346–352.

[6] D. Kotarski, P. Piljek, M. Pranjić, C. G. Grlj, and J. Kasać, "A modular multirotor unmanned aerial vehicle design approach for development of an engineering education platform," *Sensors*, vol. 21, no. 8, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/8/2737

[7] M. Ferreira, M. Tejada-Begazo, G. Lopes, A. Felipe de Oliveira, E. Colombini, and A. Simões, "Drone reconfigurable architecture (dra): a multipurpose modular architecture for unmanned aerial vehicles (uavs)," *Journal of Intelligent Robotic Systems*, vol. 99, 09 2020.

[8] R. Oung and R. D'Andrea, "The distributed flight array," *Mechatronics*, vol. 21, no. 6, pp. 908–917, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741581000139X

[9] M. I. A. Syed Ahsan Ali, Umar Farooq, "Pwm direct transmission through rf for motor speed and direction controlling," 2017.

[10] E. Iloh and C. Adelokun, "Design and construction of wireless d.c motor speed and direction control using rf technique," 05 2022.

[11] Y. Rahayu, T. A. Rahman, R. Ngah, and P. Hall, "Ultra wideband technology and its applications," in *2008 5th IFIP International Conference on Wireless and Optical Communications Networks (WOCN '08)*, 2008, pp. 1–5.

[12] T. Tsang and M. El-Gamal, "Ultra-wideband (uwb) communications systems: an overview," in *The 3rd International IEEE-NEWCAS Conference, 2005.*, 2005, pp. 381–386.

[13] H. V. Abeywickrama, B. A. Jayawickrama, Y. He, and E. Dutkiewicz, "Empirical power consumption model for uavs," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–5.