Department of Electrical and Computer Engineering
Schulich School of Engineering
University of Calgary

Fall 2012

4th Year Project Report #2
Preliminary Design Document

Group # 30

Group Name:
Team Lunch

Autonomous UAV Terrain Avoidance System

Group Members:
Rami Abou Ghanem,
Adam Dickin,
Jennifer Patterson,
James Thorne

Sponsor' Name:
CDL Systems

Dec 3, 2012

Table of Contents

Introduction

Over the past few years, there has been a huge increase in drones being used for military and police operations. Many of these drones do not have any system in place for collision detection and avoidance besides notifying the operator so that they can change course. This system is insufficient for small, highly maneuverable drones in enclosed or cluttered spaces. Additionally, operator reaction times are too slow to successfully avoid collision, especially when considering the latency introduced by a wireless control harness. Finally, many of the commercial systems available do not focus on being affordable, but rather on including every feature possible. This results in a costly drone that is equipped for many tasks, making drone adoption difficult for casual users.

Our goal as Team Lunch is to solve these problems, and to prove that every autonomous drone can be equipped with our solution. Our project will benefit our customer, CDL Systems Ltd., by demonstrating how such a system can be implemented cheaply with off-the-shelf technology available to the general consumer. Our project will also benefit the rest of the UAV market, because we will prove that such features are possible at a low cost. We hope that with time, they will become standard on every UAV. Such an adoption will greatly increase the safety of drones, not only by reducing the risk of a costly crash, but also by reducing the risk of injury to persons and damage to property.

We have decided to focus on taking a commercially available, off the shelf consumer UAV, and equipping it with an autonomous collision detection and avoidance system. This system will allow us to prove that automated accident avoidance is possible during operation of a drone, with minimal operator skill required. Our solution will allow unskilled operators to purchase and fly the UAV in close-quarter environments with minimal risk, training, and possibility of damage. Our goal is to convince drone manufacturers to adopt the use of a collision avoidance and detection system in production model aircraft. We hope that with increased availability and lower cost, many more organizations will decide to use drones in their everyday operations.

System Level Architecture

In order to help explain how our system works collectively, we have created a diagram that shows the interaction between all of the components of our system.

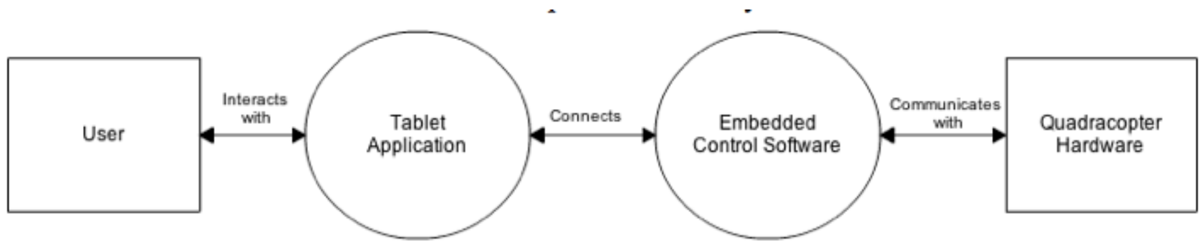

Figure 1: System Interaction Diagram

The system we have proposed is built off of a few existing systems in order for us to avoid the work that would be needed in order to have a flying quadcopter. We are building our collision avoidance software on top of a Parrot AR.Drone 2.0 quadcopter that already has its own control software, which controls all of the hardware on the drone except for the sensors that we are adding. The quadcopter has built-in Linux software, which runs the autopilot software that handles commands for the quadcopter to fly in different directions. The sensors that we have selected for our project are from a local vendor in Calgary called Phidgets. These sensors are ultrasonic sensors that provide us distance readings in between the range of 154 mm to 6.5 m [4]. Lastly, we will be using an Android tablet with an open source tablet application that we will modify to fit our needs. The tablet and application we will modify have yet to be determined since they are not needed for the first and primary phase of our project.

Preliminary Design

Design Phases

For our system we have broken the design into three core phases, with the end of each phase representing a milestone for the project. Each phase is discussed below:

Phase 1: Initial Prototyping

The initial phase of our project is a verification and validation of our functional requirements, through a series of design prototypes. Discussed further below, these prototypes will allow our team to evaluate the technical feasibility of our project, as well as ensure our customer is satisfied with our project plan. At the end of this phase, we expect to have a flight-capable AR.Drone, equipped with an ultrasonic sensor array and autopilot command proxy. This indicates we have met our Hardware Testing milestone.

While the prototype hardware and software will not be capable of meeting any of the project's functional requirements, their combination will serve as the a baseline from which the rest of the project will be developed. This baseline also serves as a synchronization point with our customer, at which we can ensure that we are developing our project according to their needs, and at which they can safely add, change, or remove requirements without invalidating any work in progress.

Phase 2: Collision Avoidance

In the second phase of our project, we will develop software for the hardware platform prototyped in Phase 1. This autonomous autopilot software will be responsible for avoiding collisions between the AR.Drone and fixed obstacles such as walls, pillars, and cabinets. The main technological challenges in this phase of the project are signal analysis from the sensor array, and development of the flight control algorithm.

Once Phase 2 of the project is complete, we will have met our Collision Avoidance milestone. This milestone is the first point at which we have delivered value to our customer, as our project will meet the basic functional requirements identified in our initial project planning and elicitation process.

Phase 3: Enhanced Operator Interaction

Once the basic collision avoidance system has been developed, our project will begin to focus on enhanced operator interaction. In Phase 3, we will develop an enhanced ground station app for an Android tablet. This ground station software will provide the vehicle operator with better situational awareness, and a more useful and intuitive control system than are possible without a collision avoidance system.

Upon completion of Phase 3, we will have met our Human Interface milestone. This milestone marks completion of our project, and delivery of our hardware schematics and software source code to our customer. Once this phase is complete, we will focus on project closeout tasks such as final budget validation and development of our Capstone Design Fair display.

Use-Case Model

Our system's use-case model is described in the following diagram.

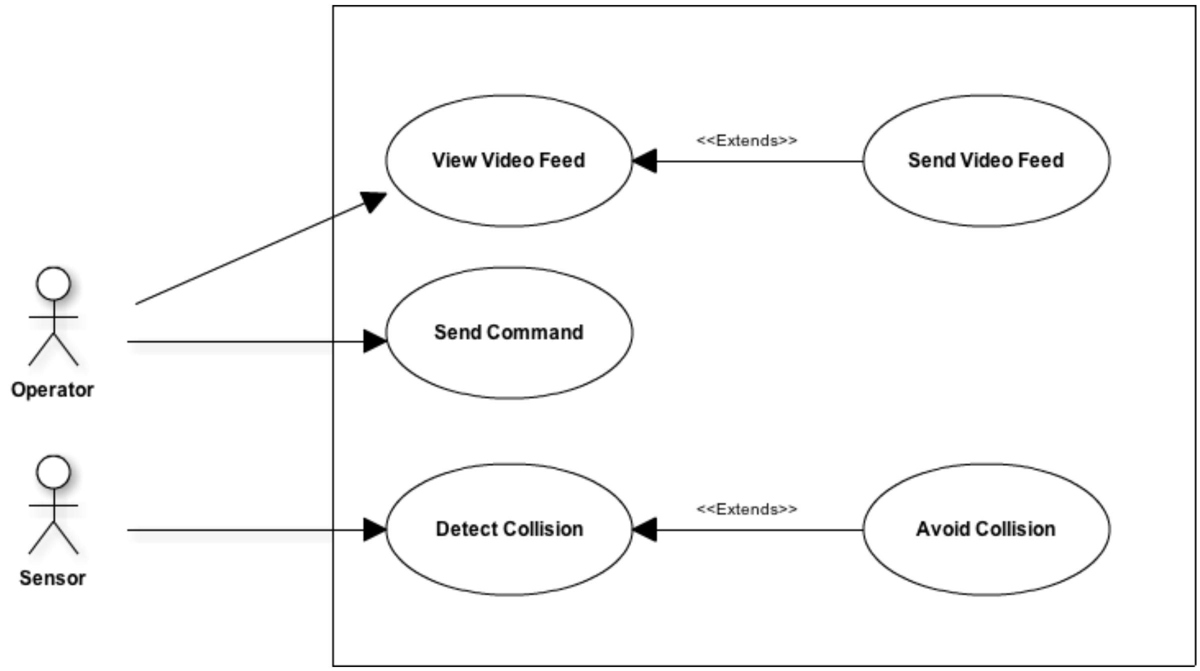

Figure 2: Use Case Diagram

The description of our use-cases is as follows:

- View Video Feed - Operator views a live video feed from the quadcopter from the application running on the Android tablet.
- Send Video Feed - The quadcopter processes the video feed from the camera and transmits it to the Android tablet
- Send Command - The Android tablet sends a command to move the quadcopter in a certain direction.
- Detect Collision - Sensors feed data into quadcopter and software uses data to detect a collision.
- Avoid Collision - Quadcopter reroutes movement to avoid an incoming collision that was detected.

Hardware Diagram

The hardware components of our system can be seen in the following diagram. Our system is essentially composed of a Parrot AR.Drone 2.0 quadcopter with 4 ultrasonic sensors wired to a microcontroller, which is attached to the quadcopter. This component will interface with an Android tablet through Wi-Fi to complete our system's hardware.

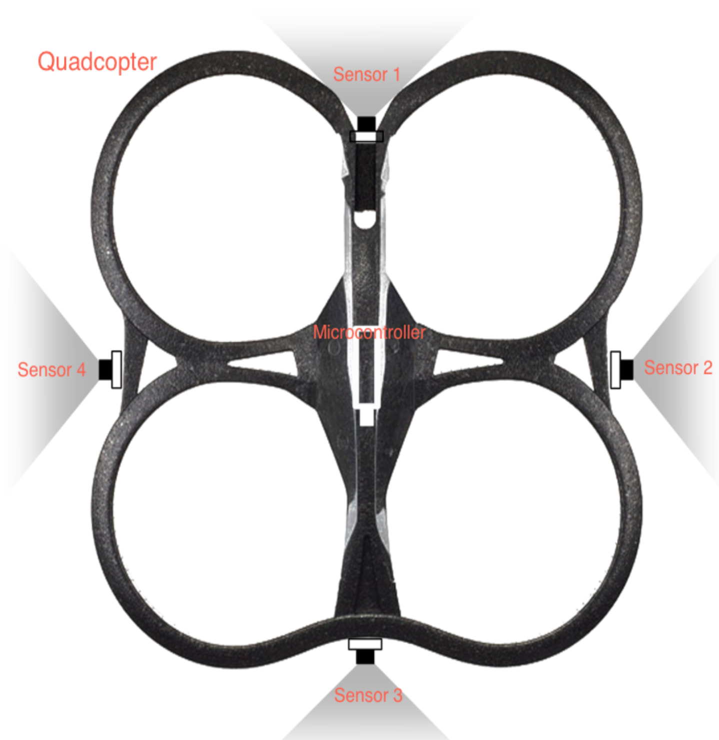

Figure 3: Hardware Diagram

Design Principles

Usability

By examining our design, it is clear that our system will be usable. Our sponsor's requirements will be met through the implementation of our design. One of our main goals for this system is to present the customer/sponsor with easy to use software that demonstrates the capability of our collision avoidance system in an effective manner.

Consistency

Our design must present a consistent experience for the end user. By building the core autonomous autopilot functionality into the quadcopter, rather than the ground control station, the vehicle will behave in a consistent manner for any combination of hardware and software. Additionally, as the AR.Drone already supports several high-level flight stabilization technologies, we are increasing the consistency of the end-user experience by filling out the vehicle's portfolio of user safety technologies.

Scalability

Our design does not rely on any specific aspects of the Parrot AR.Drone, other than its Linux-based software environment. Going forward, our project can be easily scaled to support any type of UAV by modifying only a small fraction of the overall project structure. Our key architectural design involves a separation of physical hardware (flight controls and sensor data) from the actual software algorithm, allowing us to scale the core functionality up or down to meet customer's needs.

Maintainability

Our design will be largely maintainable and modifiable. It will be able to accommodate change with regards to many features, through a system of well-architected code and automated acceptance tests. These design features and tests will allow us to refactor and change core functionality, without any risk of unintended changes to other functionality. This allows for straightforward and safe maintenance of our project, even as its scope may expand.

Manufacturability

As our system is primarily software based, it is straightforward to manufacture and duplicate. To move from a prototype to a production design, our team would work with vehicle hardware manufacturers to include the additional ultrasonic sensor suite. Once the sensors are integrated into the vehicle hardware, the full autopilot system can be easily manufactured and deployed on the vehicle's firmware.

Representability

Our design is clearly represented throughout this report. Through the use of textual descriptions and accurate diagrams, a third party will be able to make sense of what our system is and how we will be creating it. Clear design, testing and commenting of our source code will also serve as a direct representation of our functionality, allowing our customer to safely use and further develop our project after the course period has completed.

Prototyping

As requested by our customer, we have used an extensive prototyping approach to validate and verify our high-level design. We have created individual prototypes of both our key hardware (sensor platform, quadcopter) and software (command proxy, sensor data analysis) components. We have also created an end-to-end hardware prototype, which is capable of sustained flight while both recording sensor data and proxying incoming commands.

Initial Software Prototype

Our first prototype was used to ensure that we could run custom code on the quadcopter, and that our code could modify operator commands before they reached the autopilot. To ensure the prototype was simple, our code would take all commands from the control station and 'drop' them, ensuring that the autopilot would never receive commands unless we intended it to. The diagram below shows the sequence of events to ignore a command.

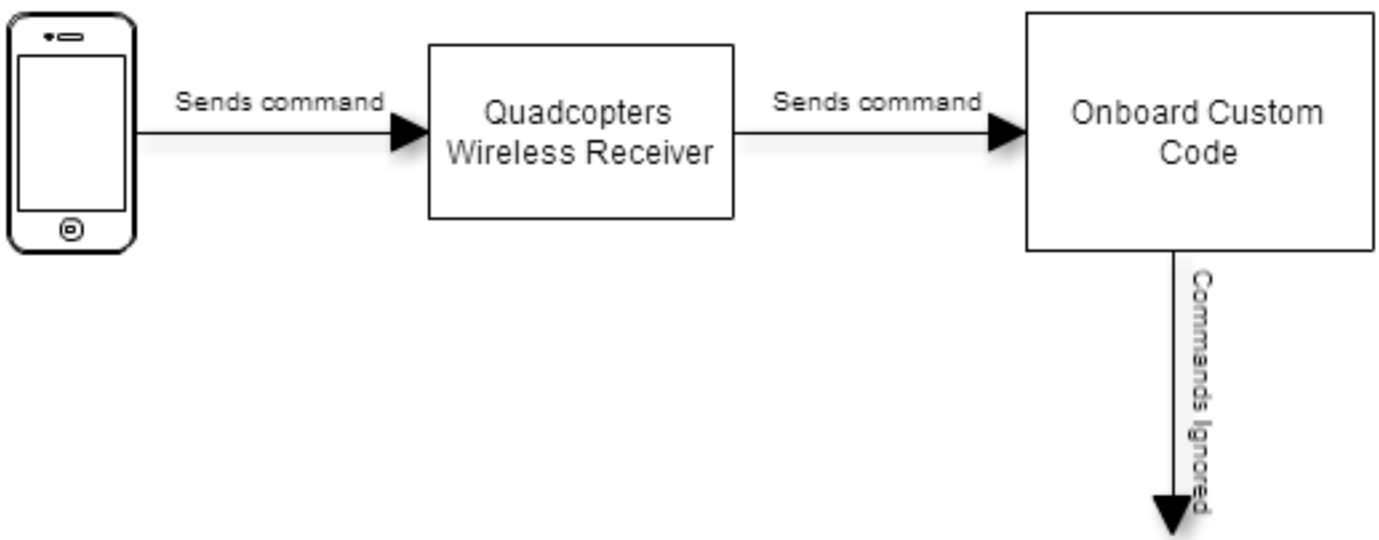

Figure 4: Software Prototype

Initial Hardware Prototype

The second prototype that we developed was a ground-based test bench, which connected all of our sensors to the development phidget interface board. The purpose of this test was to ensure that we could receive sensor data quickly enough and accurately enough to power our collision avoidance system. Below is a diagram of the prototype setup. The data sent to the laptop was simply outputted to the console, and then analyzed manually for accuracy and precision.

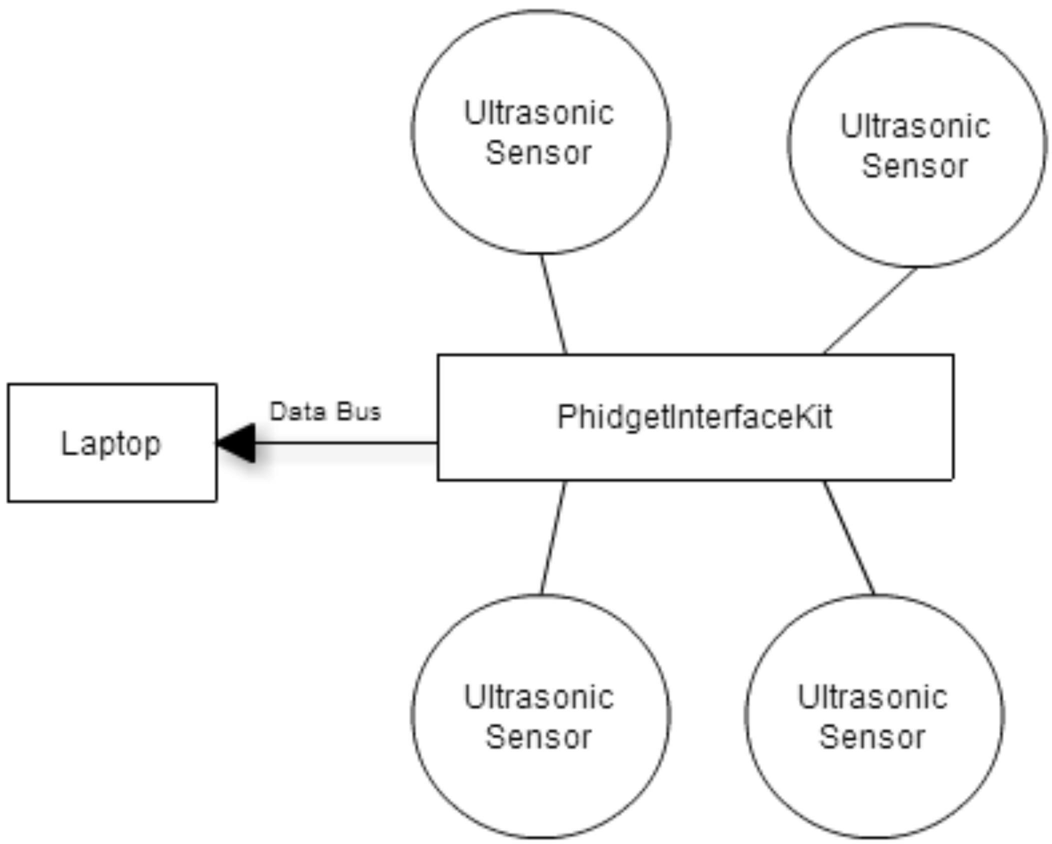

Figure 5: Hardware Prototype

List Of Materials

A complete list of our materials for this project is:

- 1 AR Drone 2.0 Quadcopter
- 1 Embedded Phidget Interface Kit
- 1 Development Phidget Interface Kit
- 4 MaxBotix EZ-1 Sonar Sensors
- 4 60cm long Sensor Cables
- 1 10 inch Android Tablet
- 1 ½ foot long USB to micro USB cable
- 2 AR Drone 1000 mAh batteries
- 1 Roll of ultra-thin double sided tape

Conclusion

We have now thoroughly outlined the requirements and design for producing a collision avoidance system for the Parrot AR.Drone. All required materials have been purchased and a prototype has been completed. After careful analysis, we have proven that our product is feasible, marketable, and beneficial to potential end users. At this point, we feel we are prepared to move on to implementation of the main product.

Glossary

CDL Systems - Our sponsor. The CDL does not stand for anything.

UAV - Unmanned Aerial Vehicle.

USB - Universal Serial Bus

Quadcopter - Also known as a quadrotor; a multicopter with 4 rotors.