|  |
| --- |
|  |
| Autonomous Landing System for a VTOL UAV on the Deck of a Ship Using Infra-Red Beacons |
|  |
| BSc (Hons) Computing with Software Development, 2014 Institute of Technology Tralee |
|  |
| Brian Clarke |
| T00154737 |

|  |
| --- |
|  |

Primary Supervisor: Ed Sheldon

Secondary Supervisor: Robert Sheehy

# Abstract

This project is a mathematical model to aid in the landing of a vertical take-off and landing unmanned aerial vehicle (VTOL UAV) on the deck of an ocean going vessel using infra-red beacons in a predetermined pattern on the deck of the vessel.

It is proposed to develop a system with output data that could be integrated with the navigational controls of a UAV as it attempts to land on the deck of a ship. The system will be required to map the space between the two bodies, both moving independently with six degrees of freedom. This will be achieved by mapping the coordinates of the infra-red beacons on the landing pad of a ship to the 2D plane of a camera located on the fuselage of a VTOL UAV. The image will then be evaluated to determine the next course correction for the UAV as it attempts to ultimately achieve a satisfactory touchdown.

The system will be aware of the ideal infra-red beacon pattern on the deck of the ship for a given altitude. Data will be streamed on a constant basis to the simulation of the UAV’s navigation control system to reduce the difference between the desired pattern and the observed pattern of the beacons on deck (these patterns will be represented mathematically). The simulations of the ship and UVA will also be influenced by their respective operational environments which will have a direct effect on their individual poise, altitude and trajectory. These UAV’s camera observations and the subsequent course corrections should result in synchronized motion between the ship and the UAV as touchdown approaches and is ultimately achieved.

# Table of Contents

[Abstract 1](#_Toc382055628)

[Table of Contents 2](#_Toc382055629)

[Table of Figures 5](#_Toc382055630)

[Declaration of Originality 7](#_Toc382055631)

[Research Question 8](#_Toc382055632)

[Terminology 9](#_Toc382055633)

[Abbreviations 9](#_Toc382055634)

[Descriptions 9](#_Toc382055635)

[Assumptions 10](#_Toc382055636)

[1 Operational Goals 11](#_Toc382055637)

[1.1 Introduction 11](#_Toc382055638)

[1.2 Maritime UVA Mission Goals 12](#_Toc382055639)

[1.3 Ship Interception 12](#_Toc382055640)

[12](#_Toc382055641)

[1.4 Hover over Deck 14](#_Toc382055642)

[1.5 Landing 14](#_Toc382055643)

[1.6 Summary 15](#_Toc382055644)

[2 Operational Characteristics of the Ship 17](#_Toc382055645)

[2.1 Overview 17](#_Toc382055646)

[2.2 Waves and Sea States 17](#_Toc382055647)

[2.3 Wind 18](#_Toc382055648)

[2.4 The Moving Platform 19](#_Toc382055649)

[2.5 The Operational Envelope 20](#_Toc382055650)

[2.6 Summary 21](#_Toc382055651)

[3 Operational Capabilities of the UAV 22](#_Toc382055652)

[3.1 Overview 22](#_Toc382055653)

[3.2 Wind 23](#_Toc382055654)

[3.3 Performance 23](#_Toc382055655)

[3.4 Navigation 26](#_Toc382055656)

[3.5 Response Times 26](#_Toc382055657)

[3.6 Camera 28](#_Toc382055658)

[3.7 The UAV Controller 29](#_Toc382055659)

[3.8 Summary 29](#_Toc382055660)

[4 Final Approach 30](#_Toc382055661)

[4.1 The Four Phases 30](#_Toc382055662)

[4.2 Landing Approaches 30](#_Toc382055663)

[4.3 Touchdown 30](#_Toc382055664)

[4.4 Summary 31](#_Toc382055665)

[5. Commercial and Academic Landing Assist Systems 32](#_Toc382055666)

[5.1 Commercial Systems 32](#_Toc382055667)

[5.2 Academic Systems 32](#_Toc382055668)

[5.3 Summary 33](#_Toc382055669)

[6. Project Design 34](#_Toc382055670)

[6.1 Scope 34](#_Toc382055671)

[6.2 Estimate of Activities 34](#_Toc382055672)

[6.3 Risk Analysis 35](#_Toc382055673)

[6.4 Use Case Diagrams 36](#_Toc382055674)

[6.4.1 Adaptive Control System Use Case Diagram 36](#_Toc382055675)

[6.4.2 Image Processor Use Case Diagram 37](#_Toc382055676)

[6.4.3 INS Simulator Use Case Diagram 38](#_Toc382055677)

[6.5 High Level State Diagram 40](#_Toc382055678)

[6.6 System Architecture 41](#_Toc382055679)

[6.7 Component Diagram 42](#_Toc382055680)

[6.8 Class Diagrams 43](#_Toc382055681)

[6.9 Methodology 44](#_Toc382055682)

[6.10 Test Plan 44](#_Toc382055683)

[6.11 Prototype 47](#_Toc382055684)

[7. Equations of Motion 50](#_Toc382055685)

[7.1 Degrees of Freedom 50](#_Toc382055686)

[7.2 Motion in 3D Space 50](#_Toc382055687)

[7.3 Summary 52](#_Toc382055688)

[8. Photogrammetry and Image Generation 53](#_Toc382055689)

[8.1 Introduction 53](#_Toc382055690)

[8.2 3 Dimensional Equations 55](#_Toc382055691)

[8. Project Management Diary 58](#_Toc382055692)

[8.1 Literature Review Diary 60](#_Toc382055693)

[8.2 Implementation Diary 61](#_Toc382055694)

[9. The UAV Model 63](#_Toc382055695)

[10. The Ship Model 63](#_Toc382055696)

[11. From Target Acquisition to Touchdown 63](#_Toc382055697)

[12. Results 63](#_Toc382055698)

[Appendix 64](#_Toc382055699)

[Appendix 1 Douglas Sea Scale 64](#_Toc382055700)

[Works Cited 65](#_Toc382055701)

# Table of Figures

[Figure 1. Helicopter landing on the deck of a ship in rough seas. 11](#_Toc374402527)

[Figure 2. Vector line diagram for the intercept path between a helicopter and a ship 12](file:///C:\Users\Brian\Documents\GitHub\Thesis\Main%20Project.docx#_Toc374402528)

[Figure 3. Landing processes overview for the proposed system 16](#_Toc374402529)

[Figure 4. Table of Wave types and their characteristics 17](#_Toc374402530)

[Figure 5. Diagram Showing 6 DoF of an ocean going vessel 19](#_Toc374402531)

[Figure 6. Table of values for 6 DoF diagram 20](#_Toc374402532)

[Figure 7. Section of hull diagram showing changes to buoyancy 21](#_Toc374402533)

[Figure 8. Forces acting on a helicopter in flight 23](#_Toc374402534)

[Figure 9. Rotomotion VTOL UAV 24](#_Toc374402535)

[Figure 10. Rotomotion’s SR200 Helicopter specification 24](#_Toc374402536)

[Figure 11. Schiebel Camcopter S-100 VTOL UAV 25](#_Toc374402537)

[Figure 12. Schiebel Camcopter S-100 specification 25](#_Toc374402538)

[Figure 13. Yuan's finding for response times versus changes in vector u,v and w 27](#_Toc374402539)

[Figure 14. Hood Tech Vision - 11EOIR1 28](#_Toc374402540)

[Figure 15. Hood Tech Vision - 11EOIR1 specifications 28](#_Toc374402541)

[Figure 16. Lobik's diagram showing lateral ‘slip’ as a helicopter is landing or taking off 31](#_Toc374402542)

[Figure 17 Use Case Diagram - Adaptive Control System 36](#_Toc374402543)

[Figure 18 Use Cases - Adaptive Control System 37](#_Toc374402544)

[Figure 19 Use Case Diagram - Image Processor 37](#_Toc374402545)

[Figure 20 Use Cases - Image Processor 38](#_Toc374402546)

[Figure 21 Use Case Diagram - INS Simulator 38](#_Toc374402547)

[Figure 22 Use Cases - INS Simulator 39](#_Toc374402548)

[Figure 23 High Level State Diagram 40](#_Toc374402549)

[Figure 24 System Architecture 41](#_Toc374402550)

[Figure 25 Component Diagram 42](#_Toc374402551)

[Figure 26 Image of ship prototype 47](#_Toc374402552)

[Figure 27 Code snippet of ship coordinates 48](#_Toc374402553)

[Figure 28 Images of landing pad prototype 49](#_Toc374402554)

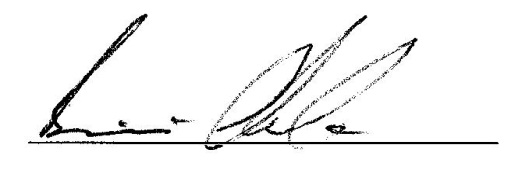
[Figure 29 Test Plan 46](#_Toc374402555)

[Figure 30 3D Plane representation 52](#_Toc374402556)

# Declaration of Originality

I hereby certify that this document titled ‘Autonomous Landing System for a VTOL UAV on the Deck of a Ship Using Infra-Red Beacons’ and the included prototype is my own work and information sourced from other sources and documents have been referenced.

Signed:



Brian Clarke.

Date: 12th December 2013

# Research Question

Given a certain sea state is it possible to land a VTOL UAV autonomously on the deck of a ship by observing infra-red beacons on the landing pad and their relative movements is three dimensional space?

# Terminology

## Abbreviations

|  |  |
| --- | --- |
| 2D | Two Dimensions |
| 3D | Three Dimensions |
| 6DoF | Six Degrees of Freedom |
| CCD | Charge Coupled Device – e.g. Camera sensor |
| CoG | Centre of Gravity |
| GM | Metacentric Height |
| GPS | Global Position System |
| INS | Inertial Navigation Sensor |
| IP/TCP | Internet Protocol – Transmission Control Protocol |
| MRAC | Model Reference Adaptive Controller. |
| PC | Personal Computer, either desktop or laptop. |
| RPAS | Remotely Piloted Air System |
| UAV | Unmanned Aerial Vehicle |
| UDP | User Datagram Protocol |
| VTOL | Vertical Take Off and Landing |
| VBN | Vision Based Navigation |

## Descriptions

|  |  |
| --- | --- |
| Centre of Gravity | One point in the ship where mass is equal in opposite directions for all 3 axes |
| Heave | Vertical motion in the direction of the z axis |
| Pitch | Rotation about the y axis |
| Roll | Rotation about the x axis |
| Sea State | Douglas Sea Scale/State – This is a number devised by Captain H.P. Douglas CMG, RN in 1929 to estimate the roughness of seas and gives a range of the expected wave heights (Met Office, 2010). The full chart can be seen in Appendix 1 |
| Surge | Thrust in the direction of the x axis |
| Sway | Lateral motion in the direction of the y axis |
| Yaw | Rotation about the z axis |

# Assumptions

It is assumed that the flight characteristics of a VTOL UAV are similar to those of its manned equivalent all other parameters being equal. This is solely for the purposes of discussion.

When the simulated image of the beacons on the ship is generated there is no allowance made for lens distortion or any lens refraction issues that may or may not occur in a single or array of lens elements. For the purposes of this project implementation a fixed point coordinate is used as the focal point of the camera simulation.

# 1 Operational Goals

## 1.1 Introduction

Unmanned aerial vehicles (UAVs) are used extensively in the military and civil sectors for a multitude of tasks including scientific endeavours, environmental data collection and military reconnaissance. Vertical Take Off and Landing (VTOL) UAV’s have an advantage over fixed wing UAV’s in that they can take off and land from any flat surface large enough, within safe operational parameters, to accommodate the vehicle. This has insured the VTOL UAV’s place in maritime operations. However landing at sea can present its own challenges if (as shown below in Figure 1) the poise of the landing platform is in a constant state of change. The issue is compounded by the helicopter position and motions relative to that of the ship. It was reported in 2012 that the first fully autonomous landing of a VTOL UAV on the deck of a ship was successfully achieved by Boeing and NovAtel in July of that year (NovAtel, 2012). This highlights the relative infancy of this new technology and is the subject of this document.



Figure 1. Helicopter landing on the deck of a ship in rough seas.

(Prism Defence, 2010)

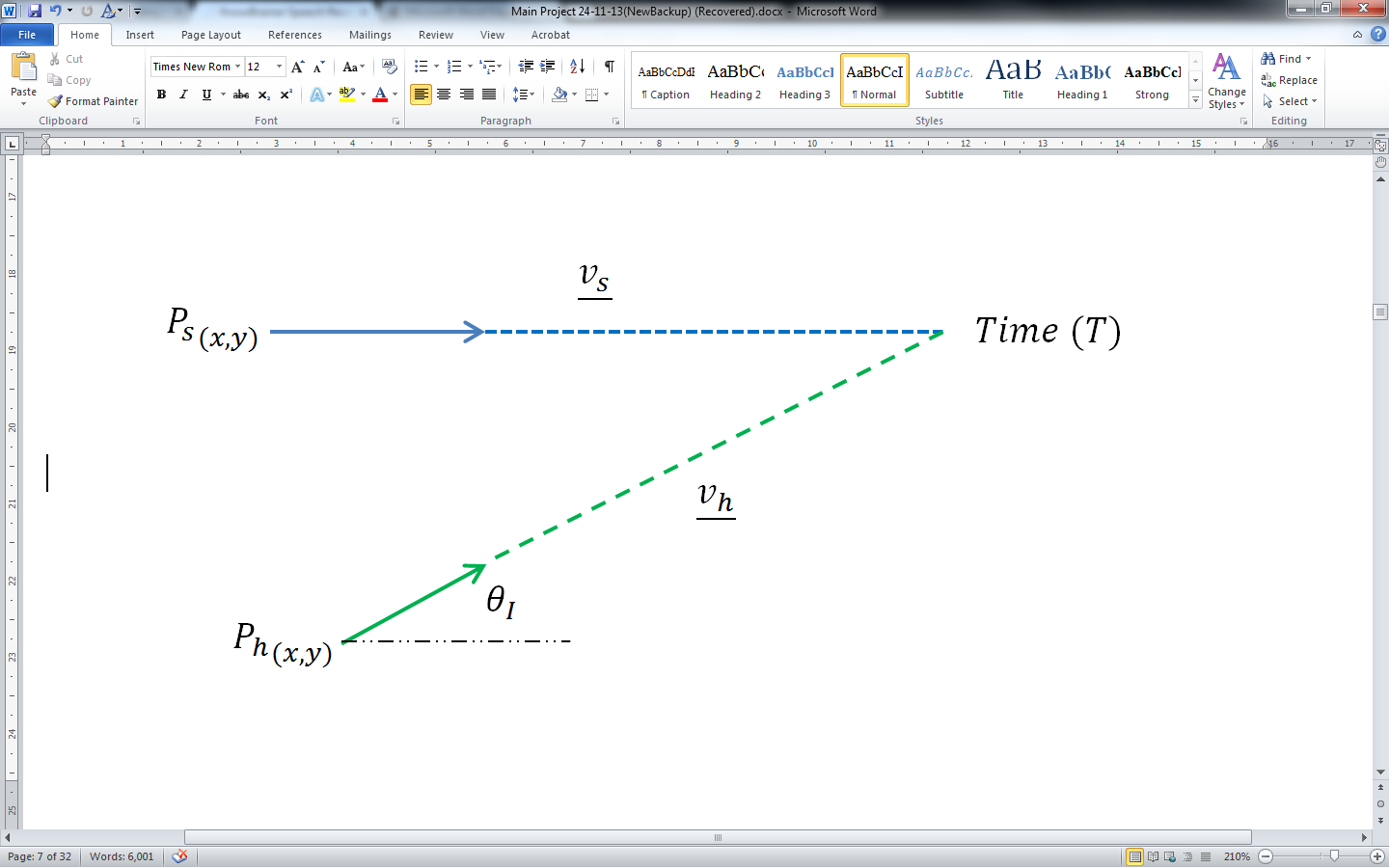
## 1.2 Maritime UVA Mission Goals

In order to complete a safe mission a UAV must take off, achieve its primary objective, return to the ship and complete a safe touchdown. During this flow of events there are many factors that have a bearing on the success or failure of the assignment. These include climatic wind conditions, the endurance (maximum operational time) of the UAV, the sea state the ship is operating in, the distance to the mission goal, and the relative speeds and directions of the ship and UAV. For a given set of determining factors, and operator’s experience, the UAV pilot will decide on the feasibility of a successful mission and act accordingly.

## 1.3 Ship Interception

In order to satisfactorily complete the mission a UAV must perform the main objective and return to the ship from which it commenced its journey. The UAV is in live contact with the UAV pilot and the ships navigational system therefore the distance and trajectory of an intercept path are calculated.

This optimum intercept path (see figure 2) is calculated based on the ships current bearing and speed, data from the UAVs Inertial Navigation Sensor (INS) system, the GPS coordinates of both ship and UAV and the UAVs relative speed when present climatic conditions have been accounted for. These calculations are computed regularly and any corrective measures are implemented as soon as detected.



## 

Figure 2. Vector line diagram for the intercept path between a helicopter and a ship

Where;

is the present location of the ship,

is the present location of the helicopter,

Time (T) is a moment in time when an interception is possible assuming interception is possible,

is the known velocity of the ship,

is the known velocity of the helicopter, and

is the unknown intercept angle relative to the ships trajectory.

Assuming the ship and helicopter meet at some point in time (T) then;

|  |  |  |
| --- | --- | --- |
|  |  | eq. 1 |
|  |  | eq. 2 |

Equation 2 can be extrapolated out in its x and y coordinates and squared to find two equations in T with other known variables;

|  |  |
| --- | --- |
|  | eq. 3 |
|  | eq. 4 |

If the assumption is made that the ships starting point is 0,0 and its traveling along the x axis, and knowing that;

Then if the two equations 3 and 4 are added;

|  |  |
| --- | --- |
|  | eq. 5 |

From equation 5 a value for (T) can be found that satisfies the other equations 3 or 4 and so the intercept angle relative to the ship can be calculated.

## 1.4 Hover over Deck

When the UAV has intercepted the vessel and landing is imminent, it may be necessary for the UAV to maintain a hover position a safe altitude above and slightly to the rear of the ship. At this juncture the UAV should match the ships bearing and speed and remain at its present altitude until it is time to begin final approach. At this point the UAV may have entered the operational envelope of the ship and could be prone to adverse effects on stability as it deals with the airwake of the ship and changes to the air vortices that provide lift to the aircraft. As the environmental factors that govern the intensity and fluctuation of the airwake are variable it would not be advisable to remain in the airwake of the ship any longer than necessary. If the landing pad on the deck of the ship is obstructed for any reason and landing is unadvisable, the UAV should ascend out of the ship’s turbulence and defer landing until the situation is resolved.

## 1.5 Landing

“Landing a UAV is the most critical and accident prone phase of the flight” (Din, et al., 2012). There are additional factors to be considered as the UAV closes the distance between itself and the ship.

* The motion of the ship in six degrees of freedom (6DoF) as it reacts to the disruptive forces induced by the waves as they strike the ship. The two motions that have the greatest bearing on a successful landing are roll and pitch (discussed further in section 2.3 The Moving Platform).
* The climatic winds and subsequent turbulence generated as winds traverses the deck of the ship.
* The flight dynamics of the UAV can change as it nears touchdown. The air wake below the main rotors begins to interfere with the normal operational vortex of the rotors which in turn can lead to reduced rotor efficiency. If not checked the rate of decent can increase and landing can become extremely hazardous (Padfield, 2007).
* Any masts, railing, antennas or other obstacles the may impede the helicopters approach and corrective action will need to be taken.

All these factors must be considered to insure the landing phase and touchdown of the UAV are performed in as safe an operational envelope as possible.

The factors that influence the landing envelope and the processes involved in landing the helicopter are shown below in figure 3, which also includes a proposed Model Reference Adaptive Controller (MRAC). This controller bases the output to the flight controllers on the system inputs and also the previous MRAC outputs. This controller system provides a degree of “stable control during parameter uncertainty” (Kamalasadan & Ghandakly, 2011) and is discussed further in section 3.7 The UAV Controller.

## 1.6 Summary

There are many factors that influence the performance of a VTOL UAV as it attempts to land on the deck of a ship. The system proposed will produce a mathematical model for the elements highlighted in the diagram of systems, shown below i.e. UAV Camera, Image Processer and MRAC. The system will interpret the coordinates of the infra-red beacons as they might appear on a 2D plane representative of an image acquired by a camera located on the under carriage of the UAV. The coordinates will then be passed to the controller which will decide on the next course correction, if any, based on all the available inputs, and also the last set of outputs the controller provided to the flight controls.



Figure 3. Landing processes overview for the proposed system

# 2 Operational Characteristics of the Ship

## 2.1 Overview

There are many ships and other ocean going vessels ‘battling the seas’ around the globe at any moment in time. They all need to overcome the adverse factors that can hamper their progress. The main force that acts on a ship as it operates at sea is the power, frequency, direction and height of the waves it encounters. To a lesser extent the force of the wind is also a factor in that it can influence the profile of the waves as they strike the ship. How a ship or another vessel responds to these forces is dependent on the craft’s design characteristics and buoyance at that time.

## 2.2 Waves and Sea States

As an ocean going vessel carries out its operational goals it is subjected to the forces of the sea and its local environment. The main forces acting on a vessel (excluding thrust) are those associated with waves and the ships reaction to its instability relative to the centre of gravity and buoyancy characteristics of the vessel.

In 1929 Captain H.P. Douglas CMG, RN estimated a set of codes to describe the roughness of seas and give a range of the expected wave heights (Met Office, 2010). The full chart can be seen in Appendix 1. These classifications range from Sea State 0 – 9 with respective wave heights of 0m to above 14m.

Waves can also be classified by their type and the table below shows the criteria for each type of wave.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Typical Periods** | **Wave Lengths** | **Forcing Mechanism** |
| Ripples | <0.2 sec | 10-2 m | wind on sea surface |
| Sea | 0.2 – 9 sec | 130 m | wind on sea surface |
| Swell | 9 – 30 sec | 100s of metres | wind on sea surface |
| Tsunamis | 15 minutes – 1 hour | 100s of km | seismic |
| Tides | Several hours | 100s – 1000s of km | gravitational |

Figure 4. Table of Wave types and their characteristics

(Florida Centre for Instructional Technology, 2005)

Ripples are the smallest of the various types of waves and are generated as a response to light winds acting on the surface of the ocean. Sea waves are generated by stronger local winds as they act on the surface and can travel in various directions and with various speeds. Swell waves on the other hand tend to travel long distances in one direction. They are more regular in amplitude and wavelength and are normally a response to climatic conditions far away. Tsunamis and tides are other types of waves and occur as a response to global forces (Florida Centre for Instructional Technology, 2005). They also suggest that to aid in the study of waves, ideal sinusoidal shaped wave structures should be considered. A sine curve can be discribed as:

where:

= the coordinate value for y at a given time t.

= the time interval during a single sinusoidal wavelength.

= amplitude of the wave, measured from the mean value of the wave i.e. center line.

= frequency of the wave, the number of times per second the wave completes a full cycle.

For a complete study however the apparently random differences in amplitude, wavelength, and frequency within the Sea State ranges would need to be accounted for, to develop a more complete model.

Individual wave types also have various characteristics based on their stage of development. During the lifecycle of a wave, it is either developing, fully developed or decaying. A wave can be said to be fully developed “when wind speed matches wave crest phase speed” (Techet A. H., 2005). A wave may travel for hundreds of miles in this state but it will start to decay if the other forces acting on the wave start to overcome the kinetic energy of the wave.

## 2.3 Wind

As described before wind plays a large part in the generation of waves which affect the vessel. To a lesser extent wind can have an effect on the performance of a ship. This effect on performance would normally not be directly attributable to the ship itself but rather the crew operating on the ship and their ability, or lack of, to perform their individual tasks. In extreme conditions however the winds impact on the ship may become more pronounced e.g. a very strong cross wind may contribute to the roll of a ship in those conditions.

One of the main concerns with wind is its interaction with the ship’s superstructure. The airwake and turbulence generated by the vessel as it passes through wind should be considered when deciding if it is appropriate to proceed with the launch or recovery of a UAV. As reported by (Snyder, 2012) wind’s “turbulent kinetic energy is significantly greater in the superstructure wake than in the free stream flow observed by the bow”. This is discussed later in this chapter during the section on the operational envelope,

## 2.4 The Moving Platform

A vessel on the surface of the ocean moves with six degrees of freedom (6DoF) and in accordance with the forces acting on it (see figure 5), which can be both external and internal. The two main external forces acting on a ship in the ocean are waves and gravity. There are others, such as wind and drag but both these are not as influential on the motion of the ship as waves and gravity. The internal forces acting on a ship are thrust, the reaction to instability relative to buoyancy and CoG and rudder induced turning moments. The diagram below gives a visual representation of the 6DoF for a vessel in water.

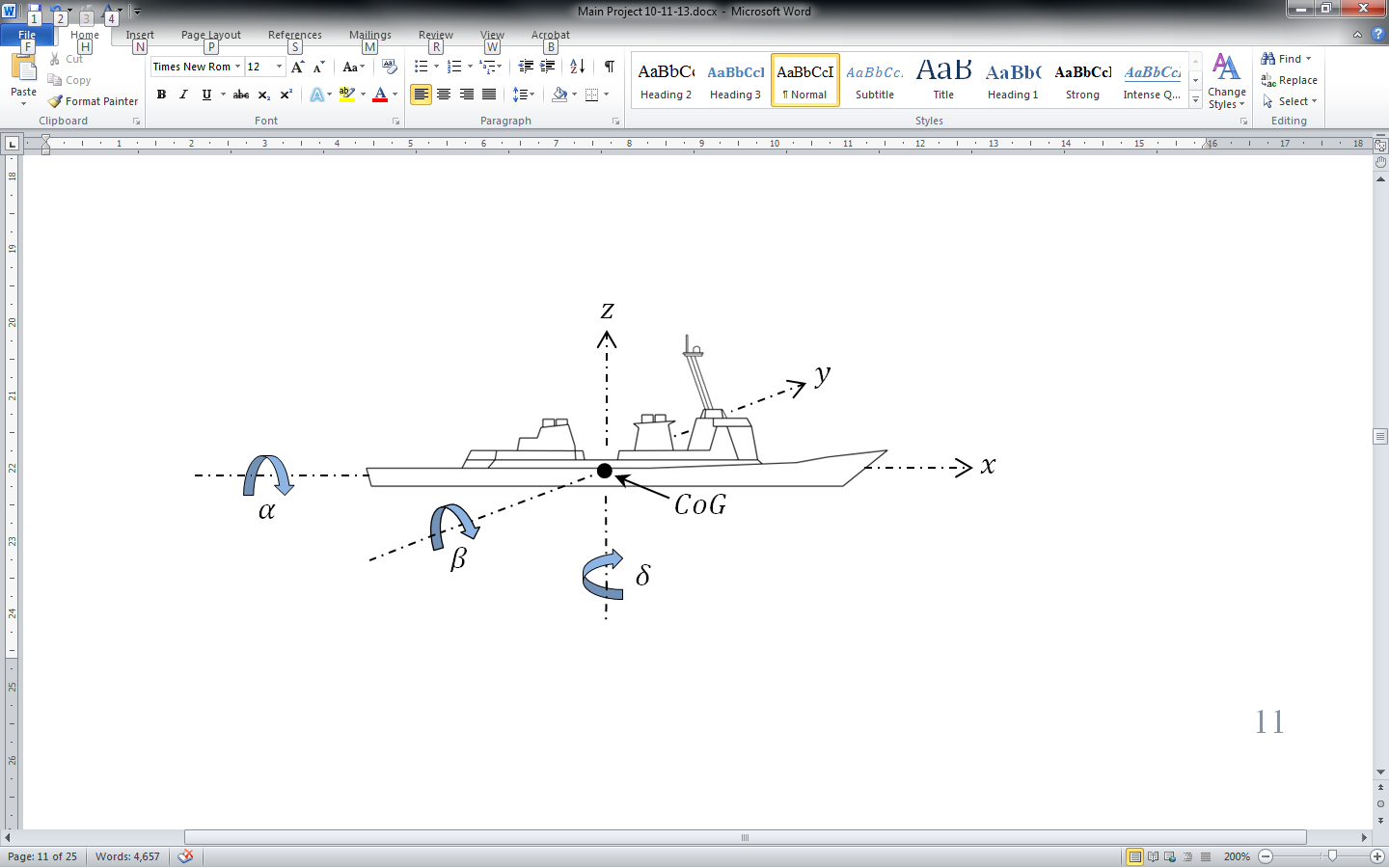


Figure 5. Diagram Showing 6 DoF of an ocean going vessel

|  |  |  |
| --- | --- | --- |
| **Name** | **Symbol** | **Description** |
| Surge |  | Thrust in the direction of |
| Sway |  | Lateral motion in the direction |
| Heave |  | Vertical motion in the direction |
| Roll |  | Rotation about the axis either |
| Pitch |  | Rotation about the axis either |
| Yaw |  | Rotation about the axis either |
| Centre of Gravity |  | One point in the ship where mass is equal in opposite directions for all 3 axes. |

Figure 6. Table of values for 6 DoF diagram

## 2.5 The Operational Envelope

Modern ocean going vessel are designed to handle a variety of different sea conditions and the larger vessels in particular would be expected to be able to handle all but the very extremist of weather conditions. While the poise of a ship is in flux there are a number of forces acting and reacting on the vessel, determining the ships responses to it local environment. Some the elements involved in this interaction include:

* Buoyancy – According to Archimedes’ Principle, a vessel immersed partly or completely in a fluid receives lifting force equal to the weight of the fluid displaced by the vessel. Buoyancy acts vertically upwards (see figure 7).
* Gravity – The earth’s gravitational force acts in a vertically downward direction and when the vessel is completely stable cancels out the buoyancy force.
* Wave forces – The force of large waves acting on the ship can have dramatic effect to the motion of a vessel. Along with amplitude and frequency, the direction of the waves relative to the ship can have very different effects on the motion characteristics of the ship.
* Ship Characteristics – The size, centre of gravity, centre of buoyancy, metacentric height, thrust capabilities of the ship and any motion damping fins all have effect on the motion of the ship as it proceeds with its objectives.

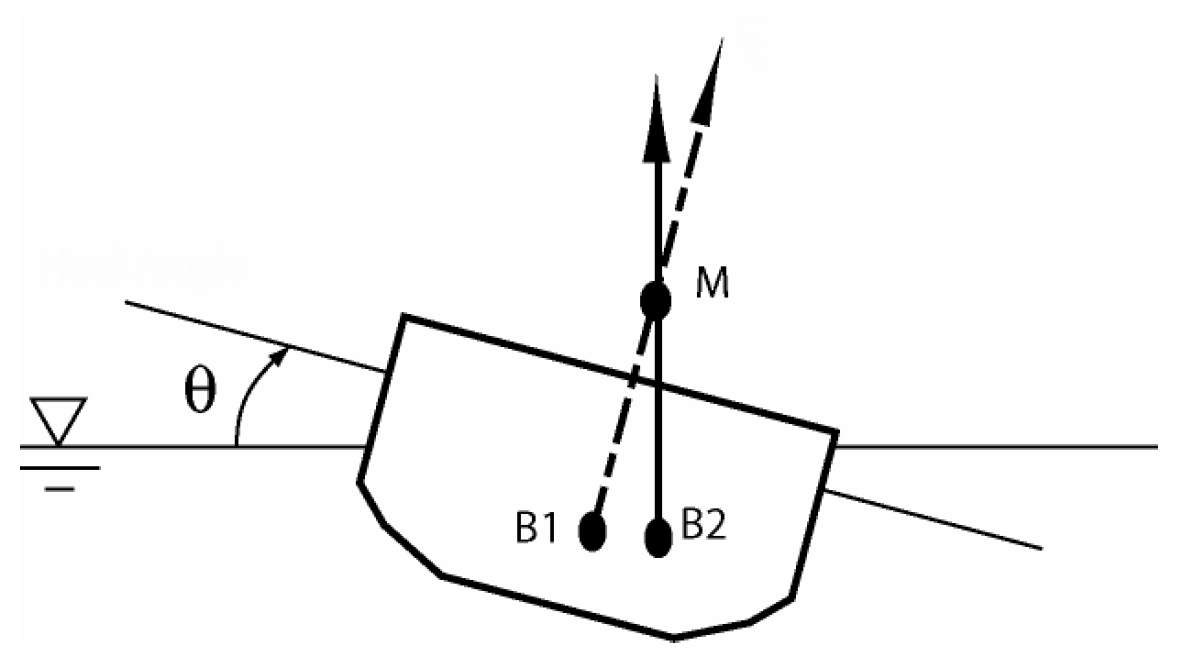


Figure 7. Section of hull diagram showing changes to buoyancy

(Techet A. H., 2004)

Where for a given roll angle of θ;

B1 = Centre of Buoyancy when the vessel was stable and horizontal.

B2 = Centre of Buoyancy after roll to angle θ

M = Metacentre

Techet explains that a ship’s metacentric height (GM) is the distance between the vessel centre of gravity and the vessels metacentre. The larger the value for GM the ‘stiffer’ the ship will be in response to the forces promoting roll motions on the vessel. The opposite is also the case if the GM is relatively small with less force needed to impose a roll on the vessel.

## 2.6 Summary

The poise of a ship at sea is in a state of constant change. Each type of vessel will have its own set of responses to the external forces acting on the vessel as it continues with its objective. In particular the action – reaction relationship between the force of the waves the vessel encounters and the vessel’s buoyancy and righting force, plays a large part in the way a ship responds. Also the direction of the waves relative to the heading of the ship will have different effects i.e. a ship heading directly into waves will have greater pitch angles and smaller roll angles when compared to a ship which is travelling parallel to the wave and being struck side on. If an aircraft is attempting to land on the deck of a ship these factors would need to be considered.

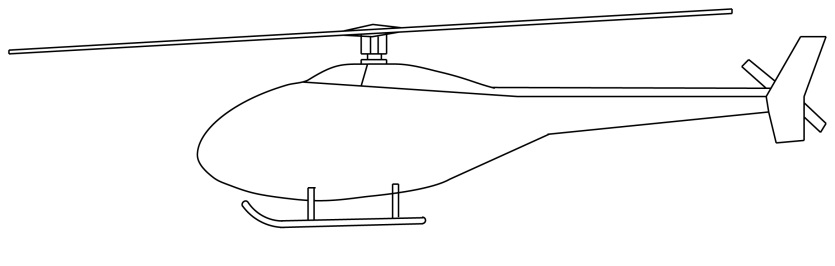
# 3 Operational Capabilities of the UAV

## 3.1 Overview

Helicopters and VTOL UAVs by extension are by nature unstable, they operate with six degrees of freedom in an environment of nonlinear dynamic variables and coupled with this a helicopter’s central body of mass is not capable of sustained flight. During flight a helicopter has principally two states, motion and hover of which hover is the most challenging. Without momentum and with the nonlinear forces acting on the aircraft, holding state involves constant change. This is a balancing act performed by a pilot or a navigational control unit.

According to (Federal Aviation Administration, 2012, pp. 2-2) once a helicopter leaves the ground it is subject to “four aerodynamic forces; thrust, drag, lift and weight” (see figure 8).

* Thrust is the self-generated force that propels the aircraft in the desired direction normally in a forward motion “parallel to the longitudinal axis”.
* Drag is resistance to thrust, it can be generated in a number of ways including; poor aerodynamics of a payload or sensory bank, wind direction and climatic or operational air turbulence. “Drag opposes thrust and acts rearward parallel to the relative wind”.
* Weight is the total mass of the operational aircraft and the forces it exerts on the aircraft as a result of gravity. “It opposes lift and acts vertically downward through the aircraft’s center of gravity (CoG)”.
* Lift is generated as the helicopter’s rotors pass through the envelope of air above the cockpit. This rotation generates negative pressure above the rotors and positive pressure below the rotors so leading to the upward vertical motion. Lift is used to counteract weight during flight and acts vertically up from the CoG.



Lift

Drag

Thrust

Weight

Figure 8. Forces acting on a helicopter in flight

## 3.2 Wind

Of all the nonlinear dynamic forces affecting a helicopter in flight wind is the force that needs the most consideration; weight is probably the largest force but is relatively constant. The flight path of the helicopter or UAV relative to the climatic winds can have a bearing on the performance of the vehicle in question. As (Biven & Guercio, 1987, p. 8) explains a helicopter has a tendency to align itself with the direction of the relative wind, this needs to be counteracted with the tail rotor and is referred to as “Yaw Weathercock Stability”. Also the wind turbulence generated by the ships airwake, as the aircraft approaches the ship can have an adverse effect on the air vortices the rotors rely on for stable flight.

These winds can have a greater effect on smaller UAVs. Referencing Newton’s Second Law of Motion, F = ma, the force of the wind relative to the UAV’s mass and acceleration is greater than that acting on a manned helicopter.

## 3.3 Performance

There are a wide variety of VTOL UAVs in civilian and military use around the world each with its own unique characteristics. Vehicle size, shape, payload, engine power, fuel consumption and design all play their part in determining the performance of the helicopter. Coupled with this is the unique environment the aircraft is deployed in each time it carries out an operation. Rotomotion’s SR200 Helicopter UAV (see figures 9 & 10) is a mid-range UAV achieving up to 5 hours of flight and with a maximum payload of almost 23kgs, details listed below. This UAV would meet a lot of the needs for a maritime surveillance operation.



Figure 9. Rotomotion VTOL UAV

(Rotomotion, 2011)

|  |  |
| --- | --- |
| **UAV Specification Item** | **Specification** |
| Length | 2790 mm |
| Width | 760 mm |
| Height | 860 mm |
| Main Rotor (M/R) Diameter | 3000 mm |
| Tail Rotor (M/R) Diameter | 700 mm |
| Dry Weight | 25 kg |
| Fuel Capacity | 2 litres (18 litres available) |
| Engine | 150cc, 2-stroke engine |
| Generator | 800 W, 24 V alternator |
| Climb Rate | 2 ms-1 |
| Maximum Speed | Up to 60 km/h |
| Endurance | Up to 5 hours |
| Maximum Payload | 22.7 kg |
| Operating Ceiling | 1,500 m |

Figure 10. Rotomotion’s SR200 Helicopter specification

(Rotomotion, 2009)

A larger UAV, predominantly in military use, is the Schiebel Camcopter S-100 (see figures 11 & 12). This UAV has been deployed by the French, Italian, Russian, and United Arab Emirate militaries. A partnership deal between Schiebel and Boeing helped expose the S-100 to the American market, where it was also presented to “U.S. Army Special Operations Command (USASOC) at Ft. Bragg” (Schiebel, 2009)



Figure 11. Schiebel Camcopter S-100 VTOL UAV

(Schiebel, 2013)

|  |  |
| --- | --- |
| **UAV Specification Item** | **Specification** |
| Length | 3110 mm |
| Height | 1120 mm |
| Main Rotor (M/R) Diameter | 3400 mm |
| Dry Weight | 110 kg |
| Maximum Payload | 50 kg |
| Fuel Capacity | 57 litres |
| Engine | 41 kW (55 HP) rotary engine |
| Maximum Speed | Up to 240 km/h |
| Endurance | Up to 6 hours |
| Operating Ceiling | 5400 m |
| Wind (takeoff and landing) | Up to 46 km/h |
| Data Link Range | Up to 180km |

Figure 12. Schiebel Camcopter S-100 specification

(Schiebel, 2013)

## 3.4 Navigation

All aircraft have an Inertial Navigation Systems (INS) and VTOL UAVs are no different. The deviation though may be in the destination of that information. If the UAV is fully autonomous then the information is sent to the UAVs flight control system where the system determines the next control response to be undertaken. However if the UAV is under the control of a pilot then the INS information will be displayed on the pilots interface, which may be many kilometres away, for example Schiebel’s Camcopter S-100 can communicate with its pilot up to 180 km away (Schiebel, 2013).

The information processed by an aircraft’s INS can include; altitude, heading, pitch, roll, velocity and GPS. These values in conjunction with the mission objectives will determine the UAVs next course of action. Some INSs also employ secondary or redundant systems as fail safes. For example the KN-4072A unit, available from Kearfoot, also records barometric altitude which can be included in the “vertical loop in the absence of valid GPS data” (Kearfoot, 2011), which could be extremely important if a hostile force is interfering with a GPS signal.

Other navigational tools that are deployed on some UAVs include; radar, way points (the systems generate their own longitudinal/latitudinal points for the return leg of the operation) Ethernet, cameras, and high altitude imagery.

Combined, this array of sensors, receivers and equipment is known as the sensory bank and is often fitted under the aircraft away from the engine and rotors, reducing vibration as much as possible.

## 3.5 Response Times

The response time of helicopters and UAVs to flight control changes is difficult to provide. There are a number of factors that will affect the performance characteristics of the aircraft. These include; mass and payload, CoG and its location, wind and its direction, thrust at the time of control change and momentum relative to the desired control change. For example the helicopter or UAV may respond almost instantaneously, if the desired control change corresponds to the relative wind direction, however the same control applied in the opposite direction may have little or no impact if the force applied is not substantial enough to overcome the force of the relative wind

In his doctoral thesis on flight control for VTOL UAVs Yuan observed the dynamics of u, v and w the longitudinal, lateral and vertical velocities (see figure 13). When a closed-loop controller was implemented, the expected change rates for u, v and w were recorded. These are shown below.

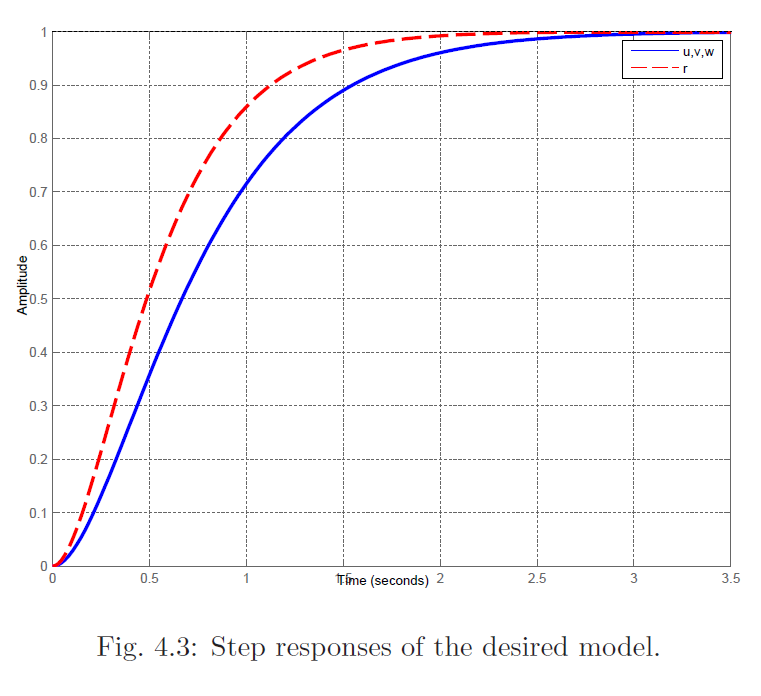


Figure 13. Yuan's finding for response times versus changes in vector u,v and w

(Yuan, 2013, p. 50)

As can be seen from the plot the rate of changes for the vector forces u, v and w are nonlinear. This is expected as the internal and external forces that affect these rates of change are also nonlinear e.g. acceleration, momentum, climatic and turbulence induced winds etc.

As the aircraft approaches the point at which a required alteration is nearing completion it is common for a flight controller to apply a dampening effect to the flight controls. In Yuan’s implementation he also investigates the need to “increase the damping ratio of the closed-loop system as the system output approaches the target reference to reduce the overshoot” (Yuan, 2013, p. 15)

Another factor in the control of UAVs (piloted) is the human response times which (Biven & Guercio, 1987, p. 45) states is “0.3 sec, which is representative of the human neuromuscular time delay”. This delay could be further extended if the pilot is under pressure to perform multiple tasks at once which is normally the case when operating the controls of a helicopter UAV.

## 3.6 Camera

Like the sensory array available to the UAV there is also a wide variety of cameras that are available to the aircraft. One of the only disqualifying factors would be the net mass of the camera and the UAVs payload capabilities. A high end camera system is supplied by Hood Tech Vision (see figures 14 & 15). The 11EOIR1 is a “mid-wave infrared and electro-optical imaging system … designed for small unmanned aerial vehicles” (Hood Tech Vision, 2013) selected details are below.



Figure 14. Hood Tech Vision - 11EOIR1

(Hood Tech Vision, 2013)

|  |  |
| --- | --- |
| **Specification Item** | **Specification** |
| Field of View | MWIR: 2°-25° EO: 1.7°-57° |
| Pixels | MWIR: 640 x 480 EO: 720 x 480 |
| Wavelength | MWIR: 3-5 µm EO: 400-900 nm |
| Zoom (optical) | MWIR: 12.5X EO: 36X |
| Communications | Nominal: Serial communication, 57,600 bps |
| Video | Composite NTSC |
| Weight | 5,700 g with enclosure |

Figure 15. Hood Tech Vision - 11EOIR1 specifications

(Hood Tech Vision, 2013)

## 3.7 The UAV Controller

The proposed control system for the UAV simulation will be model a closed loop system where feedback is reviewed and alterations to commands are made if required. It is also proposed to develop this closed loop system as a Model Reference Adaptive Controller (MRAC). The MRAC will base any required changes on the controller’s previous commands and the controller’s reference model or baseline. These systems are suited to the task in hand due to their “improved performance and increased robustness to uncertainties by virtue of their ability to adjust control parameters” (Thompson Dydek, 2010). The controller can make allowances for any momentary unknown parameter which can be the case when dealing with the nonlinear variables in the UAV operational envelope.

## 3.8 Summary

Helicopters and VTOL UAVs are naturally unstable aircraft. Their operation is affected by a list of nonlinear variables as they operate in six degrees of freedom. That been said they are ideally suited for certain operations that cannot be performed by winged aircraft. One of these is landing in a confined area. This project proposes to examine the interaction between some of these nonlinear variables and the projected movements of a UAV in 3D space as it approaches touchdown on the deck of a ship. Although the movement and responses of the aircraft are hard to establish definitively, an operational response envelope of 0.5ms-1 in all 3 axes of 3 dimensional space is expected. This expectation is a minimum would be relative to any acceleration or momentum already acting on the aircraft.

# 4 Final Approach

## 4.1 The Four Phases

As discussed by (Shin, You, & Shim, 2013) and (Fang, et al., 2003) the landing of a helicopter on the deck of a ship can be described in four distinct phases. The helicopter is in “Tail-Chasing Mode” when there is greater than 350m between the helicopter and the ship. When the helicopter enters the range between 350m and 50m, the aircraft is in the “Approach Mode”, at this stage it is also recommended to lower altitude to 30m. To proceed, the helicopter then enters the “Descending Mode” reducing altitude, matching velocity with the ship and finally touching down in the “Landing Mode”.

## 4.2 Landing Approaches

With reference to the landing of a helicopter, Fang, et al. also goes on to explain from page 1-18 that to their “knowledge there are six different procedures which are being applied worldwide”. Three of the main procedures are described as;

“Fore/aft or forward facing procedure”

“Relative-wind or into wind procedure”

“Cross-deck procedure”.

When deciding on a desired and/or required landing approach the polit of a helicopter must consider many factors, one of the main considerations when evaluating a flight path is the crosswind and the direction of the crosswind relative to the ship (Shin, You, & Shim, 2013, p. 771).

## 4.3 Touchdown

When the UAV approches touchdown it can be in a percarious state. The aircraft can be prone to slip and rotational momentium as the movements of the ship are now imposed on the aircraft. If a helicopter in the process of landing, touches down on one side while still amost airbourne it is possible the aircraft may ‘slip’ laterally. This issue could then be compounded if the landing gear strikes an obstacle (see figure 16), adding momentum about the landing gear pivot point.

“When the helicopter comes in contact with the ground and a new pivot point is established the moment of inertia about the roll axis increases nearly five-fold due to this displaced pivot point” (Lobik, 2003). The rotational momentum of the ship can compound this issue, but it can also have adverse affects on the aircraft alone. If the centre of gravity and the tipping point of the aircraft approach each other as the ship rolls the aircraft will be prone to dynamic rollover. Any state of instability will continue until the aircraft has lost suffient lift to insure gravity will hold it in place on the deck of the shipor the aircraft mechanically secured or stowed away.

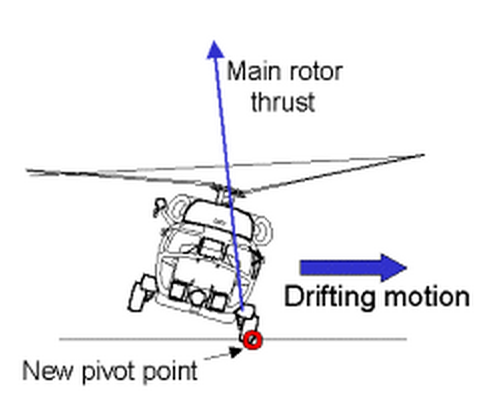


Figure 16. Lobik's diagram showing lateral ‘slip’ as a helicopter is landing or taking off

(Lobik, 2003)

## 4.4 Summary

The final stages of a helicopter or UAV flight, as it attempts to touch down on the deck of a ship involve complex operations. The skill and coordination of the pilot or UAV operator must be up to the task at hand. The proposed system will focus on this stage of VTOL aircraft flight, and in particular UAV landings.

# 5. Commercial and Academic Landing Assist Systems

There are a number of commercial systems on the market that offer similar functionality as that proposed for this project. There are also many academic papers that have investigated the issues discussed here. The commercial products discussed below use positioning systems either local or global. The academic papers outlined in section 5.2 employ various methods to aid in the landing.

## 5.1 Commercial Systems

Astrium’s ‘DeckFinder’ system uses deck mounted “Pseudo-Satellites” to provide a local positioning system which operates without the need for Global Positioning Systems. The system delivers positional accuracy of 20cm at a rate of up to 15Hz to a range of up to 1.1km (Astrium, 2013).

NovAtel have developed an “Automated Flight Control System” which uses their navigational systems to autonomously land a drone helicopter. These navigational systems use GPS and INS data from the ship and the UAV. This system was tested successfully in conjunction with Boeing’s H-6U VTOL UAV on 4th July 2012 when the UAV was autonomously landed on the deck of a moving ship (NovAtel, 2012).

## 5.2 Academic Systems

At the 2013 International Conference on Unmanned Aircraft Systems, (Shin, You, & Shim, 2013) presented a paper which evaluated an algorithm for the landing of a UAV autonomously. The paper investigates the use of a Time-Delay Control and a MatLab simulation to validate the results. The algorithm makes allowances for crosswind, and the different landing phases a VTOL UAV must negotiate to touch down successfully. The performance of the time delay controller, which is modelled on a model reference adaptive controller, is compare to the results from a conventional proportional derivative controller. Shin’s finding showed that “the landing system generated optimal landing procedures based on the Navy procedures”.

Another paper presented at the Unmanned Aircraft Systems conference by (Sanchez-Lopez, Saripalli, Campoy, Pestana, & Fu, 2013) discusses the use of a colour camera fixed to a UAV to aid in the its landing on the deck of a ship. The UAV in this case also uses sonar to determine the difference in altitude between the ship and the aircraft. Their simulation for the ship was based on an Oliver Hazard Perry Class FFG Frigate operating in sea state six. The system uses image recognition and extraction to acquire the optimal landing coordinates and the model’s performance was satisfactory in “the first steps required to achieve a solution to the challenge of autonomously landing on a ship”.

In the IEEE journal, Transactions on Robotics (Herissé, Hamel, Mahony, & Russotto, 2012) reported on the findings of their paper on VTOL UVA landings on a moving target using optical flow. Their control algorithms used the optical flow measurements obtained from a camera fixed to the centre of mass of the UAV. They explained that the results “show that the proposed scheme is effective” although it was observed that improvements in the response times for the camera would help improve the effectiveness of the closed loop controller.

In their paper Esmailifar and Saghafi propose a UAV landing system which employs a multilevel controller to track the target and land on a platform which moves with six DoF. This is achieved in two distinct phases, the “Supervisor level” which deals with intercepting the ship and the “Tracking level” which tracks the attitude differences between the aircraft and the ship. In evaluating the landing system’s performance the motion of the ship is modelled on a sinusoidal wave. They concluded that the performance of the simulation was satisfactory as “the helicopter is able to track the ship states in both landing phases” (Esmailifar & Saghafi, 2009).

## 5.3 Summary

There have been many academic papers on proposed landing systems for helicopters and VTOL UAVs but the same does not seem to be the case for their commercial equivalents. On 4th July 2012 NovAtel and Boeing claimed to have been the first to successfully land a UAV on the deck of a moving ship (NovAtel, 2012). This is surprising considering the level of research into this sector of aerial systems, and could indicate the complexity involved in implementing the research models in real life applications.

# 6. Project Design

## 6.1 Scope

This project will determine course correction of a UAV as it attempts to land, based on simulated images of the deck of a ship. To aid in this task infra-red beacons in a predetermined pattern will be analysed. It is expected that these course corrections will be generated in good time allowing for sufficient correction to the UAVs mathematical model. It is proposed to achieve operational responses for the UAV model of 0.5ms-1 in all 3 axes.

The ship will also be represented by a mathematical model and its movements displaying sinusoidal motion tendencies as it reacts to waves in Sea State 6. A dataset has been obtained for a list of ships operating in sea state 6 which is at the upper end of the operational envelope of the UAV. These datasets contain maximum angles of displacement and accelerations which will need to be transposed to 3D geometry relative to time.

It is planned to present the results as a simulated display in Unity where the UVA model will attempt to land by reducing the displacement between itself and the ship and ultimately resulting in a successful touchdown.

## 6.2 Estimate of Activities

1. Produce a working mathematical model of the ship relative to the dataset already obtained (McDonald, 1993).

2. Produce a working mathematical model for a UAV based on the performance characteristics stated above and relative to the ships motion.

3. Produce statistical dataset of the require control alteration for the UAV as it attempts to land on the ship, resulting in the touchdown.

4. Implement a comprehensive test plan to validate the system results.

4. Present the results in a visual display to confirm the finding of the project.

## 6.3 Risk Analysis

|  |  |
| --- | --- |
| Risk | Alternative Action |
| The dataset is for the motion of the ship was recorded in Sea State 6. This is at the upper limits for the operation of mid-sized UAVs and the results may be unsatisfactory. | If landing the UAV in this sea state proves problematic the expected motion of the ship will be altered proportionally to represent a lower sea state. |
| It is proposed to produce the mathematical models for the ship and helicopter with Unity and Visual Studio. Knowledge of Unity is limited and a steep learning curve is anticipated. | There are ample online resources to mitigate this risk. |
| The format for the presentation of the results has not yet been decided. | It is hoped to develop this project with two PCs that communicate via interconnected sockets. However as the scope for this project is significant and it is proposed to develop the implementation on one machine first. |
| The scope of the project as a whole is a risk as there are many nonlinear variables that need to be computed to achieve the ultimate goal. | Unknown and dynamic variables such as deck draft and wind turbulence could be assumed as near constants to help compute the models. |
| It may not be possible to randomly generate all the variable forces acting on the models as the application tries to process the datasets. | A more sterile version of the forces could be used; these though should still include the major variables. |
| At the time of writing this report it is not known if the hardware that will be used to deploy this project is powerful enough for the task in the time required. | If this turns out to be the case, processing demand could be lowered if certain minor or nominal variable forces are set as constants in the interest of evaluating the bigger issues. |

## 

## 6.4 Use Case Diagrams

### 6.4.1 Adaptive Control System Use Case Diagram



Figure 17 Use Case Diagram - Adaptive Control System

|  |  |
| --- | --- |
| **Use Case** | **Receive Latest INS Data** |
| Actor | INS Simulator |
| Description | The coordinates and rotations of the UAV relative to the spatial reference plane are received from the INS Simulator |
| Flow of Events | The INS posts new coordinate and rotation data on a looped basis. The controller records the new data received so it is available for evaluation. |
| **Use Case** | **Receive Image Processor Data** |
| Actor | Image Processor |
| Description | The image processor provides data on the location of the infra-red beacons relative to the image plane. |
| Flow of Events | The Image Processor post new coordinate data on a looped basis. The controller records the new data received so it is available for evaluation. |
| **Use Case** | **Compute Course Correction** |
| Actor | UAV Flight Controller |
| Description | Course correct data is posted from the adaptive control unit to the UAV flight controller. |
| Flow of Events | The adaptive control unit reads the INS coordinate data. The adaptive control unit also reads the image processor data and references the last set of outputs. Based to this data and the expected data patterns a decision on the next course correction is made. These corrections are then posted to the flight control unit. |

Figure 18 Use Cases - Adaptive Control System

### 6.4.2 Image Processor Use Case Diagram



Figure 19 Use Case Diagram - Image Processor

|  |  |
| --- | --- |
| **Use Case** | **Process Latest Image** |
| Actor | Camera Simulator |
| Description | The camera simulator provides bitmap images to the image processor as they are available on a looped basis. The image processor reads the images and records the coordinates displayed in the image. |
| Flow of Events | The Image Processor receives an image form the camera simulator. The image is the scanned using the search algorithm. When the coordinates are found they are recorded locally. |
| **Use Case** | **Provide Image Data** |
| Actor | Adaptive controller |
| Description | The image processor posts the image coordinates to the adaptive control system for evaluation |
| Flow of Events | As the images are processed the coordinates are posted to the adaptive control system. The image processor then repeats the process on a looped basis |

Figure 20 Use Cases - Image Processor

### 6.4.3 INS Simulator Use Case Diagram



Figure 21 Use Case Diagram - INS Simulator

|  |  |
| --- | --- |
| **Use Case** | **Receive Latest Data** |
| Actor | Adaptive Control System |
| Description | The INS simulator receives the last flight control requests from the adaptive controller and records the data locally. |
| Flow of Events | The INS system receives data from the control unit and persists it locally. This continues on a looped basis as the inputs are received from the control unit. |
| **Use Case** | **Provide Latest INS Data** |
| Actors | Adaptive controller  Camera Simulator |
| Description | The INS simulator posts the post the UAV coordinates to the adaptive control system and the camera simulator for evaluation. |
| Flow of Events | The INS simulator reads the data provided by the adaptive controller and adjusts the recorded coordinates for the UAV accordingly. These new coordinates are the returned to the adaptive controller and the camera simulator for evaluation. This process repeats on a looped basis. |

Figure 22 Use Cases - INS Simulator

## 6.5 High Level State Diagram



Figure 23 High Level State Diagram

## 6.6 System Architecture

It is proposed to use a three tier architectural design, consisting of a GUI presentation layer, a control layer and two business layers (see figure 24). The implementation of two business layers will provide a layer of separation between the business logic layers for the two models.

It is proposed to build the presentation layer as a collection of Unity (Unity, 2013) ‘scripts’. These C Sharp classes will be developed in the Visual Studio editor which is accessible directly from Unity. In order to harness the functionality of the Unity game engine, the presentation layer classes need to implement the Unity interface ‘MonoDevelop’. The implication of using Unity with this configuration is the difficulty testing this element of the project. The testing of ‘gameObjects’ and their relative motion in a virtual 3D space would involve substantial effort. As an alternative it is proposed to remove any business logic from presentation layer and implement it in the lower layers of the architecture. The lower levels of the architecture will not implement MonoDevelop and so NUnit (NUnit, 2013) or a similar testing framework will be available to the system.



Figure 24 System Architecture

## 6.7 Component Diagram

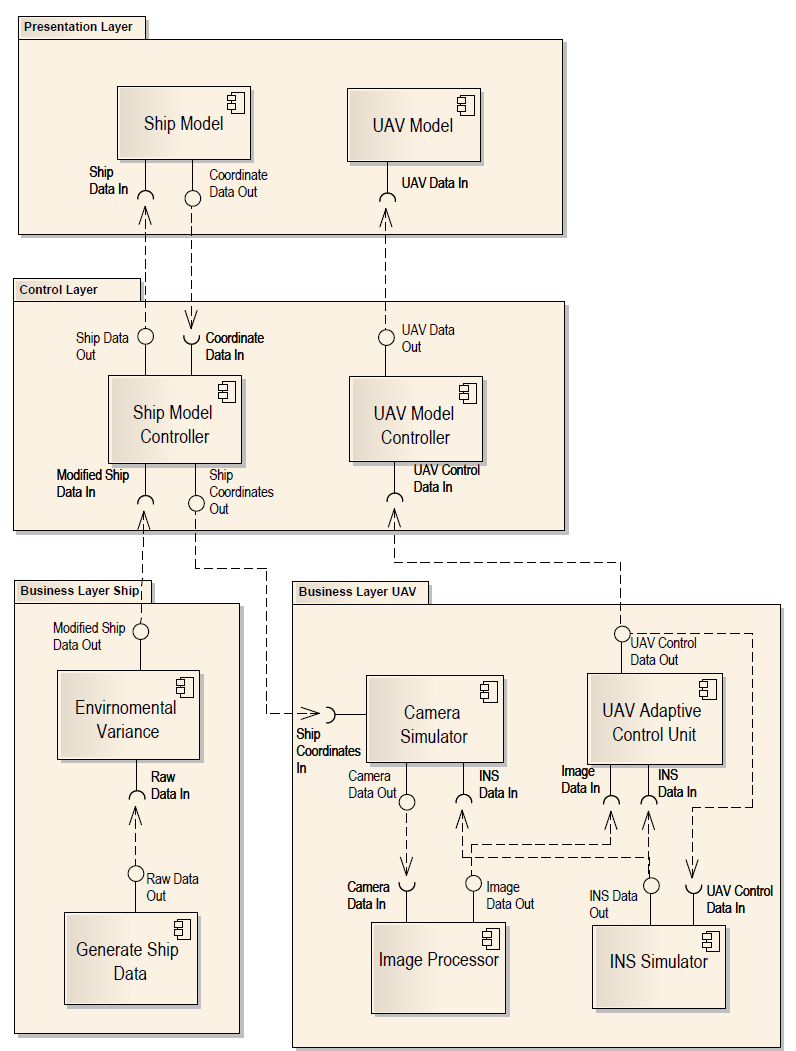


Figure 25 Component Diagram

## 6.8 Class Diagrams

Placeholder

## 6.9 Methodology

The research methodology for this project involved investigating previous works on related subjects, in particular white papers which were sourced on the internet. To this end, access to IEEE proved extremely productive. For reference purposes commercial product were reviewed and documented. Other information was sourced form reputable online sources including;

* MIT
* National Aeronautics and Space Administration
* North Atlantic Treaty Organisation
* United States Department of Transportation
* United States Navy
* UK, Met Office
* American Society of Naval Engineers

It is proposed to implement this project with a waterfall methodology. The system components will be developed and documented on a weekly basis and presented to the primary supervisor for evaluation. Alongside each project component, unit tests will be developed and then grouped into test suites where appropriate.

## 6.10 Test Plan

It is proposed to use NUnit as the test environment for this project. As discussed earlier in section 6.4 System Architecture, there are difficulties running unit tests against Unity game objects and how they interact with the scene that contains them. It is proposed to remove all business logic, where possible, from this layer and perform the required testing in the lower layers of the system. At the presentation layer manual test will be performed at a baseline of sea state 0 and also at the desired sea state 6.

To aid in the achievement of this goal it is proposed to implement a test plan which will include a suit of test for both the ship and UAV motions.

Ship Test Suite;

* Test the pitch of the ship is within parameters.
* Test the roll of the ship is within parameters.
* Test the yaw of the ship is within parameters.
* Test the surge (propulsion) of the ship is within parameters.
* Test the heave of the ship is within parameters.
* Test the sway of the ship is within parameters.

In the case of all the tests above, the movement of the 3D points allocated to the location of the infrared beacons will be tested.

UAV Test Suite;

* Response times for the controller outputs will be measured as the system processes the perceived pattern of the infrared beacons on the deck of the ship. There will be tests for each of the motions listed above.

Integration Test Suite;

* Tests will be deployed to measure the success of the interaction between the ship and UAV models.

Below is the test plan for the proposed project (see figure 29).

|  |  |  |
| --- | --- | --- |
| **Test Number** | **Description** | **Expected Result** |
|  | **Non Test Framework Environment** |  |
| **1** | Manual test of model at sea state 0 | Satisfactory touchdown |
| **2** | Manual test of model at sea state 6 | Satisfactory touchdown |
|  |  |  |
|  | **Test Framework Environment** |  |
| **3** | **Test data generated for ship motion** |  |
| 3.1 | Test sample data for heave | Below sea state 6 maximum values |
| 3.2 | Test sample data for roll | Below maximums for ship concerned |
| 3.3 | Test sample data for pitch | Below maximums for ship concerned |
|  |  |  |
| **4** | **Test ship data following adjustments for environmental variance** |  |
| 4.1 | Test sample data for heave | Within sea state 6 limits |
| 4.2 | Test sample data for roll | Within limits for ship |
| 4.3 | Test sample data for pitch | Within limits for ship |
|  |  |  |
| **5** | **Coordinates for IR beacons** |  |
| 5.1 | Test coordinates for heave | Within sea state 6 limits |
| 5.2 | Test coordinates for roll | Within limits for ship |
| 5.3 | Test coordinates for pitch | Within limits for ship |
|  |  |  |
| **6** | **Test output from camera simulator** |  |
| 6.1 | Test 3D – 2D conversion for coordinates | Pass the test previously verified data for comparison |
| 6.2 | Test simulated frame rate of the camera | 24 – 30 fps |
|  |  |  |
| **7** | **Test output from Image Processor** |  |
| 7.1 | Test 2D – 3D reconversion of coordinates | Pass the test previously verified data for comparison |
| 7.2 | Test new 3D coordinates relative to camera poise | Pass the test previously verified data for comparison |
| 7.3 | Test processing speed | 15 fps (commercial products) |
|  |  |  |
| **8** | **Test the INS simulator** |  |
| 8.1 | Test responses to changes to relative wind direction | Pass the test previously verified data for comparison |
| 8.2 | Test simulated GPS coordinates | Pass the test previously verified data for comparison |
| 8.3 | Test simulated inertial responses | 0.5ms-1 |
|  |  |  |
| **9** | **Test Adaptive Controller** |  |
| 9.1 | Test processing speed | 15 fps (commercial products) |
| 9.2 | Test Output control instructions | Pass the test previously verified data for comparison |

Figure 26 Test Plan

## 6.11 Prototype

To develop a prototype a number of different platforms were investigated. First the MatLab IDE (MathWorks, 2013) was evaluated and although the platform is suitable to perform the task in hand it is not available as an open source application and only offers a 30 free trial. Following on from the evaluation of MatLab, iPython (IPython, 2013) and Pythonxy (pythonxy, 2013) were investigated but the initial results proved disappointing. It was an aspiration to provide a 3D plot from two live streams but it was felt that the visual results may not demonstrate the interaction between the UAV and the ship to an acceptable level of detail.

Finally Unity was evaluated. This game engine is open source and proved to have an acceptable level of processing power, visual display elements and coupled with the option of using Visual Studio and the main script editor it was felt that this configuration offered advantages over the other platforms.

Another application, Blender (Blender, 2013) was used to create a basic model of a ship (see figure 26). This scale model approximates the size of DDG 51 Arleigh Burke destroyer and was then imported into Unity as a program asset. The model of the helicopter was retrieved from a 3D model library and imported into Unity (TF3DM, 2013). To demonstration the helicopter moving, a basic script was attached to the model in Visual Studio.

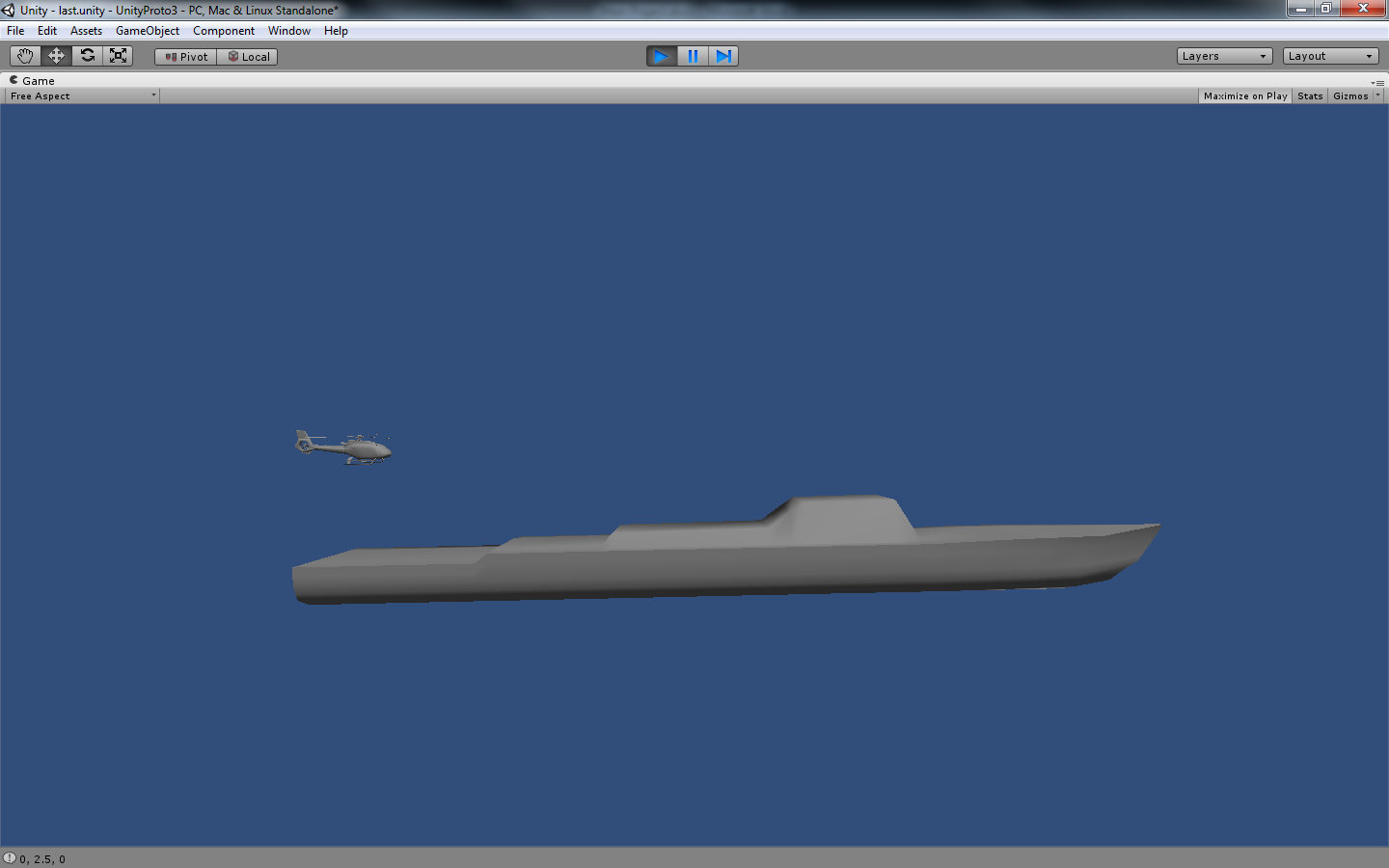


Figure 27 Image of ship prototype

When the model is ‘playing’ the application can produce coordinates for points on ship model as it virtualises movement in 3D space. It is proposed to harness these coordinates to provide a dataset for the movement of the landing pad relative to the ships poise.

Vector3 point = gameObject.transform.position;

print(point.x + ", " + point.y + ", " + point.z);

A sample of the data recorded for the motion of the model is shown below. This particular dataset only applies to the centre point of the model and so does not show responses to rotation, as the model rotates around it centre point. It is proposed to add a number of reference points to the model that can then be tracked.

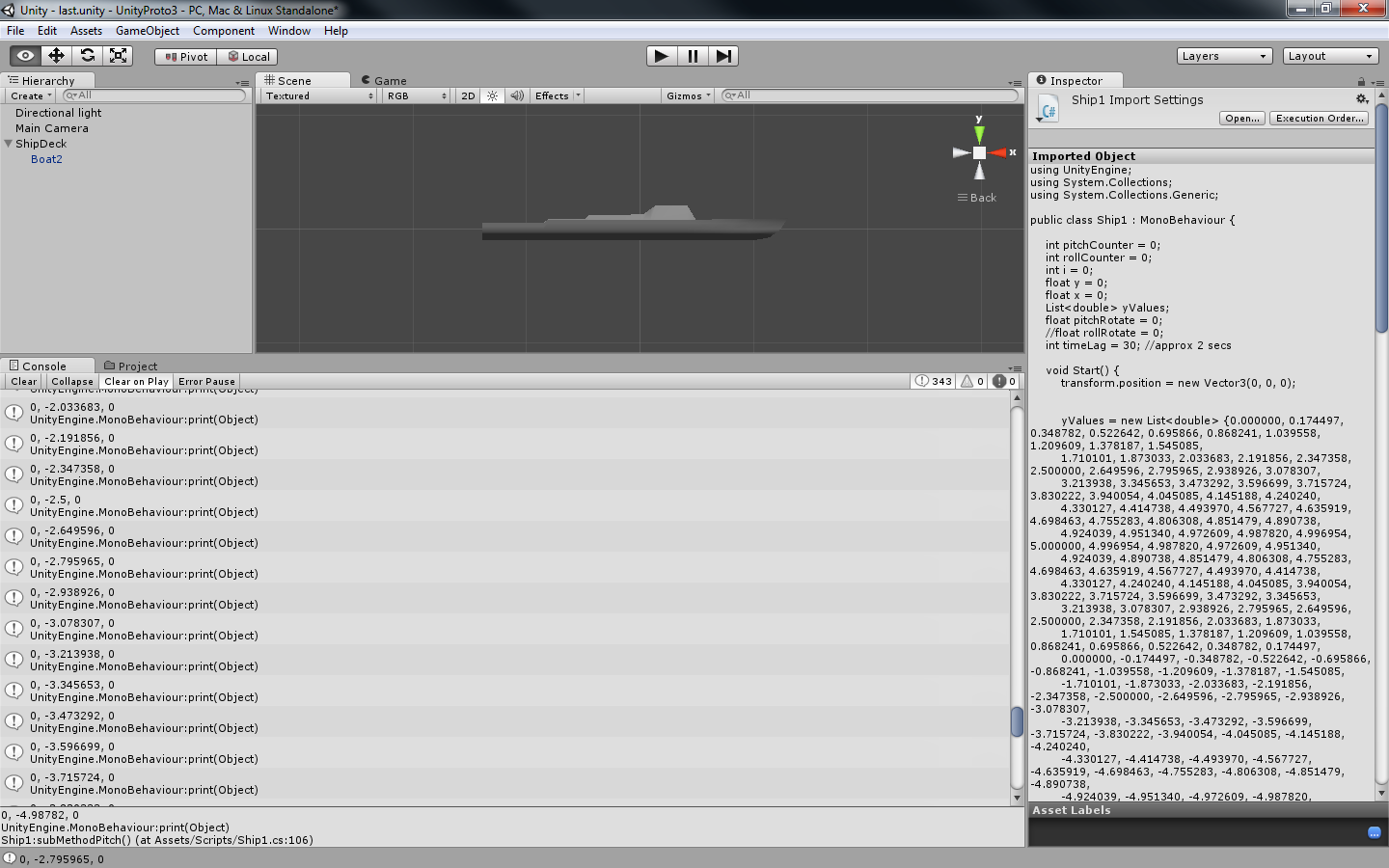


Figure 28 Code snippet of ship coordinates

A prototype was also created for the infra-red beacons on the ship as they would appear to a camera situated directly above the landing platform (see figure 28). The same script was run against this model and the poise of the landing platform was observed as the model moved in a virtual environment symbolic of sea state 6.

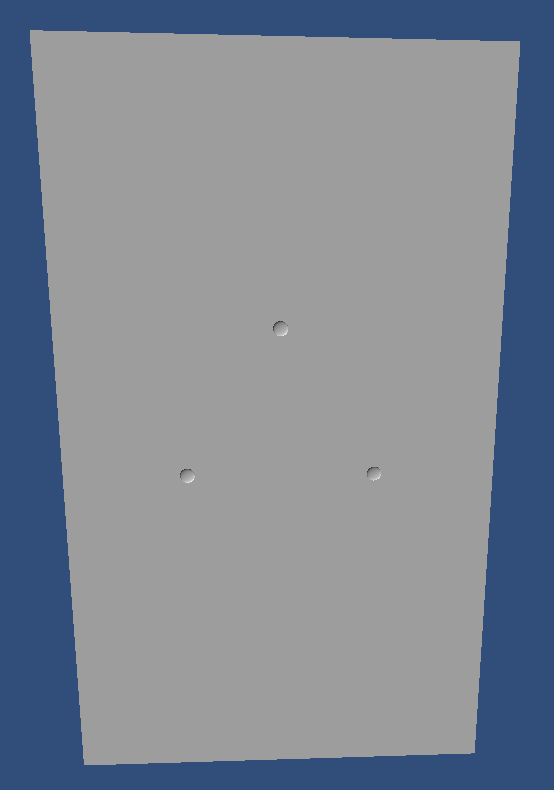
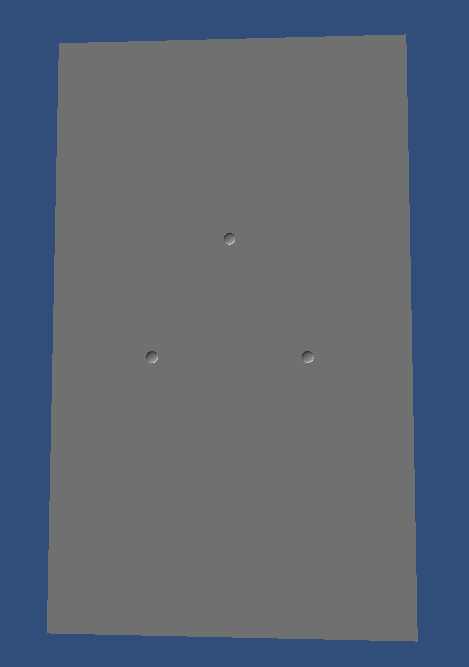
 

Figure 29 Images of landing pad prototype

# 7. Equations of Motion

## 7.1 Degrees of Freedom

If a body can move in any direction in space, it is said to have six degrees of freedom (6DoF). There are three possible translations in the x, y and z axes and also three possible rotations in a 3D space. Both aircraft and ocean going vessels have 6DoF as they operate in their respective environments. The buoyancy of the ship counteracts gravity as it sails on the ocean and the lift exerted by an aircraft allows it to stay airborne, neutralising gravity. Both vehicles exhibit motion in all three translational and rotational vectors.

## 7.2 Motion in 3D Space

A body in 3D space can be represented as a set of Cartesian coordinates such that

where:

= the x, y and z coordinate in a spatial plane, and

n = the number of points on the body.

This set of coordinates can also be represented as a set of 1 x 3 matrix or column vectors where n is the number of points on the body so that

When matrix computation is then used it is possible to translate and rotate the coordinates of the body with relative ease using variations of the transformation matrix A. However to aid in this computation a homogeneous coordinate system is used, such that;

and

Translation of coordinate (x, y, z, 1) by scalar quantities a, b, c will result in new coordinates when a translation homogeneous matrix is applied, such that;

Rotation of coordinate (x, y, z, 1) in the x axis by an angle will result in new coordinates

Rotation of coordinate (x, y, z, 1) in the y axis by an angle will result in new coordinates

Rotation of coordinate (x, y, z, 1) in the z axis by an angle will result in new coordinates

where

roll rotational angle

pitch rotational angle

yaw rotational angle

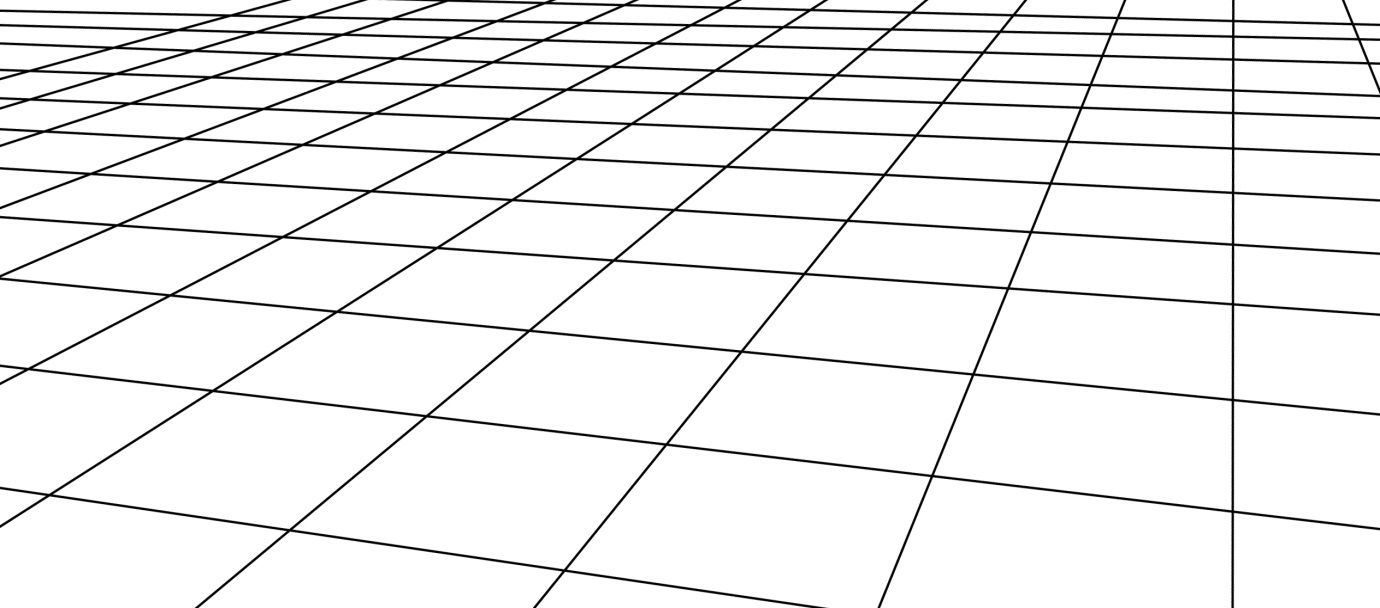
To calculate the respective angles of rotation coordinate geometry will be utilised. This will involve comparing the actual location of the coordinates with the corresponding coordinates, in the know pattern of infra-red beacons. From there the rotations can be extrapolated using the Sine or Cosine rules depending on the information that is known at the time.

## 7.3 Summary

In the computational model proposed, it is expected to evaluated the coordinates of the ship based infra-red beacons and the coordinate of the UAV using calculations which include the translations described above or variations and combinations of them.

The ship’s infra-red beacons will have coordinates in the ‘World Plane’ whereas the camera is fixed to the UAV which will have its own ‘Object Plane’ (see figure 30). As it is proposed to simulate the camera output, the ship based coordinates will need to be mapped to the to the object plane, i.e. the UAV object plane.

UAV Object Plane



World Plane (x,y,z)

Figure 30 3D Plane representation

# 8. Photogrammetry and Image Generation

## 8.1 Introduction

The essence of the project is to track infrared beacons on the deck of a ship and compute course corrections for an airborne VTOL UAV based on the location and poise of those beacons relative to the UAV. As this is a simulation, the logic behind these course corrections uses a visual representation of the beacons as they would appear in images taken with a camera fitted to the fuselage of a UAV. The camera’s images are processed and the target’s parameters are established. These parameters provide the flight control unit with a vector distance to the landing pad on the deck of the ship.

For this implementation the image data is simulated by computing the coordinates of the beacons as they would appear on an image based on the relative poise of the two crafts at any given time. If the focal point and length of the camera, the Charge Coupled Device (CCD) size and resolution, and the positions of the camera and the beacons in 3D space are known, then an image simulation can be generated. All the camera’s characteristics are known, the VTOL UAV’s INS provides the camera’s location and the coordinates of the beacons on the deck of the ship are streamed to the UAV. Using this data and parameterizes equations for lines from the individual beacons to the focal point of the camera, the image of the beacons can be simulated. Fig ########## below shows a model that represents this process. Where represents a beacon on the deck of the ship, is the focal point of the camera and is the corresponding point on the camera’s CCD. Initially the z coordinate is not shown but is included later in the chapter.



•

*Camera Model*

*CCD plane*

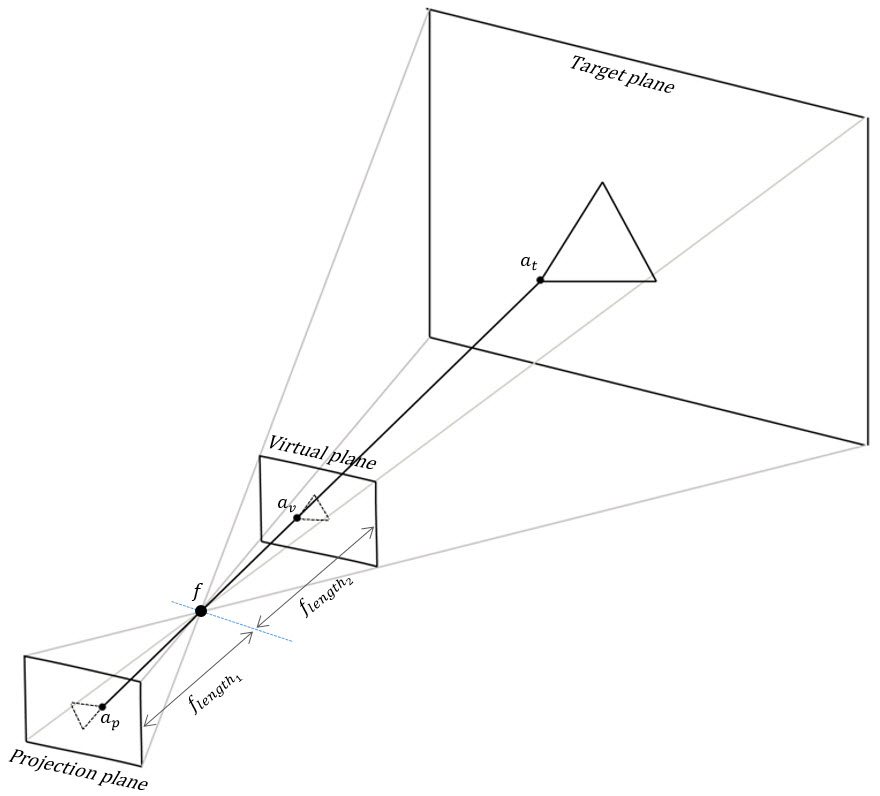
•



•

*Target plane*

To simplify the calculations when simulating the beacons locations as they would appear in the image, a virtual plane and the single focal point are considered. This virtual plane is positioned exactly the same distance from , the focal point, as the CCD plane but on the target side of the focal point as shown in Fig ############. This arrangement of the virtual plane and focal point provides a representation of the target as it would appear in an image provided by a camera, all things being equal. By moving the virtual plane in front of the focal point the image will not be inverted and by placing it at exactly the same distance from the focal point the coordinates of anything appearing on the CCD will have matching (but not inverted) coordinates on the virtual projection plane. In this example, a point on the target plane can be represented on the cameras CCD at point and on the virtual plane discussed earlier at point provide the distance between the focal point and the virtual plane i.e. equals the distance from the CCD to the focal point.



## 8.2 Parameterized Equations

Coordinates on the virtual plane can now be represented mathematically by using parameterized equations. Considering Fig ###########

•



*Virtual Projection plane*

•



•

*Target plane*

Where is the focal point of a camera and has three dimensional coordinates and is a point on the target plane, e.g. one of the infrared beacons with coordinates then the coordinates of the point on the virtual plane where the line segment insects the virtual plane can be calculated if the focal length is known and the coordinates for and are known. This point can be expressed with three equations;

Where

These values are the values for the components of the directional vector for the line segment and as shown above are equal to.

In the model described above the value for can be calculated if the camera is set perpendicular to the plane. By taking this approach the value for is known as it is less the focal length of the camera;

therefore

The value for can then be used to equate the coordinates on the virtual plane for

So, given two points, one on the target plane (infrared beacon) and the other the focal point of the camera and also knowing the details of the camera it is possible to recreate an image as it would be produced by the camera.

## 8.3 Conclusion

## This was the process that was used to create simulated images of the infrared beacons as they would appear on the deck of a ship see Fig ######## below. represent the image coordinates . The camera was penpendicular to the world plane simply the required calculations. If the camera is not perpendicular then the plane and is in fact rotated then a translation or series of translations described in section 7.2 Motion in 3D Space would also need to be considered. A rotation translation would be performed on the two known points and , it is only then that the parameterized equations could be used to give the coordinates.



# 9. Project Implementation – The Ship Model

## 9.1 Ship’s Motion Data

The initial framework for the structure of the waves the ship responses to was based on a sinusoidal wave. The data generated for these waves was added to a set of collections and the individual collections were then added to the parent collection \_motionArray. The wave sampling was performed at 30Hz. The decision was made to concentrate on the three motions of a ship that display the greatest variation, these being pitch, roll, and heave. Although the thrust of the ship i.e. it’s surge is constantly under external forces to change from e.g. wind, waves and drag, the velocity and acceleration of the UAV is substantially greater than that of the ship and the overall relative effect is not as significant as the relative motions of pitch, roll and heave. For similar reasons the effects of sway and yaw are not included in the calculations for motion of the ship model. Sway being a lateral sideways motion and yaw a rotation in the axis pointing vertically up and down from the ship’s centre of buoyancy.

When generating the motion data for the ship model a particular vessel was considered. This vessel is the DDG-51 Arleigh Burke Class and is shown in Fig#############



(Sakhal, 2014)

## A number of the characteristics of the ship were referenced when the motion data was being generated.

|  |  |
| --- | --- |
| **Item** | **Ship Details \*at Sea State Six** |
| Name | DDG-51 Arleigh Burke Class |
| Length | 142m |
| Beam | 18m |
| Top Speed\* | 16.5 ms-1 |
| Maximum Roll Angle\* | 16 degrees |
| Maximum Pitch Angle\* | 6 degrees |

## (McDonald, 1993)

The data generated by the application is for the purposes of a test platform for the UAV application and does not allow for the possibility of the vessel’s length being greater the then wavelength of a wave system. When this happens there would be an averaging affect relative to the buoyancy of the ship and any moments about the ships centre of buoyancy in effect at the time.

## 9.2 Environmental Variations

The data for the waves is obtained by sampling a set of six waves at 30Hz each. These waves have random amplitudes, relative frequency and wavelength within the expected ranges for Sea State Six. Other considerations are the ship’s speed and the direction of the waves relative to the ship. A decision was also made to supply datasets for the motions of the Arleigh Burke in Sea State Three and Zero. The ship’s speed, direction relative to the wave direction and the sea state are selections available each time the application is run.

When the process begins a strategy pattern is introduced and the algorithm specific to the sea state selected is implemented. First the heave array is populated.

private void PopulateHeaveArray()

{

if (\_currentSeaState != SeaState.SeaState0)

{

for (int i = 0; i < NumberOfWaves; i++)

{

var samples = (int)(\_stateStrategy.Wavelength \* \_relativeWavelengthFactor

/ \_relativeSpeed \* WaveSampleFactor);

for (int j = 0; j < samples; j++)

{

\_heaveValue = \_stateStrategy.WaveHeight \* (float)Math.Sin(2 \* Math.PI \* j / samples);

\_heaveArray.Add(\_heaveValue);

}

\_samplesPerWaveList.Add(samples);

\_stateStrategy.NewWaveStatistics();

}

}

else

{

\_heaveArray.Add(\_stateStrategy.WaveHeight);

}

}

## With reference to the code above initially a check is performed for Sea State Zero in which case there is no heave motion. Next the outer ‘for’ loop sets the number of iterations of the inner algorithms to the number of waves in a set. Then the number of samples required to provide an approximation of 30 fps is computed. This value is dependent on the relative speed of the ship to the waves, the wavelength range expected for the selected sea state and the direction of the waves.

The inner for loop above calculates the heave displacements for the ship using the randomly generated wave amplitude and the equation for a Sine Wave

Where:

is the corresponding y coordinate for a given time displacement

*A* is the amplitude of the wave,

is the frequency of the wave,

is a given time period, and

is a phase factor, which is only required to shift the phase of the wave and is not used in the heave algorithm.

## Then the number of samples for an individual wave is recorded as this will be required to process the pitch and roll collection and ensure that the three collections are in sync with one another. This is followed by a call to the method that generates a new wave for the next iteration. The PopulateHeaveArray() method terminates with a corresponding ‘else’ statement for the original ‘if’ statement and adds the zero wave height to the collection if the user has selected Sea State Zero when starting the application, this ensure synchronization through the processes.

## Similar operations are then performed to compute the data for the pitch collection and the roll collection, the main difference being the parameters that are used in the calculations. As stated above the number of samples per wave is used when populating these collections ensuring synchronization. The three collections are added to the parent \_motionArray and passed back to the Control Layer which in turn passes the coordinates for the ship model back to the presentation layer.

These three datasets are applied to a one metre cube at the approximate centre of the ship model. As the alterations are applied to the cube the ship model responses equally as it is a child object of the one metre cube, even though it is substantially bigger.

This arrangement posed some initial challenges with Unity’s Mono Develop environment. When applying the various alterations to the ships poise there was expected results depending on the transform function implemented.

Example 1 gameObject.transform.position = new Vector3(0, heaveArray[modCount], 0);

Example 2 Transform myTransform = transform;

myTransform.position = gameObject.transform.position + (Quaternion.Euler(0, heaveArray[modCount], 0) \* new Vector3(0, 0, 0));

Example 3 myTransform.position = gameObject.transform.position + (Quaternion.Euler(0, 0, 0) \* new Vector3(0, heaveArray[modCount], 0));

Example 4 gameObject.transform.position = new Vector3(0, heaveArray[modCount], 40);

Each of these setups worked as expected until they were combined with the transform operations for the pitch and roll rotations. There was also a number of ways this could be approached e.g.

Example 5 gameObject.transform.Rotate(new Vector3(pitchArray[modCount], 0, 0));

Example 6 gameObject.transform.Rotate(-pitchArray[modCount], 0, 0, Space.World);

Example 7 gameObject.transform.eulerAngles =

new Vector3(pitchArray[modCount], 0, 0);

The configuration that finally worked is shown below. Some of the errors there were observed included rotation from the wrong pivot point, single direction motion i.e. the ship model would flip over, erratic motion and the model freezing. During the development period the same set of tested data was used to check the responses of a configuration.

void Update()

{

\_modCount = \_count % \_heaveArray.Count;

gameObject.transform.Translate(0, \_heaveArray[\_modCount] \* Time.deltaTime, 0,Space.World);

transform.eulerAngles = new Vector3(-\_pitchArray[\_modCount], 0, - \_rollArray[\_modCount]);

\_count++;

RecordCoordinates();

}

This Update method shown above uses Time.deltaTime as a factor and Space.World as a parameter within the Translate method for the model responses to the heave data. However if Time.deltaTime was used in conjunction with the Vector3 class instantiation the model’s pivot point altered significantly and so was omitted.

For each iteration of the Update() method the coordinates of the 6 child points on the landing pad, i.e. the infrared beacons, are recorded and added to the \_shipCoordinates collection. This collection is then passed to the UAV application via the applications Service Layer.

## 9.3 UDP Communication

Part of the essential framework of this project implementation includes two UDP channels between the two applications, one in each direction. The decision was made to use UDP because of the reduced overhead associated with UDP packets when compare with IP/TCP communication, Internet Protocol – Transmission Control Protocol. Part of the reason for this speed difference is due to the lack of error checking performed by UDP; this however is acceptable in the proposed model as each new set of data posted to the recipient is complete and independent of the previous or next sets of data. The PostShipCoordinates() method shown below is responsible for sending the coordinates of the infrared beacons to the UAV application. The length of the byte array is set to 4 times the length of the float array as a float is 4 bytes long. The IP end point is also included in the SendTo() method for the server socket.

public void PostShipCoordinates(float [] coordinates)

{

try

{

\_byteArray = new byte[coordinates.Length \* BytesPerFloat];

Buffer.BlockCopy(coordinates, 0, \_byteArray, 0, \_byteArray.Length);

\_server.SendTo(\_byteArray, \_byteArray.Length, SocketFlags.None, \_ipep);

}

catch (Exception ex)

{

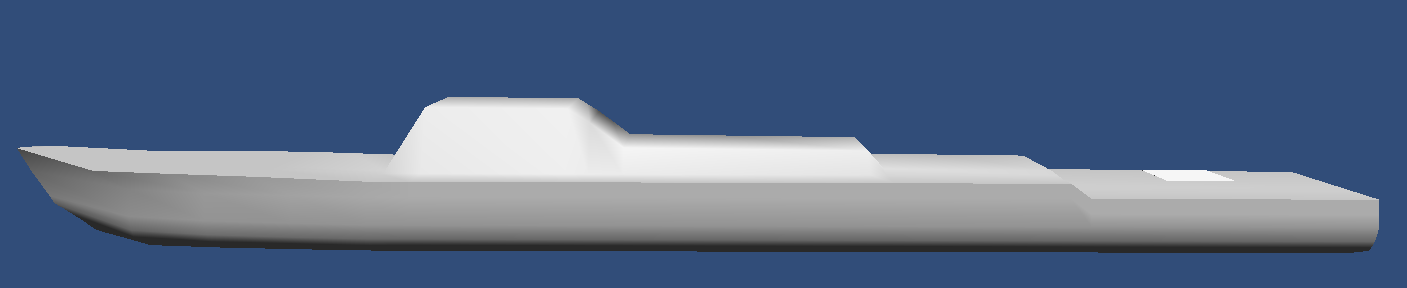
Console.WriteLine(ex.Message);

}

}

As a receive method is normally a blocking method an asynchronous UDP socket is used to receive the data from the UAV application for the course corrections needed to update the UAV model in Unity.

## 9.4 The ship model in Unity

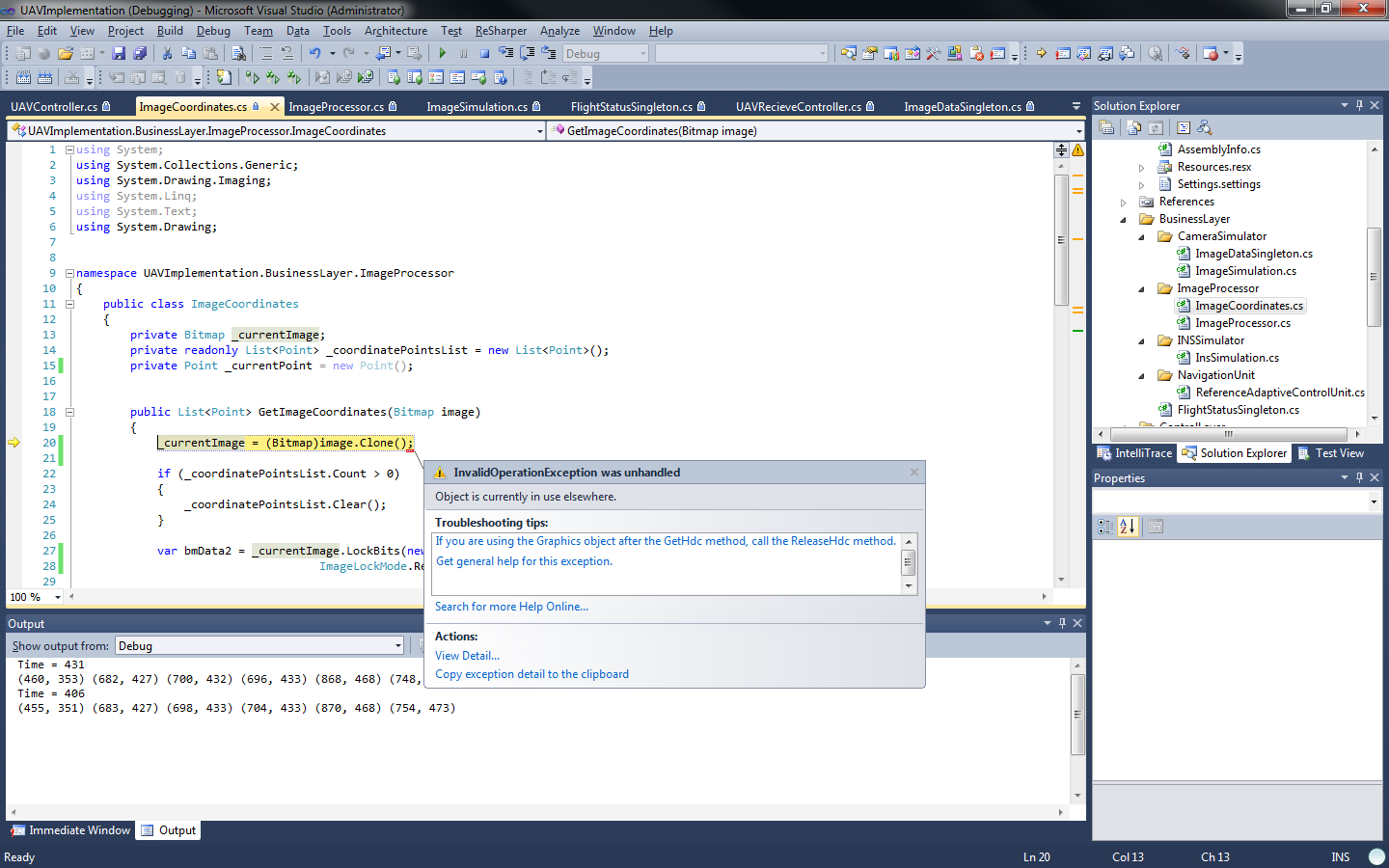


# 10. Project Implementation – The UAV Model

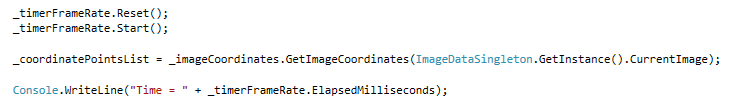
Image Processing

To simulate the images that would be generated by a camera fixed to the fuselage of the UAV a succession of bitmaps were generated (see chapter 9 Fig ###############). These images were generated on a dedicated thread in stored in a local variable \_currentImage. On another thread these images were then interpreted and the coordinated of the simulated beacons were established.

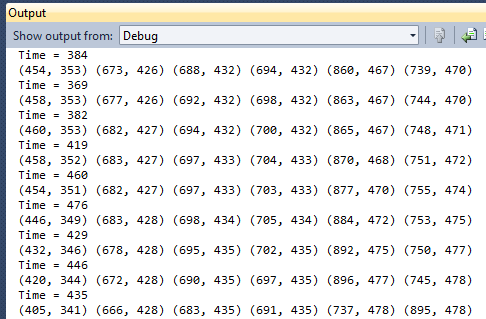
There were issues with this process as the system struggled to read images in memory when they were open in another process i.e. the ImageSimulation.cs class. For speed purposes the ImageSimulation.cs class locks the location in memory to performed the required tasks and the processing speeds were satisfactory. To overcome this problem initially an attempt was made to clone the image and allocate the image to a new location in memory however the problem still persisted, see Fig ############ below. The issue reported was related to graphics objects being used elsewhere.



This issue was eventually resolved by added the bitmap image to the ImageDataSingleton.cs class and the required processing in the ImageProcessor.cs class was executed see Fig ########## below.



The system performed without exception however the time required to process the images was unsatisfactory with the execution time taking approximately 0.4 seconds, see Fig ########## below (time is in milliseconds).



In a real world system this parallel processing issue would not arise as separate units would handle the image creation (i.e. the camera) and the image processing. For this reason it is proposed to exclude the image processing from this simulation and pass the coordinates directly to the flight control unit. The only steps that it is proposed to exclude from the system are the image simulation and the corresponding image processing to retrieve the initial beacon coordinates stored the image which have been shown to operate albeit with high levels of processing required.

This processing problem may be overcome with the use of GPU programming and libraries like OpenCL (AMD, 2014).

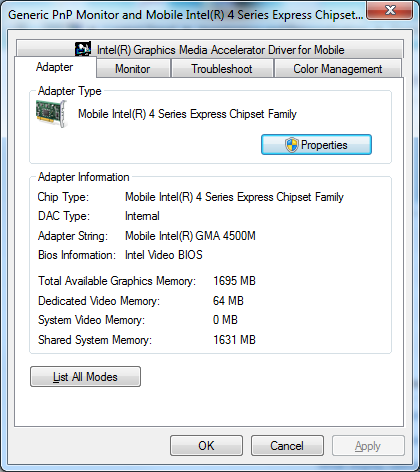
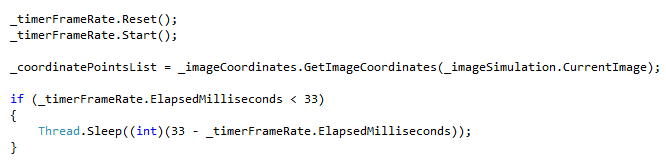
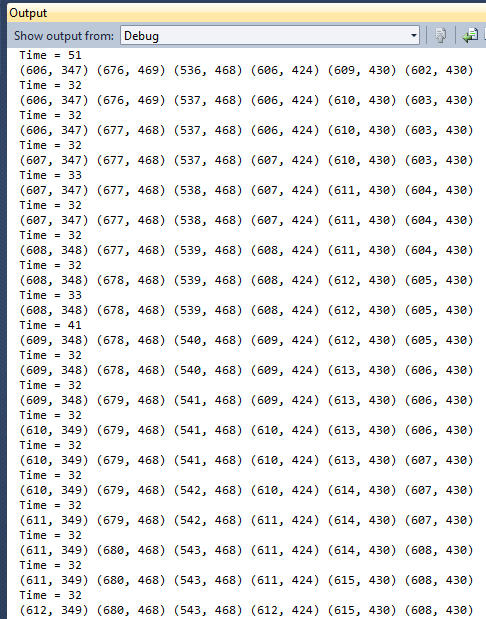
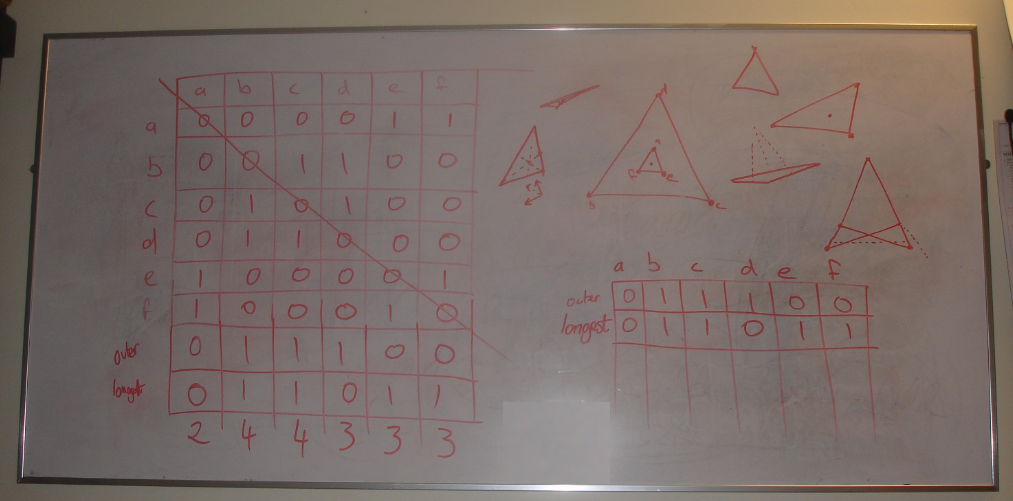


Image processor skipped







# 8. Project Management Diary

## 8.1 Literature Review Diary

|  |  |
| --- | --- |
| **Date (M-S)** | **Work Items** |
| **September** |  |
| 16-22 | Formulated project idea and submitted the project vision statement |
| 23-29 | Initial research carried out for the project. Concerns where expressed about the scope of the project. |
| **October** |  |
| 30-6 | Research into the operational parameters of ships and UAVs |
| 7-13 | Further research and document development |
| 14-20 | Further research and document development |
| 21-27 | Submitted the Project Proposal Document including an outline TOC, introduction and the project research question |
| 28-3 | Evaluated trajectory computations and further research |
| **November** |  |
| 4-10 | Development of risk analysis and design documents |
| 11-17 | Submission of the Project Plan including one complete chapter, risk analysis, implementation objectives and preliminary design documents |
| 18-24 | Developed a prototype is the proposed system with Unity and Visual Studio |
| 25-1 | Evaluation of the mathematical models involved in the project and further development of design documents |
| **December** |  |
| 2-8 | Document development and editing |
| 9-15 | Literature Review submission |
| 16-22 | Literature Review and prototype presentation |

## 8.2 Implementation Diary

|  |  |
| --- | --- |
| **Date (M-S)** | **Work Items** |
| **January** |  |
| 13-19 | The initial setup up of the ship implementation was established. This involved a tiered framework with the presentation layer alone implementing the Unity ‘MonoDevelop’ interface. |
| 20-26 | Work continued on the project framework. In the Ship implementation data for the motion of the ship model is generated considering external forces and passed to the ship model in the presentation layer model. Work also commenced on testing for the business layer. |
| 27-2 | The motion algorithms for the ship motion where further refined and integrated into the Unity model script. This proved complicated and the dependencies between the different ‘GameObject’ movement controls had undesirable affects. Further work on the test for this implementation. |
| **February** |  |
| 3-9 | The bi-directional UDP communication was established between the UAV implementation and the ship implementation on the two PCs |
| 10-16 | An application test rig for the image generation was developed. |
| 17-23 | Further refinement on the image test rig. Started work on testing for the image simulation component of the project. |
| 24-2 | The image test rig was integrated into the UAV implementation. Further work was also carried out on the testing of the UAV business layer. |
| **March** |  |
| 3-9 | The image generation component to the project and testing is complet and tested. Presentation to superior and demonstration of partially complete video log for the project. |
| 10-16 |  |
| 17-23 |  |
| 24-30 |  |
| **April** |  |
| 31-6 |  |
| 7-13 |  |
| 14-20 |  |
| 21-27 |  |
| **May** |  |
| 28-4 |  |
| 5-11 |  |
|  |  |
|  |  |

# 9. The UAV Model

Placeholder

# 10. The Ship Model

Placeholder

# 11. From Target Acquisition to Touchdown

Placeholder

# 12. Results

Placeholder

# Appendix

## Appendix 1 Douglas Sea Scale

Douglas Sea Scale (Met Office, 2010)

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Height (m) | Sea State | Swell Description |
| 0 | 0 | Calm (glassy) | No Swell |
| 1 | 0 - 0.1 | Calm (rippled) | Very low (short and low wave) |
| 2 | 0.1 - 0.5 | Smooth (wavelets) | Low (long and low wave) |
| 3 | 0.5 - 1.25 | Slight | Light (short and moderate wave) |
| 4 | 1.25 - 2.5 | Moderate | Moderate (average and moderate wave) |
| 5 | 2.5 - 4.0 | Rough | Moderate rough (long and moderate wave) |
| 6 | 4.0 - 6.0 | Very rough | Rough (short and heavy wave) |
| 7 | 6.0 - 9.0 | High | High (average and heavy wave) |
| 8 | 9.0 - 14.0 | Very high | Very high (long and heavy wave) |
| 9 | Over 14.0 | Phenomenal | Confused (wave length and height indefinable) |

# Works Cited

Astrium. (2013, July 10). *Key Features*. Retrieved November 28, 2013, from DeckFinder: http://deckfinder.net/key-features/

Biven, C. C., & Guercio, J. G. (1987). *A Simulation Investigation of Scout/Attack Helicopter Directional Control Requirements for Hover and Low-Speed Tasks.* California: National Aeronautics and Space Administration. Retrieved November 4, 2013, from https://www.google.ie/url?sa=t&rct=j&q=&esrc=s&source=web&cd=10&cad=rja&sqi=2&ved=0CGUQFjAJ&url=http%3A%2F%2Fwww.dtic.mil%2Fcgi-bin%2FGetTRDoc%3FAD%3DADA185874&ei=Mw59UuHCIIqV7Qa5s4DAAg&usg=AFQjCNHJ8Z4-ncgeOfiU-7ki8WCqHVN4Iw&bvm=bv.56146854,d.ZGU

Blender. (2013, October 31). *Blender 2.69*. Retrieved November 24, 2013, from Blender: http://www.blender.org/download/

Din, A., Bona, B., Morrissette, J., Hussain, M., Violante, M., & Naseem, F. (2012). Embedded Low Power Controller for Autonomous Landing of Small UAVs using Neural Networks. *10th International Conference on Frontiers of Information Technology* (pp. 196-203). Islamabad: Conference Publishing Services & IEEE Computer Society. Retrieved October 6, 2013, from http://0-ieeexplore.ieee.org.acpmil13web.ancheim.ie/stamp/stamp.jsp?tp=&arnumber=6424321

Esmailifar, S. M., & Saghafi, F. (2009). Autonomous Unmanned Helicopter Landing System Design for Safe Touchdown on 6DOF Moving Platform. *Fifth International Conference on Autonomic and Autonomus Systems* (pp. 245-250). Valencia: IEEE. Retrieved October 26, 2013, from http://0-ieeexplore.ieee.org.acpmil13web.ancheim.ie/stamp/stamp.jsp?tp=&arnumber=4976611

Fang, R., Krijns, W., Finch, R. S., Geyer, W. P., Long, K., & Carico, D. (2003). *Helicopter/Ship Qualification Testing.* North Atlantic Treaty Organisation/Research and Technology Organisation, The Research and Technology Organisation (RTO) of NATO. Ottawa: St. Joseph Print Group Inc. Retrieved October 19, 2013, from https://www.google.ie/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CCwQFjAA&url=http%3A%2F%2Fwww.dtic.mil%2Fcgi-bin%2FGetTRDoc%3FAD%3DADA411975&ei=W6hiUoalA8eshQe1yICYCw&usg=AFQjCNH8bYfwU1lJw8j0ueLwmMsj89JhLQ&bvm=bv.54934254,d.ZG4

Federal Aviation Administration. (2012). *Helicopter Flying Handbook.* Oklahoma City: United States Department of Transportation. Retrieved November 4, 2013, from http://www.faa.gov/regulations\_policies/handbooks\_manuals/aviation/helicopter\_flying\_handbook/media/helicopter\_flying\_handbook.pdf

Florida Centre for Instructional Technology. (2005). *Waves in the Ocean.* Retrieved November 15, 2013, from fcit.usf.edu: http://fcit.usf.edu/florida/teacher/science/mod2/resources/waves.pdf

Herissé, B., Hamel, T., Mahony, R., & Russotto, F.-X. (2012). Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow. *Robotics, IEEE Transactions on (Volume:28 , Issue: 1 ), 28*(1), 77-89. doi:10.1109/TRO.2011.2163435

Hood Tech Vision. (2013). *Hood Tech Vision Product Summary Table.* Retrieved November 9, 2013, from www.hoodtechvision.com: http://www.hoodtechvision.com/products.html

IPython. (2013, September). *IPython Interactive Computing*. Retrieved November 24, 2013, from ipython.org: http://ipython.org/

Kamalasadan, S., & Ghandakly, A. A. (2011). A Neural Network Parallel Adaptive Controller for Fighter Aircraft Pitch-Rate Tracking. *Instrumentation and Measurement, IEEE Transactions on* , 258-267. doi:10.1109/TIM.2010.2047310

Kearfoot. (2011, August). *KN-4072A Airborne INS/GPS*. Retrieved November 9, 2013, from www.kearfott.com: http://www.kearfott.com/images/stories/pdf/DATASHEETS\_KGN\_NJ/AIR/kn-4072a\_avionics\_ins-gps.pdf

Lobik, M. D. (2003, March 14). *Helicopter Dynamic Rollover.* Retrieved November 20, 2013, from Nav Air: http://www.navair.navy.mil/safety/toolkit.cfm

MathWorks. (2013). *MatLab 30-Day Free Trial*. Retrieved November 24, 2013, from MathWorks: http://www.mathworks.co.uk/programs/nrd/matlab-trial-request.html?ref=ggl&s\_eid=ppc\_6060

McDonald, M. (1993). *SHF SATCOM Terminal Ship-Motion Study.* Technical Report, United States Navy, Naval Command, Control and Ocean Surveillance Centre, San Diego. Retrieved September 27, 2013, from http://www.spawar.navy.mil/sti/publications/pubs/tr/1578/tr1578.pdf

Met Office. (2010). *The Beaufort Scale.* Retrieved October 10, 2013, from www.metoffice.gov.uk: http://www.metoffice.gov.uk/media/pdf/b/7/Fact\_sheet\_No.\_6.pdf

NovAtel. (2012). *Automated Flight Control System*. Retrieved November 28, 2013, from www.novatel.com: http://www.novatel.com/technology-in-action/automated-flight-control/

NUnit. (2013, October 10). *NUnit - Home*. Retrieved from www.nunit.org: http://nunit.org/index.php?p=home

Padfield, G. D. (2007). *Helicopter Flight Dynamics* (2nd ed.). Oxford: Blackwell Publishing. Retrieved October 12, 2013, from http://www.foinikas.org/ftp/public/DCS%20Blackshark/HELICOPTER%20FLIGHT%20DYNAMICS.pdf

Prism Defence. (2010). *Prism Defence*. Retrieved November 20, 2013, from www.prismdefence.com: http://www.prismdefence.com/index.html

pythonxy. (2013, November 21). *Downloads*. Retrieved November 24, 2013, from pythonxy Scientific-oriented Python Distribution based on Qt and Spyder: https://code.google.com/p/pythonxy/wiki/Downloads?tm=2

Rotomotion. (2009, November). *SR200 Helicopter UAV Specifications.* Retrieved from www.rotomotion.com: http://www.rotomotion.com/datasheets/sr200\_uav\_sheet.pdf

Rotomotion. (2011, Feburary 14). *SR 200*. Retrieved from www.Rotomotion.com: http://www.rotomotion.com/r\_product\_5\_sr200.html

Sakhal. (2014). *Arleigh Burke class*. Retrieved March 7, 2014, from American Cruisers and Destroyers: http://www.shippictures.byethost22.com/american%20destroyers.htm

Sanchez-Lopez, J. L., Saripalli, S., Campoy, P., Pestana, J., & Fu, C. (2013). Toward visual autonomous ship board landing of a VTOL UAV. *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on* (pp. 779-788). Atlanta: IEEE. doi:10.1109/ICUAS.2013.6564760

Sandino, L. A., Bejar, M., & Ollero, A. (2011). On the applicability of linear control techniques for autonomous landing of helicopters on the deck of a ship. *Mechatronics (ICM), 2011 IEEE International Conference on*, 363-368. doi:10.1109/ICMECH.2011.5971312

Schiebel. (2009, December 16). *Schiebel Press*. Retrieved November 9, 2013, from www.schiebel.net: http://www.schiebel.net/AcmsFile/1311/0/550/2009-12-16\_Schiebel\_demonstrates\_CAMCOPTER\_S-100\_t.pdf

Schiebel. (2013). *CAMCOPTER-S-100*. Retrieved November 8, 2013, from www.schiebel.net: http://www.schiebel.net/Products/Unmanned-Air-Systems/CAMCOPTER-S-100/System.aspx

Schiebel. (2013). *Image-Gallery*. Retrieved from www.schiebel.net: http://www.schiebel.net/AcmsFile/1782/0/550/CAMCOPTER\_S-100\_084.jpg

Shin, H., You, D., & Shim, D. H. (2013). An autonomous shipboard landing algorithm for unmanned helicopters. *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on* (pp. 769-778). Atlanta: IEEE. doi:10.1109/ICUAS.2013.6564759

Snyder, C. M. (2012). Validation of Ship Air Wake Simulations and Investigation of Ship Air Wake Impact on Rotary Wing Aircraft. *Launch & Recovery.* Maryland: American Society Naval Engineers. Retrieved November 15, 2013, from http://www.cobaltcfd.com/pdfs/ASNE\_2012\_ship\_airwake\_snyder.pdf

Tang, Y.-R., & Li, Y. (2012). Design of an optimal flight control system with integral augmented compensator for a nonlinear UAV helicopter. *Intelligent Control and Automation (WCICA), 2012 10th World Congress on* (pp. 3927-3932). Beijing: IEEE Conference Publications. doi:10.1109/WCICA.2012.6359128

Techet, A. H. (2004, September 8). *Hydrodynamics for Ocean Engineers.* Retrieved November 20, 2013, from web.mit.edu: http://web.mit.edu/13.012/www/handouts/Reading3.pdf

Techet, A. H. (2005). *Ocean Waves.* Retrieved November 15, 2013, from web.mit.edu: http://web.mit.edu/13.42/www/handouts/wave\_spectra\_slides2.pdf

TF3DM. (2013). *Helicopter N916MU 3d model*. Retrieved from http://tf3dm.com/3d-model/helicopter-n916mu-72852.html

Thompson Dydek, Z. (2010, June 30). Adaptive Control of Unmanned Aerial Systems. Massachusetts, USA. Retrieved from DSpace@MIT: https://www.google.ie/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&ved=0CEYQFjAD&url=http%3A%2F%2Fdspace.mit.edu%2Fbitstream%2Fhandle%2F1721.1%2F62324%2F712601752.pdf&ei=GPKoUq-ZDMe6hAff-IGQBw&usg=AFQjCNFzDOSgQzqQYVIh-GcGBobgo98B-g&bvm=bv.57799294,d.Z

Unity. (2013). *What is Unity and what can I do with it?* Retrieved November 24, 2013, from Unity: http://unity3d.com/pages/create-games?gclid=CMbP-6Pi\_roCFWd72wodkgkAgA

Yu, Z., Nonami, K., Shin, J., & Celestino, D. (2007). 3d vision based landing control of a small scale autonomous helicopter. *International Journal of Advanced Robotic Systems, 4*(1), 51-56. Retrieved October 12, 2013, from http://cdn.intechopen.com/pdfs/4211/InTech-3d\_vision\_based\_landing\_control\_of\_a\_small\_scale\_autonomous\_helicopter.pdf

Yuan, W. (2013, August 5). Dynamic Modelling and Flight Control Methodologies for Vertical Take-Off and Landing Unmanned Aerial Vehicles. Sydney, New South Wales, Australia. Retrieved November 9, 2013, from https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=23&cad=rja&ved=0CDIQFjACOBQ&url=http%3A%2F%2Funsworks.unsw.edu.au%2Ffapi%2Fdatastream%2Funsworks%3A11442%2FSOURCE01&ei=i9Z\_Up21JceQhQfsp4DgCg&usg=AFQjCNEB8SQgD8Xr4TLfM9\_YGd-WK4ipYw&bvm=bv.56146