A Combined Approach for 3D Formation Control in a Multi-UAV System using ROS

9^{th} International Micro Air Vehicle Conference IMAV 2017

Rafael Gomes Braga[1] Roberto Claudino Silva[1] Felix Mora-Camino[2] Alexandre Carlos Brandão Ramos[1]

[1]Institute of Mathematics and Computing
Federal University of Itajubá, UNIFEI
Itajubá - MG, Brazil
(rafaelbraga, claudino, ramos)@unifei.edu.br

[2]Air Transport Department
Ecole Nationale de l'Aviation Civile, ENAC
Toulouse, France

Toulouse, France - September, 2017

Table of contents

Introduction

Introduction

- There is a growing interest in UAV research
- Many applications are possible because of UAV flying capabilities
- Many types of missions can benefit from using a group of UAVs

Introduction

- A group of UAVs present advantages compared to a single unit:
  - Carry more sensors and thus have a better area coverage
  - Mission can continue even even if some UAVs are lost
  - Robots can be simpler and less energy consuming

Main Objectives

This research aims to:

- Develop an algorithm to control a group of UAVs that:
  - Move together keeping a formation
  - Avoid collisions with each other
- Run simulations to evaluate the system in different scenarios

Combined Approach

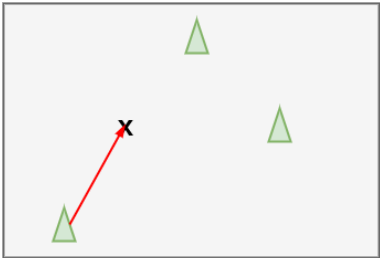
Many strategies can be used to control groups of UAVs

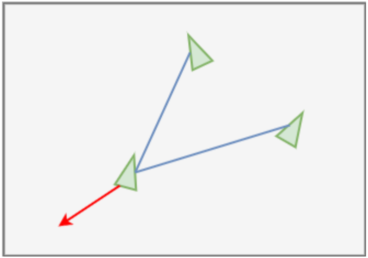
- **Behavior based:** Program UAVs to have desired behaviors, such as move together and avoid collisions
- **Virtual Structure:** Define the formation as an array of points in a virtual structure
- **Leader Follower:** Follower UAVs base their movement in the leader's position

Our idea is to combine these three strategies

Combined Approach

- The formation is represented by an array of points
- The UAVs have two behavioral rules: Move to desired position and avoid collisions





Combined Approach

We define a global reference frame S_G with X, Y and Z Cartesian coordinates, fixed on the ground. The position of each UAV i in the S_G frame is given by:

$$p_i^G = (X_i^G, Y_i^G, Z_i^G) \quad (1)$$

Combined Approach

- Each robot uses its embedded sensors to locate itself in the global frame
- It comminucates this position with the others, so they are able to know each other's positions
- Each UAV receives an unique ID, which is a positive integer number starting at 0 and increasing in increments of 1
- The points in the formation are relative to a formation reference frame, S_F

Combined Approach

Each UAV has to compute its desired position relative to the formation frame. We defined two methods to make this calculation:

- **Leader-follower method:** In this method, one UAV is considered the leader of the group (the one with ID 0). The leader ignores its position in the formation and instead is able to move freely, being controlled by an operator or following a sequence of waypoints. The other UAVs calculate their desired position relative to the leader, considering it the origin of the formation frame.

$$p_{iD}^{G} = p_{0}^{G} + p_{i}^{F} \tag{2}$$

Combined Approach

- **Waypoint method:** In this method, all UAVs receive a waypoint to follow and treat it as the origin of the formation frame. There is no leader in this method. As such, this approach is more tolerant to UAV failure than the first one, however is more difficult for a human operator to drive the group.

$$p_{iD}^{G} = p_{w}^{G} + p_{i}^{F} \tag{3}$$

Development

System Overview

- Our UAVs use open source software and hardware:

pixhawk

RaspberryPi

ubuntu

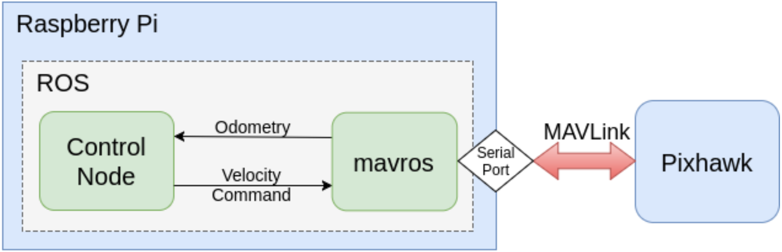
ROS

System Overview

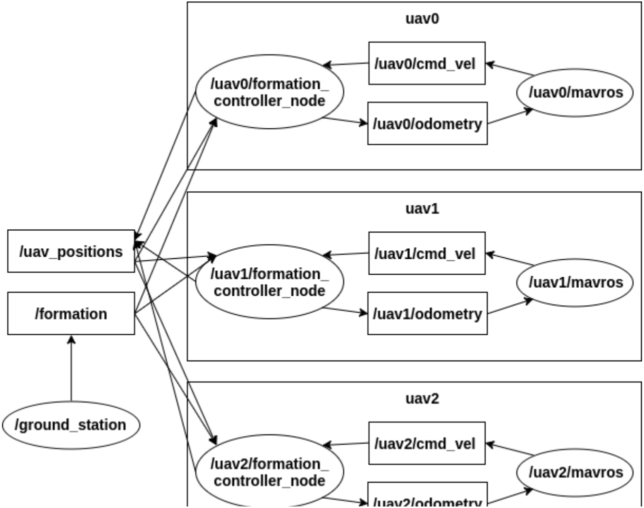
- The control program was written in C++ using the ROS platform
- Each UAV has a Pixhawk board, responsible for the low level control of the rotors and a Raspberry Pi 2, responsible for running the ROS system
- The Pixhawk and the Raspberry Pi communicate through the MAVLink protocol
- A node called mavros converts MAVLink messages into ROS messages and vice-versa

System Overview

Diagram representing the system running in each UAV:

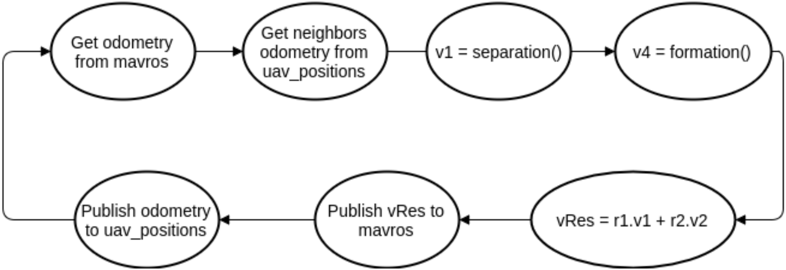


ROS Diagram


uav0
/uav0/cmd_vel
/uav0/formation_
controller_node
/uav0/mavros
/uav0/odometry
uav1
/uav_positions
/uav1/cmd_vel
/uav1/formation_
controller_node
/uav1/mavros
/formation
/uav1/odometry
uav2
/ground_station
/uav2/cmd_vel
/uav2/formation_
controller_node
/uav2/mavros

ROS Application

- General data flow of the application



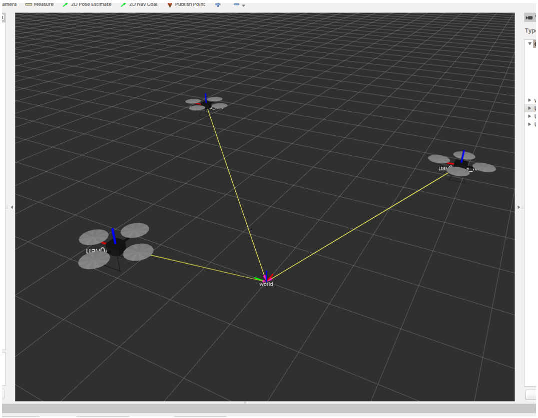
Simulation

Simulation

- We defined a scenario where the three UAVs, with IDs of 0, 1, and 2, have to move together maintaining a formation
- The UAVs were commanded to take off to an altitude of 2 meters. As soon as they are stabilized in this altitude, we started the control node and they start moving
- The system was simulated in the Pixhawk SITL simulator and in Gazebo

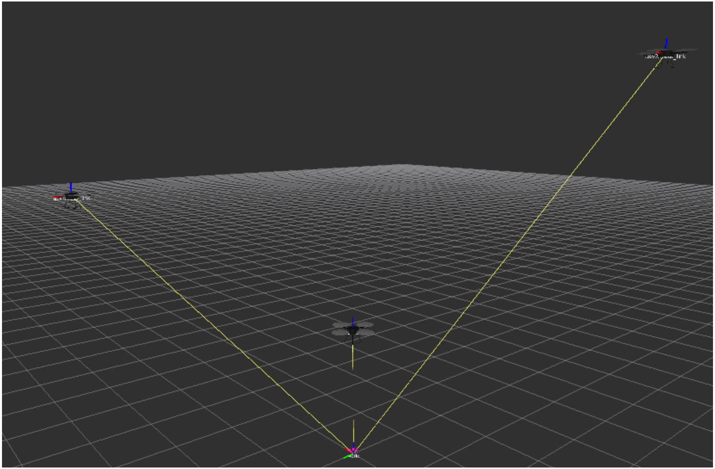
Simulation

- Visualization of the simulated UAVs in RViz



Simulation

- Example of a 3D formation

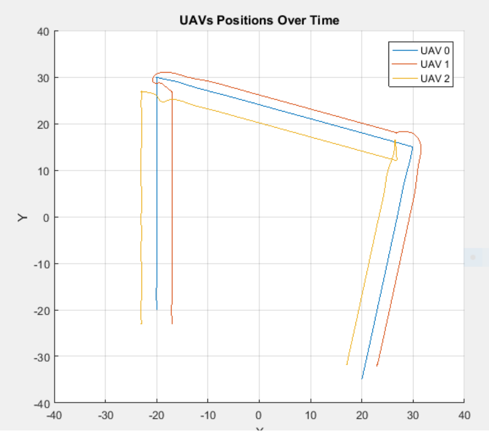


Simulation

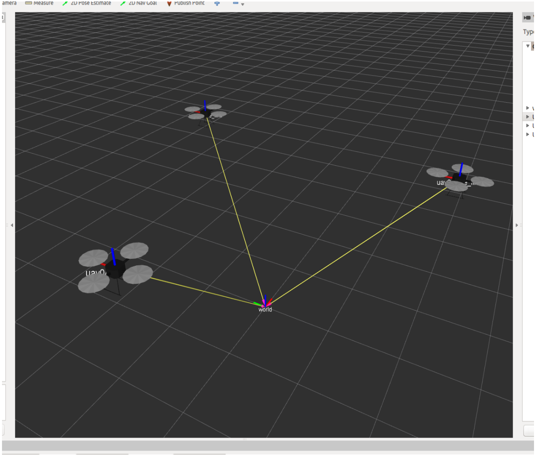
- We defined three formations
  - Horizontal line along the Y axis
  - Vertical column along the Z axis
  - Triangle on the X-Y plane
- We defined two main tests to evaluate the capability of the UAVs to maintain formation and to change formations

First Scenario: Moving in formation

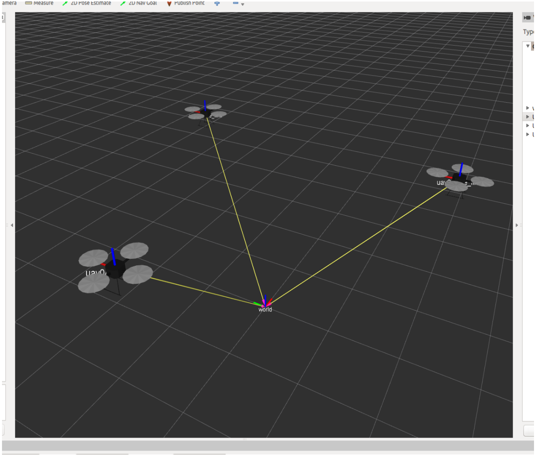
- The robots are commanded to assume a formation and then move to two waypoints



First Scenario: Moving in formation



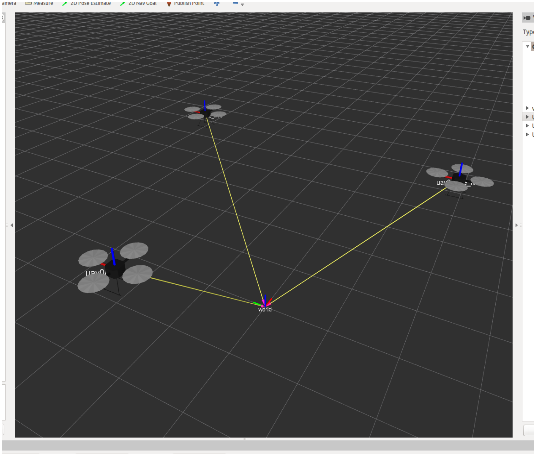
First Scenario: Moving in formation



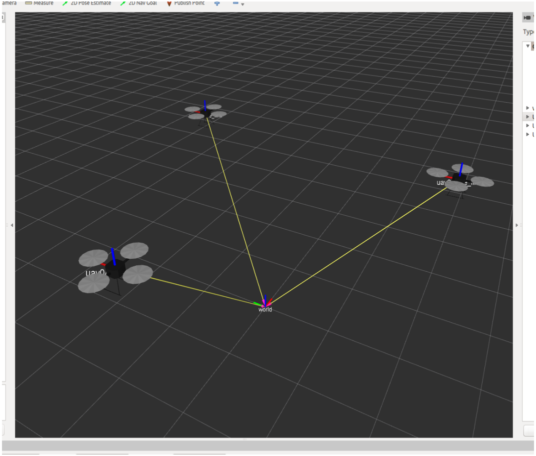
Second Scenario: Changing Formations

- The UAVs started at the line formation, then changed to other formations following this sequence: column, triangle, line, triangle, column and then line.

Second Scenario: Changing Formations



Second Scenario: Changing Formations



Second Scenario: Changing Formations

Transition	Time [s]
line $\rightarrow$ column	7.16
line $\rightarrow$ triangle	3.66
column $\rightarrow$ triangle	4.90
column $\rightarrow$ line	4.70
triangle $\rightarrow$ line	3.53
triangle $\rightarrow$ column	7.17

Longer times appear in the transitions to the column formation. This is caused by the fact that this formation is actually vertical

Conclusion

Conclusion

- Our algorithm combined elements from three different multi-UAV control strategies: leader-follower, virtual structure and behaviour-based
- The implementation was tested in simulations in the Pixhawk SITL simualtor and in Gazebo

Future Work

- Implementation on real robots
- The use of cameras or range sensors to detect nearby robots
- This would eliminate the need of communication between the robots, avoiding delays and enable the possibility to detect obstacles in the path
- Use AI techniques to tune the parameters

Thank you for your attention!
Questions?