

Bibliography

Master in Computer Science and Engineering
Instituto Superior Técnico, Universidade de Lisboa

Pedro Maria da Costa Almeida Martins — 99303*
pedromcamartins@tecnico.ulisboa.pt

Advisor: Prof. Paulo Oliveira
Co-advisor: Pedro Santos

*I declare that this document is an original work of our own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa (<https://nape.tecnico.ulisboa.pt/en/apoio-ao-estudante/documentos-importantes/regulamentos-da-universidade-de-lisboa/>).

Contents

| | | |
|---|--|---|
| 1 | A novel distributed architecture for unmanned aircraft systems based on Robot Operating System 2 | 3 |
| 2 | Design and test of electromechanical actuators for thrust vector control | 3 |
| 3 | A Structured Approach for Modular Design in Robotics and Automation Environments | 3 |
| | References | 4 |

1 A novel distributed architecture for unmanned aircraft systems based on Robot Operating System 2

Good historical context of ROS and DDS. Example of multi-layered architecture for vision-based navigation. Used PX4 internal control, sending setpoints from offboard computer.

[1]

Robotic software architectures aim to solve the need for scalability, modularity and parallelism inside robotic systems. The Robot Operating System (ROS) was created to address this need. However, it does not satisfy real-time constraints and is not natively fault tolerant. ROS2 addresses these issues, focusing on crossplatform, soft real-time systems. To satisfy soft real-time systems, new middleware technologies were developed. DDS has been adopted as the default middleware for ROS2, allowing developers to use Quality of Service configuration (deadlines, reliability, durability), and support scalability, with no overhead [discussed in a later section of the paper].

Today, electronics for drones are composed of several SoCs, depending on the complexity of the system. It is usual to delegate low-level control algorithms to small, real-time SoCs, while high-level algorithms are run on more powerful systems. Therefore, the problem of establishing a connection between these two systems arises. DDS is one solution for this problem.

The Application of Autonomous Quadcopter was used, following a layered architecture, composed of modules (ROS2 nodes). - Lowest row: soft real-time modules - Middle row: high-level, slower algorithms - Top row: supervision logic

—

Suggestion for our system

Soft real-time modules: Flight Control. It offers ROS2 services that enable other modules to request flight operations: - mode services: arm, disarm, offboard

Soft real-time modules: Sensor Fusion. It is responsible for fusing data from the IMU and GPS.

Soft real-time modules: Actuator. It is responsible for controlling the motors and servos.

High-level modules: Controller. It is responsible for controlling the drone. It uses the data from the sensor fusion module and sets the actuators accordingly. It is composed of two loops: - Outer loop: high-level control, responsible for the trajectory generation. - Inner loop: low-level control, responsible for the attitude control.

2 Design and test of electromechanical actuators for thrust vector control

[2]

3 A Structured Approach for Modular Design in Robotics and Automation Environments

[3]

References

- [1] L. Bianchi, D. Carnevale, F. Del Frate, R. Masocco, S. Mattogno, F. Romanelli, and A. Tenaglia, “A novel distributed architecture for unmanned aircraft systems based on robot operating system 2,” *IET Cyber-Systems and Robotics*, vol. 5, no. 1, p. e12083, 2023.
- [2] J. Cowan and R. A. Weir, “Design and test of electromechanical actuators for thrust vector control,” in *NASA. Ames Research Center, The 27th Aerospace Mechanisms Symposium*, 1993.
- [3] A. Elkady, J. Joy, T. Sobh, and K. Valavanis, “A structured approach for modular design in robotics and automation environments,” *Journal of intelligent & robotic systems*, vol. 72, pp. 5–19, 2013.