Alma Mater Studiorum · Università di Bologna

Scuola di Ingegneria e Architettura

Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione
"Guglielmo Marconi" – DEI

Corso di Laurea in Ingegneria dell'Automazione

Agricultural Field Detection and Path Planning for an Unmanned Aerial Vehicle

Progetto di Laurea

Relatore:
Chiar.mo Prof. Lorenzo Marconi

Correlatore:
Prof. Nicola Mimmo

Presentato da:
Michael Rimondi

II Sessione
Anno Accademico 2017/2018

Todo list

Abstract

Maecenas accumsan dapibus sapien. Duis pretium iaculis arcu. Curabitur ut lacus. Aliquam vulputate. Suspendisse ut purus sed sem tempor rhoncus. Ut quam dui, fringilla at, dictum eget, ultricies quis, quam. Etiam sem est, pharetra non, vulputate in, pretium at, ipsum. Nunc semper sagittis orci. Sed scelerisque suscipit diam. Ut volutpat, dolor at ullamcorper tristique, eros purus mollis quam, sit amet ornare ante nunc et enim.

Keywords: One, Two

Abstract (italiano)

Phasellus fringilla, metus id feugiat consectetuer, lacus wisi ultrices tellus, quis lobortis nibh lorem quis tortor. Donec egestas ornare nulla. Mauris mi tellus, porta faucibus, dictum vel, nonummy in, est. Aliquam erat volutpat. In tellus magna, porttitor lacinia, molestie vitae, pellentesque eu, justo. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Sed orci nibh, scelerisque sit amet, suscipit sed, placerat vel, diam. Vestibulum nonummy vulputate orci. Donec et velit ac arcu interdum semper. Morbi pede orci, cursus ac, elementum non, vehicula ut, lacus. Cras volutpat. Nam vel wisi quis libero venenatis placerat. Aenean sed odio. Quisque posuere purus ac orci. Vivamus odio. Vivamus varius, nulla sit amet semper viverra, odio mauris consequat lacus, at vestibulum neque arcu eu tortor. Donec iaculis tincidunt tellus. Aliquam erat volutpat. Curabitur magna lorem, dignissim volutpat, viverra et, adipiscing nec, dolor. Praesent lacus mauris, dapibus vitae, sollicitudin sit amet, nonummy eget, ligula.

Keywords: Uno, Due

Contents

Chapter I

Introduction

In this introduction it is first briefly described the motivation, objective and scope of the project this thesis has been developed in. Once the background has been well set and explained,it will be describe the specific subject of the thesis.

1 Motivation

Deers gives birth to their offspring during April and May [1], often choosing meadows as they consider it a safe spot. This period is unfortunately the same in which meadows are cut. The results is that every year a great number of young deers fall victim of combine harvesters cutting hay. Germany counts about 100000 death every harvest season [1]. The BambiSaver project was born aiming to provide an autonomous, fast and user friendly device able to search and locate, living creatures in agricultural areas. It is, as a matter of fact, difficult to locate small animals hidden in vast grasslands especially if they freeze when they feel under threat. For this specific reason the proposal design is based on a UAV (Unmanned Aerial Vehicle) [2] and more precisely a quad-copter equipped with a thermal camera. An aerial vehicle is, in fact, able to efficiently cover large surfaces way faster then any other ground alternatives and guarantees the best viewpoint for the specific kind of wildlife research. Moreover it has been chosen a copter rather than a fixed wing model for its holonomic properties which turns out to be very useful in upland regions as well as for small and irregular fields. It is basically why in search and rescue operations helicopters are often adopted instead of planes.

1.1 Importance for Wildlife and for Agricultural

In early hunting literature from as far back as the mid-19th century, references can be found to significant losses of breeding partridges and pheasants from the use of sickle and scythe. Due to the fierce competition in the agricultural sector, developments in agricultural technology have brought about a tremendous acceleration in mowing techniques, with tendency still rising. Today, mowing speed can even exceed 15 km/h, while at the same time ever-wider mowers are used. Nesting birds, young hares and fawns are regular victims of such mowers and even full-grown wild animals cannot always escape. Ever since the 1950s, the importance of silage meadows has increasingly taken precedence over the traditional hay harvest.

1.2 Affected Species

Grassland is used by countless species of wildlife as food, cover and reproduction habitat. Apart from leverets, fawns and various field birds, small mammals, amphibians and insects all fall victim to the practice of early and more frequent mowing. Formerly reliable survival strategies proven successful over thousands of years have a devastating effect in mowing situations. The instinct of the brooding partridge hen to sit tight on her nest, or of the hare or fawn to freeze motionless, now prove fatal. The optimized patterns of predator avoidance behavior which wild animals have evolved can no longer keep up with the developments in modern cultivation techniques. 5de0697a03

1.3 Measure to Reduce Wildlife Losses

The most important factor influencing wildlife mortality is without doubt the time of mowing. On the other hand, economic considerations make this a crucial factor for the farmer, too. A late mowing is good for wildlife but not ideal for the farmer from the point of view of yield and quality. Yet there are some other factors in the mowing of arable land which offer potential for reducing wildlife losses [1]:

- *Cutting height*: the higher the cut, the lower will be the losses suffered by crouching animals and nesting birds.
- *Mowing direction*: mowing the field from the center outwards gives fully-grown wild animals the opportunity to escape.
- *Mowing date*: late cuts,from mid-July onwards, reduce the losses to wildlife during the nesting and rearing period.
- *Mowing strategy*: mowing of partial areas, leaving edge strips unmown.
- *Mowing frequency*: a longer interval between first and second cuttings reduces the mortality rate, especially for ground-nesting birds.
- *Mowing technology*: cutter-bar mowers cause less harm to wild animals than rotary mowers.

Another practical approach is the one adopted in a German wildlife rescue project which consists in deploying small aerial drones to find young deer hiding in tall grass.

2 State of the art

During last years, the increasing interest and development of UAS (Unmanned Aircraft System) makes them affordable and suitable for many different applications. Specifically, in wildlife research, drones are frequently use as they are less expensive, quieter, and safer than traditional manned aircraft. Most studies we reviewed recorded minimal or no visible behavioral responses to UAS; however, UAS are capable of causing behavioral and physiological responses in wildlife when observing at close range. [3].
For the case under-study, the "Flying Wildlife Finder", represents the state-of-the-art. The project, developed by the German Aerospace Center (DLR), is an application system which prevents accidents by detecting animals hidden in tall grass during the hay harvest.

The "Flying Wildlife Finder", a remotely controlled aerial drone equipped with sensors and a GPS link, is sent on a reconnaissance flight before mowing starts. A high resolution thermal imaging camera detects the temperature of animals hidden in the grass, which is higher than the ambient temperature of the field. Once the animal is located a search party is led to the fawn's resting place with the help of GPS. This solution proves to be good as it is less time-consuming then using trained dog, while maintaining even higher "hit-rate", but it has the main drawback of requiring at least two specialists: one pilot and one camera operator that must be focused on the thermal video stream during the whole mission duration.

3 Proposal Solution

In the following section it is briefly explained the goals and design guidelines of the project.

3.1 Project Goals

The scope of BambiSaver is to design an integrated system capable to autonomously handle every steps of the mission so that it can be operated also by untrained personals. This must be guaranteed even in adverse environments, especially in uneven fields as it is in upland regions. Along with this, it is of particular interest to keep modularity in every hardware and software components. This is to permit easier implementation of new features or improvements. All this guidelines have been taken into account during the design and implementation stages (i.e. ROS as framework for the on-board computer).

????the idea was developed having the mountain of Trentino Altoadige in mind????

sinonimo per dire idee alla bass

Software Design concept

The raspberry Pi 3 it is used as on-board computer, it runs ROS and manages the transition between the different mission's phases. The software develops according to the usual ROS philosophy [4], thus it is composed of five nodes implementing every mission component as an independent module. Figure I.1 displays all the nodes and the related topics used to let them communicate.
In the following chapter the thesis will focus, exclusively on two problems:

- *Geo-referencing the mission's environment and get the field boundary.* This is of great importance, as it define the mission's range which is fundamental for every successive tasks.
- *Planning the coverage path.*It regards planning the geometric path so that the whole field of interest is covered by the sensor footprint.

4 Innovation

The approach The proposal device is an UAV [2]

$$\vec{F} = \vec{a} \times \vec{b} \tag{I.1}$$

$$dof_{rot} = \sum_{i=1}^{n}(i-1) = n\frac{n-1}{2} \tag{I.2}$$

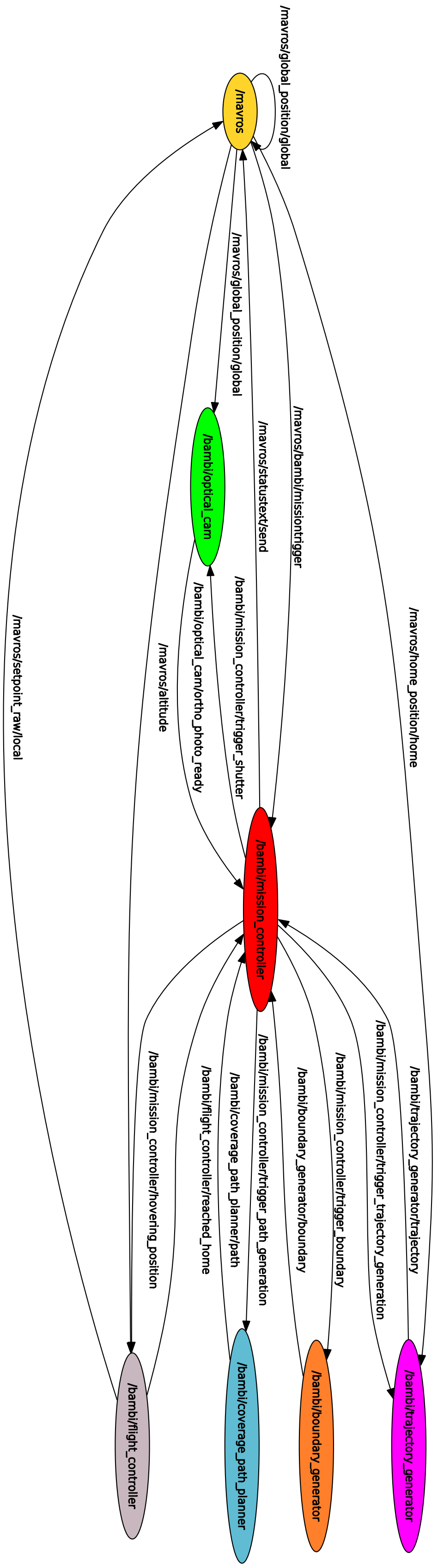

Figure I.1: Nodes and Topics graph

Chapter II

Georeferencing the mission's environment

In this chapter it is explained in details how the device obtain the information of the field boundary in form of a list of geographic coordinates. This important task will define the mission range of action which is used as input to the Coverage Path Planning module. It is thus important that the gathered information are precise and consistent or it will impact the successive steps of the mission.
The process could be divided in two tasks:

- Obtain a georeferenced photo displaying the entire meadow.
- Detect and trace the field boundary over the georeferenced image.
- Store the boundary as a set of geographic coordinate points.

1 Georeferencing a Photo

Before an aerial image can be used to support a site-specific application it is essential to perform the geometric corrections and geocoding. This is commonly called *georeferencing* which enables the assignment of ground coordinates to the different features in the datasets. If the map projection (and map projection parameters) of the ground coordinates are known, equivalent geographic coordinates can be produced which enables positioning the features of the coverage into a World context. [5]. For this specific use case the final result is an image where every point it is associated to a geographic coordinate.

1.1 Theory Background

Spatial datasets, like any type of data, are prone to errors. Thus, three fundamental concepts have to be kept in mind – precision, bias and accuracy. Precision refers to the dispersion of positional random errors and it is usually expressed by a standard deviation. Bias, on the other hand, is associated with systematic errors and is usually measured by an average error that ideally should equal zero. Accuracy depends on both precision and bias and defines how close features on the map are from their true positions on the ground [6]. So, despite being frequently confused concepts, high precision does not necessarily mean high accuracy. But both depend greatly on the map scale. All maps have inherent

CHAPTER 1. GEOREFERENCING A PHOTO

positional errors, which depend on the methods used in the construction of the map. The scale is the ratio between a distance on the map and the corresponding distance on the ground. The maximum acceptable positional error (established by cartographic standards) is determined by the map scale.

Fist of all a georeferenced photo containing the whole field is

Chapter III

Simulation results

The PX4 firmware offer a great Software-In-The-Loop simulation environment which was fundamental for the software development. It guarantee a real-time, safe and convenient way to test the software implementation without all the risks related to a real flight.
Simulation environments:

- PX4 SITL: The PX4 simulated hardware which reacts to the simulated given input exactly as it would react in the reality and issues the output as a percentage of the total thrust that every rotor has to provide.

- Gazebo: The dynamics simulator that is used by the SITL. This software reads the PX4 output and, by elaborating the modeled dynamics, provides the simulated input to the PX4. This allows for a quite accurate simulation of the real model behavior during flight.

The different parts of the system (Figure III.1) are connected via UDP, and can be run on either the same computer or another computer on the same network. The communication protocol is MAVLink (see appendix C).

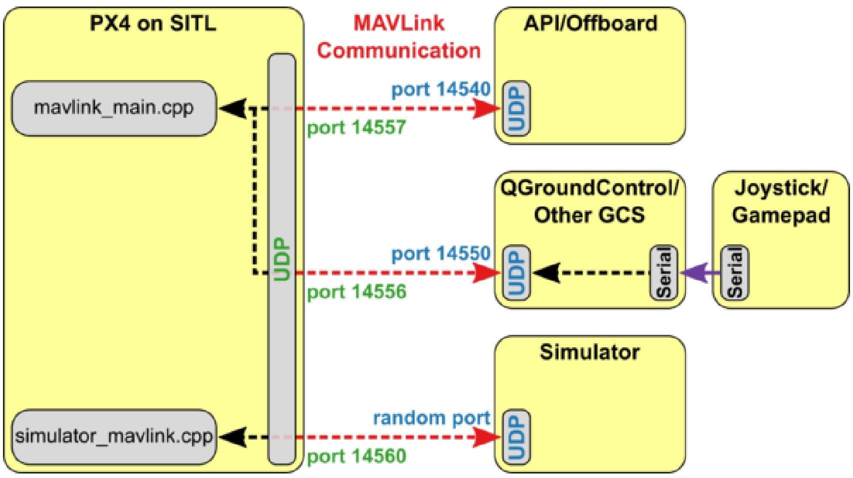

Figure III.1: SITL with Gazebo architectural scheme

1 Development Setup

2 section name

Chapter IV

Coverage Path Planning

Appendix A

Bibliography

[1] D. W. Stiftung, “Mowing mortality in grassland ecosystems,” 2011.

[2] in *ICAO. Global air traffic management operational concept*, 2005.

[3] K. S. Christie, S. L. Gilbert, C. L. Brown, M. Hatfield, and L. Hanson, “Unmanned aircraft systems in wildlife research: current and future applications of a transformative technology,” *Frontiers in Ecology and the Environment*, vol. 14, no. 5, pp. 241–251.

[4] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA Workshop on Open Source Software*, 2009.

[5] A. D. Chapman and J. Wieczorek, *Guide to Best Practices for Georeferencing*. Copenhagen: Global Biodiversity Information Facility, 8 2006.

[6] L. Meier, “Pixhawk main page.”

[7] L. Meier, “Mavlink developer guide main page.”

Appendix B

Pixhawk Autopilot

1 Hardware

"Pixhawk is an independent, open-hardware project aiming at providing high-end autopilot hardware to the academic, hobby and industrial communities at low costs and high availability. It provides hardware for the Linux Foundation DroneCode project. It originated from the PIXHAWK Project of the Computer Vision and Geometry Lab of ETH Zurich (Swiss Federal Institute of Technology) and Autonomous Systems Lab as well from a number of excellent individuals." [6]

Figure B.1

The Pixhawk hardware weights 38g and it is provided with a 32-bit ARM Cortex M4 core with FPU with 256 KB of RAM and a 32-bit fail-safe co-processor; it is also equipped with a compass, a barometer, an accelerometer and a gyro sensor.

2 Softwere

Modern, sophisticated flight controllers share a commonality in architecture. We can divide their functionality into three distinct layers, illustrated in Figure B.2.

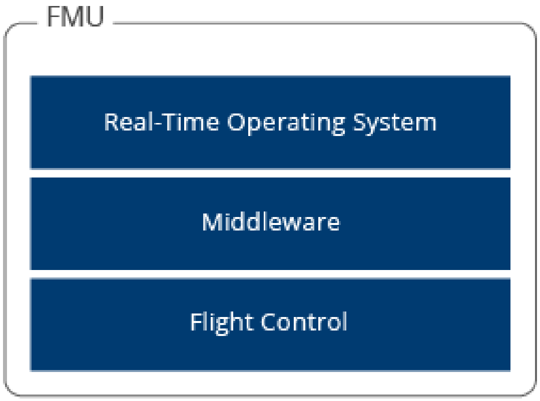

Figure B.2: The architecture of a modern flight controller

Layer 1: Real Time Operating System
The real time operating system is the back bone of the flight firmware, providing basic hardware abstraction and concurrency. Real time systems are critical for flight control performance and safety, as they guarantee that flight control tasks will be completed in a certain amount of time, and are essential for the safety and time-critical performance of UAVs. Luci uses a real time operating system called NuttX, which is highly expansive and configurable.

Layer 2: Middleware
The middleware is a collection of tools, drivers, and libraries that relate to flight control. It contains device drivers that handle sensors and other peripherals. It also contains flight control libraries such as RC protocols, math utilities, and control filters.

Layer 3: Flight Control
The flight control layer is the brains of the operation; this layer contains all of the command and control routines. Things like state estimation, flight control, system calibration, telemetry, motor control, and other flight control aspects reside in this layer.

2.1 PX4 Stacks

The Pixhawk platform can be flashed with two very popular flight stack: ArduPilot and PX4. In this thesis it has been adopted the PX4 software because:

- Its software-in-the-loop (SITL) simulation is much more developed and matured.
- Supports a much larger number of peripherals, including more IMU sensors, lidar, range finders, status indicators, optical flow, and motion capture units. PX4 supports the most advanced sensing peripherals for drones.
- Contains advanced command and control functionality, including things like terrain estimation, and indoor flight correction.
- More ubiquitous and built with advanced drone applications in mind. It can be compiled for POSIX (Linux) systems, and it can also integrate with ROS to run

flight applications in a hybrid system, with some running on an underlying real-time OS, and others running on Linux using ROS to communicate.

The choice was mainly driven by the more developed SITL environment and framework provided by the PX4 community and for the more advanced integration with ROS rather then a matter of performance or features, where ArduPilot stack prove to be good as well. The diagram in Figure B.3 provides a detailed overview of the building blocks of PX4. The top part of the diagram contains middleware blocks, while the lower section shows the components of the flight stack.

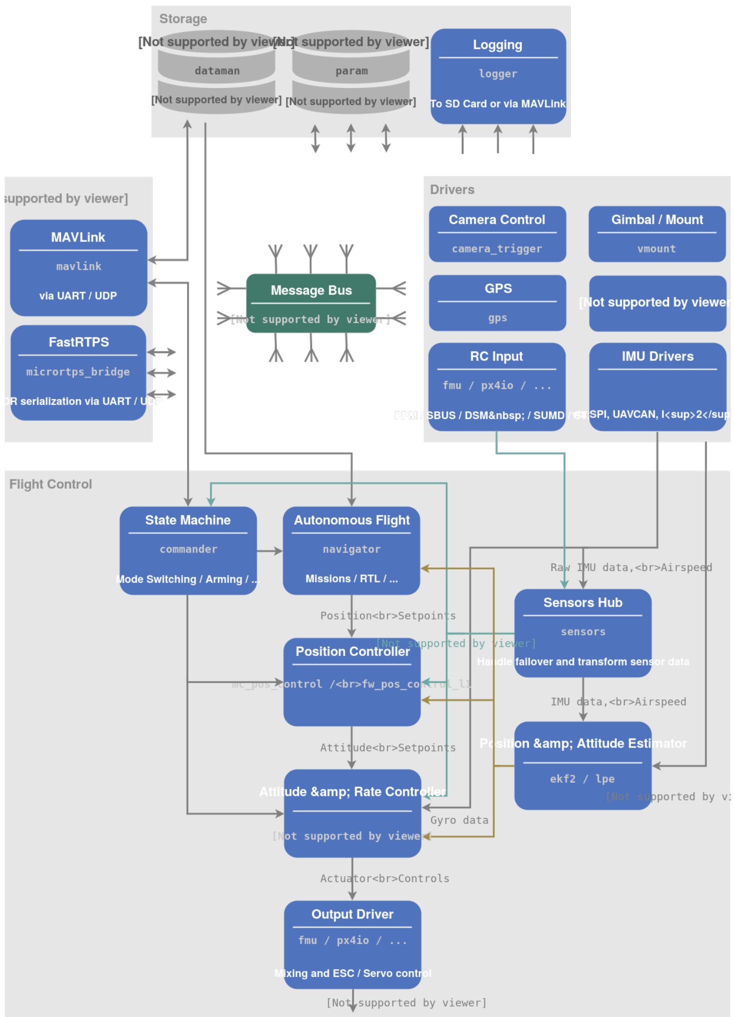

Figure B.3: PX4 Architecture

Appendix C

MAVLink Protocol

"MAVLink is a very lightweight messaging protocol for communicating with drones (and between onboard drone components).
MAVLink follows a modern hybrid publish-subscribe and point-to-point design pattern: Data streams are sent / published as topics while configuration sub-protocols such as the mission protocol or parameter protocol are point-to-point with retransmission.
Messages are defined within XML files. Each XML file defines the message set supported by a particular MAVLink system, also referred to as a "dialect". The reference message set that is implemented by most ground control stations and autopilots is defined in common.xml (most dialects build on top of this definition).
The MAVLink toolchain uses the XML message definitions to generate MAVLink libraries for each of the supported programming languages. Drones, ground control stations, and other MAVLink systems use the generated libraries to communicate. These are typically MIT-licensed, and can therefore be used without limits in any closed-source application without publishing the source code of the closed-source application." [7]

Key Features

- Very efficient. MAVLink 1 has just 8 bytes overhead per packet, including start sign and packet drop detection. MAVLink 2 has just 14 bytes of overhead (but is a much more secure and extensible protocol). Because MAVLink doesn't require any additional framing it is very well suited for applications with very limited communication bandwidth.

- Very reliable. MAVLink has been used since 2009 to communicate between many different vehicles, ground stations (and other nodes) over varied and challenging communication channels (high latency/noise). It provides methods for detecting packet drops, corruption, and for packet authentication.

- Supports many programming languages, running on numerous microcontrollers/operating systems (including ARM7, ATMega, dsPic, STM32 and Windows, Linux, MacOS, Android and iOS).

- Allows up to 255 concurrent systems on the network (vehicles, ground stations, etc.)

- Enables both offboard and onboard communications (e.g. between a GCS and drone, and between drone autopilot and MAVLink enabled drone camera).

1 List of Figures

2 List of Tables

3 Listings

4 Index