DOI: 10.13009/EUCASS2022-4878
9TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)

Attitude Control of a Ducted VTOL UAV with a Certifiable (DO-178C/ED-12C) Adaptive and Learning Avionic Agent Trained with Reinforcement Learning

John Pyrgies and Moaad Yacoubi***
**SkyAngels Lab*
20 Rue de l'Industrie, 1400 Nivelles, Belgium
john.pyrgies@skyangels.eu
***Aéro-Thermo-Mécanique, Université Libre de Bruxelles*
50 Avenue F. Roosevelt, 1050 Bruxelles, Belgium
moaad.yacoubi@ulb.be

Abstract
Ducted-Fan Vertical Taking Off and Landing (VTOL) Unmanned Aerial Vehicles (UAV) combine the advantages of high precision landing and taking off without the need of runways, like helicopter and multi-copter UAVs, with high speed horizontal flight, like fixed-wing UAVs. The duct offers protection to their rotor, improves their hovering, decreases their power consumption and increase their stealth capabilities. However, the control of their attitude i.e. the orientation of the aircraft on the yaw, pitch and roll axis taking into account engine thrust represents challenges mainly in the transitions between the vertical and the horizontal flight modes. Aircraft attitude is typically controlled with a Proportional-Integral-Derivative (PID) control law which delivers acceptable performance for 'simple' flight conditions in stable environments. However, it is expected that a single PID for all flight modes shows limitations in more complex use cases like the one of a ducted VTOL UAV used for dropping a bomb to blast a nascent forest fire.

The research pursues a State-of-the-Art whose first exploration shows interest in finding alternatives to a single PID control law for ducted VTOL UAVs and the relevancy of leveraging Reinforcement Learning for multi-copter UAVs attitude control. 'System Requirements allocated to Software' are elicited from a research project team (Aéro-Thermo-Mécanique Dpt, Université Libre de Bruxelles) busy developing a ducted VTOL UAVs. Software Requirements are specified with SysML diagrams to avoid pitfalls of Natural Language Specification and to facilitate Validation by Aeronautic Engineers. Software Architecture is designed with SysML block diagrams. A first model of the flight controller software is produced, exploiting both data flows and state-machines diagrams. A training environment is identified to train the Flight Controller with Reinforcement Learning algorithms in order to compute the parameters that optimize a reward function (minimize error between observed attitude and target attitude). Throughout the whole Software Development Life Cycle, Safety requirements will be taken into account as the UAV attitude control software delivered by the research is aimed at being certified against the DO-178C (Safety) standards. The rationale is that this ducted VTOL UAV shall operate in non-segregated airspaces (EASA 'Certified' category) and in hostile environments (battle fields). The article will conclude with a presentation of the first design model of the ducted VTOL UAV flight and with future works in the scope of the PhD thesis of the authors, in progress.

1. Introduction

Ducted fan Vertical Taking Off and Landing (VTOL) Unmanned Aerial Vehicles (UAV) combine the advantages of high precision landing, taking off without runways and hovering, like helicopter and multi-copter UAVs, with high speed horizontal flight, like fixed-wing UAVs. The duct protects the rotor, increases thrust, provides lift in horizontal flight, decreases the power consumption and increase their stealth capabilities. However, the control of their attitude i.e. the orientation of the aircraft on the yaw, pitch and roll axis taking into account engine thrust represents challenges mainly in the transitions between the vertical and the horizontal flight modes. UAV attitude is typically controlled by a software, the *firmware*, loaded and executed in the UAV autopilot hardware. UAV attitude and thrust controllers typically implement a single set of Proportional-Integral-Derivative (PID) control laws which deliver acceptable performance for UAVs with 'simple' aerodynamic flying in stable environments.

DOI: 10.13009/EUCASS2022-4878

Aero-Thermo-Mechanics (ATM) department of the University Libre de Bruxelles (ULB) has designed and prototyped *SaFly*, a ducted fan VTOL integrated with a quadrotor [1], which is a unique UAV of its kind. It has been observed from the first trials of *SaFly* that its piloting can be challenging and a perceived track of improvement is the enhancement of its attitude control software. It is indeed assumed that a single set of PID control laws, common to all flight modes of *SaFly*, shows limitations for critical phases of the flight or difficult conditions encountered in the environment, like a gust. In this research, we propose to design the *SaFly* attitude controller as an Adaptive and Learning Avionic Agent (ALAA).
SaFly is aimed at carrying critical civil security missions, like dropping a bomb to blast a nascent forest fire, or military missions, like destroying an armoured vehicle on a battlefield. Therefore, a *Safety and Security by design* approach for its embedded software, including its attitude controller, shall facilitate its future Safety (DO-178C/ED-12C) certifications.
This paper is organized as followed : In Chapter 2, we present a State-Of-The-Art on Reinforcement Learning (RL) applied to UAVs Flight Control and on UAV RL Training Environments. In Chapter 3, we present *Safly's* aerodynamic and specify the Flight Control 'System Requirements allocated to Software'. In Chapter 4, we design *SaFly* Flight Control Software as an Adaptive and Learning Avionic Agent (ALAA). In chapter 5, we present the environment to train *SaFly* Flight Control ALAA with Reinforcement Learning and the Software Development Environment. In chapter 6, we conclude and mention future works.

2. Related Work

2.1 Reinforcement Learning applied to UAVs Flight Control

In [2], which has been the most influential to our work, Koch and al. train a Quadcopter attitude controller with state-of-the-art Policy Gradient Reinforcement Learning (RL) algorithms : PPO, TRPO and DDPG. The attitude control function is implemented as an Artificial Neural Network (ANN) whose synaptic weights are tuned by the RL training. They compare their result with an attitude control function implemented as a classical Proportional-Integral-Derivative (PID): The ANN controller trained with PPO outperforms the PID controller. They conclude that a controller trained with RL meet the stringent precision and accuracy requirements for a software executing in the UAV autopilot ("inner loop"). Waslander and al. [3] apply Model Based Reinforcement Learning and Integral Sliding Mode to improve UAV Altitude control in the "inner loop" when compared to standard integral Linear Quadratic Regulator (LQR) techniques. In [4] Taher Azar and al. provide a review of the usage of Deep Reinforcement Learning (DRL) techniques to improve UAV Path Planning, Navigation and Control functions. For those latter, in the "inner loop", PPO, TRPO and DDPG algorithms are used for Altitude Control and PPO for Longitudinal & Lateral Control. It shall be noted that PPO and DDPG are part of the Stable-Baselines3 [5] which provides open-source implementations of DRL algorithms in Python. [6] leverages a hierarchy of Deep Q-Networks (DQN) to allow an autonomous landing of a Quadrotor on a static ground marker.

2.2 UAV Reinforcement Learning Training Environment

[7] is a PhD thesis whose contribution consists in 3 GymFCs (v1, v1.2, v1.5) Reinforcement Learning Training Environments. GymFC combines OpenAPI Gym ([8], a common API for RL environments) with Gazebo ([9], an open source high fidelity simulator) for synthetizing attitude flights controllers. The most relevant version to our work is GymFC v1.2 which allows to tune flight controllers in simulation (Simulation In The Loop). It shall be noted that v1.2 extends the previous GymFCs to support any aircrafts and any control algorithms like PIDs and not only Neural Networks flight controllers. It also provides a methodology for creating new multicopter digital twins. In [10], Panerati and al. present 'Learning to Fly', an Open API Gym environment [8] whose underlying simulation engine is PyBullet, the Python interface to the Bullet Physics and that aims at training Multi-agent Quadcopter Controllers with Reinforcement Learning. Controllers can be handwritten controllers like PID or controllers trained with RL algorithms like PPO. PyBullet offers a vision based Gym and a multi-agent RL interface which is suited to control problems. It can be used either in single agent mode or in multi-agent to develop Multi-Agent Reinforcement Learning applications.

DOI: 10.13009/EUCASS2022-4878

3. Software Requirements : SaFly Flight Control

3.1 SaFly Configuration

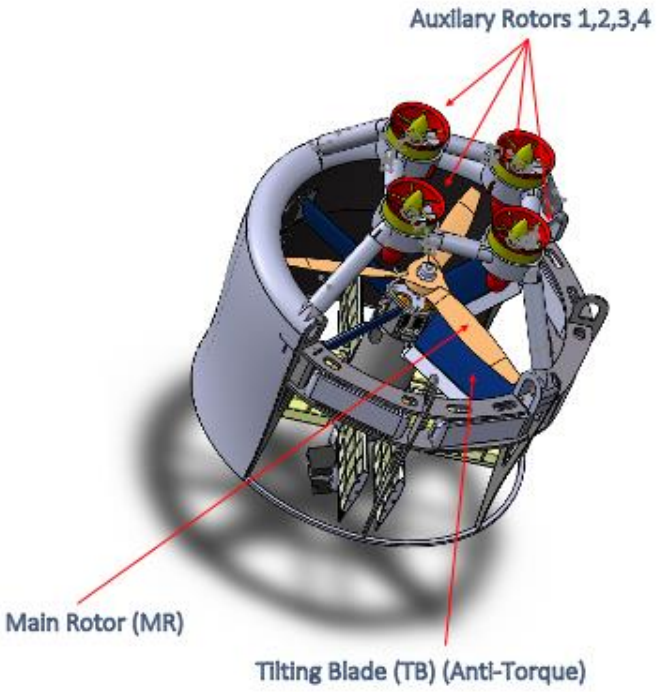


Figure 1: SaFly, a ducted fan VTOL UAV

[1] provides a description of SaFly, a ducted fan VTOL UAV designed and prototyped by the Aero-Thermo-Mechanics department of the School of Engineering (Polytechnic School) at the Université Libre de Bruxelles (ULB). SaFly is a ducted fan VTOL UAV integrated with a quadrotor, which makes it a unique UAV of its kind. Thrust is delivered by the Main Rotor and a Tilting Blade controls yaw. The role of the Quadrotor standing at the top of the aircraft is to control roll and pitch. It has been observed from the first trials of *SaFly* that its piloting can be challenging and **a perceived track of improvement is the enhancement of its Flight Control System**.

3.2 SaFly's Flight Control System

Flight Control System is an avionic system which, according to [11], has following functions : "*Aircraft flight controls are the means by which a pilot controls the direction and attitude of an aircraft in flight"*. Attitude is the orientation of the aircraft in regards of the Yaw, Pitch and Roll axis. SaFly's Flight Control System, the *Autopilot*, is implemented as a Hex Cube Black Flight Controller (formerly Pixhawk 2.1) whose main characteristics are [12]:

- Processor
  - 32-bit ARM Cortex M4 core with FPU
  - 168 Mhz
  - 256 KB RAM
  - 2 MB Flash
  - 32-bit failsafe co-processor
- Sensors
  - Three redundant IMUs (accels, gyros and compass)
  - InvenSense MPU9250, ICM20948 and/or ICM20648 as first and third IMU (accel and gyro)
  - ST Micro L3GD20+LSM303D or InvenSense ICM2076xx as backup IMU (accel and gyro)
  - Two redundant MS5611 barometers

DOI: 10.13009/EUCASS2022-4878

Figure 2: Hex Cube Black Flight Controller (Pixhawk 2.1)

Control Software executable, the *Autopilot's Firmware,* is loaded ('flashed') into the Autopilot flash memory. There are several Open Source software (source code) that can be leveraged to build the *Autopilot's Firmware* for the Hex Cube Black Flight Controller. The most notable are the ArduPilot Project [13] and the PX4 Project [14].

3.3 Flight Control Software Development Life Cycle (SDLC)

The DO-178C (and its European equivalent standard, the ED-12C) certification process [15] determines the level of a software in regards of the effects of a failure on the crew, passengers and aircraft (no effect, minor, major, hazardous, catastrophic) and the rigor, details and traceability of the certification artefacts are related to this level.

DO-178C / ED-12C Software Development and Integral Life Cycle Processes implements the best practices in Software Engineering:

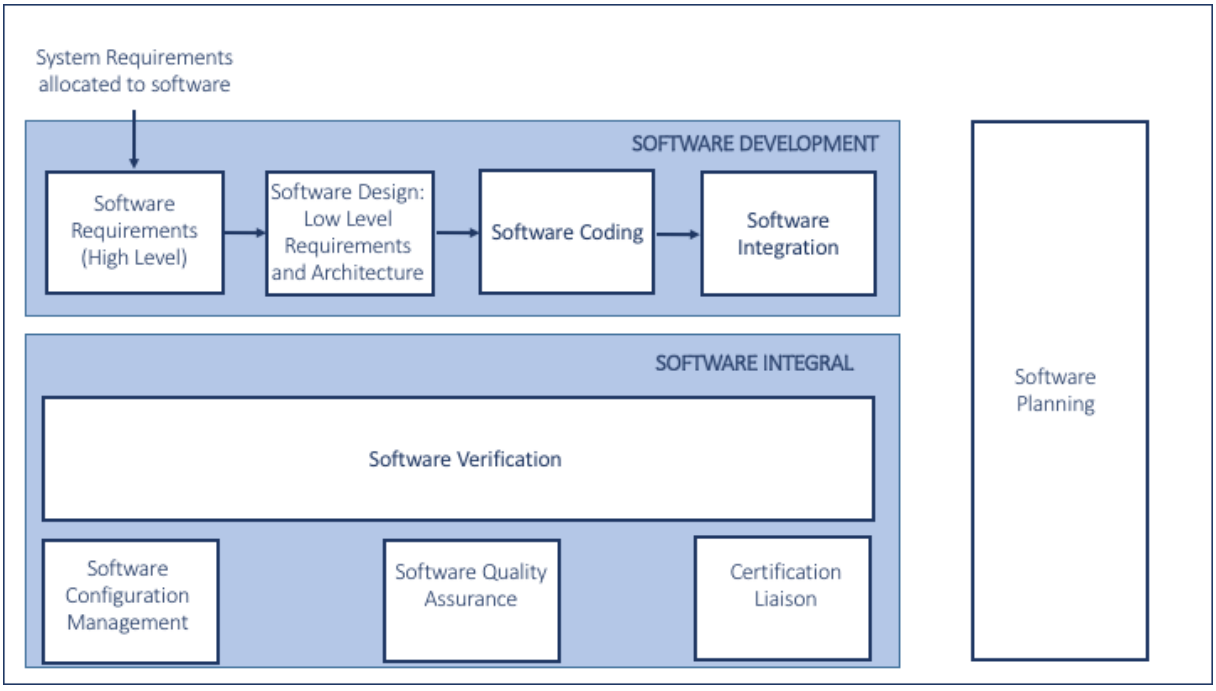


Figure 3: DO-178C / ED-12C Software Development and Integral Life Cycle Processes

In [16], the author and al. performed a literature review on DO-178C applied to safety critical UAV avionic software and proposed an innovative approach for the certification of an 'intelligent' software implementing a sense-and-avoid capability in a UAV.

SaFly is aimed at carrying critical civil security missions, like dropping a bomb to blast a nascent forest fire, or military missions, like destroying an armoured vehicle on a battlefield. Therefore, a *Safety by design* approach for its embedded software, including its flight control software, shall facilitate its future DO-178C / ED-12C certification.

In [16], following Software Development Life Cycle has been proposed and will be leveraged to specify and design the SaFly's Flight Control Software:

DOI: 10.13009/EUCASS2022-4878

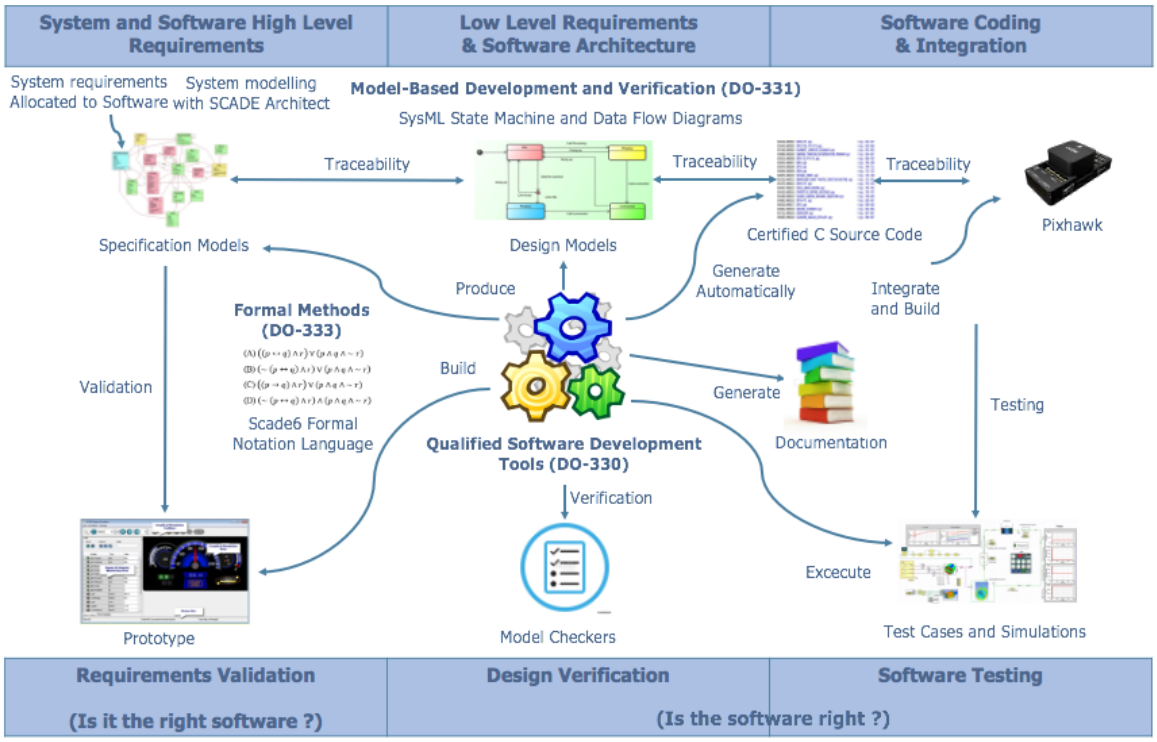


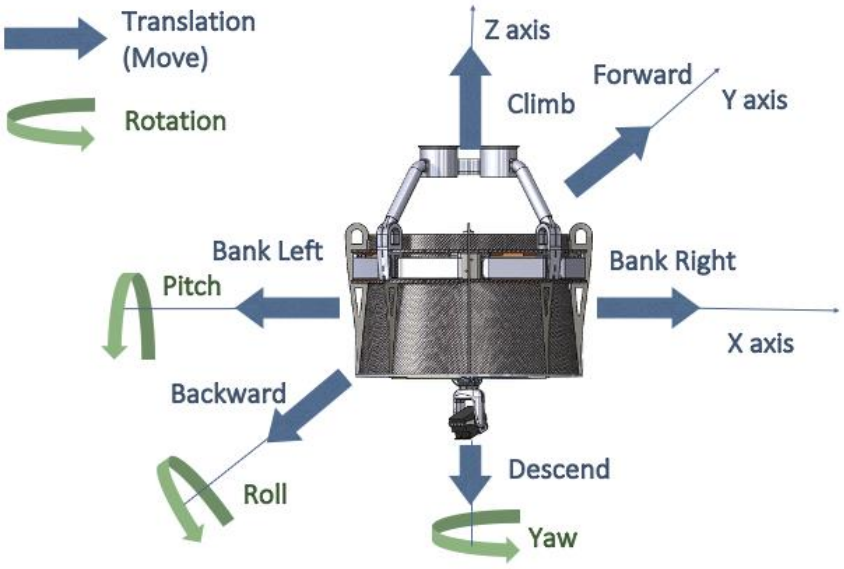
Figure 4: SaFly's Flight Control Software Development Life Cycle

3.4 High Level System Requirements allocated to Software

The first step in our method consists in specifying the High Level Functional Requirements of the Flight Control System functions that will be delegated to the Flight Control Software, in our case, the *autopilot's firmware*.

3.4.1 - High Level Functional Requirement 1 : Control the translational and rotational moves of SaFly

Safly's Flight Control Software shall control the translational moves of the UAV in the tri-dimensional space (typically in a North-East-Down reference system) and the rotational moves (Euler angles) of the aircraft over its yaw, pitch and roll axis. This represents a six degrees of freedom system.



DOI: 10.13009/EUCASS2022-4878

Figure 5: SaFly six degrees of freedom translational and rotational moves

Safly's Flight Control those 3 translational and 3 rotational moves with following actuators :

- The Main Rotor that controls thrust : MR.
- The 4 Auxiliary Rotor that controls pitch and roll: AR1, AR2, AR3 and AR4.
- The Anti-Torque Tilting Blade that controls yaw: TB.

In [2], Koch and al. provide, for a quadcopter, the aerodynamic effect U that each rotor speed ω_i has on thrust and Euler angles of the pitch, roll and yaw axis.

We have adapted those formulae to SaFly :

$$U_{Thrust} = f(\omega^2_{MR}) \quad (1)$$

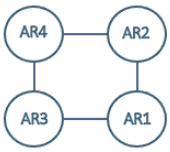
$$U_{Roll} = f(\omega^2_{AR1} + \omega^2_{AR2} - \omega^2_{AR3} - \omega^2_{AR4}) \quad (2)$$

$$U_{Pitch} = f(\omega^2_{AR1} - \omega^2_{AR2} + \omega^2_{AR3} - \omega^2_{AR4}) \quad (3)$$

$$U_{Yaw} = f(\theta_{TB}) \quad (4)$$

Where :

- U_{Thrust}, U_{Roll}, U_{Pitch}, U_{Yaw} are the thrust, roll, pitch and yaw effect.
- ω^2_{MR}, ω^2_{AR1}, ω^2_{AR2}, ω^2_{AR3}, ω^2_{AR4} are the rotor speed of the main rotor and auxiliary rotors 1, 2, 3 and 4.
- θ_{TB} is the tilting angle of the Anti-Torque Tilting Blade.
- The 4 auxiliary rotors configuration is :



State-Of-The-Art Flight Control Software use PID control laws to control the aerodynamic effect U such that, as in [2]:

$$\mathrm{UThrust}(t) = \mathrm{KpThrust}\, e(t) + \mathrm{KiThrust} \int_0^t e(t)dt + KdThrust \frac{de(t)}{dt} \quad (5)$$

$$\mathrm{URoll}(t) = \mathrm{KpRoll}\, e(t) + \mathrm{KiRoll} \int_0^t e(t)dt + Kd\mathrm{Roll} \frac{de(t)}{dt} \quad (6)$$

$$\mathrm{UPitch}(t) = \mathrm{KpPitch}\, e(t) + \mathrm{KiPitch} \int_0^t e(t)dt + Kd\mathrm{Pitch} \frac{de(t)}{dt} \quad (7)$$

$$\mathrm{UYaw}(t) = \mathrm{KpYaw}\ e(t) + \mathrm{KiYaw}\ \int_0^t e(t)dt + Kd\mathrm{Yaw}\ \frac{de(t)}{dt} \quad (8)$$

As mentioned in [7], standard method to tune a PID controller is the Ziegler-Nichols method.

3.4.2 - High Level Functional Requirement 2 : Adapt Flight Control to SaFly Flight Modes

To enable SaFly's critical missions (here illustrated with an armoured vehicle neutralisation), Safly's Flight Control Software shall adapt to the different flight modes of SaFly.

In [17], Zamri provides a good overview of flight modes of a Ducted-Fan VTOL UAV which could be applied to SaFly:

DOI: 10.13009/EUCASS2022-4878

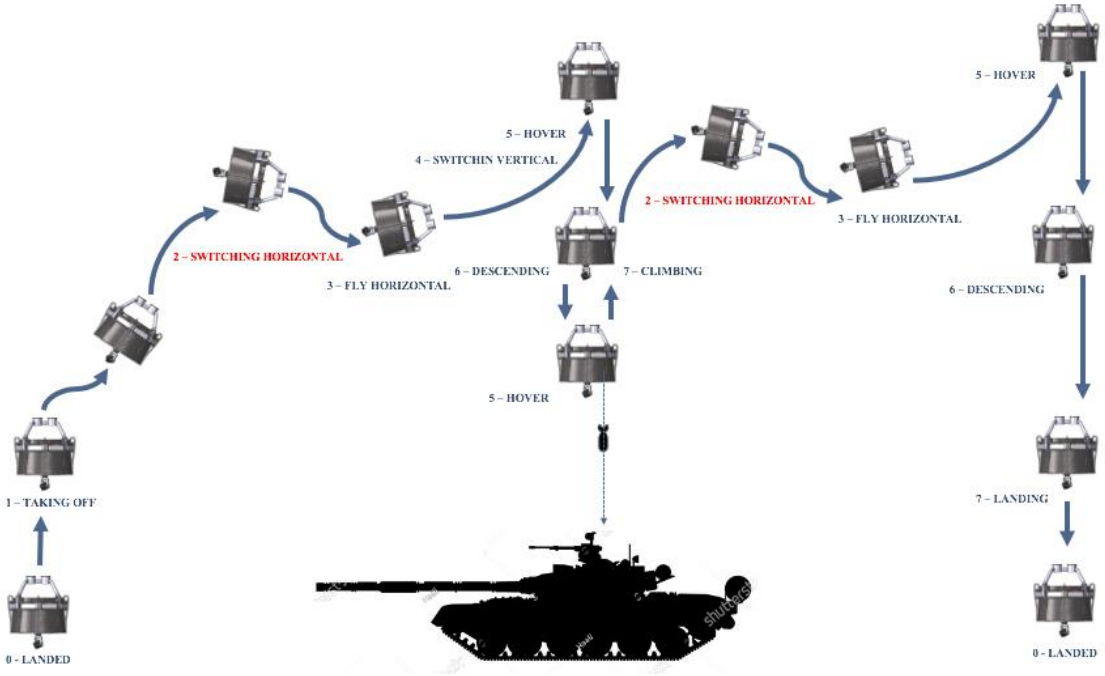


Figure 6: SaFly Flight Modes

4 - Software Design : An Adaptive and Learning Avionic Agent (ALAA)

4.1 SaFly Flight Control Software Architecture

Following Block Diagram specifies the SaFly Flight Control Software Architecture:

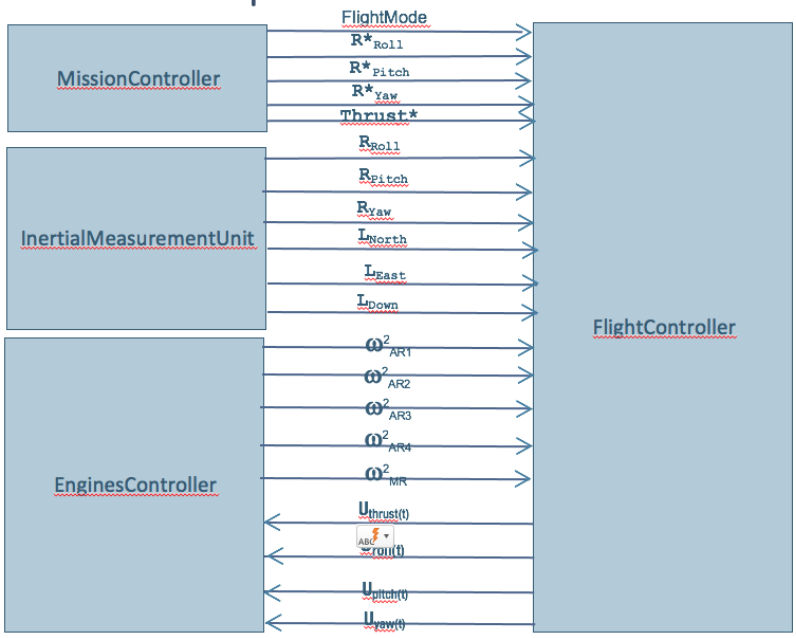


Figure 7: SaFly Flight Control Software Architecture

Where :

- FlightMode is the FlightMode dictated by the Mission Controller.
- R^*_{Roll}, R^*_{Pitch}, R^*_{Yaw}, Thrust* are the Roll, Pitch, Yaw and Thrust setpoints dictated by the Mission Controller.

DOI: 10.13009/EUCASS2022-4878

- ω^2_{MR}, ω^2_{AR1}, ω^2_{AR2}, ω^2_{AR3}, ω^2_{AR4} are the rotor speed of the main rotor and auxiliary rotors 1, 2, 3 and 4.
- R_{Roll}, R_{Pitch}, R_{Yaw}, L_{North}, L_{East}, L_{Down} are the real rotational and transactional speed measured by the gyroscopes and the accelerometers of the Inertial Measurement Unit.
- $U_{Thrust(t)}$, $U_{Roll(t)}$, $U_{Pitch(t)}$, $U_{Yaw(t)}$ are the thrust, roll, pitch and yaw effect dictated by the Flight Controller (computed by the PID Control laws).

4.2 SaFly Flight Control Adaptation

SaFly Flight Control is Adaptive thanks to its State-Machine whose States match the different fly modes of SaFly.

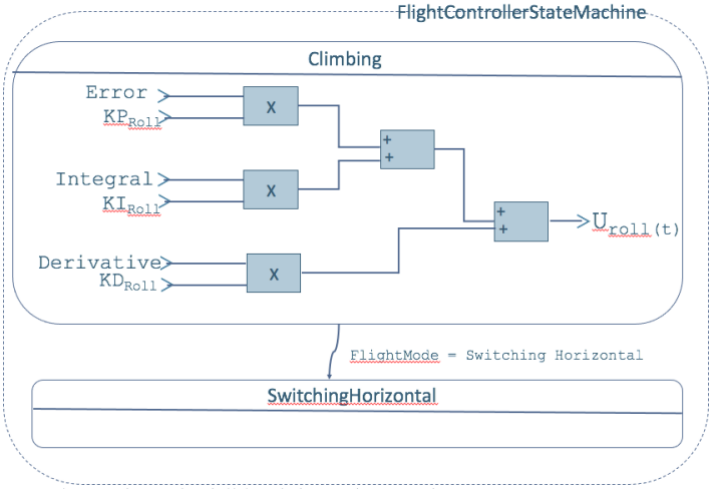
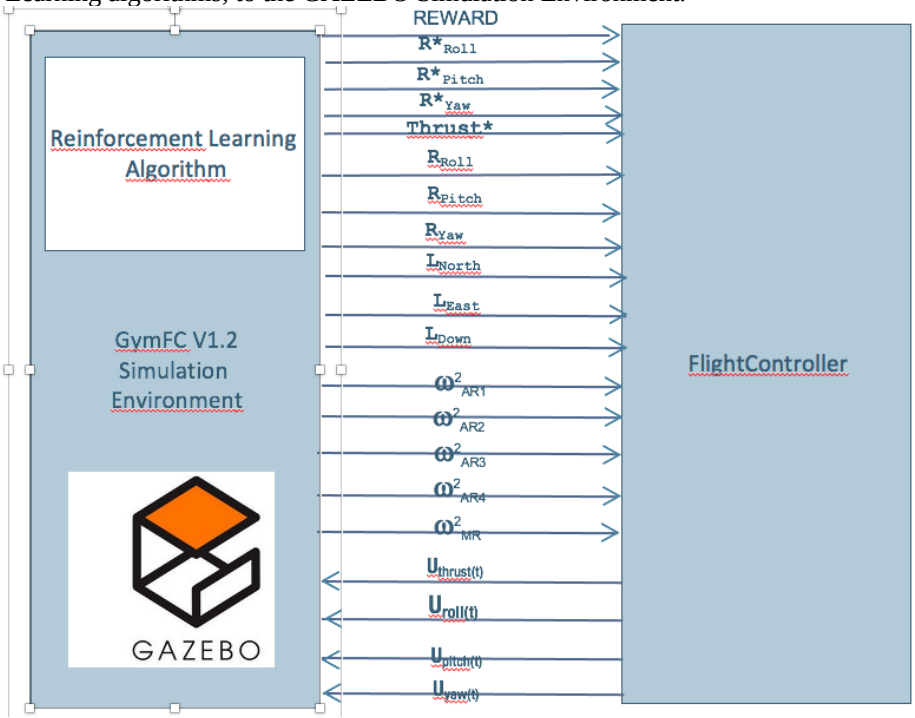


Figure 8: SaFly Flight Control Software State Machine

This State-Machine approach allows to have PID control laws whose gain parameters are adapted to each flight mode. This approach would also allow to have alternative control laws e.g. LQR.

5 – Training SaFly Flight Controller with Reinforcement Learning

SaFly Flight Control is Learning thanks to GymFC V1.2 which leverages a Gym interface, and its State-Of-The-Art Reinforcement Learning algorithms, to the GAZEBO Simulation Environment.



DOI: 10.13009/EUCASS2022-4878

Reinforcement Learning teaches a Control Function by tuning its parameters on the basis of the (current and future) rewards returned by the Environment.

In our case, this is the PID Control Laws parameters Kp, Ki and Kd that are tuned by the Reinforcement Learning algorithms.

6 – Conclusions and Future Works

In this research, we have identified a Software Development Life Cycle, a Development environment and a Reinforcement Learning Training environment to specify, design, train and certify (DO-178C) an Adaptive and Learning Flight Controller Software Agent that addresses the complex Flight Control requirement of SaFly, A ducted fan UAV with a quadrotor.

Future works will consist in deploying this Reinforcement Learning Training environment to train the Safly Flight Controller with Reinforcement Learning algorithms. We will consider control laws which are more powerful and complex than PID, like Artificial Neural Networks. We will also compare SISO and MIMO approaches.

References

[1] Yacoubi, M., A. Jlassi, J. Karoun, T. Ben Kirane, L. Bekkali, H. Jhabli, and P. Hendrick. 2019. Design and optimization of a ducted fan VTOL MAV controlled by Electric Ducted Fans. In: *8TH EUROPEAN CONFERENCE FOR AERONAUTICS AND AEROSPACE SCIENCES (EUCASS).*

[2] Koch, W., R. MANCUSO, R. WEST, and A. BESTAVROS. 2019. Reinforcement Learning for UAV Attitude Control. In: *ACM Transactions on Cyber-Physical Systems, Vol. 3, No. 2, Article 22.*

[3] Waslander, S.L., G.M. Hoffmann, J. Soon Jang, and C.J. Tomlin. 2005. Multi-Agent Quadrotor Testbed Control Design: Integral Sliding Mode vs. Reinforcement Learning. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems.*

[4] Taher Azar, A., A. Koubaa, N. Ali Mohamed , H. A. Ibrahim, Z. Fathy Ibrahim, M. Kazim, A. Ammar, B. Benjdira, A. M. Khamis, I. A. Hameed, and G. Casalino. 2021. Drone Deep Reinforcement Learning: A Review. In: *Electronics 2021, 10, 999*.

[5] Raffin, A., A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. 2021. Stable-Baselines3: Reliable Reinforcement Learning Implementations. In: *Journal of Machine Learning Research 22 (2021).*

[6] Polvara, R., M. Patacchiola, S. Sharma, J. Wan, A. Manning, R. Sutton, and A. Cangelosi. 2018. Autonomous Quadrotor Landing using Deep Reinforcement Learning. In: *arXiv:1709.03339*.

[7] Koch, W., Flight Controller Synthesis via Deep Reinforcement Learning. 2019. PhD Dissertation. Boston University.

[8] OpenAPI Gym.

[9] Koenig, N., and A. Howard. 2002. Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator. In: *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[10] Panerati, J., H. Zheng, S. Zhou, J. Xu, A. Prorok, and A. P. Schoellig. 2021. Learning to Fly—a Gym Environment with PyBullet Physics for Reinforcement Learning of Multi-agent Quadcopter Control. In:

[11] https://skybrary.aero/articles/flight-controls

[12] https://ardupilot.org/copter/docs/common-thecube-overview.html

[13] https://ardupilot.org/

[14] https://px4.io/project/

[15] Software Considerations in Airborne Systems and Equipment Certification, RTCA DO-178C.

[16] Pyrgies, J., D. Gigan, and R. Haelterman. 2017. An innovative approach for achieving DO-178C certification of an intelligent system implementing sense-and-avoid function in UAVs. In: Air Transport Research Society World Conference. Antwerp (2017).

[17] Zamri, O., 2010. Intelligent Control of a Ducted-Fan VTOL UAV with Conventional Control Surfaces. PhD Thesis. RMIT University.