Software Architecture for Low-Cost UAVs: an Application Considering Automatic Target Tracking Mission Scenarios

João Alves, Tiago Oliveira, Gonçalo Cruz and Diogo Silva

October 26, 2022

Software Architecture for Low-cost UAVs: an application considering automatic target tracking mission scenarios

Authors: J. Alves (1), T. Oliveira (2), G. Cruz (3), D. Silva (4)

(1) Air Force Academy, jfalves@academiafa.edu.pt

(2) Air Force Academy, tmoliveira@emfa.pt

(3) Air Force Academy, gccruz@emfa.pt

(4) Air Force Academy, dasilva@emfa.pt

ABSTRACT

Since 2009, the Portuguese Air Force Academy Research Centre (CIAFA) has been involved in several research projects (e.g., PITVANT) using UAVs. Typically, Piccolo (Collins Aerospace, 2022) autopilots have been used by CIAFA for research purposes. However, this autopilot is relatively expensive and resorts to closed-source software. Therefore, CIAFA is exploring new open-source hardware and software options that allow for rapid prototyping and flight testing of UAVs, such as the Pixhawk (Kortunov et al., 2015).

This work aims to implement an adequately validated and documented software architecture compliant with the low-cost hardware architecture previously proposed by Silva et al. (2020), by considering open-source software tools available in the scientific community. Once the software architecture is outlined, an autonomous ground target tracking mission is implemented using the proposed software tools, thus illustrating the effectiveness of the proposed solution.

The software architecture proposed in this research is presented in Figure 1. The architecture comprises four main blocks, PX4 autopilot (including a feature for Software in the Loop Testing - SILT), Gazebo (only used for SILT purposes), QGroundControl (QGC) and Robot Operating System (ROS). The PX4 block corresponds to the autopilot's software that enables the conversion of the thrust, pitch and bank reference-values provided by the guidance algorithms at the ROS environment (detailed in the sequel) to the deflection of the aircraft's flight surfaces and motor settings. The Gazebo simulator module provides the physical simulation environment, and the QGC corresponds to the software running at the Ground Station. Finally, the ROS environment contains a set of modular nodes, namely Guidance and Control, Video Acquisition, Target Detector, Target Geolocation and Target Estimate that allows a UAV equipped with a video camera to detect, extract target's features and compute a set of control references to autonomously follow that target.

In the Guidance and Control node, the outer loop controller uses the PX4 Offboard mode to compute roll, pitch, yaw and thrust commands, which are then used by the PX4 autopilot's inner loop controller. The trajectory control for the UAV is separated into lateral and longitudinal control of the UAV. The lateral control was implemented based on Oliveira & Encarnação (2013) Moving Path Following (MPF) method, and the longitudinal control was based on Proportional Integral (PI) controllers.

The Video Acquisition node imports the camera video from the simulator and provides it to the Target Detector node, which uses a neural network, YOLOv3 (Redmon & Farhadi, 2018), to detect the vehicle. The Target Geolocation block implements a Geolocation algorithm from Barber et al. (2006) to compute the position of a ground target through its location and motion streamed in the video sequence. Next, a Kalman Filter is used in the Target Estimate node to estimate and predict the other parameters required by the MPF.

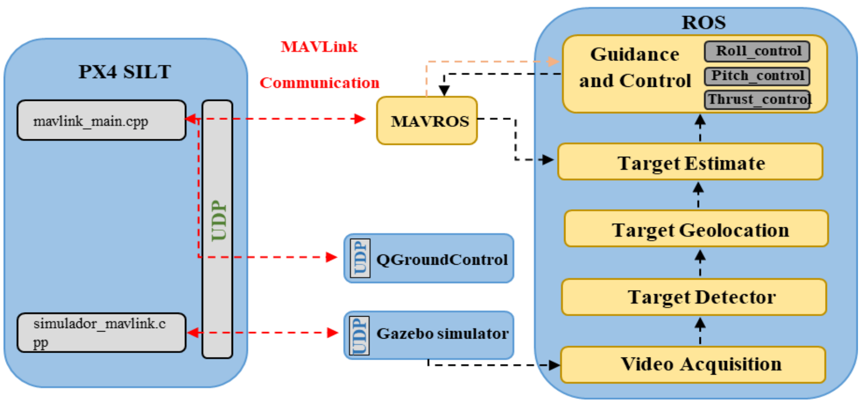

Figure 1. Proposed system software architecture.

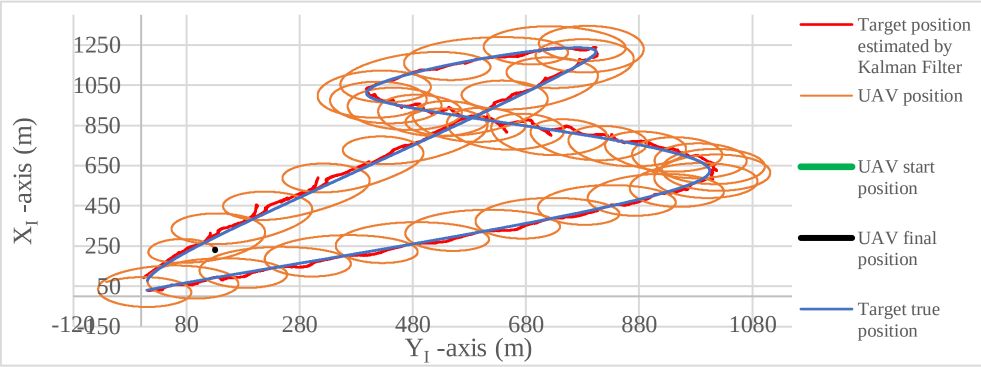

Figure 2. 2-Dimensional view of the UAV, target estimated position and target's true position.

In this work, a modular and incremental validation method was adopted, until the entire closed-loop system was successfully validated. Figure 2 shows the trajectory of a UAV following a ground vehicle in a simulated environment, using a video camera to detect and autonomously track a ground vehicle. This simulation demonstrates the effectiveness of the proposed architecture, considering both the selected software tools and the implemented control system.

KEYWORDS
Software architecture; Unmanned Aircraft Systems; YOLOv3, Moving Path Following; PX4 autopilot.

BIBLIOGRAPHY

Barber, D. B., Redding, J. D., McLain, T. W., Beard, R. W., & Taylor, C. N. (2006). Vision-based Target Geo-location using a Fixed-wing Miniature Air Vehicle. *Journal of Intelligent and Robotic Systems*, *47*(4), 361–382. https://doi.org/10.1007/s10846-006-9088-7

Collins Aerospace. (2022). *Piccolo™ Autopilots*. Collins Aerospace. https://www.cloudcaptech.com/products/piccolo-autopilots

Kortunov, V. I., Mazurenko, O. V., Gorbenko, A. V., Mohammed, W., & Hussein, A. (2015). Review and comparative analysis of mini- and micro-UAV autopilots. *2015 IEEE International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*, 284–289. https://doi.org/10.1109/APUAVD.2015.7346622

Oliveira, T., & Encarnação, P. (2013). Ground Target Tracking Control System for Unmanned Aerial Vehicles. *Journal of Intelligent & Robotic Systems*, *69*(1–4), 373–387. https://doi.org/10.1007/s10846-012-9719-0

Rafael Silva, Luís Félix, & Tiago Oliveira. (2020). *Projeto Conceptual de uma Aeronave Pequena de Baixo Custo para Aplicações em Controlo Cooperativo*. Academia da Força Aérea Portuguesa.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *ArXiv:1804.02767 [Cs]*. http://arxiv.org/abs/1804.02767