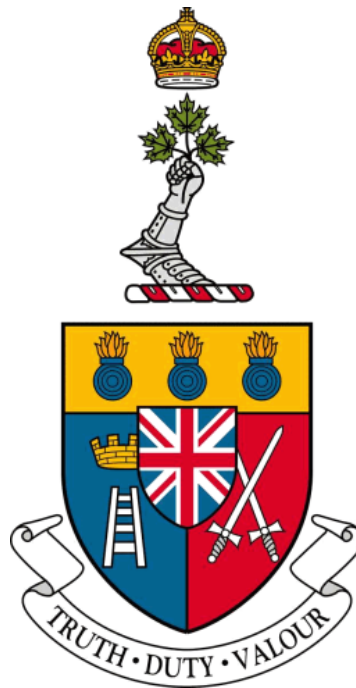


ROYAL MILITARY COLLEGE OF CANADA

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING



DID-03 - Statement of Requirements

Presented by:

Amos Navarre HEBB & Kara STEPHAN

Presented to:

Dr. Sidney GIVIGI

September 27, 2018

Contents

1	Introduction	4
1.1	Document Purpose	4
1.2	Background	4
1.2.1	Obstacle Avoidance	4
1.2.2	Uncrewed Aerial Vehicles	4
1.3	Definitions	4
1.3.1	Flying Robot	4
1.3.2	Marker	5
1.4	Aim	5
1.4.1	Targets of Interest	5
1.4.2	Complex Environment	5
1.5	Scope	5
2	Requirement Definition Activities	5
2.1	Information	5
3	Product Requirements	6
3.1	Functional Requirements (FR)	6
3.1.1	FR-01: State Machine	6
3.1.2	FR-02: Movement Toward Target	6
3.1.3	FR-03: Trajectory Library	6
3.1.4	FR-04: Trajectory Library Updating State Machine	6
3.1.5	FR-05: Identification of Target	6
3.1.6	FR-06: Avoiding Obstacle	6
3.1.7	FR-07: Identification of Obstacle	6
3.1.8	FR-08: Multiple Obstacles	6
3.2	Performance Requirements (PR)	6
3.2.1	PR-01: Target Identification	6
3.2.2	PR-02: State Machine Corrections	6
3.3	Interface Requirements (IR)	7
3.3.1	IR-01: Turtlebot Communication through ROS	7
3.3.2	IR-02: Air Robot Communication	7
3.3.3	IR-03: Air Robot Radio Denied	7
3.4	Simulation Requirements (SimR)	7
3.4.1	SimR-01: Air Robot	7
3.5	Implementation Requirements (ImpR)	7
3.5.1	ImpR-01: Gazebo Simulation	7
3.5.2	ImpR-02: Turtlebot Robot Operating System	7
3.6	Schedule Restrictions (SR)	7
3.6.1	SchR-01: Simulation	7
3.6.2	SchR-02: TurtleBot Prototype	7
3.6.3	SchR-03: Flying Prototype	7
4	Risk Assessment	8
4.1	Identifying Markers	8
4.1.1	Likelihood	8
4.1.2	Impact	8
4.1.3	Process Solution	8
4.2	Hardware Limitations	8
4.2.1	Likelihood	8

4.2.2	Impact	8
4.2.3	Process Solution	8
5	Conclusion	9

1 Introduction

1.1 Document Purpose

The purpose of this document is to outline the project requirements. That is, what the project is to accomplish once all of the requirements outlined in this document have been completed and to what standard they shall be considered done. The benefits of meeting these requirements and solving this problem will be. This document will then identify the constraints that these requirements impose on this project will be.

1.2 Background

1.2.1 Obstacle Avoidance

Both in the consumer and professional sectors the use of autonomous aerial vehicles is growing quickly. Obstacle avoidance is the task of satisfying a control objective, in this case moving toward a visual target, while subject to non-intersection or non-collision position constraints. Those later constraints are, in this case, to be dynamically created while moving in a reactive manner, instead of being pre-computed.

1.2.2 Uncrewed Aerial Vehicles

Very generally any powered vehicle that uses aerodynamic forces to provide lift that does not carry a human operator can be considered an uncrewed aerial vehicle. Currently most of these vehicles make up a single component of a larger Uncrewed aircraft system.

An Uncrewed aircraft system (UAS), or Remotely piloted aircraft system (RPAS), is an aircraft without a human pilot on-board, instead controlled from an operator on the ground.

Such a system can have varying levels of automation, something as simple as a model aircraft has limited capability. An UAS capable of detecting, recognizing, identifying, and tracking targets of interest in complex environments and integrate with the systems required to process and fuse the collected information into actionable intelligence while operating in a low-to-medium threat environment is the current goal of the RPAS project by the RCAF [1].

Flying a UAS requires a secure link to the operator off-board. Maintaining this link, particularly while flying close to the ground where more opportunities for interference are introduced, and in environments where potentially hostile actors may be attempting to jam communications, necessitate a level of automation on-board capable of maintaining flight while denied navigation information.

There are many different types of approaches for this problem with UAVs, one for example is the pushbroom algorithm. This has proven successful on a flying robot traveling at high speeds [2], [3]. This system successfully combined trajectory libraries and a state machine to avoid obstacles using very little computational power even at very high speeds [2]. Another solution to obstacle avoidance on flying robots was the creation of NanoMap [4]. This allows for 3D data to be processed at a much faster rate allowing for higher speeds of the robot [4].

1.3 Definitions

Many words used during this project have many synonyms, and similarly there are many different ways to word many of the things in this project and often no conventions. For this purpose we will define terms which we intend to have more specific meanings than in normal English, or will clarify what we mean by the often vague descriptions used for many of the components of this project.

1.3.1 Flying Robot

A Flying Robot refers to any vehicle that is able to execute instructions while in the air under its own power.

1.3.2 Marker

A marker will be specifically a high contrast shape, such as one output by ArUco or a QR Code generator, that is optimized for identification by a camera.

1.4 Aim

The aim of this project is to build an air robot that is able to identify a target and move to the target, avoiding any obstacles that are in the way.

Tracking targets of interest in complex environments with an uncrewed aerial vehicle is the ultimate goal of this project.

1.4.1 Targets of Interest

This target will be an object in the environment that has already been identified visually by an operator before navigation information is denied, or pre-programmed into the UAV before operation.

1.4.2 Complex Environment

The proposed use case of this project would be an environment with obstacles that the UAV could potentially collide with. Flying at sufficiently low altitude that trees or buildings could come between the UAV and the target of interest is the core of the project.

1.5 Scope

The scope of the project is limited due to only being able to testing indoors. The range for target set up from the robot for identification is limited due this, and it constraints certain characteristics of the obstacles. To fit everything inside for testing we will be using smaller obstacles. A possible end goal of this project would be to be able to identify a target and avoid obstacles in realtime in a non-simulation environment.

2 Requirement Definition Activities

2.1 Information

To better define the requirements for this project the CAF RPAS project requirements were researched, to attempt to familiarize ourselves with the type of system outlined in the project [1]. Previously created obstacle avoidance systems implemented on similar UAVs were also researched to gain more background information for our knowledge. An article that was identified to be beneficial is about both target tracking and obstacle avoidance on a drone [5]. This outlined some specific problems that this project will likely face as well, and a possible solution to these problems. The Discussions with our project supervisor Dr. Givigi revealed that it was possible to build a monocular detection system that can identify targets and multiple obstacles at once. This then can be implemented on an air robot which can take proper action to avoid these obstacles and move towards the identified target. Those discussions inspired the project requirements detailed in the following section.

3 Product Requirements

3.1 Functional Requirements (FR)

3.1.1 FR-01: State Machine

Create a state machine for both land and air robots that allows for a movement decision to be made based on an input from the robot.

3.1.2 FR-02: Movement Toward Target

The air and land robots will be able to move toward a target under control.

3.1.3 FR-03: Trajectory Library

Create a trajectory library that will contain all possible allowed movements for both the land and air robots.

3.1.4 FR-04: Trajectory Library Updating State Machine

The Trajectory Library will update the state machine when a movement was made. The state machine will then account for this displacement and later correct the movement to keep the robot on the path to the target.

3.1.5 FR-05: Identification of Target

The robot shall recognize optically a goal and be able to give information on the targets location relative to the robot.

3.1.6 FR-06: Avoiding Obstacle

The robot shall be able to make a deviation from its current movement pattern to avoid an obstacle in its path and then return to this pattern.

3.1.7 FR-07: Identification of Obstacle

The robot shall recognize optically obstacles in its environment and identify where they are relative to itself.

3.1.8 FR-08: Multiple Obstacles

The robot should be able identify multiple obstacles and determine which is the most dangerous and avoid it accordingly.

3.2 Performance Requirements (PR)

3.2.1 PR-01: Target Identification

The robot will be able to identify and locate OpenCV's ArCuo shapes within a 15m radius. See FR-05.

3.2.2 PR-02: State Machine Corrections

The state machine will be able to correct for movements made and put the robot back on its original path without deviating more than 15cm (we need to pick a acceptable deviation). See FR-04.

3.3 Interface Requirements (IR)

3.3.1 IR-01: Turtlebot Communication through ROS

Communication to the Turtlebot will be done through Robot Operating System through USB.

3.3.2 IR-02: Air Robot Communication

The air robot will be communicating through Robot Operating System over wireless network.

3.3.3 IR-03: Air Robot Radio Denied

The air robot shall be able to identify a pre-determined marker and fly toward it while avoiding obstacles.

3.4 Simulation Requirements (SimR)

3.4.1 SimR-01: Air Robot

The final product will be an air robot, but for simulation we will be using Gazebo to create and test our systems.

3.5 Implementation Requirements (ImpR)

3.5.1 ImpR-01: Gazebo Simulation

Prior to testing on the Turtlebot, the program shall be implemented on the gazebo simulation.

3.5.2 ImpR-02: Turtlebot Robot Operating System

The simplest obstacle avoidance algorithm must be implemented on a Turtlebot using the Robot Operating System.

3.6 Schedule Restrictions (SR)

3.6.1 SchR-01: Simulation

The first simulation shall be able to operate in a Gazebo simulation environment no later than November 5th. It shall be able to identify a marker, move toward the marker, and avoid an obstacle placed into the simulation environment.

3.6.2 SchR-02: TurtleBot Prototype

The first functional prototype shall be a TurtleBot robot capable of positively identifying a marker, moving toward the marker, and avoiding an obstacle placed into its environment no later than December 18th.

3.6.3 SchR-03: Flying Prototype

The first functional flying prototype shall be capable of identifying a marker, moving toward the marker, and avoiding an obstacle placed into its environment no later than February 18th.

4 Risk Assessment

4.1 Identifying Markers

The markers we intend to use, at least initially, are intended for use in Augmented Reality purposes. Tests of these shapes show that they are capable of being identified at many angles, but the reliability with which they can be identified decreases quickly as the viewing angle changes.

There is potential that markers which are reliably identified at low speeds and in simulation may not be detectable on a flying vehicle moving faster.

4.1.1 Likelihood

Low likelihood due to low speeds that even the flying robot will be operated at.

4.1.2 Impact

High impact, the primary task that our robot must accomplish is moving toward a visual target. If the robot is not able to reliably identify a marker then there will be no way to verify its capability.

4.1.3 Process Solution

Alternatives to the ArUco markers can be tested, or if these prove truly unreliable then coloured areas or illuminated targets could be incorporated. These are less desirable as ideally the robot would be able to fly toward an arbitrary target.

4.2 Hardware Limitations

Image processing, especially quickly and with multiple goals, is computationally expensive. While this should not be an issue on a ground vehicle moving at slower speeds, it may become more of an issue on a flying robot if it operates at higher speeds and is incapable of carrying as much on-board computational capability.

4.2.1 Likelihood

High likelihood, although hardware for flying robot has not yet been chosen there are currently very few hardware devices capable of executing complex computer vision instructions quickly.

4.2.2 Impact

Medium impact, presumably we would still be able to prove our systems on the ground robot which is capable of carrying as much processing power as needed. It may also only be a limitation at higher speeds or may impose limits on what the air robot is capable of identifying and tracking.

4.2.3 Process Solution

Testing many algorithms and working to streamline the algorithm used, especially for the flying robot, as well as selecting hardware that is complimentary to the kind of loads created by running such an algorithm can mitigate these risks.

5 Conclusion

The project requirements have been listed for the air robot obstacle avoidance system. Background and requirement defining activities have been given to lay the groundwork and a basic understanding of the current research in this domain. This document will be used for guiding the project and aiding in the creation of the preliminary design.

References

- [1] N. Defence and R. C. A. Force, *Royal Canadian Air Force — news article — update and new name for the joint unmanned surveillance target acquisition system (justas) project*, 2018. [Online]. Available: <http://www.rcaf-arc.forces.gc.ca/en/article-template-standard.page?doc=update-and-new-name-for-the-joint-unmanned-surveillance-target-acquisition-system-justas-project/j9u7rzyf>.
- [2] A. J. Barry, P. R. Florence, and R. Tedrake, “High-speed autonomous obstacle avoidance with pushbroom stereo”, *Journal of Field Robotics*, vol. 35, no. 1, pp. 52–68, 2018.
- [3] A. J. Barry and R. Tedrake, “Pushbroom stereo for high-speed navigation in cluttered environments”, in *Robotics and automation (icra), 2015 IEEE international conference on*, IEEE, 2015, pp. 3046–3052.
- [4] P. R. Florence, J. Carter, J. Ware, and R. Tedrake, “Nanomap: Fast, uncertainty-aware proximity queries with lazy search over local 3d data”, *arXiv preprint arXiv:1802.09076*, 2018.
- [5] A. C. Woods and H. M. La, “Dynamic target tracking and obstacle avoidance using a drone”, in *International Symposium on Visual Computing*, Springer, 2015, pp. 857–866.