University of Glasgow

University of Glasgow (2017/18)

Cooperative Motion Control of Autonomous Surface Vehicles and Quadcopters

ENG5041P: Individual Project 5

Matthew Cole 2070583

MEng Mechanical Engineering with Aeronautics

First Supervisor: Dr. Euan McGookin

Second Supervisor: Dr. Dave Anderson

Abstract

Presented in the following are methods for the cooperative motion control of two distinct classes of vehicle; the Autonomous (Marine) Surface Vehicle, and the Quadcopter. A system such as this has many potential applications in data acquisition in marine and aquatic environments.

The dynamic and kinematic models are presented, low level state control of the vehicles is proposed and simulation results shown. Satisfactory results are seen in both vehicles, with improved performance in the Quadcopter controllers.

Autonomy is introduced to the system by means of a path following algorithm, which uses cross track error to make changes in the vehicles' heading references. A constant nominal velocity reference is provided to the vehicles throughout each mission. Further simulation results are shown in following of paths consisting of straight lines and arcs.

The path following cooperative controller produced uses normalised coordination states in each of the vehicles, which should all be equal at all times, to generate a coordination error term. Commands to achieve coordination are in corrective speed references, applied differently to each of the vehicles.

In the simulations shown. up to four vehicles follow parallel paths in a constant formation. The algorithms produced include the possibility to extend this simplified definition of cooperation to more complex scenarios by choice of coordination states, and more complex path references.

Results suggest the proposed system is feasible, but further work needs to be carried out to test the controllers in trials in real vehicles.

Acknowledgements

I would like to thank all of the staff and students at the Institute for Systems and Robotics (Técnico) in Lisbon, for their knowledge and guidance throughout my time with them in the autumn semester of 2017.

Particularly, I wish to thank Professor Antonio Pascoal, who offered me the opportunity to work with his students, and offered me his great experience throughout.

I would also like to thank my friends and family, who have supported me throughout the last six months, and through the last 5 years at the University of Glasgow & TU Delft.

Contents

1 Introduction

1.1 Background

Modern developments in computing technologies and robotics have allowed for the application of autonomous vehicles in a wide range of complex missions. Marine vehicles have been utilised in; finding subsea hydrothermal vents with position and CH_4 (Methane) sensors, and studying dispersal and spawning of fish using telemetry devices [1]. Quadcopters and other forms of Unmanned Aerial Vehicles (UAVs) are already seeing commercial applications in a wide range of areas; collecting radiation data and assessing damage after the Fukushima nuclear disaster [2], forest fire monitoring, humanitarian observations, agricultural data collection, and gas pipeline monitoring [3]. The growth of the market for UAVs [4], and even the popularisation of drone racing [5] is causing rapid improvement in the technology.

Advanced motion control systems have already been applied in both Marine Vehicles and UAVs with a high level of success.

The Widely Scalable Mobile Underwater Sonar Technology (WiMUST) project (2015-18) undertaken at the Institute for Systems and Robotics (ISR), Lisbon aims to create a reconfigurable acoustic array for geophysical surveying at sea [6]. A group of marine robots manoeuvre together, an emitter produces high power sonic shockwaves, which are reflected on the seabed and received by the vehicles. The measured signals and time delays can then be converted into spatial information. In established methods, there is no possibility to change the geometry of the receivers (known as the "virtual streamer") so re-configurability of the receivers' positions is an exciting development which is likely to improve results of these surveys

Missions which combine Aerial and Marine vehicles have a greatly widened scope, due to the increased potential for data collection introduced by each new element in the system. Although not directly associated with WiMUST, the work presented here could form an extension to their, or similar, systems. Introducing an aerial element to networks of marine vehicles has the potential to collect new and useful data about the environment and the participating vehicles.

An example of an Autonomous Surface Vehicle (ASV) and UAV demonstrating cooperative control may be seem in [7], deployed to assess damage to property after Hurricane Wilma, in Florida. The UAV used in this example is a micro helicopter, but promising results were produced, and a number of areas for future investigation were highlighted.

The reducing cost of components including batteries, integrated circuits, and electric motors will likely see the commercial market share increasing – leading to more varied and widespread applications of autonomous vehicles.

An extensive overview of autonomous vehicles in [8] demonstrates the state of the art of ground, aerial and underwater robots in 1996, this may help us understand the rate of progress over the last 20 years in the field. The understanding of many of the problems to face was strong, but examples of implementation were still rudimentary. Both hardware, and software, have improved to such a degree that some of the experimental concepts mentioned in the sections on UAVs are now mass-produced, commercially available devices.

1.2 Proposed Scenario

A use case for a cooperative system of ASVs and Quadcopters is in assessing the extent of a pollutant spillage from an oil rig or other marine installation. The ASV would be fitted with chemical sensors which measure concentration of harmful substances at points in space, allowing maps to be produced. The Quadcopter can carry visual sensors which can record an overview of the spill area, this may be used to dynamically change the mission planning, and direct the ASV toward problem areas.

Additionally, the Quadcopter's height may be utilised, and the vehicle used as a communications relay with a command centre onshore or on board a manned vessel. This should provide a continuous data stream to engineers, and the ability to command and re-plan the mission remotely.

1.3 Literature Review

This literature review intends to give an outline of work carried out up to the present day in the fields encompassed by this project. It is broken down into four subsections; Autonomous Surface Vehicles, Quadcopters, Path Following, and Cooperative Motion. Information about the vehicles and their methods of control are required before it is possible to complete complex missions with them, including path following and cooperation.

More specific information about the vehicles' dynamics, algorithms for path following and cooperation, which are used within the project are discussed in the relevant chapters later.

1.3.1 Autonomous Surface Vehicles

A brief overview of ASV technology since its beginnings in 1993 can be found in [9]. First it details the work of early teams at the Massachusetts Institute for Technology and elsewhere,

who produced catamaran and kayak like surface craft capable of tracking a tagged fish, and measurement of water depth. The 5 years leading up to 2008 showed promising signs of improvement and interest in the technology, new research teams began producing their own vehicles for tests. The report explains the US Navy's interest in ASVs as an interface between underwater and aerial vehicles in networked communication, this is of particular relevance to the work here. Also acknowledged is the regulatory restrictions ASVs face, as they operate in environments with traditional vehicles.

The ASV considered within this project is the MEDUSA, a class of autonomous marine vehicle developed at the Institute for Systems of Robotics (ISR) in Lisbon. The overview of the MEDUSA seen in [10], explains its design process and choices made based on five key requirements. The vehicle is intended to be a versatile, low cost test bed for research in the field of surface *and* submersible marine robotics. Two variants exist, the $MEDUSA_S$ for surface missions and the $MEDUSA_U$ for underwater, only the former is considered within this report.

This report provides a useful background into the work of ISR, and many of the components of this project, specifically the ASV, path following, and cooperation.

The double cylinder structure seen in Figure 1 provides space for sensors to be placed both sub-surface and above surface, and the modularity of the system allows for modifications to be made depending on mission requirements. Information about navigation systems is provided, for surface missions this is an easier task using GPS (Global Positioning System) at 10Hz with Kalman filtering applied. This report also provides the building blocks for high level mission controllers for future missions.

Figure 1 - Examples of the Medusa class of vehicle during sea trials in Sesimbra, Portugal

1.3.2 Quadcopter

The concept of a four bladed Vertical Take-off and Landing (VTOL) rotorcraft has existed for over one hundred years. In 1907 the aviation pioneer Louis Breguet built the *Breguet-Richet Gyroplane.* The article [11] (*French*) shows a large single passenger vehicle with four powered rotors. The craft was demonstrated in public and successfully left the ground, before a structural failure caused it to break apart, and the prototype vehicle to be abandoned.

In the subsequent century of aviation, huge progression has been made in VTOL rotorcraft, with helicopters being commonplace vehicles for many applications. Quadrotors however saw little development until more recently, largely due to Size, Weight and Power (SWAP) constraints. Exceptions, where development of large Quadcopters has taken place, include the *Boeing Quad TiltRotor* proposed in 2006 [12].

In the meantime, other types of aerial vehicle were introduced with autonomous control. A trajectory tracking autonomous helicopter was proposed in [13] which produced satisfactory simulation results, but using a simplified model, and no real test results are available as validation. Autonomous *auto-pilot* systems are used extensively throughout the industry.

Particularly in military applications, fixed wing UAVs have been in operation for a number of years, often controlled remotely by pilots. [14] Presents the technology used in a small lightweight fixed wing aircraft with a degree of autonomy. A method of path planning and trajectory tracking is described, which produces a path more suitable to the vehicle's constraints than a simple waypoint based method. The achievement here is the ability to calculate paths in-flight using lightweight computers.

Technology for autonomous control of Quadcopters has seen rapid improvement in recent years and is reaching maturity. Early results are seen in [15] where a primitive quadrotor or *"X4-flyer"* was developed and tested. Components available at the time typically had a lower Power to Weight ratio than those that are available today, and so power was supplied via a tether, limiting potential applications and range. The control system required pilot input and did not have autonomy, and measurement of orientation proved difficult due to lateral accelerations causing interference in the rate gyro.

In [16] is presented a model for Quadcopters from positions with considerable experience using these vehicles. A number of important factors are highlighted, including aerodynamics effects such as blade flapping, SWAP constraints, and state estimation. Sensible control methods,

including in trajectory control, are explained but no results either in simulation or implementation are provided in this tutorial overview.

Advanced trajectory tracking control is presented in [17] where a back-stepping method is applied to improve disturbance rejection and aggressive manoeuvre performance. Back-stepping is achieved as in [18], and Lyapunov Functions are used to ensure asymptotic stability in the region of the desired equilibrium point in attitude control. The control system was tested indoors using a *VICON* advanced motion capture system, this has the advantage of very high accuracy results when properly calibrated, and as such, the results shown should be relied upon.

The Quadcopter was commanded to follow a 3D figure of eight trajectory, initial errors are high, but after $5\ seconds$ the vehicle converges on the trajectory. It appears in the results that the vehicle does not converge accurately on the spatial path throughout the entirety of its cycle, instead favouring a steady state trajectory with minor adjustments to control inputs.

1.3.3 Path Following

To test the cooperation of the vehicles in this system, it is necessary to command them to follow pre-determined paths in space. Two methodologies are commonly suggested within the literature; *Trajectory Tracking* (TT) and *Path Following* (PF). In TT, the vehicle follows an instantaneous position signal which is pre-defined and fixed in time. In PF there are no time constraints, and the vehicle is commanded to follow a path with a fixed velocity profile.

[19] Explains that when un-planned disturbances occur, simple TT algorithms may provide speed commands above system capabilities, or too low, leading to instability in the systems. Due to the lack of spatial awareness relative to the path ahead, motions produced can be unnatural, when typically a smooth path is desired and time is not critical.

It is generally understood in the newer literature that for more intelligent systems, PF provides a useful degree of flexibility. Otherwise, increasingly intelligent TT systems are employed which alter the trajectory based on performance and conditions.

[20] Describe a path following controller at dynamics level, with an architecture useful to this project, where the path following algorithm does not greatly depend on the vehicle dynamics, allowing for only one to be developed. Unfortunately this system requires the inner loop dynamics to react more quickly than the outer loop, and typically Quadcopters are more manoeuvrable than ASVs, so tuning of this controller to suit multiple vehicles may prove difficult.

[21] Compares two methods of path following in ASVs; the *Vector Field Method* (VF), and the *Line of Sight Method* (LOS), both of these methods are well understood with practical examples of applications. VF creates a mathematical field of direction vectors in the region of the desired path, which are translated into vehicle heading commands, this method was initially developed for use in UAVs.

LOS has seen more applications, a circle with constant radius, representing the "line of sight" of the vehicle is defined. When the vehicle is sufficiently close, there are two points of intersection between the body fixed circle and the path, and the vehicle is commanded to head towards the point where the path directions agree. This method is displayed in Figure 2 for straight line and circular paths. The body fixed circle is shown by a dotted line, and the path is in blue, R denotes radii, p denotes positions in 2D space, χ denotes angles.

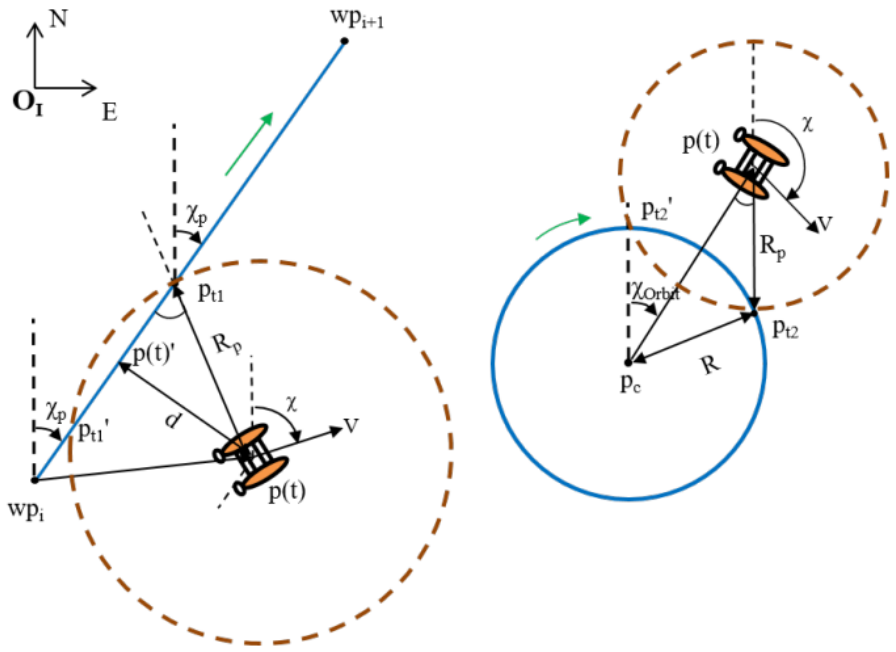


Figure 2 - Line of sight path following for ASVs [21]

[21] also compares results from trials with VF and LOS methods applied. In simulations of straight line paths, LOS provided faster convergence, but at the expense of using twice the *steering energy* (as defined in the paper) when compared with VF. In circular paths, VF produces inconsistent results depending on arc radius, but typically with lower oscillations that LOS. In implementation, LOS provided better and faster path following, with VF producing less accurate tracking while consuming less energy. There is perhaps a bias towards proving VF a viable method, significant work needs to be done on the algorithms for applying this method to ASVs.

1.3.4 Cooperation

Cooperative control aims to command a group of two or more vehicles to move in a way in which they are co-dependent. Cooperation can take many forms, vehicles can mimic motions of another, or behave like a flock of birds. Early work on formation control of mobile robots is seen in [22] where feedback control laws are presented. Formation control intends on keeping a group of autonomous vehicles in a fixed formation as they move along a pre-planned route or otherwise.

One leader vehicle is provided with a planned path, and all other vehicles use on-board sensors to measure distances between one another in order to maintain a fixed formation. The two control algorithms proposed may be mathematically proven to have asymptotic convergence on the equilibrium, but do not account for any complex vehicle dynamics as only simplified kinematics are simulated.

A common distinction made is *centralized* or *decentralized* control. Presented in [22] is an example of decentralized cooperative control, where computations are made on-board each of the vehicles. This method typically improves redundancy properties, the trade-off is in the requirement for more processing power, adding weight which is particularly critical in Quadcopters.

There are a number of methodologies applied to the cooperative control of multi-agent systems. The *Leader Tracker System* (LTS) has vehicles tracking the reference trajectory taken by a leader based on relative position measurements [23]. LTS has been demonstrated successfully, but as both classes of vehicle in this system will have access to simple GPS positioning data there is no need for the vehicles to be reliant on a single leader vehicle.

Behaviour Coordination (BC) specifies high level goals such as flocking or swarming, where rules define vehicle behaviour as in automata theory [24], [25]. BC is unsuitable for the typical mission profiles proposed for the system designed, as it produces complex and somewhat unpredictable motion. This method has however been tested successfully, and produces interesting results.

The method chosen here is a cooperative path-following controller. This is a system where a group of vehicles are required to follow pre-defined spatial paths while maintaining a formation pattern in time [26]. The paths (as in Section 5) are defined spatially, with an associated speed profile which is constant between vehicles. The controller provides a correction velocity which

adjusts the velocity command received by each vehicle, providing temporal control to the vehicles. This method has been demonstrated with success in [27] for UAVs and [28] for ASVs.

Communication between vehicles is of particular importance in the quality of control in all of these potential algorithms, as data about other vehicles' states are required. In almost all test cases there will be communication failures of some form, usually a drop-out of the link between one or more vehicles. It is therefore important that intermittent communication failures do not cause the mission to require abortion [29].

2 Coordinate Systems

The motion of the ASV and Quadcopter are defined by distinct sets of equations with 6 degrees of freedom, which must be defined in order to simulate and control their motions. It is important that a consistent coordinate system is established for each of the vehicles, and for the global frame.

2.1 Coordinate System

The global frame chosen is the *North East Down* coordinate system (NED) [30] which may be easily measured with a magnetic sensor installed in the vehicles. It makes sense for the altitude component (*Down*, or Z) to have zero value at sea level, as the ASV will operate on this plane, and as such the relative height of the air-borne Quadcopter is easily found.

A diagram of the chosen coordinate system is seen in Figure 3, the ASV and Quadcopter body fixed frames are included.

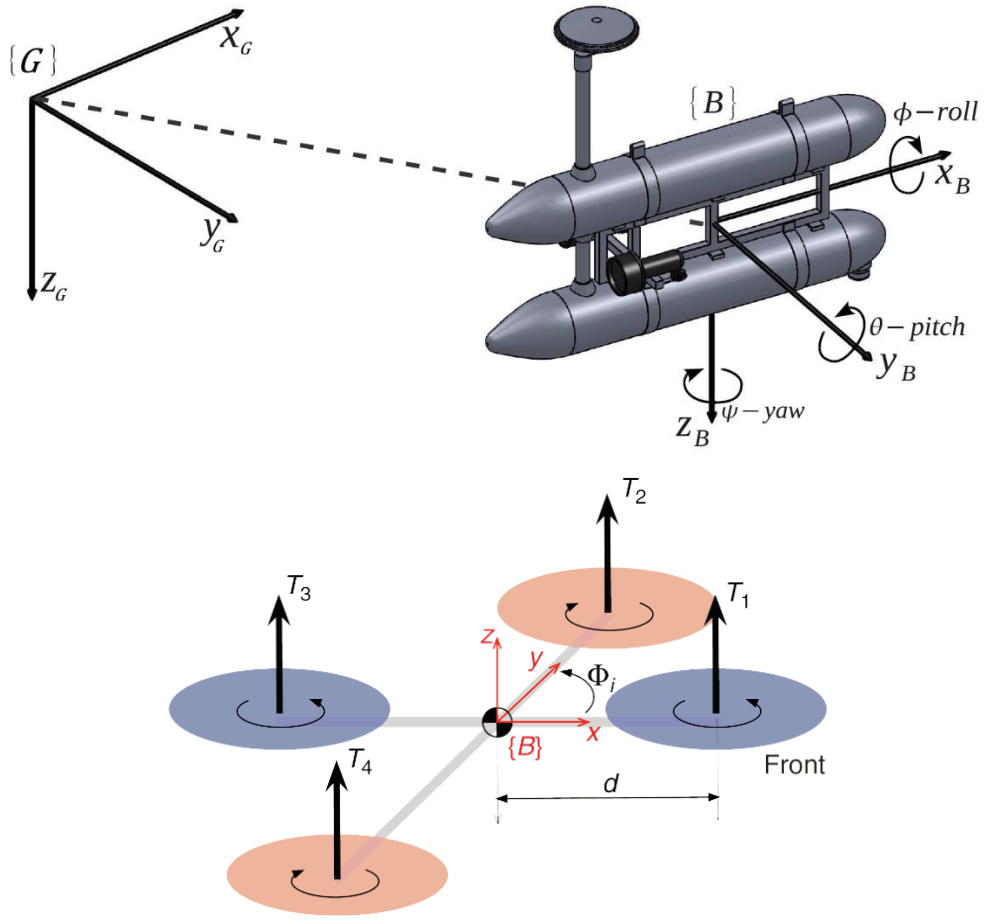


Figure 3 - Global coordinate frame, ASV and Quadcopter body fixed frames, taken from [16] & [28]

The origins of $\{B\}$ for both vehicle classes are located on the centre of mass.

The x-axis of the ASV is aligned along the line of action of the actuators, the z-axis is oriented downward, and the y-axis is the resultant bi-normal.

The Quadcopter x-axis is oriented along one of the arms of the vehicle, y-axis along an adjacent (perpendicular) arm, and z-axis downwards. It does not make sense to align the x-axis of the UAV along the line of action of the actuators as in the ASV, as this points vertically and is of little relation to the x-axis of the inertial frame or of the ASV's body frame.

Attitude angles are defined in the same way for each, and left out of the Quadcopter diagram for clarity.

Position information will be denoted as follows:

$$[X, Y, Z]^G_{UAV} = 3\ axis\ position\ of\ the\ UAV\ (m)\ in\ \{G\}$$
$$[X, Y, Z]^G_{ASV} = 3\ axis\ position\ of\ the\ ASV\ (m)\ in\ \{G\}$$
$$[x, y, z]^B_{UAV} = Local\ position\ of\ the\ UAV\ in\ \{B\}$$
$$[x, y, z]^B_{ASV} = Local\ position\ of\ the\ ASV\ in\ \{B\}$$

Superscripts indicate the reference frame in which the variable is measured. Subscripts indicate the vehicle class - this is omitted if the vehicle class is made clear by context to improve readability. UAV (Unmanned Aerial Vehicle) may be used as a shorthand for Quadcopter when pertinent.

Attitude information for both vehicle classes is defined by:

$$\phi = roll\ angle\ about\ x^B$$
$$\theta = pitch\ angle\ about\ y^B$$
$$\psi = yaw\ angle\ about\ z^B$$

$$p = roll\ rate\ about\ x^B$$
$$q = pitch\ rate\ about\ y^B$$
$$r = yaw\ rate\ about\ z^B$$

All translational velocities and accelerations will be indicated by their states, with the derivatives with respect to time denoted by the dot operator in the usual fashion e.g. $\dot{X^G}$ or $\ddot{Y^G}$.

Transformation matrices between the two reference frames will need to be defined, this can be achieved with Euler angles. Here, R^B_G denotes a rotation matrix from $\{G\}$ to $\{B\}$.

$$x^B = R_G^B X^G = R(\phi)R(\theta)R(\psi)X^G \tag{1}$$

$$R(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{C}(\phi) & \mathrm{S}(\phi) \\ 0 & -\mathrm{S}(\phi) & \mathrm{S}(\phi) \end{bmatrix}, \qquad R(\theta) = \begin{bmatrix} \mathrm{C}(\theta) & 0 & -\mathrm{S}(\theta) \\ 0 & 1 & 0 \\ \mathrm{S}(\theta) & 0 & \mathrm{C}(\theta) \end{bmatrix}$$
$$R(\psi) = \begin{bmatrix} \mathrm{C}(\psi) & \mathrm{S}(\psi) & 0 \\ -\mathrm{S}(\psi) & \mathrm{C}(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$R_G^B = \begin{bmatrix} \mathrm{C}(\psi)\,\mathrm{C}(\theta) & \mathrm{S}(\psi)\,\mathrm{C}(\theta) & -\mathrm{S}(\theta) \\ \mathrm{C}(\psi)\,\mathrm{S}(\phi)\,\mathrm{S}(\theta) - \mathrm{C}(\phi)\,\mathrm{S}(\psi) & \mathrm{S}(\phi)\,\mathrm{S}(\psi)\,\mathrm{S}(\theta) + \mathrm{C}(\phi)\,\mathrm{C}(\psi) & \mathrm{C}(\theta)\,\mathrm{S}(\phi) \\ \mathrm{C}(\phi)\,\mathrm{C}(\psi)\,\mathrm{S}(\theta) + \mathrm{S}(\phi)\,\mathrm{S}(\psi) & \mathrm{C}(\phi)\,\mathrm{S}(\psi)\,\mathrm{S}(\theta) - \mathrm{C}(\psi)\,\mathrm{S}(\phi) & \mathrm{C}(\phi)\,\mathrm{C}(\theta) \end{bmatrix} \tag{3}$$

The short-hands $S(x)$ and $C(x)$ are equivalent to $sin(x)$ and $cos(x)$ respectively.

The inverse transformation from $\{B\}$ to $\{G\}$ is then found:

$$\boldsymbol{X}^G = R_B^G \boldsymbol{x}^B = R(\phi)^T R(\theta)^T R(\psi)^T x^B = (R_G^B)^T \boldsymbol{x}^B \tag{4}$$

$$R_B^G = \begin{bmatrix} \mathrm{C}(\psi)\,\mathrm{C}(\theta) & \mathrm{C}(\psi)\,\mathrm{S}(\phi)\,\mathrm{S}(\theta) - \mathrm{C}(\phi)\,\mathrm{S}(\psi) & \mathrm{C}(\phi)\,\mathrm{C}(\psi)\,\mathrm{S}(\theta) + \mathrm{S}(\phi)\,\mathrm{S}(\psi) \\ \mathrm{S}(\psi)\,\mathrm{C}(\theta) & \mathrm{S}(\phi)\,\mathrm{S}(\psi)\,\mathrm{S}(\theta) + \mathrm{C}(\phi)\,\mathrm{C}(\psi) & \mathrm{C}(\phi)\,\mathrm{S}(\psi)\,\mathrm{S}(\theta) - \mathrm{C}(\psi)\,\mathrm{S}(\phi) \\ -\mathrm{S}(\theta) & \mathrm{C}(\theta)\,\mathrm{S}(\phi) & \mathrm{C}(\phi)\,\mathrm{C}(\theta) \end{bmatrix} \tag{5}$$

3 Vehicle Models

Kinematic and Dynamic models for both vehicle classes need to be produced in order to produce simulations. In the following two sub sections, the models used for the ASV and Quadcopter are explained.

3.1 Autonomous Surface Vehicle Model

First, a generalised model for a marine vehicle with 6 Degrees of Freedom (DoF) will be shown, following this, assumptions will be made to simplify this model into a 3DoF system. The model presented here is based on work in [28], by a research scientist at ISR who has continued to work on the MEDUSA to present.

It is convenient to express the forces and moments of marine vehicles in their body frame. In addition, due to their coupled behaviour in the chosen model, it is convenient to stack the rate of change of state variables, and forces & moments as follows:

$$\boldsymbol{\upsilon} = \begin{bmatrix} \dot{x}^B \\ \dot{y}^B \\ \dot{z}^B \\ p^B \\ q^B \\ r^B \end{bmatrix}, \text{ and } \boldsymbol{\tau^B} = \begin{bmatrix} F_x^B \\ F_y^B \\ F_z^B \\ \tau_\phi^B \\ \tau_\theta^B \\ \tau_\psi^B \end{bmatrix} \tag{6}$$

Here, F and τ represent reaction forces and moments respectively, with relevant directions indicated in subscripts. The rigid body can then be expressed by the following equation.

$$\boldsymbol{\tau}^B = M_{RB}\dot{\boldsymbol{\upsilon}} + \boldsymbol{C_{RB}}(\boldsymbol{\upsilon})\boldsymbol{\upsilon} \tag{7}$$

Where $M_{RB}$is a rigid body inertia matrix with dimension $(6,6)$ of the form:

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m & 0 & -mz_g & 0 & mx_g \\ 0 & 0 & m & my_g & -mx_g & - \\ 0 & -mz_g & my_g & I_x & -I_{xy} & -I_{xz} \\ mz_g & 0 & -mx_g & -I_{yx} & I_y & -I_{yz} \\ -my_g & mx_g & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix} \tag{8}$$

$\boldsymbol{C_{RB}}$ is a skew symmetric matrix with dimensions $(6,6)$ which takes the form:

$$\boldsymbol{C_{RB}}(\boldsymbol{\upsilon}) = \begin{bmatrix} 0 & -mr & mq & 0 & -Z_{\ddot{z}}\dot{z} & Y_{\ddot{y}}\dot{y} \\ mr & 0 & -mp & Z_{\ddot{z}}\dot{z} & 0 & -X_{\ddot{x}}\dot{x} \\ -mq & mp & 0 & -Y_{\ddot{y}}\dot{y} & X_{\ddot{x}}\dot{x} & 0 \\ 0 & -Z_{\ddot{z}}\dot{z} & Y_{\ddot{y}}\dot{y} & I_x & -I_{xy} & -I_{xz} \\ Z_{\ddot{z}}\dot{z} & 0 & -X_{\ddot{x}}\dot{x} & -I_{yx} & I_y & -I_{yz} \\ -Y_{\ddot{y}}\dot{y} & X_{\ddot{x}}\dot{x} & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix} \tag{9}$$

Equation 7 accounts for both Newton's laws ($\boldsymbol{M_{RB}}$) and Coriolis forces ($\boldsymbol{C_{RB}}$). The left hand side term $\boldsymbol{\tau}^B$ can be broken down into component forces and moments, which are:

$$\boldsymbol{\tau}^B = \tau_T^B + \tau_A^B + \tau_D^B + \tau_R^B + \tau_{dist}^B \tag{10}$$

From left to right these are the terms due to; control inputs, added mass, hydrodynamics (lift, drag etc.), restoring forces, and disturbances. Each of these terms will be looked at separately. First the thruster/actuator force is defined by:

$$\tau_T^B = T_{\text{sb}} + T_{\text{ps}} \tag{11}$$

Where T is thrust, and subscripts denote the starboard and port-side thrusters of the MEDUSA vehicle (Nautical terms used to match general literature). These two thrusters are capable of creating a force in the x_{ASV}^B direction, and a yawing moment in ψ.

The forces and moments due to added mass, represented by τ_A^B, are related to the kinetic energy imparted on the surrounding displaced fluid by the vehicle.

$$F_A^B = -M_A\dot{\boldsymbol{\upsilon}} - C_A(\boldsymbol{\upsilon})\boldsymbol{\upsilon} \tag{12}$$

$$M_A = \begin{bmatrix} F_{\dot{x}} & 0 & 0 & 0 & 0 & 0 \\ 0 & F_{\dot{y}} & 0 & 0 & 0 & 0 \\ 0 & 0 & F_{\dot{z}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \tau_{\phi,\dot{p}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \tau_{\theta,\dot{q}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \tau_{\psi,\dot{r}} \end{bmatrix}, \; C_A(\boldsymbol{\upsilon}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -F_{\dot{z}}\dot{z} & F_{\dot{y}}\dot{y} \\ 0 & 0 & 0 & F_{\dot{z}}\dot{z} & 0 & -F_{\dot{x}}\dot{x} \\ 0 & 0 & 0 & -F_{\dot{y}}\dot{y} & F_{\dot{x}}\dot{x} & 0 \\ 0 & -F_{\dot{z}}\dot{z} & 0 & 0 & -\tau_{\psi,\dot{r}}\dot{r} & \tau_{\theta,\dot{q}}\dot{q} \\ F_{\dot{z}}\dot{z} & 0 & -F_{\dot{x}}\dot{x} & \tau_{\psi,\dot{r}}\dot{r} & 0 & -\tau_{\phi,\dot{p}}\dot{p} \\ -F_{\dot{y}}\dot{y} & F_{\dot{x}}\dot{x} & 0 & -\tau_{\theta,\dot{q}}\dot{q} & \tau_{\phi,\dot{p}}\dot{p} & 0 \end{bmatrix}$$

Where the coefficient terms in the M_A and $C_A(\boldsymbol{\upsilon})$ matrices may be found experimentally, and take the general forms:

$$F_{\dot{s}} = \frac{\partial F_s^B}{\partial \dot{s}} \text{ and } \tau_{\omega,\dot{k}} = \frac{\partial \tau_\omega}{\partial \dot{k}} \tag{13}$$

Where s is some general translational direction, and $\dot{k}$ is the rotational acceleration in ω, i.e. $\dot{k} = \ddot{\omega}$.

The hydrodynamic terms expressed in τ_D^B are more difficult to quantify but are generally expressed using the positive definite *hydrodynamic damping* matrix D, which is a function of the vehicle's state variables:

$$F_D^B = -D(\boldsymbol{v})\boldsymbol{v} \tag{14}$$

The term for restoring forces which are related to buoyancy and gravity may be explained via Figure 4.

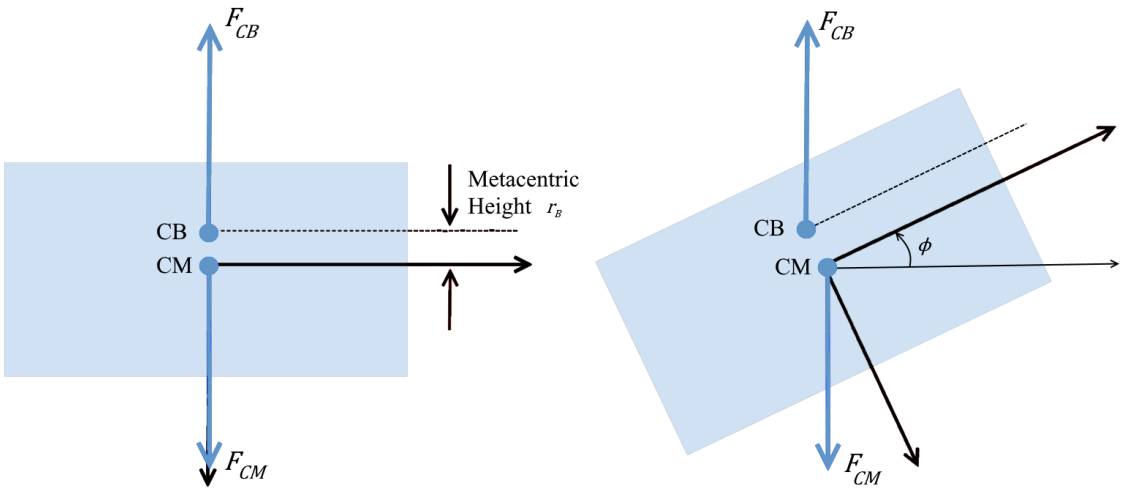


Figure 4- Restoring forces for a buoyant vehicle pitching upwards

It is seen that the centre of buoyancy (CB) and centre of mass (CM) are at different locations, and that a force acts on each (blue arrows). These forces are:

$$F_{CB} = \rho_{fluid} V_d g \tag{15}$$

$$F_{CM} = mg \tag{16}$$

Where each is defined in the global frame, ρ_{fluid} is fluid density ($kg\ m^{-3}$), V_d is volume of fluid displaced (m^3), $R(\phi)$ is an euler rotation matrix in pitch, the resultant force is found by:

$$F_R^B = R(\phi)^{-1} \begin{bmatrix} 0 \\ 0 \\ F_{CM} - F_{CB} \end{bmatrix} \tag{17}$$

The disturbance forces F_{dist}^{B} can be caused by any number of outside factors, including currents and waves, and so no equation exists unless in a heavily controlled environment.

3.1.1 Simplified Model

To reduce the complexity of the control system for the marine vehicles, we will reduce the kinematics to a 3 degree of freedom system $[x, y, \psi]$. This models a vehicle which will only operate in the horizontal plane of the water surface. The assumptions made are as follows:

- $\dot{Z}^{G} = 0$ i.e. zero velocity in Z^{G} axis
- $p = 0$ i.e. no pitching motion
- $q = 0$ i.e. no rolling motion

Where rates are assumed zero, one may also assume that the absolute values of the states are also zero. The simplified kinematic model is now:

$$\begin{aligned} \dot{X}^{G} &= \dot{x^{B}}\, cos(\psi) - \dot{y^{B}}\, sin(\psi) \\ \dot{Y}^{G} &= \dot{x^{B}}\, sin(\psi) + \dot{y^{B}}\, cos(\psi) \end{aligned} \tag{18}$$

With the starboard and port-side thrusters' thrusts denoted by T_{sb} and T_{ps}, and the length of the moment arm between them, l (m). The external force and moment caused by the actuators are found by:

$$\begin{aligned} \tau_{x}^{B} &= T_{sb} + T_{ps} \\ \tau_{\psi}^{B} &= \mathrm{l}\left(T_{\mathrm{ps}} - \mathrm{T_{sb}}\right) \end{aligned} \tag{19}$$

When accounting for a constant ocean current $v_c = [\dot{x}_c, \dot{y}_c, \dot{z}_c]^T$, the dynamic equations take the form:

$$\begin{aligned} m_{\dot{x}}\ddot{x} - m_{\dot{y}}(\dot{y}_r + \dot{y}_c)r + d_{\dot{x}}\dot{x}_r &= \tau_{x}^{B} \\ m_{\dot{y}}\ddot{y} - m_{\dot{x}}(\dot{x}_r + \dot{x}_c)r + d_{\dot{y}}\dot{y}_r &= 0 \\ \mathrm{m_r}\dot{\mathrm{r}} - \mathrm{m_{\dot{x}\dot{y}}}(\dot{x}_{\mathrm{r}} + \dot{x}_{\mathrm{c}})(\dot{y}_{\mathrm{r}} + \dot{y}_{\mathrm{c}}) + \mathrm{d_r r} &= \tau_{\psi}^{B} \end{aligned} \tag{20}$$

The terms $[m_u, m_v, m_r, m_{uv}]$ are mass and hydrodynamic added mass terms, and $[d_u, d_v, d_r]$ represent hydrodynamic damping effects. The expressions for these have been left out for conciseness, but they can be considered constant coefficients.

3.1.2 Thruster Model

An accurate thruster model was available for the ASV, this was previously produced using experimental data, this allows for accurate modelling and simulation. For each of the thrusters the following open loop system is applied:

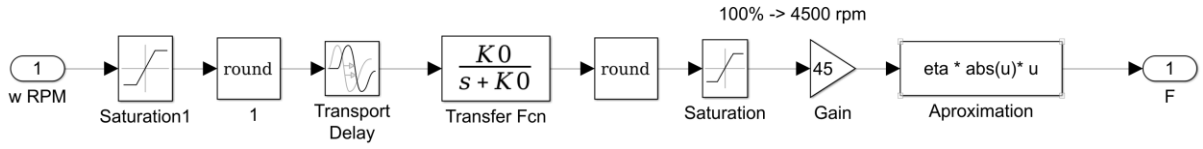


Figure 5 - Simulink model of ASV thruster [28]

The input is a Revolutions per Minute (RPM) value as a percentage provided by the motor power controller. The first stage is a saturation value which is rounded to model internal quantization. There is a short time delay of $0.346s$, and the transfer function takes the form.

$$TF = \frac{K0}{s + K0} \tag{21}$$

Where $K0 = 7.2115$. Maximum RPM of the thruster is $4500rpm$. The conversion from RPM to thrust is achieved with.

$$F_T^B = \frac{4}{1500^2}|RPM| \cdot RPM \tag{22}$$

3.2 Quadcopter Model

The Quadcopter consists of a rigid frame and four rotors which produce the forces and torques which act to manoeuvre the vehicle in the air.

A schematic diagram of a Quadcopter was seen in Figure 3. Each rotor is placed at a distance d in the xy^B plane from the centre of mass. It is assumed that the blades are equally spaced, i.e. perpendicular to one another. Adjacent rotors rotate in the opposite direction from one another, to counteract the resultant moments produced.

Much of the following is based on the models detailed in [16] [31], and the simulation example in [32].

3.2.1 Motor Equations

Typically, out-runner brushless motors are used in Quadcopters, for their high torque, which allows for faster rotational accelerations. Power is provided by the on-board batteries, and

motors are controlled with speed controllers. First, we calculate the power produced by the motor based on electrical properties [32].

$$\tau_i = K_t(I - I_0) \tag{23}$$

$$V = IR_m + K_v\omega \tag{24}$$

$$P_{m,i} = IV = \frac{(\tau_i + K_t I_0)(R_t I_0 R_m + \tau_i R_m + K_t K_v \omega)}{{K_t}^2} \tag{25}$$

τ_i is torque of motor i in (Nm) and, ω is rotational speed of the same motor $(rad\ s^{-1})$, K_t is the torque proportionality constant $(Nm\ A^{-1})$, I is input current (A), I_0 is current with no load (A), V is voltage across the motor (V), R_m is motor resistance (Ω), K_v is back EMF coefficient, P_m is motor power (W). Assuming that $R_m \approx I_0 \approx 0$

$$P_{m,i} = \frac{K_v \tau_i \omega_i}{K_t} \tag{26}$$

The motor power in hover can be calculated by looking at the induced velocity and thrust. T_i (N) is the motor thrust, v_h $(m\ s^{-1})$ is the induced velocity in hover, and P_h (W) is the power required for hover.

$$P_h = T_i v_h \tag{27}$$

$$T_i = 2\rho_{air} A {v_h}^2 \tag{28}$$

$$P_h = \frac{{T_i}^{3/2}}{\sqrt{2\rho_{air}A}} \tag{29}$$

The symbols take their usual meaning, ρ_{air} is air density $(kg\ m^{-3})$, A (m^2) is rotor swept area. Torque can be related to thrust via the coefficient K_T. We then replace terms in the equation for $P_{m,i}$.

$$\tau_i = K_\tau T_i \tag{30}$$

$$P_{m,i} = \frac{K_v \tau \omega}{K_t} = \frac{K_v K_\tau T_i \omega}{K_t} \tag{31}$$

Assuming translational velocities are small, these equations may be extended to the general case, by equating $P_h = P_m$.

$$\frac{K_v K_\tau T \omega}{K_t} = \frac{T_i^{3/2}}{\sqrt{2\rho_{air} A}} \tag{32}$$

$$T = \left[\left(\frac{K_v K_T \sqrt{2\rho_{air} A}}{K_t}\right)\omega\right]^2 \tag{33}$$

$$T = K_T \omega^2 \tag{34}$$

3.2.2 Dynamics

Translational and rotational dynamics will be considered separately, first we will consider translational dynamics as a result of vehicle forces. The motor thrusts and other forces can be used to manoeuvre the vehicle translationally in $\{G\}$, and can be related using the following dynamics equations, m is the vehicle mass (kg) all forces are (N):

$$m\begin{bmatrix}\ddot{X}^G \\ \ddot{Y}^G \\ \ddot{Z}^G\end{bmatrix} = F_g^G - F_T^G - F_d^G \tag{35}$$

F_g^G represents gravitational force, note it acts in the positive (downwards) Z^G direction.

$$F_g^G = \begin{bmatrix}0 \\ 0 \\ mg\end{bmatrix} \tag{36}$$

F_T^G is the sum total of all rotor thrusts T_i, transformed into the global frame.

$$F_T^G = R_B^G F_T^B = R_B^G \begin{bmatrix}0 \\ 0 \\ K_T \sum_{i=1}^{4} {\omega_i}^2\end{bmatrix} \tag{37}$$

Drag force F_d^G is defined by drag coefficients in each direction in $\{B\}$, these are $[K_{dx}, K_{dy}, K_{dz}]$. This is rotated to $\{G\}$ to confirm with the other force components.

$$F_d = R_B^G \begin{bmatrix}K_{dx} & 0 & 0 \\ 0 & K_{dy} & 0 \\ 0 & 0 & K_{dz}\end{bmatrix}\begin{bmatrix}\dot{X}^G \\ \dot{Y}^G \\ \dot{Z}^G\end{bmatrix} \tag{38}$$

To relate the time derivatives of the Euler angles to the rotation rates we can use the following expression:

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}=\begin{bmatrix}1 & \sin\phi\tan\theta & \cos\phi\\ 0 & \cos\phi & -\sin\phi\\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta}\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix} \tag{39}$$

Equation (39) has a singularity at $\theta = \pm 90°$, some cases (e.g. aerial acrobatics) have been investigated in other literature when this singularity cannot be ignored, but in normal operation it will not become an issue. Limits have been imposed in the software to ensure this situation is not reached.

It is assumed in this case that the vehicle is symmetric about x^B and y^B and so the Quadcopter moment of inertia matrix J_b is also symmetric. The torques and angular velocities are related by the following differential equation in ω, angular velocities of the vehicle in the body frame.

$$J_b\dot{\omega} = \tau_m - \tau_g - (\omega \times J_b\omega) \tag{40}$$

Where $J_b(\text{kg m}^2)$ is the moment of inertia of the vehicle, τ_m is the torques produced by the motors (Nm), τ_g are the torques due to gyroscopic effects (Nm), the final term accounts for the coupling of the gyroscopic effects and these rotation of the vehicle. By defining the left-hand side of the differential equation in terms of the state variables, we find that:

$$J_b=\begin{bmatrix}J_x & 0 & 0\\ 0 & J_y & 0\\ 0 & 0 & J_z\end{bmatrix}\dot{\omega}=\begin{bmatrix}\ddot{\phi}\\ \ddot{\theta}\\ \ddot{\psi}\end{bmatrix}(\omega \times J_b\omega)=\begin{bmatrix}\theta\psi(J_z - J_y)\\ \psi\phi(J_x - J_z)\\ \theta\phi(J_y - J_z)\end{bmatrix} \tag{41}$$

The motor's moment of inertia is $J_m(\text{kgm}^2)$, and τ_ψ is the torque along the relevant body axis (Nm).

$$\begin{gathered}J_m\dot{\omega} = \tau_m - \tau_\psi\\ \text{At hover, } \dot{\omega} = 0 \text{ so } \tau_m = \tau_\psi\end{gathered} \tag{42}$$

The torque due to drag (τ_D) is defined in the following equations. Note, that it is assumed the rotors all have a constant speed ω_i, and therefore the torque about Z^G is equal to τ_D.

$$\tau_D = \frac{1}{2}R_\rho C_D A(\omega R)^2 = K_d\omega^2 \tag{43}$$

$$\tau_\psi = \tau_D = (-1)^{i+1}K_d{\omega_i}^2 = K_d({\omega_1}^2 - {\omega_2}^2 + {\omega_3}^2 - {\omega_4}^2) \tag{44}$$

The $(-1)^{i+1}$ term in the above equation is a mathematical convenience which represents the counter-rotation of adjacent rotors discussed previously. It can be seen in the result that $\omega_{1,3}$ rotate in the positive direction, and $\omega_{2,4}$ in the negative direction.

The rotors placed on each of the axes can be chosen somewhat arbitrarily, here $\omega_{1,3}$ are placed on the body pitch axis, $\omega_{2,4}$ on roll. The term d (m) is the length of the moment arm of the rotors. The resulting torque matrix due to the motors is defined by:

$$\tau_m = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} dK_T({\omega_4}^2 - {\omega_2}^2) \\ dK_T({\omega_1}^2 - {\omega_3}^2) \\ K_d({\omega_1}^2 - {\omega_2}^2 + {\omega_3}^2 - {\omega_4}^2) \end{bmatrix} \tag{45}$$

Gyroscopic effects can be defined with the following:

$$\tau_g = \omega \times G_Z \sum_{i=1}^{4} J_r \omega_i \tag{46}$$

$$\tau_g = \begin{bmatrix} \dot{\theta} \\ -\dot{\phi} \\ 0 \end{bmatrix} J_r \sum_{i=1}^{4} (-1)^{i+1} \omega_i = \begin{bmatrix} J_r \dot{\theta}(\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ -J_r \dot{\phi}(\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ 0 \end{bmatrix} \tag{47}$$

3.2.3 Equations of Motion

To find the Quadcopter's global translational acceleration we can use the following system of equations. We choose the global frame in this case as it allows for coordination between vehicles, and additionally available sensors (GPS etc.) measure translation in a global frame.

$$\begin{bmatrix} \dot{X}^G \\ \dot{Y}^G \\ \dot{Z}^G \end{bmatrix} = R_B^G \begin{bmatrix} \dot{x}^B \\ \dot{y}^B \\ \dot{z}^B \end{bmatrix} \tag{48}$$

$$\begin{bmatrix} \ddot{X}^G \\ \ddot{Y}^G \\ \ddot{Z}^G \end{bmatrix} = \begin{bmatrix} \frac{1}{m}\left(-(\cos(\phi)\cos(\psi)\sin(\theta) + \sin(\phi)\sin(\psi))F_T^B - K_{dx}\dot{X^G}\right) \\ \frac{1}{m}\left(-(\cos(\phi)\cos(\psi)\sin(\theta) + \sin(\phi)\sin(\psi))F_T^B - K_{dy}\dot{Y^G}\right) \\ \frac{1}{m}\left(-\cos(\phi)\cos(\theta)\,F_T^B - K_{dz}\dot{Z^G}\right) + g \end{bmatrix} \tag{49}$$

Data about angular rotation in Quadcopters is typically measured with an Inertial Measurement Unit (IMU) [33], which measures these rotations in the body frame with a magnetometer, gyroscope and accelerometer.

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}=\begin{bmatrix}1 & \sin\phi\tan\theta & \cos\phi\\ 0 & \cos\phi & -\sin\phi\\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta}\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix} \tag{50}$$

$$\begin{bmatrix}\ddot{\phi}\\ \ddot{\theta}\\ \ddot{\psi}\end{bmatrix}=\begin{bmatrix}\dfrac{1}{J_x}\Big(\big(J_y-J_z\big)qr-J_rq(\omega_1-\omega_2+\omega_3-\omega_4)+rK_T({\omega_4}^2-{\omega_2}^2)\Big)\\ \dfrac{1}{J_y}\big((J_z-J_x)pr-J_rp(\omega_1-\omega_2+\omega_3-\omega_4)+rK_T({\omega_1}^2-{\omega_3}^2)\big)\\ \dfrac{1}{J_z}\Big(\big(J_x-J_y\big)pq+K_d(\omega_1-\omega_2+\omega_3-\omega_4)\Big)\end{bmatrix} \tag{51}$$

4 Vehicle Control and Simulation

Now that the vehicle models are defined, it important to be able to directly control their states. This will allow for higher level mission control discussed in the chapters on Path Following and Cooperation.

Throughout, a PID (Proportional, Integral, Derivative) control method will be used, as its behaviour is well understood and documented. This method generates an error term, which undergoes each of the relevant operations, which are also subjected to unique gain values, the control term is the sum of these values.

For some arbitrary variable σ, at time t, with a desired value of σ_d. Error is defined by:

$$\varepsilon_\sigma(t) = \sigma_d(t) - \sigma(t) \tag{52}$$

The control term may be found by summing each of the PID terms. $K_{P,I,D}$ are all gain values chosen for the controller:

$$\begin{gathered} u = P_\sigma + I_\sigma + D_\sigma \\ \text{Where,} \quad P_\sigma = K_P \varepsilon_\sigma(t) \quad I_\sigma = K_I \int_0^t \varepsilon_\sigma(t) \quad \text{and} \quad D_\sigma = K_D \dot{\varepsilon_\sigma}(t) \end{gathered} \tag{53}$$

For brevity in the following sections, the controller equations will not be stated explicitly unless significantly altered, for PID controllers only the gains will be displayed, as seen in Table 1. Column one is the error variable, and the last column is the resulting control term.

Table 1 - Gain Table Layout Template

	K_P	K_I	K_D	
σ	1.00	1.00	1.00	u

A simple methodology was typically used when tuning the gain values.

- Increase K_P until a steady state oscillation is observed
- Increase K_I until the oscillation is centred on $\varepsilon_\sigma = 0$
- Increase K_D until the oscillation is suitably damped
- Repeat until response is satisfactory

When required, methods other than the one above were used, but this formed the basis of tuning for all controllers.

4.1 ASV Control

The full controller architecture is shown in Figure 6, a larger version is shown in Appendix C for improved readability.

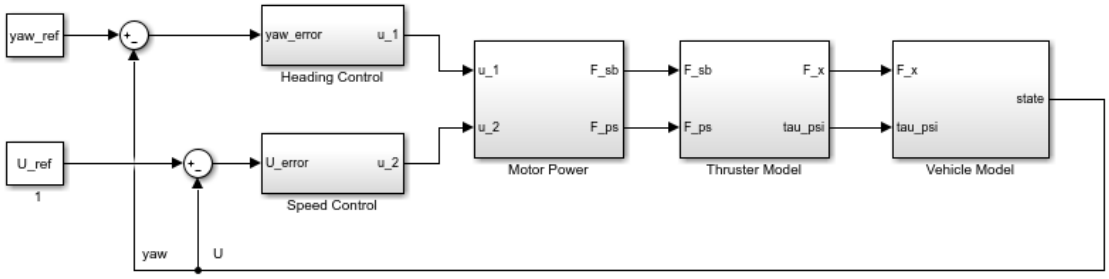


Figure 6 - Autonomous Surface Vehicle state controller architecture

4.1.1 Control Inputs

Two control inputs are used in the ASV model, representing the two possible forces and moments the thruster combination is able to produce.

- u_1: Difference in thrust between motors resulting in change in yaw
- u_2: Total thrust

4.1.2 Heading Control

A PID heading controller provides the control term u_1 to the thrusters.

Table 2 - ASV Heading Controller Gains

	K_P	K_I	K_D	
ψ	750	0.10	-100	u_1

High gain values were required to produce reasonable response times, these were kept as low as was possible to prevent the yaw control from completely over-riding the speed control in the motor power control. Still, some interference between these controllers is experienced.

4.1.3 Speed Control

In a typical manoeuvre there will be a constant *nominal* velocity reference, so that the vehicle progresses at a continuous rate. Introduced later will be correction velocity terms, which will affect this both by increasing and decreasing the speed reference U_{ref}.

For speed control we need to calculate the absolute velocity in the local frame.

$$U_{abs} = \sqrt{(\dot{X}^G)^2 + (\dot{Y}^G)^2} \tag{54}$$

We then define the error term as $\varepsilon_{U_{abs}}$. This is only controlled with a PI controller as negligible oscillatory motion was apparent.

Table 3 - ASV Speed Controller Gains

	K_P	K_I	K_D	
U_{abs}	60	2.00	0	u_2

The MEDUSA vehicle's maximum water velocity is 1m/s, which is implemented as a saturation value in the thruster model, and with a saturation in the speed controller.

$$|u_2| \leq 60 \tag{55}$$

Commanding the vehicle to go any faster than $1m\,s^{-1}$ will have no effect, therefore all simulations should have a limit imposed on the speed reference.

4.2 ASV Simulations & Results

4.2.1 Heading Controller

The response of the ASV's heading controller to the stepped input seen in Figure 7 is satisfactory, with a sufficiently short rise time of $10s$, and stable steady state. When stepping ψ_{ref} up or down by the same value, the ASV behaves similarly due to symmetry in x^B.

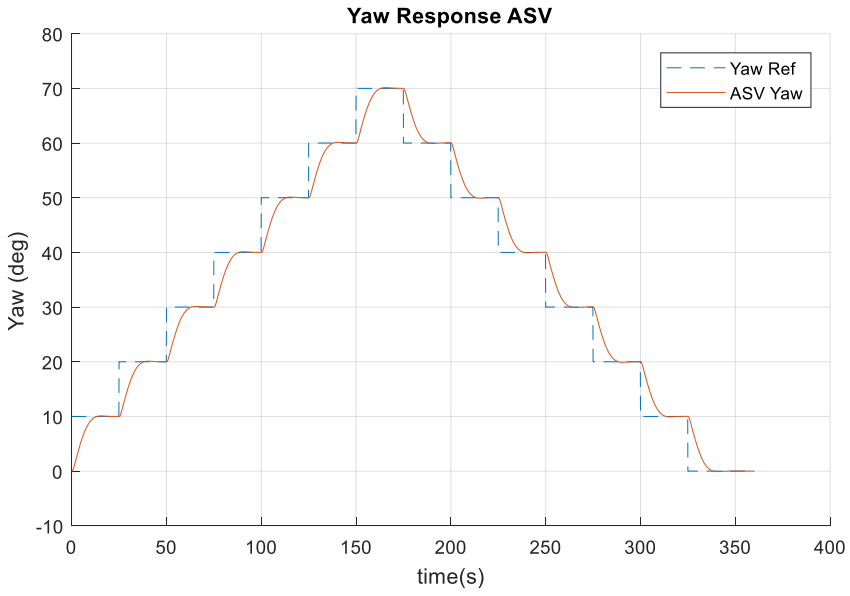


Figure 7 - Yaw response of ASV subjected to a stepped input with increments of 10° per 25 seconds

4.2.2 Speed Controller

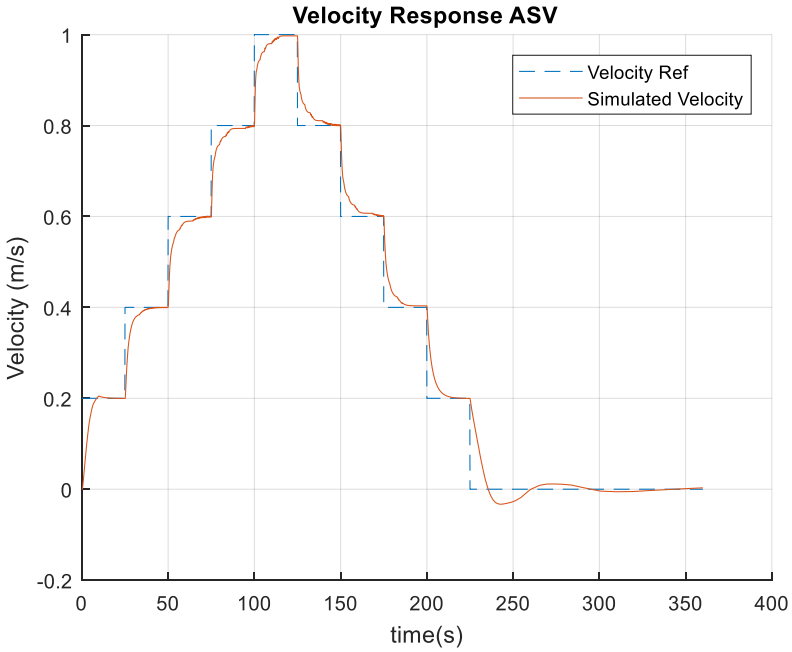


Figure 8 - Velocity response of ASV subjected to a stepped input with increments of 0.2m/s each 25 seconds

Results of tests on the speed controller are seen in Figure 8, we see acceptable tracking of the reference input, with a high initial response rate, and a rise time of $8s$. When decelerating to zero there is some overshoot and the vehicle reverses at a low speed, this is due to the need to counteract all the forwards momentum. The transient response, however, is sufficiently damped.

4.3 Quadcopter Control

The controller architecture for the Quadcopter inner loop is shown in Figure 9, a larger version is shown in Appendix B for improved readability.

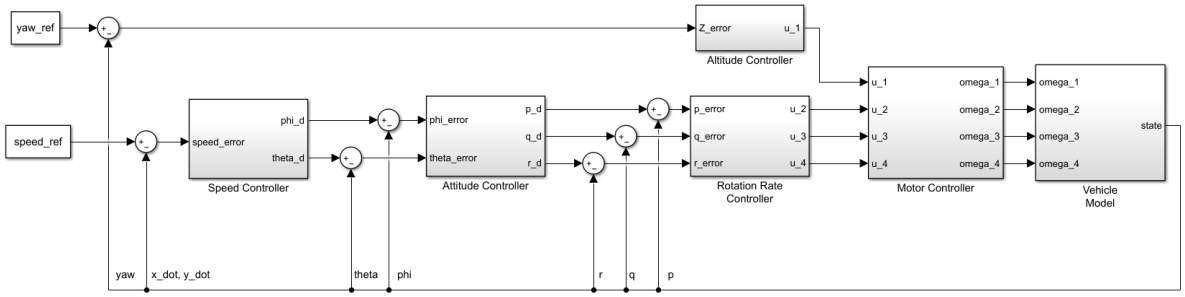


Figure 9 - Quadcopter state controller architecture

The following sections work their way through this controller, explaining the methods adopted.

4.3.1 Control Inputs

The vehicle can only be controlled by the speed of its four rotors, all oriented in one direction, whereas it is a six degree of freedom system, this is an *underactuated* system [34]. Four direct control inputs were chosen:

- u_1: Thrust of all four rotors
- u_2: Difference in thrust between motors on the x^B axis, resulting in change in roll
- u_3: Difference in thrust between motors on the y^B axis, resulting in change in pitch
- u_4: Difference in torque between motors rotating clockwise and counter clockwise, resulting in a moment in the z^B axis.

These control terms may be related to the forces and torques in $\{B\}$ with the following system of equations.

$$\begin{bmatrix} F_T^B \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} K_T & K_T & K_T & K_T \\ 0 & -rK_T & 0 & rK_T \\ rK_T & 0 & -rK_T & 0 \\ K_d & -K_d & K_d & -K_d \end{bmatrix} \begin{bmatrix} {\omega_1}^2 \\ {\omega_2}^2 \\ {\omega_3}^2 \\ {\omega_4}^2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \tag{56}$$

These direct control inputs in place in the equations of motion take the following form;

$$\begin{bmatrix} \ddot{X}^G \\ \ddot{Y}^G \\ \ddot{Z}^G \end{bmatrix} = \begin{bmatrix} \frac{1}{m}\left(-(\cos(\phi)\cos(\psi)\sin(\theta) + \sin(\phi)\sin(\psi))u_1 - K_{dx}\dot{X^G}\right) \\ \frac{1}{m}\left(-(\cos(\phi)\cos(\psi)\sin(\theta) + \sin(\phi)\sin(\psi))u_1 - K_{dy}\dot{Y^G}\right) \\ \frac{1}{m}\left(-\cos(\phi)\cos(\theta)\,u_1 - K_{dz}\dot{Z^G}\right) + g \end{bmatrix} \tag{57}$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{1}{J_x}\left((J_y - J_z)qr - J_r q(\omega_1 - \omega_2 + \omega_3 - \omega_4) + ru_2\right) \\ \frac{1}{J_y}\left((J_z - J_x)pr - J_r p(\omega_1 - \omega_2 + \omega_3 - \omega_4) + ru_3\right) \\ \frac{1}{J_z}\left((J_x - J_y)pq + u_4\right) \end{bmatrix} \tag{58}$$

4.3.2 Speed Controller

A controller was produced which commands the vehicle to move at set speeds in each of the X^G and Y^G directions. This is achieved by providing desired orientations in roll and pitch which are then fed into an attitude controller.

In contrast with the ASV, the apparent yaw angle of a Quadcopter is not necessarily indicative of its heading in space, as it has no *"fore"* or *"aft"* direction in its body structure, and 3D

velocities may be achieved in a range of orientations. The reference heading angle ψ_{ref} therefore may be used to parameterize the global speed reference into components by simple means.

$$\dot{X}_{ref}^{G} = U_{ref} \cos \psi_{ref}$$
$$\dot{Y}_{ref}^{G} = U_{ref} \sin \psi_{ref} \quad (59)$$

The reference velocities are transformed into $\{B\}$ $[\dot{x}_{ref}, \dot{y}_{ref}]$ to provide useful information related to the vehicle's orientation. The gain values used in the speed controller are detailed in Table 4. Due to the symmetry of the vehicle and the desire for equivalent performance in all directions the gains have the same absolute values, although it should be noted that $\dot{x}^B$ and θ_d are inversely related.

Table 4 – Quadcopter Speed Controller Gains

	K_P	K_I	K_D	
$\dot{x}^B$	-2.5	-0.50	0.10	θ_d
$\dot{y}^B$	2.5	0.50	-0.10	ϕ_d

4.3.3 Position Controller

A position controller can be created using a similar methodology to that of the speed controller in Section 4.3.2 by replacing the speed error terms with a position error in the same directions.

This controller was initially used for a waypoint based path follower, not used in the final version of the system. There are still use cases for this controller, when we want the vehicle to hold its position at the beginning or end of a path following mission, this controller can be engaged and the Quadcopter will head directly to a point in space and hold its position.

Table 5 - – Quadcopter Position Controller Gains

	K_P	K_I	K_D	
x^B	-0.30	-0.22	0.35	θ_d
y^B	0.30	0.22	-0.35	ϕ_d

4.3.4 Altitude Controller

The altitude controller was specified separately from speed and position control as it only directly affects the control input in thrust u_1, and not the required orientation.

$$u_1 = \frac{-(P_z + I_Z + D_Z) + mg}{\cos\theta \cos\phi} \tag{60}$$

Table 6 – Quadcopter Altitude Controller Gains

	K_P	K_I	K_D	
Z^G	5.88	0.00	-5.05	u_1

It should be clear that a high total thrust, controlled by u_1, will lead to more aggressive manoeuvres in all directions as this is the only source of translational actuation. When changing altitude, there will be disturbances in motion in the horizontal plane.

4.3.5 Attitude and Rotation Rate Controllers

The attitude errors are used to find the desired angular rates $[p, q, r]$. The gain values are shown in Table 7, it was found that these controllers did not require Integral or Derivative terms in their controllers in simulation. Perhaps in implementation this will need to be reassessed.

Table 7 – Quadcopter Attitude Controller Gains

	K_P	K_I	K_D	
ϕ	4.25	0.00	0.00	p_d
θ	4.25	0.00	0.00	q_d
ψ	9.80	0.00	0.00	r_d

A PID controller calculates the remaining control terms $u_{2,3,4}$ based on these desired angular rates. Gains in all directions are equal.

Table 8 – Quadcopter Rotation Rate Controller Gains

	K_P	K_I	K_D	
p	2.70	1.00	-0.01	u_2
q	2.70	1.00	-0.01	u_3
r	2.70	1.00	-0.01	u_4

4.3.6 Motor Control

We must also relate the control inputs to real motor speed, this can be achieved with a simple set of equations, inverse to Equation 56, for the control inputs.

$$\begin{bmatrix} {\omega_1}^2 \\ {\omega_2}^2 \\ {\omega_3}^2 \\ {\omega_4}^2 \end{bmatrix} = \begin{bmatrix} K_T & K_T & K_T & K_T \\ 0 & -rK_T & 0 & rK_T \\ rK_T & 0 & -rK_T & 0 \\ K_d & -K_d & K_d & -K_d \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (61)$$

All motors have a maximum operating speed, and so a saturation limit can be imposed at this stage. An accurate motor model obtained experimentally was available for the ASV but not for the Quadcopter. This limit, when implemented at the control level, helps prevent burn-out of expensive motors in operation, and also helps generate more accurate simulations.

4.4 Quadcopter Simulations & Results

All simulations were carried out using software developed in *Matlab*, note that all source code has been made available, details in Appendix A. Thorough tests were carried out to prove the controllers' performance before applying them to path following missions. The Quadcopter responds quickly to changes in control inputs, providing a manoeuvrable vehicle.

4.4.1 Speed Controller

To test the basic performance of the controllers acting upon the Quadcopter model, a stepped input in U_{ref} was provided. The value ψ_{ref} was held constant in the simulation shown in Figure 10.

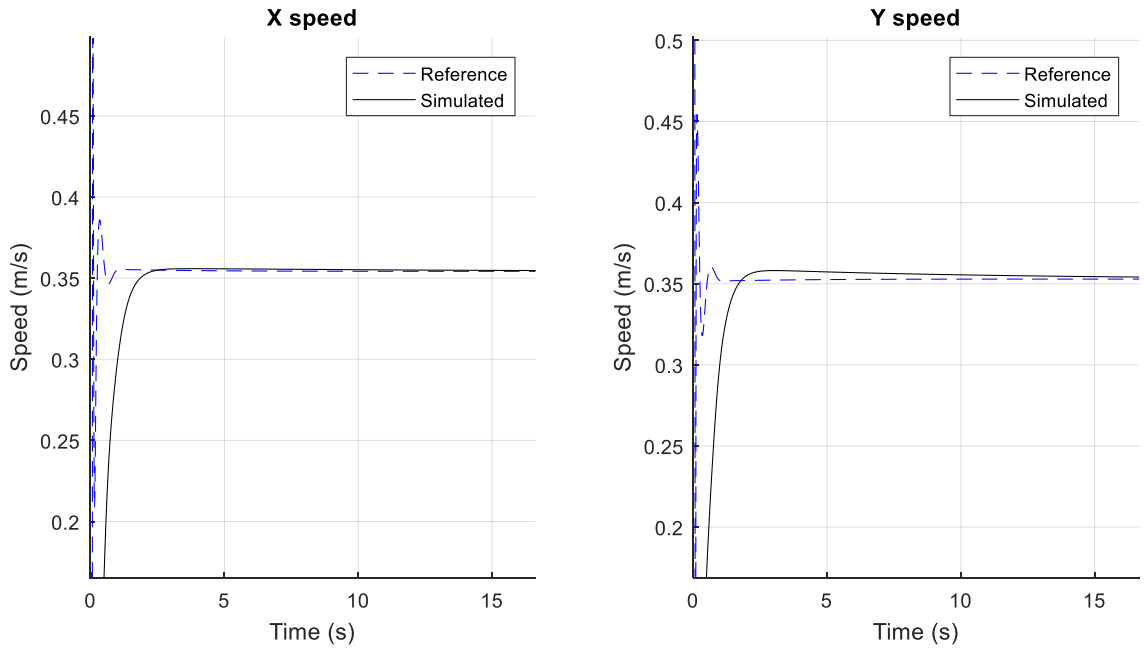


Figure 10 - Response of speed controller to constant desired velocity starting from stationary

4.4.2 Position Controller

The results in Figure 11 show the response to stepped inputs in position. At $t = 5s$, both directions in the horizontal plane are provided with a stepped input $= 1m$, and respond in a steady way in $4s$.

Then at $t = 10s$ another step is provided, only in Y^G, response in this direction is similar to before, a small disturbance is seen in X^G, but this does not demonstrate any unstable behaviour, and an accurate steady state is reached.

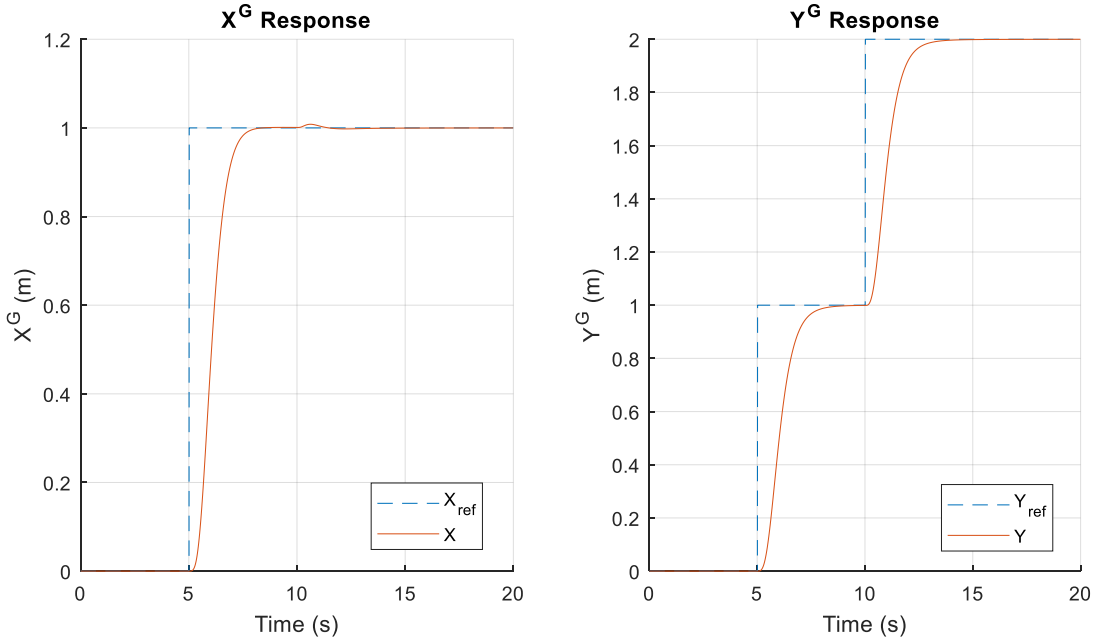


Figure 11 - Response of position controller to stepped inputs

4.4.3 Altitude Controller

Figure 12 shows the response of the altitude controller. Remember that due to the NED coordinate system, the negative values seen in the y axis of this figure are in fact above sea level.

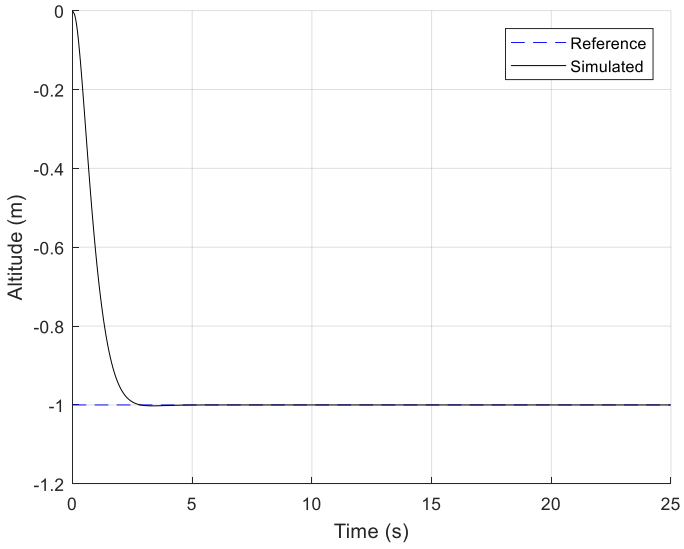


Figure 12- Response of altitude controller to a constant desired altitude -1m starting from origin

4.4.4 Attitude and Rotation Rate Controllers

The simulation shown in Figure 13 shows the response of the Quadcopter to three stepped inputs in each of roll, pitch and yaw. It shows the inter-play of the three orientation directions, as well as response characteristics of each.

At $t = 5s$ a stepped input with value $\phi_{ref} = \pi/4$ is provided and a rapid response is seen, with negligible transient oscillations. Note that responses are similar in all directions when the other two orientation angles are held at zero value.

At $t = 10s$ a stepped input with value $\theta_{ref} = \pi/4$ is provided. This causes disturbances in both roll and yaw, due to the coupling of motion, particularly between roll and pitch. A transient is seen in pitch, but is highly damped, with a decay time of $2s$.

At $t = 15s$ a stepped input with value $\psi_{ref} = \pi/4$ is provided. This causes disturbances in all of the rotational axes, similar to seen at $t = 10s$, but with greater amplitudes due to both the roll and pitch values being held well away from their stable equilibrium positions.

During the majority of the simulations within this report, a fixed desired yaw value $\psi_{ref} = 45°$ was provided as it may be arbitrarily set, and any direction of translational motion still achieved.

In missions with sensors mounted on-board, ψ_{ref} may be altered dynamically to suit requirements. For example; a sensor could be commanded to point towards a fixed point in space while the Quadcopter orbits around it.

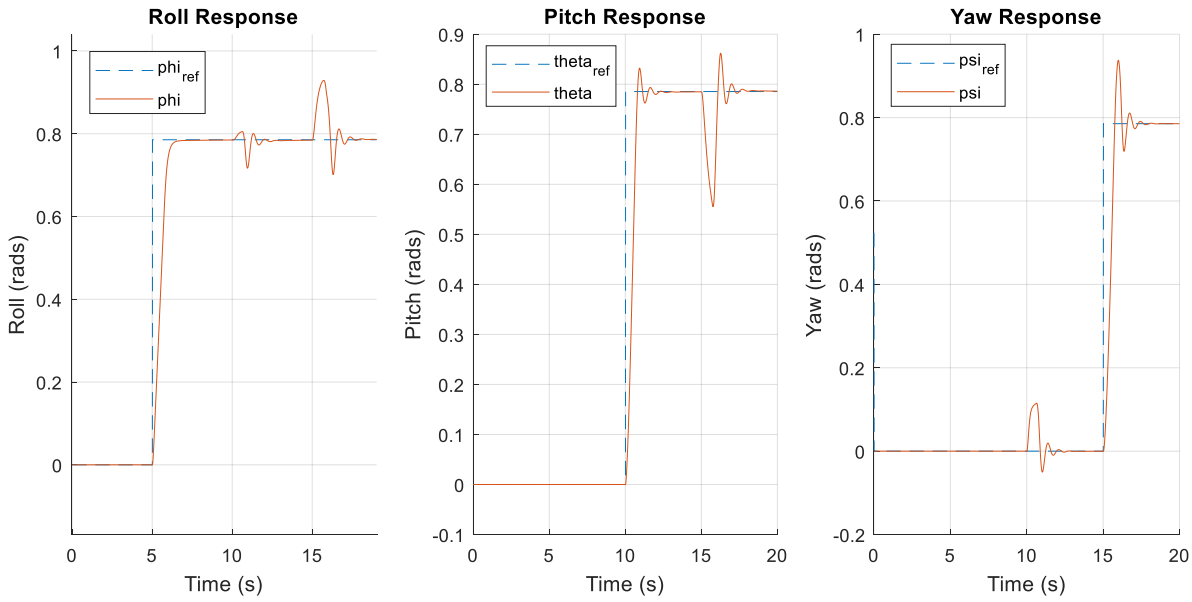


Figure 13 - Response of attitude controller to step inputs

The rotation rate reference and response for the same simulation is shown in Figure 14. One can see the rapid response this controller has to complex, high frequency signals in all three directions simultaneously. The rapid response is due to the control inputs directly actuating the rate commands, to great advantage of the overall manoeuvrability of the vehicle.

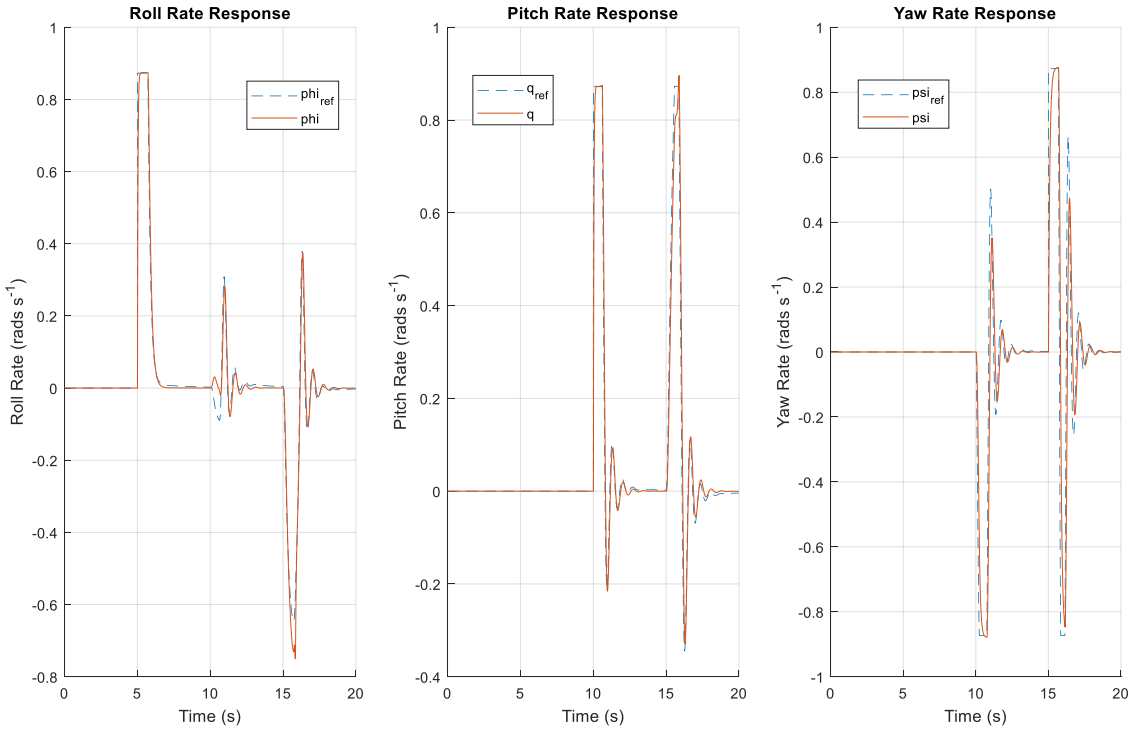


Figure 14 - Response of rotation rate controller to step inputs in orientation (from previous figure)

4.5 Vehicle Comparison

Response of the Quadcopter to all control inputs is faster than with the ASV, this led to questioning the simulated performance of the ASV, and whether it was suitable for the task. By comparison with previous simulations and practical tests of the MEDUSA vehicle, the results presented for the ASV are valid. There are a number of reasons thought to impact on its slower response:

Firstly, the ASV model includes the effects of moving through water, which is clearly a more resistive medium to motion than air. The ASV thruster model also includes the real performance and limitations of the vehicle's actuators, shown in Section 3.1.2. There was no specific thruster model available for the Quadcopter, and so the overall response does not include these un-modelled dynamics, which will likely introduce time delays.

Although the vehicles and their respective controllers have different responses over time, they both accurately track reference inputs, and are deemed to be suitable for path following missions.

5 Path Following

Path following, as explained in Section 1.3.3, is a means of commanding an autonomous vehicle to follow a pre-specified route. This section develops the control algorithm used to achieve consistent path following in straight lines and arcs, which may be built up to follow any general path.

Results of simulations of the algorithm developed are shown and discussed for both vehicle types. Simulations are carried out using simple path types and lawnmower paths consisting of multiple components joined together consecutively. Limitations of the vehicles in path following are also highlighted.

A simplified controller architecture is seen in Figure 15, where the inner loop represents the vehicle model and state controllers seen in the previous chapters.

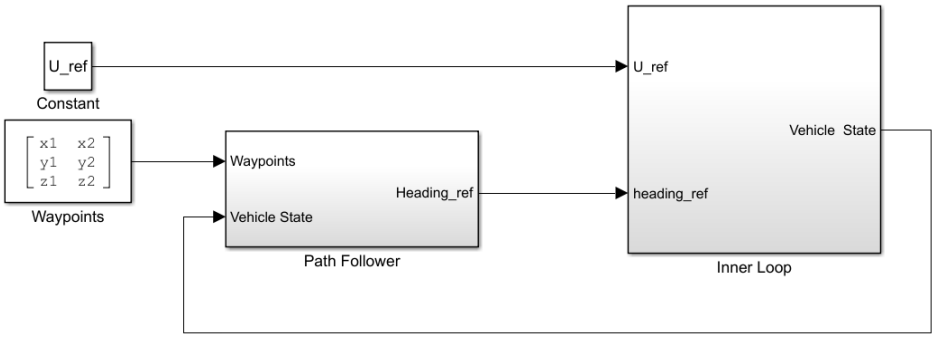


Figure 15 - Simplified controller architecture for path following vehicle

5.1 Methodology

The most fundamental component of path following for any arbitrary path is, that of following a straight line at a constant velocity, seen in *Figure 16.*

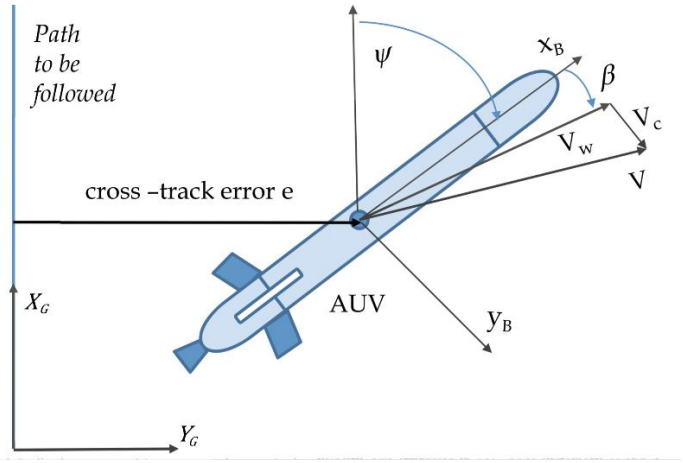


Figure 16 - Cross track error for straight line following [20] for a general vehicle

Where e is the cross-track error, β is sideslip angle, and ψ is the current yaw angle. The overall vehicle velocity V is defined by summing velocity with respect to the water V_w, and disturbance velocity V_c. For a fixed desired velocity $V = U_{ref}$, the desired heading is given by;

$$\psi_d = \beta + \sin^{-1}(\psi_c) \tag{62}$$

Where the control term ψ_c is found using:

$$\psi_c = -\frac{K_{P,e}e}{U_{ref}} - \frac{K_{I,e}}{U_{ref}} \int_0^t e(\tau)d\tau \tag{63}$$

It must be specified that the absolute value $|\psi_c| \leq 1$. Gain values $K_{P,e}$ and $K_{I,e}$ may be tuned for natural frequency and damping factor of this 2nd order system.

Cross-track error e is defined as the distance to the nearest point on the path. The method of calculating this property varies slightly between path types

Straight lines may be defined by two points, and for a spatial path this is typically the start $[X_1, Y_1]$ and the end $[X_2, Y_2]$. The angle this line makes with the X^G is the path angle, found by:

$$\psi_{path} = atan2\big((Y_2 - Y_1), (X_2 - X_1)\big) \tag{64}$$

Typical test missions include 180° arcs defined by a midpoint $[x_M, y_M]$ and radius r. We need to find the current angular displacement along the arc using the four-quadrant inverse tangent function $atan2d(Y, X)$ [35].

$$\theta_{arc} = atan2d\big((y_{vehicle} - y_M), (x_{vehicle} - x_M)\big) \tag{65}$$

Desired yaw angle depends on direction of travel around the arc, Clockwise (CW) or Counter-Clockwise (CCW).

$$\psi_{path,CW} = \theta_{arc} - 90° \text{ or } \psi_{path,CCW} = \theta_{arc} + 90° \tag{66}$$

Now the desired yaw angle is known (for straight line and arc paths) it is straight forward to find e. First, rotate the vehicle's position around the Z^G axis by $-\psi_d$:

$$\begin{bmatrix} X'^G \\ Y'^G \end{bmatrix} = \begin{bmatrix} \cos(-\psi_{path}) & -\sin(-\psi_{path}) \\ \sin(-\psi_{path}) & \cos(-\psi_{path}) \end{bmatrix} \begin{bmatrix} X^G \\ Y^G \end{bmatrix} \tag{67}$$

The modified position $\left[X'^G, Y'^G\right]^T$ represents the vehicle with respect to a straight line path along the X^G axis. Therefore cross track error is found:

$$e = Y'^{G} \tag{68}$$

It is useful to note that when e is *positive*, the yaw angle should be *decreased*, and vice-versa.

Although this work on path following focuses on a marine vehicle (pictured in Figure 16), there is in fact no direct relation to the inner-loop vehicle dynamics. Therefore, the same methods (with minor adjustments) may be used in controlling both UAVs and Marine Vehicles. Of course, with the UAV there must be an additional constraint on the vehicle's height.

5.1.1 Component Paths

There are two types of path commonly used in tests of systems similar to the one proposed here; Lawnmower and Racetrack. They consist of combinations of straight lines and arcs, which are joined together at the waypoints which define the path.

Lawnmower Paths for single vehicles can be defined with two variables; swath length l_{swath} and arc diameter d_{arc}. For a path beginning at the origin of the chosen global coordinate system, the path's waypoints are defined as follows for the first full cycle.

$$Waypoints_{lawnmower} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} 0 & 0 & d_{arc} & d_{arc} & 2d_{arc} \\ 0 & l_{swath} & l_{swath} & 0 & 0 \end{bmatrix} \tag{69}$$

This pattern can be repeated for any number of path sections, with an overall positive progression along the X axis, seen in Figure 17. The racetrack path is a repeating loop which is more compact for experiments at sea, similar in shape to a NASCAR track. Defined by the same two variables, this takes the repeating form:

$$Waypoints_{racetrack} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} 0 & 0 & d_{arc} & d_{arc} \\ 0 & l_{swath} & l_{swath} & 0 \end{bmatrix} \tag{70}$$

When multiple vehicles are following component paths in cooperation, we specify the path waypoints in a slightly altered way, with an offset between parallel paths l_{off}, and a minimum arc diameter $d_{arc_{min}}$. The reason for specifying the *minimum* arc diameter is discussed in Section 5.3.

5.2 Path Following Simulations & Results

The path following algorithm developed is suitable for all classes of vehicle, the only difference between Quadcopter and ASV versions are gain values. This was an intentional effort to demonstrate the inner/outer loop control [20], and the path follower's decoupled nature from

the vehicle dynamics. One advantage of this is the ability to incorporate new classes and models of vehicle controlled in heading angle and speed.

All simulations were carried out using *Matlab* code developed by the author.

5.2.1 Quadcopter

The gain values used here are $K_{P,e} = 10$ and $K_{I,e} = 1$

The 3D trajectory seen in Figure 17 shows one cycle of a lawnmower path being followed by a Quadcopter. The vehicle clearly follows the path with great accuracy throughout, in Figure 18 the cross track error over time of the vehicle is shown, with absolute error in the mm range.

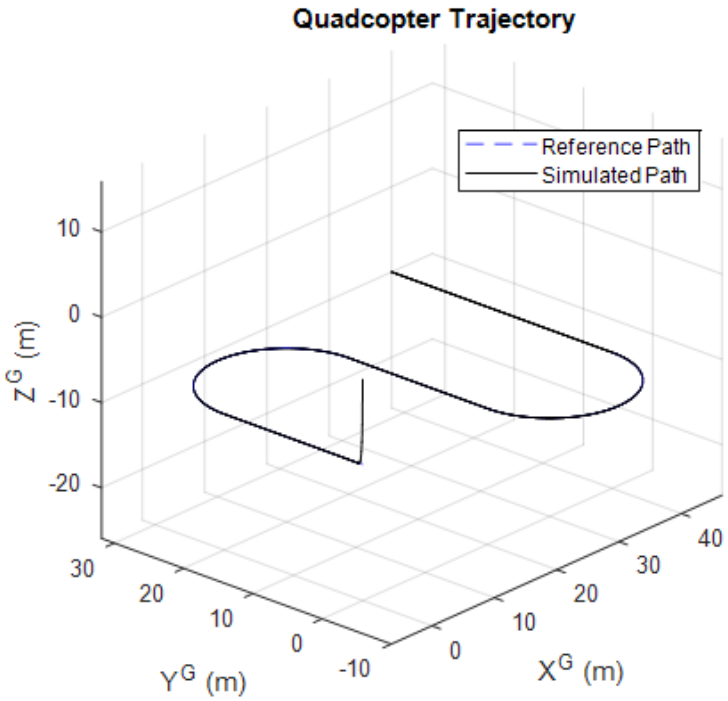


Figure 17 - Trajectory of a Quadcopter following a lawnmower path

The initial steep rise in e seen in Figure 18 is due to the altitude controller attempting to reach a steady altitude $= 10m$, causing more aggressive manoeuvring in other directions. Transients from one path component to the next occur at $t = 80, 205, 290, and\ 410s$. Performance is better in straight line components as the dynamics involved are much simpler.

In the arc sections, cross track error is highest half way round, but with only $4.50mm$ maximum value, this is acceptable.

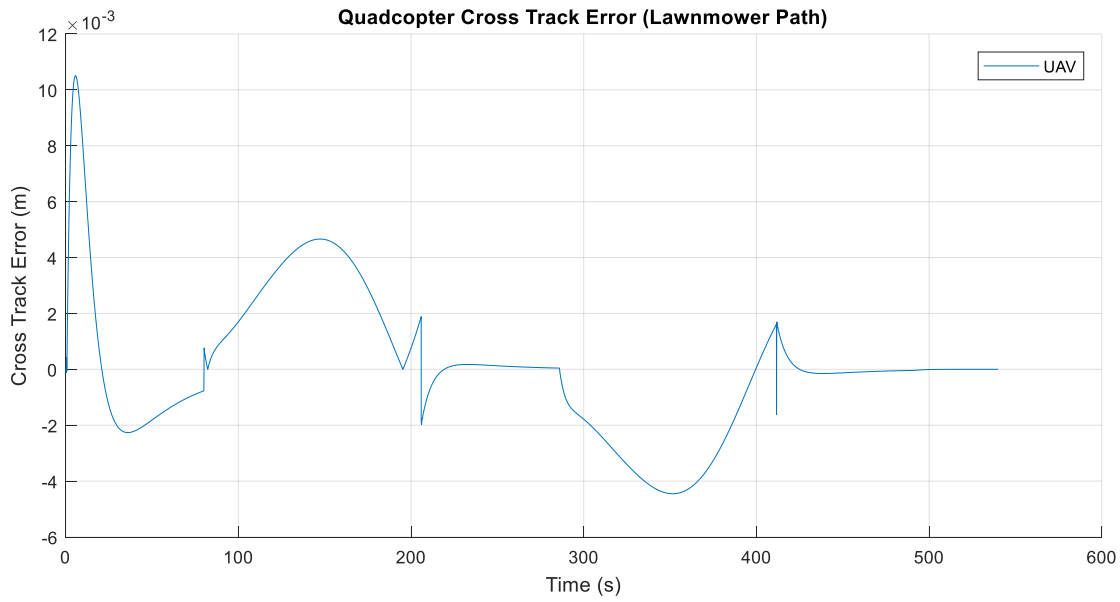


Figure 18 - Cross track error of a Quadcopter following a lawnmower path

5.2.2 Autonomous Surface Vehicle

The gain values used here are $K_{P,e} = 5$ and $K_{I,e} = 0.2$

The ASV is provided with a constant speed reference $U_{ref} = 0.25m\ s^{-1}$ commanded to follow a lawnmower path defined by:

$$l_{swath} = 25m, d_{arc} = 25$$

The path following achieved by the ASV is of lower quality than that of the Quadcopter, this can be seen in the cross track error versus time plot Figure 20. Maximum cross track error is an order of magnitude greater than in the Quadcopter simulations. There are also oscillations present, this appears to be due to the different time delays in the speed and heading controllers, and so corrective motions work against one another to some degree.

In arc paths the vehicle typically has greater cross track error halfway round, as was seen in the Quadcopter.

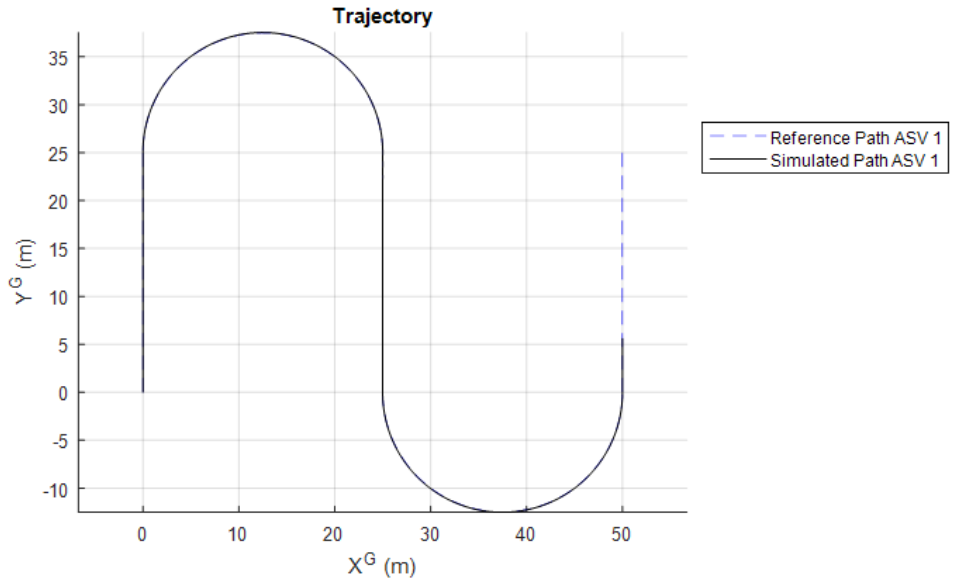
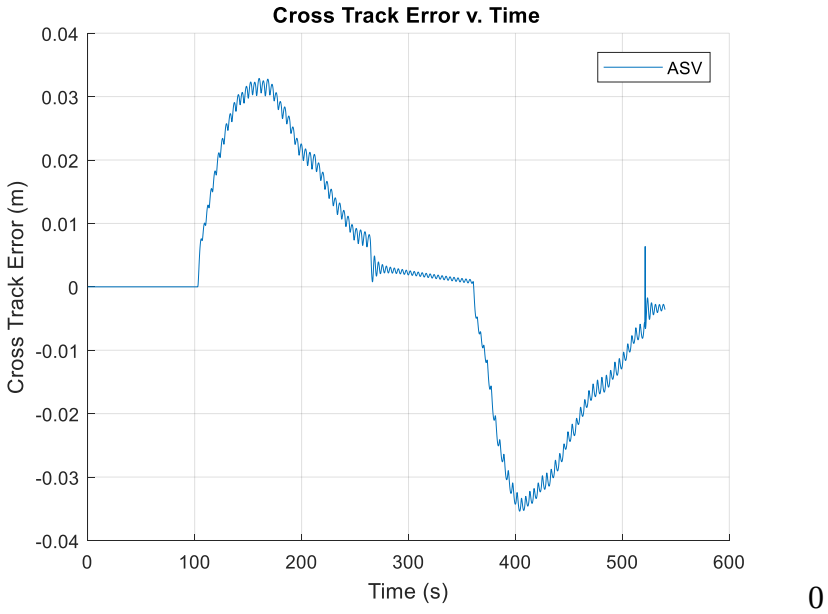


Figure 19 - Trajectory of an Autonomous Surface Vehicle following a lawnmower path



0

Figure 20 - Cross track error of an Autonomous Surface Vehicle following a lawnmower path

The path following controller is satisfactory, the simplified model of the vehicle stays within a small margin of the path throughout, and the oscillations in cross track error are small in amplitude. Smoothing by filtering may be required if deploying this control algorithm in tests at sea, as oscillatory commands often lead to instability in real vehicles

5.3 Limitations

It was hypothesised that the vehicles would have a minimum turn radius due to actuator limitations, to test this 4 ASVs were simulated with incrementally decreasing turn diameters,

each with a constant speed reference of $0.5m/s$. Maximum and Route Mean Squared (RMS) cross track errors are shown in Table 9, and the followed trajectories can be seen in Figure 21.

Table 9 - ASV minimum arc diameter tests, maximum cross track error

Vehicle	d_{arc} **(m)**	e_{max} **(m)**	e_{RMS} **(m)**
ASV1	16	0.1555	0.1037
ASV2	12	0.1969	0.1495
ASV3	8	0.4229	0.3121
ASV4	4	0.8014	0.5025

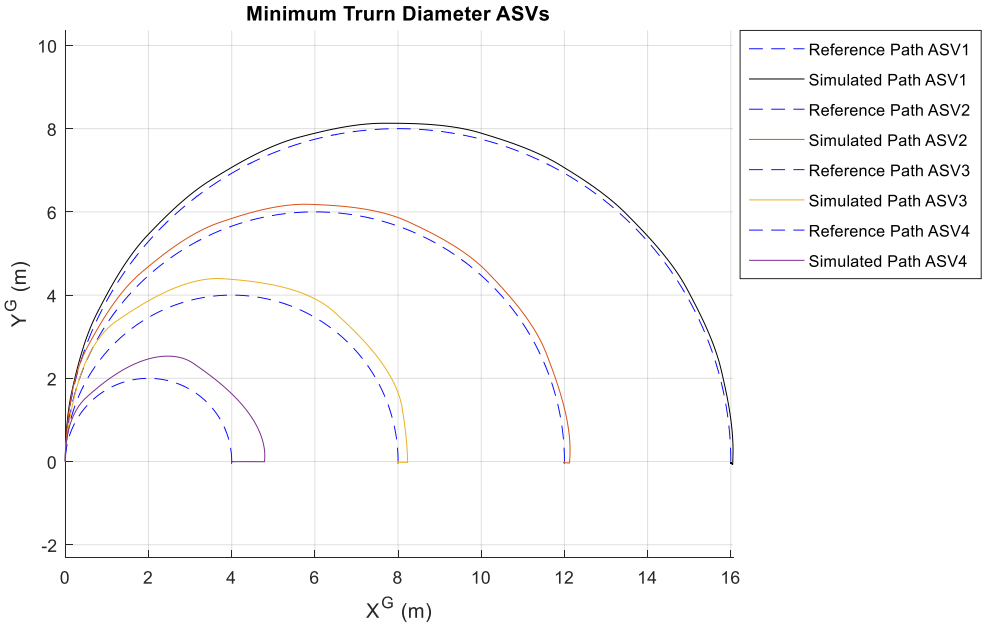


Figure 21 - Finding ASV minimum turn radius

An arc diameter $d_{arc} = 10m$ was deemed to be the minimum turn radius for the ASV due to the large increase in e_{RMS} seen between $8m$ and $12m$ arcs. Particularly, $d_{arc} = 4m$ demonstrates the loss of accurate control small arcs produce.

The same investigation was carried out with the Quadcopter, with results presented in Table 10, a smaller minimum turn radius was found: $d_{arc_{min}} = 3m$, with $e_{rms} = 0.046$. Generally the UAV followed less distorted paths, with lower errors, which increase more quickly when d_{arc} is reduced below $3m$.

Table 10 - Quadcopter minimum arc diameter tests, maximum and RMS cross track error

Vehicle	d_{arc} **(m)**	e_{max} **(m)**	e_{RMS} **(m)**
UAV1	4	0.0963	0.0378
UAV2	3	0.1164	0.0460
UAV3	2	0.1674	0.0551
UAV4	1	0.2417	0.0638

It is also noted that the turning performance deteriorates with higher velocities for both vehicles, slower reference speeds may be provided in arc sections to combat this.

The overall minimum arc diameter is set to accommodate the requirements of both vehicles:

$$d_{arc_{min}} = 10m$$

6 Cooperative Controller Design & Analysis

Now that the vehicles are capable of following pre-defined routes, they can be controlled in cooperation, and results may be validated by assessing cooperation and path following simultaneously.

In this section the method of cooperation is described, and a control algorithm is proposed. Results from a variety of simulations are shown for combinations of up to four vehicles.

A simplified architecture of cooperative control of 2 vehicles is seen in Figure 22. Where the box *Vehicle 1* is equivalent to *Vehicle 2*, which has been reduced for ease of reading. In cases with n vehicles, the cooperative controller will also have n inputs and outputs.

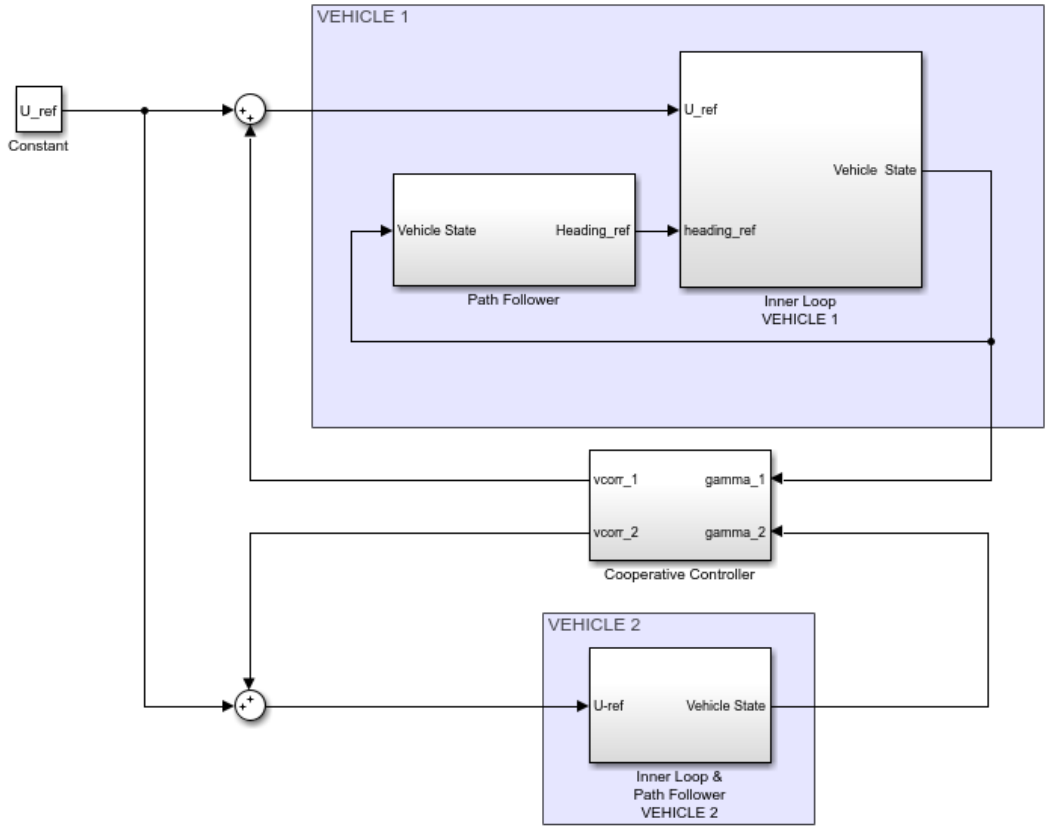


Figure 22 - Simplified architecture of cooperative control of two vehicles

6.1 Controller Design

Cooperative path-following between $i, j \in 1, \ldots, n$ vehicles is achieved by specifying a coordination state γ, where vehicles $[i, j]$ are said to be in coordination if $\gamma_i = \gamma_j$ [29]. The coordination state may be defined in any number of ways depending on the mission requirements.

The vehicles are not necessarily prescribed to follow paths of the same shape, for example one vehicle could follow a sine curve intersecting another vehicle following straight line, and both coordination states could be defined along the axis of the straight line.

The coordination is defined within this report as a normalized variable along path length, stated below for each of the component path types considered.

***Arc*:** $$\gamma = {}^{\theta}/_{\theta_{total}} \tag{71}$$

Straight Line: $$\gamma = {}^{d}/_{d_{total}} \tag{72}$$

In many scenarios, communication between all vehicles is not possible due to physical constraints (especially in the case of a sea-surface link [36]), and momentary communication failures. The set of vehicles that agent i can communicate with is denoted N_i. This allows us to define the overall coordination error of each vehicle as a pseudo-average value.

$$\varepsilon_{\gamma,i} = \gamma_i - \frac{1}{|N_i|}\sum_{j \in N_i} \gamma_j \tag{73}$$

This definition for coordination error may be driven to zero using the controller;

$$v_{corr_i} = -k_{sync}\, sin^{-1}\left(\frac{\varepsilon_{\gamma,i}}{|\gamma_i| + k_s}\right)\frac{2}{\pi} \tag{74}$$

Where the gain value k_{sync} affects the maximum value of speed correction, so must be lower than designated path speed to prevent vehicles from reversing along the path, k_s affects the controller's sensitivity to coordination error.

6.2 Results

Simulations were run for different combinations of vehicles and path types in order to produce a stable cooperative controller. In the following sections we will build up the simulations from two Quadcopters following parallel lines, to four vehicles (2 Quadcopters and 2 ASVs) controlled in cooperation along lawnmower paths.

The gain values which were settled on for all of the following simulations are:

$$\mathrm{k_{sync}} = U_{ref,nominal}\text{ , and } \mathrm{k_s} = 0.1$$

6.2.1 2 Quadcopter Cooperation

Straight line coordination for parallel paths of equal length is trivial in simulation, the coordination states remain equal throughout, regardless of the cooperative controller setup. This is due to the mathematical equality of the simulations of the vehicles.

To increase the usefulness of the results, the vehicles were given randomly assigned initial locations in the region of the start point (Figure 23). The vehicles reach perfect coordination quickly and remain in that state throughout.

There are some artefacts in Figure 23 at the end of the mission ($\gamma = 1$) when the inner loop switches to a position controller headed for the path end point. There is some 3D oscillatory motion seen when the Quadcopters are settling on a fixed point whilst decelerating, this behaviour is acceptable as the vehicles settle in $5s$.

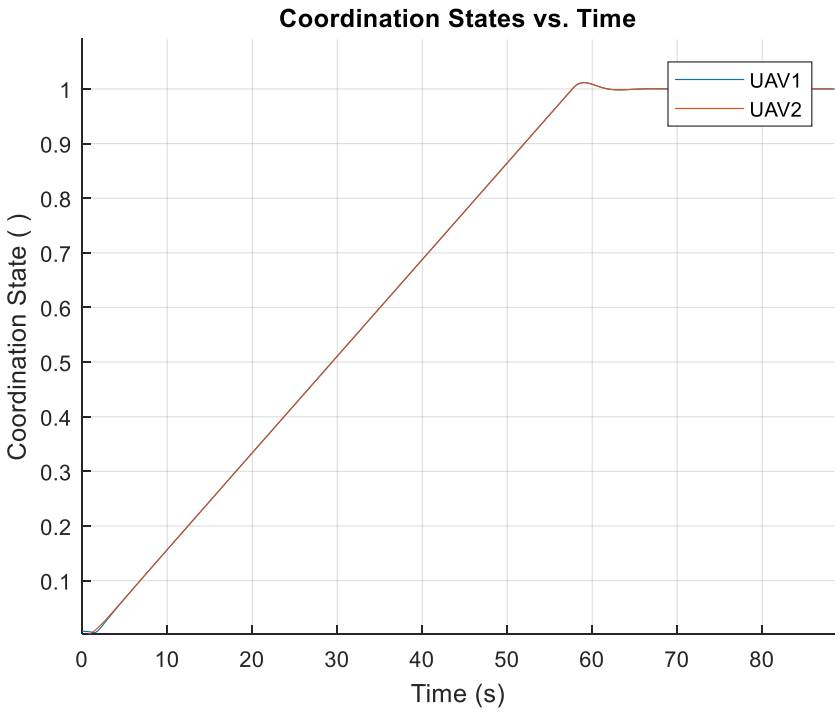


Figure 23 Coordination States of a 2 Quadcopter straight line simulation vs. time. (with randomly assigned initial position)

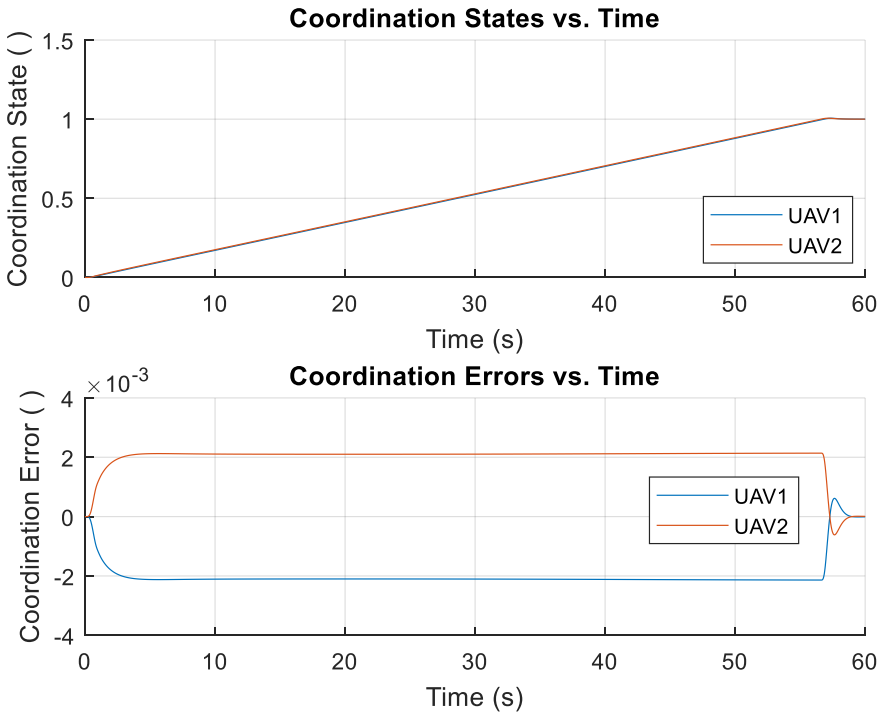


Figure 24 - Coordination States and errors of a 2 Quadcopter concentric circle cooperative simulation

Figure 24 shows coordination states and error values for two Quadcopters, *UAV1* following an arc with diameter 20m, and *UAV2* an arc of 18m. The equal gradients of the coordination states suggest good cooperation throughout. The coordination error of *UAV2* remains high throughout, and *UAV1* remains low, due to the difference in path *lengths* of the two arcs.

Vehicles on arcs with larger diameters require higher velocities than those with lower radii if coordination is to be maintained throughout.

6.2.2 2 ASV Cooperation

For the basic paths demonstrated by parallel lines and concentric circles, the behaviour of ASV-ASV systems was almost identical to that seen in Figure 23 for Quadcopters. Coordination is overall good, and the same steady state behaviour is seen along arc paths.

To provide more interesting results, a lawnmower path will be followed, specified as in Section 5.1.1 by:

$$l_{swath} = 100m,\ d_{arc_{min}} = 25, \text{ and } \ l_{off} = 5m.$$

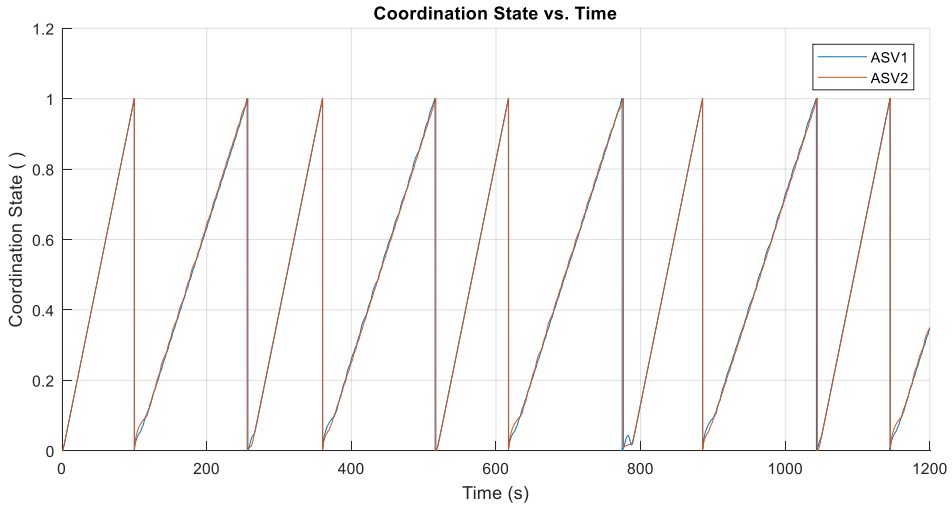


Figure 25 - Coordination States of a UAV-UAV lawnmower simulation vs. time

Each of the path segments is seen as a ramp from 0 to 1 in Figure 25, these reset as each segment is completed. Overall coordination performance is good, as demonstrated by similar coordination gradients in both vehicles, with no signs of drifting or accumulation of errors over time.

The straight-line segments (odd numbered segments when reading Figure 25 left to right) display consistent cooperative performance as was seen previously. There is minimal coordination error in the arc segments.

The cooperative control algorithm is enacted upon only vehicles on the same path segment. This prevents the case where the trailing vehicle near the end of one segment sees the leading vehicle has a much lower coordination state (along the following segment). This would lead the trailing vehicle to slow down, producing an accumulative error.

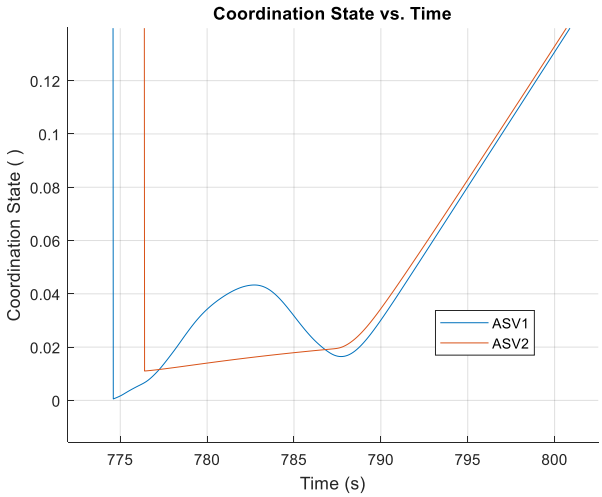


Figure 26 - Transient performance of ASV-ASV cooperative lawnmower path (cutaway from Figure 25)

There exists the possibility to have one single coordination state normalized along the entire component path, but the method shown here allows for simple implementation of any general path made of lines and arcs.

The issue with the presented method of γ parameterisation is in the transients between path sections, reversing behaviour (negative coordination state gradient) is sometimes seen if there is delay between the completion of sections as in Figure 26.

6.2.3 Inhomogeneous 2 Vehicle Systems

A cooperative lawnmower simulation with one Quadcopter and one ASV produced good quality results with consistent coordination throughout. In Figure 27 the lawnmower path is defined by the following:

$$l_{swath} = 10m,\ l_{off} = 5m, \text{ and } d_{arc_{min}} = 25m$$

A constant nominal speed reference was provided $U_{ref,nominal} = 0.25\ m\ s^{-1}$.

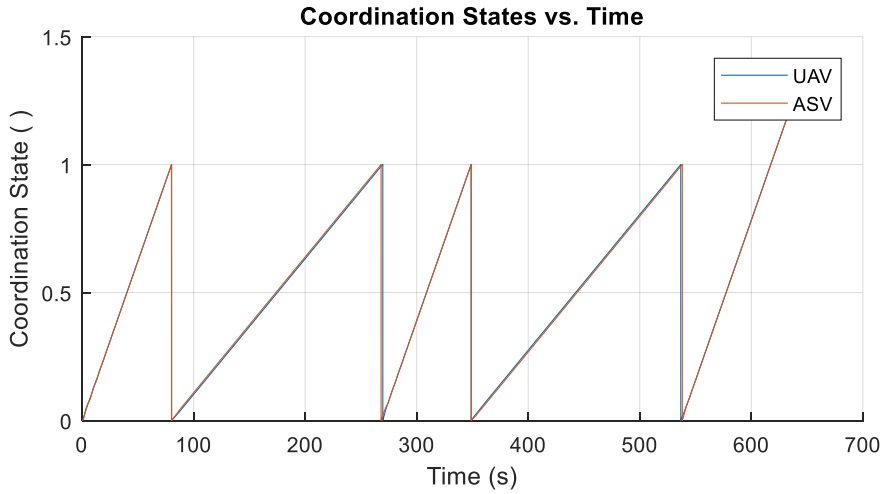


Figure 27 - Coordination states versus time of a simulated cooperative lawnmower path between an ASV and Quadcopter

It can be useful to look at $\upsilon_{\mathrm{corr_i}}$ over time to understand what effect the coordination controller has on the vehicles, this is presented in Figure 28, $\upsilon_{\mathrm{corr_i}}$ is equivalent in character to ε_γ.

Due to the controller design, when only two vehicles are cooperating, the $\upsilon_{\mathrm{corr_i}}$ values are always equal in value with switched signs.

The three damped oscillatory motions seen (e.g. 0 to $80s$) take place during the straight line segments of the path, at these times the coordination controller is attempting to reach a steady state with zero coordination error, results show that this is typically achieved in $50s$. This varies depending on path offset in preceding arc sections.

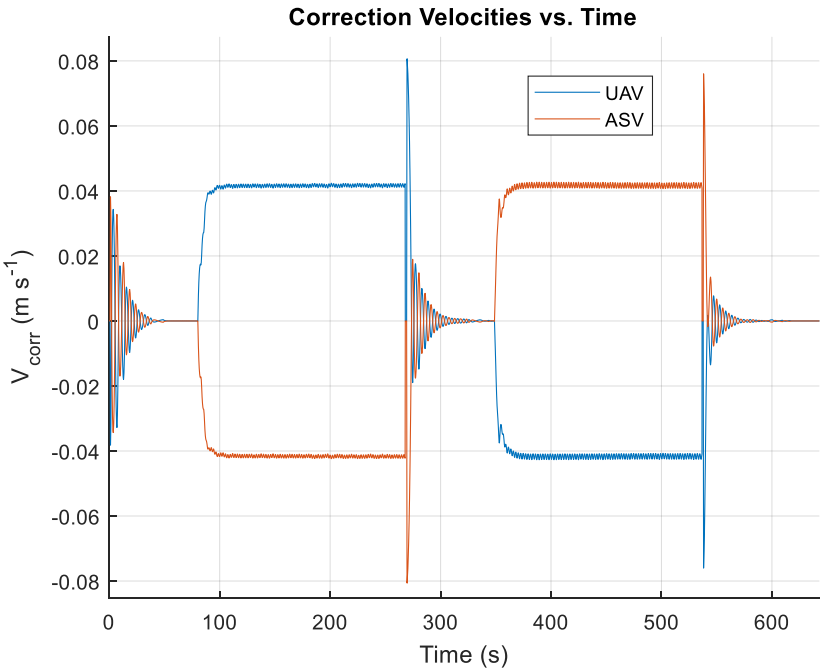


Figure 28 - Correction velocities versus time of a simulated cooperative lawnmower path between an ASV and Quadcopter

The other segments, where one $\upsilon_{\mathrm{corr_i}}$ is high and the other is low occurs during the arc sections of the path as before. The high and low values switch each time due to the method of parameterising parallel lawnmower paths, alternating large and small diameters.

A similar quality of result was seen in a range of simulations with this vehicle configuration.

6.3 Multi-Vehicle Systems

It was hypothesised that the complexity brought on by the control of more than two vehicles simultaneously would lead to poor results in cooperation. Any errors brought in by the individual vehicle controllers or path followers could lead to instability in the overall system.

6.3.1 4 Quadcopters

Initial tests with 4 Quadcopters produced promising results, correction velocities over time for a typical lawnmower manoeuvre may be seen in Figure 29. More complex behaviour is seen than in previous simulations, but all of the basic principles still apply.

There are high amplitude spikes in v_{corr} at the end of the arc segments, this is related to the delay in completion, when there are points in time when not all vehicles are in cooperation with one another. This causes an increase in error, but is quickly controlled, and error is brought to zero during straight line sections.

Coordination states for an arc segments is shown in Figure 30, where there is a small steady state error seen between the vehicles.

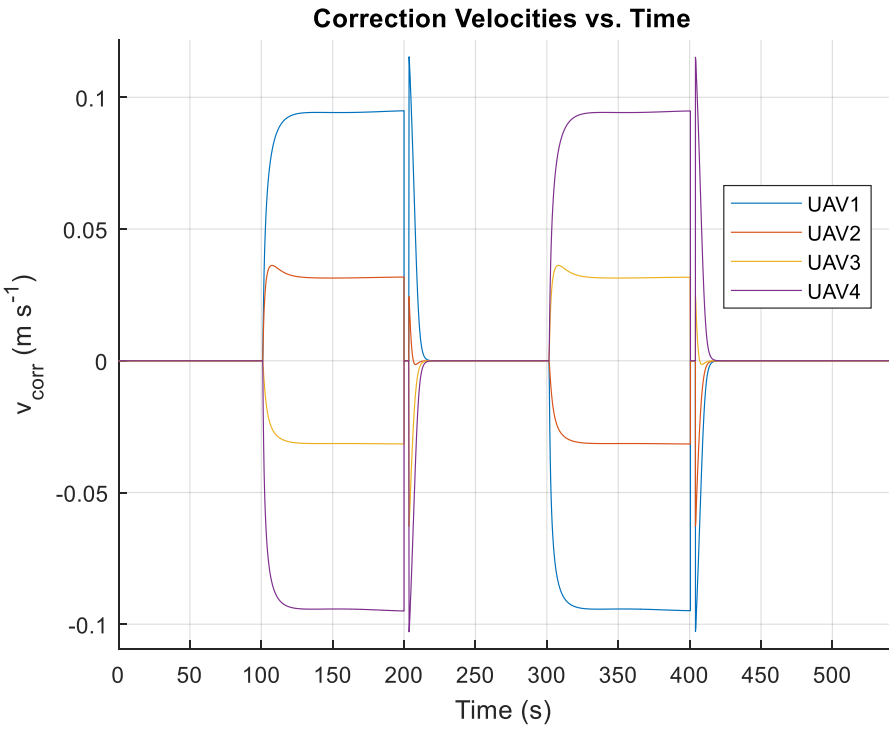


Figure 29 - Correction velocities of four Quadcopters cooperating along a lawnmower path

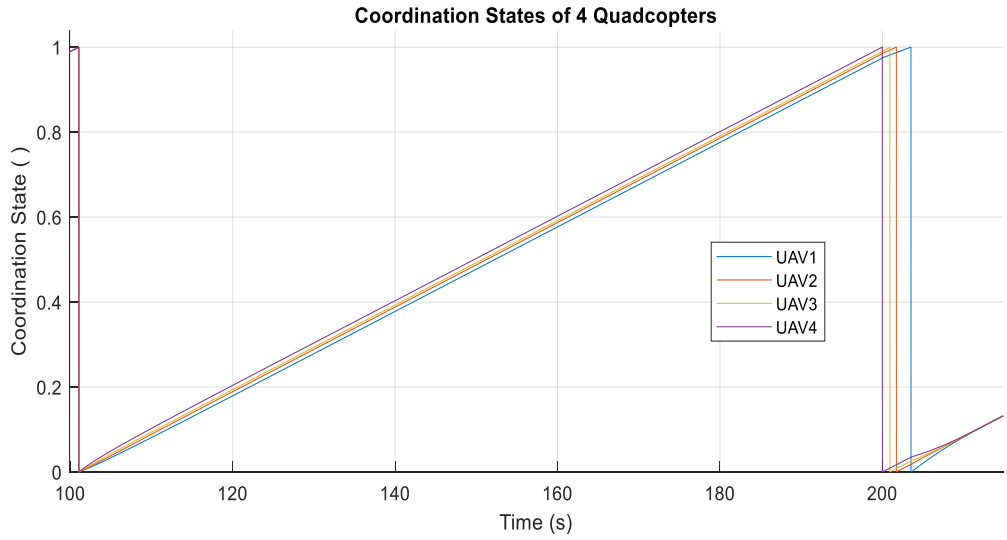


Figure 30 - Coordination states of four quadcopters along an arc

6.3.2 2 ASVs, 2 Quadcopters

Tests with two ASVs and two Quadcopters were conducted. Overall coordination is okay throughout a lawnmower mission, with all vehicles completing every section within a $2s$ margin of one another, as seen in Figure 31 (right).

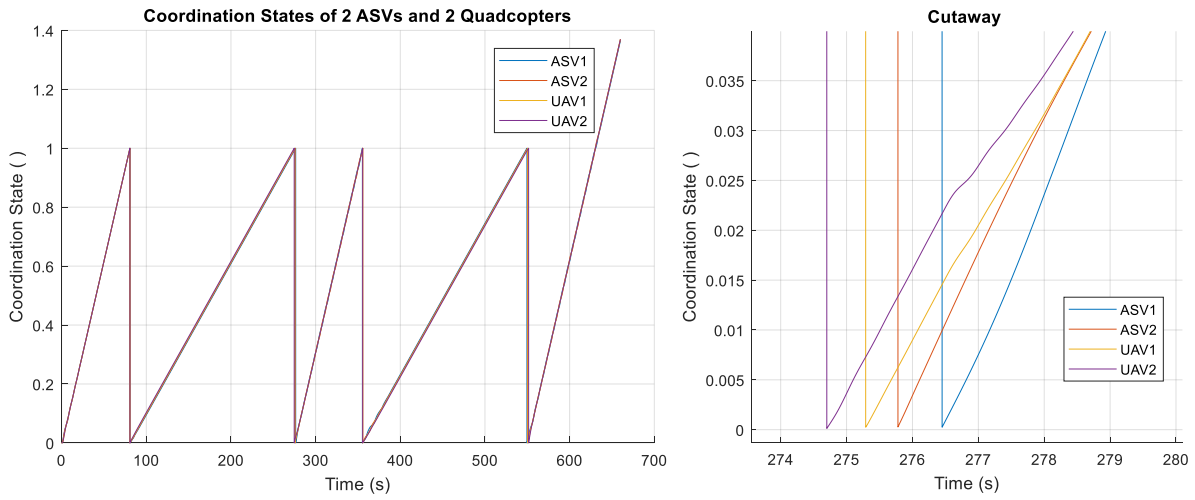


Figure 31 - Coordination states of a cooperative lawnmower mission with two ASVs and two Quadcopters, left, full simulation, right, transient cutaway

As seen in Figure 32, the v_{corr} plot exhibits high frequency oscillations, in simulation this has not affected performance in path following or degraded state control, but in practice may lead to unstable behaviour.

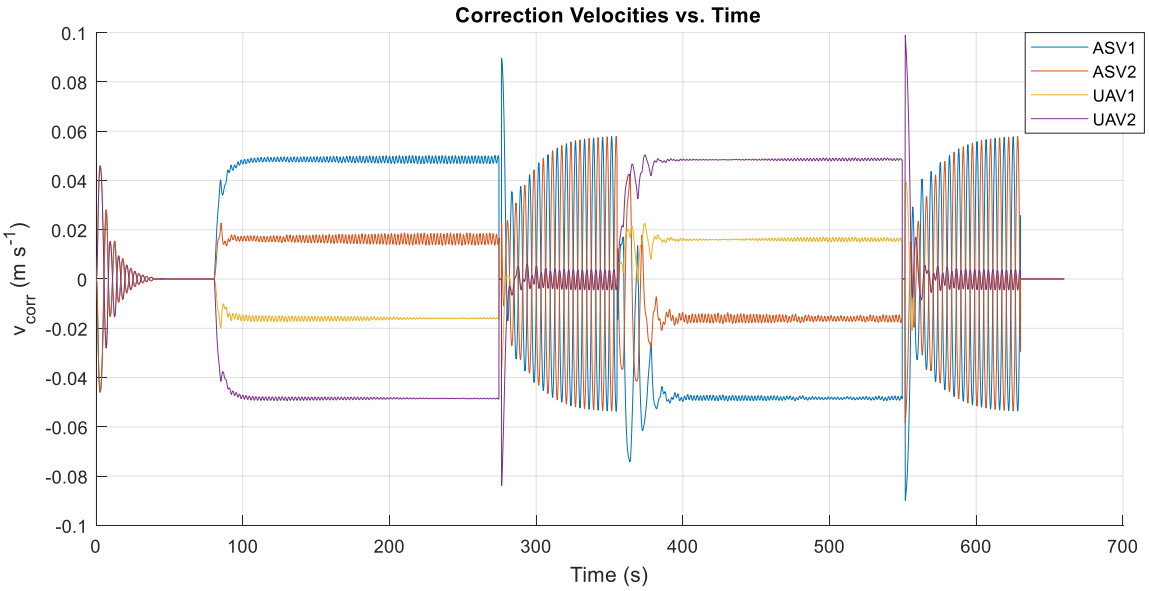


Figure 32 - Correction velocities for a 2 ASV, 2 Quadcopter lawnmower path simulation

The oscillations are introduced by the inclusion of ASVs in the 4 vehicle system, and the various time delays these add.

The measured speed of the Quadcopters remains steadier than the ASVs throughout. This suggests that Quadcopters are capable of performing well in a variety of situations with disturbances, particularly variations caused by other vehicles in cooperation. This is a reasonable conclusion, given the manoeuvrability they demonstrate.

6.3.3 Modified Cooperative Controller

To combat the issues seen in the results of cooperative mission simulations, additions were made to the controller:

- Derivative term for smoothing of v_{corr}
- Integral term to reduce steady state errors

This resulted in the following controller equation

$$\upsilon_{\mathrm{corr_i}} = -\mathrm{k_{sync}} \sin^{-1}\left(\frac{\varepsilon_{\gamma,\mathrm{i}}}{|\gamma_\mathrm{i}| + \mathrm{k_s}}\right)\frac{2}{\pi} - K_D\dot{\varepsilon}_{\gamma,\mathrm{i}}(t) - K_I\int_0^t \varepsilon_{\gamma,\mathrm{i}}(t)dt \qquad (75)$$

Where $K_D = 5$, and $K_I = 1$, this produced the following results in coordination states, where all other mission parameters were kept constant:

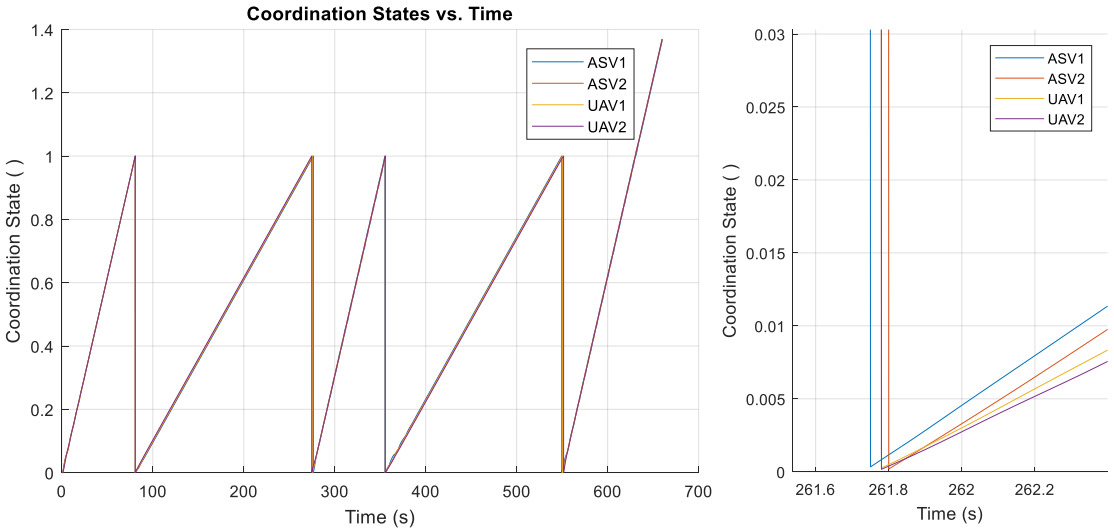


Figure 33 - Coordination states for a 2 ASV, 2 Quadcopter cooperative lawnmower manoeuvre (with altered coordination controller). Left, full simulation, right, cutaway.

Figure 33 represents an improvement, with a maximum delay between the first vehicle and the last vehicle completing arc sections at $0.05s$, compared with $2s$ seen in Figure 31. The steady state error has all but vanished.

The behaviour in v_{corr} has different characteristics from that seen before, a bias is introduced to the arc segments, this appears to improve overall coordination performance in these sections.

Oscillatory behaviour is still seen, tuning of gains and a better estimation of $\dot{\varepsilon}_{\gamma,\mathrm{i}}$ may again reduce oscillations in the speed reference provided to the vehicles. It is also possible to apply filtering to remove the high frequency components of the signal. It should be noted that the

greatest amplitude seen in v_{corr} is $0.025m\,s^{-1}$. Which is only 10% of the nominal speed reference provided in this case.

Unstable v_{corr} responses are seen in the second and third straight line sections, it is unclear at this time what is causing this instability. Further experimentation in how the controller is applied may prove fruitful.

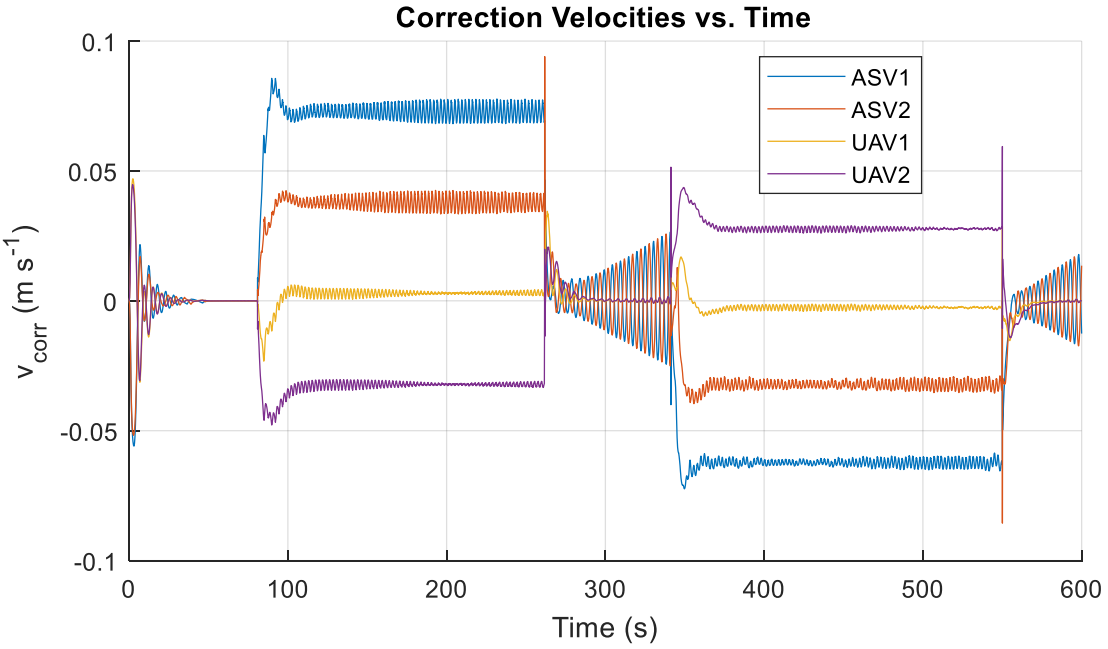


Figure 34 - Correction velocities for a 2 ASV, 2 Quadcopter cooperative lawnmower manoeuvre (with altered coordination controller)

Unfortunately, due to time constraints, this modified version of the controller was not tested on all vehicle combinations. Results from this test, which includes both vehicle and path types, strongly suggest that similar improvements will be seen in other mission types. Preliminary results confirm this.

Results for up to four vehicles have been shown within this chapter, the algorithms developed are capable of handling any number of vehicles, given the required processing power. Results here present qualitative behaviour, and the methodology provides a framework to replicate these results.

7 Conclusions

Models of both Autonomous Surface Vehicles and Quadcopters have been produced, and controllers for their states have been tested with success. The vehicles exhibit different characteristics, with the Quadcopter being more manoeuvrable. The controllers produce good results, suitable for use in high level control of these vehicles.

Path following of predefined routes consisting of lines and arcs has been achieved in simulation in both vehicles. Cross track error of both vehicles in most path types is acceptably low, with the Quadcopter performing better than the ASV. Constraints in arc diameter have been set, such that the vehicles will not be commanded to follow paths they are incapable of.

Once controllable models of both vehicles were produced, and a stable algorithm for path following developed, it was possible to produce a cooperative path following controller. By defining an appropriate coordination state along path length (or similar) the coordination error between vehicles may be found. This coordination error term is used to provide a correction velocity reference to the path following vehicles.

Valid simulation results have been acquired for the cooperation of combinations of Autonomous Surface Vehicles and Quadcopters. An altered version of the controller initially presented was proposed after results were analysed, this led to reduced coordination error in simulations of 4 heterogeneous vehicles.

The simulations show successful cooperation in paths made up of lines and arcs, for various combinations of vehicles. Results provided may be useful guides for any future implementation of the proposed system.

Mathematical approximations have led to the simulation of ideal systems, results will need to be validated with experiments, or error models generated. Following any re-evaluation, it is likely methods will need to be assessed and gain values altered.

This work provides a guide for developing cooperative controllers for Autonomous Surface Vehicles and Quadcopters. Successful simulations suggest that implementation in these vehicles will be possible, if the relevant vehicles, sensors and communications systems are available.

8 Recommendations for Further Work

The work presented here requires validation by implementation in the vehicles, and rigorous trials to assess performance. This will require knowledge of the systems in question.

There are some features missing from the control system, I have not specified a collision avoidance algorithm to prevent vehicles impacting upon one another. Obstacle avoidance is also an important feature to include to prevent avoidable crashes, but relies on suitable sensors being installed. These are common modules in mobile robotics control systems which can be included in the overall control architecture.

This work does not include a model for communications failures during cooperative missions, previous work at ISR and elsewhere provide guides for how this problem should be addressed. Probabilistic models may be used to simulate communications failures.

Bibliography

[1] A. Aguiar, J. Almeida, M. Bayat, B. Cardeira, R. Cunha, A. Hausler, P. Maurya, A. Oliveira, A. Pascoal, M. Rufino, L. Sebastiao, C. Silvestre and F. Vanni, "Cooperative Autonomous Marine Vehicle Motion Control in the scope of the EU GREX Project: Theory and Practice".

[2] D. Hambling, "Japan sends robots into Fukushima nuclear plant," New Scientist, 18 March 2011. [Online]. Available: https://www.newscientist.com/blogs/onepercent/2011/03/japanese-send-robots-into-fuku.html. [Accessed 11 January 2018].

[3] E. Salami, C. Barrado and E. Pastor, "UAV Flight Experiments Applied to the Remote Sensing of Vegetated Areas," *Remote Sens.,* vol. 11, no. 6, 2014.

[4] Gartner, "Gartner Says Almost 3 Million Personal and Commercial Drones Will Be Shipped in 2017," Gartner, Stamford, Conneticut, 2017.

[5] J. Cao, "New Drone Racing League Wants to Be the Next Nascar," Bloomberg, 11 January 2016. [Online]. Available: https://www.bloomberg.com/news/features/2016-01-26/new-drone-racing-league-wants-to-be-the-next-nascar. [Accessed 27 January 2018].

[6] C. R. a. D. I. Service, H. Al-Khatib, G. Antonelli, A. Caffaz, G. Casalino, I. B. de Jong, H. Duarte, G. Indiveri, S. Jesus, K. Kebkal, A. Pascoal and D. Polani, "Widely scalable Mobile Underwater Sonar Technology (WiMUST) project: An overview," in *OCEANS 2015*, Genoa, Italy, 2015.

[7] R. R. Murphy, E. Steimie, C. Griffin, M. Hall, C. Cullins and K. Pratt, "Cooperative Use of Unmanned Sea Surface and Micro Aerial Vehicles at Hurricane Wilma," *Journal of Field Robotics,* vol. 25, no. 3, pp. 164-180, 2008.

[8] A. L. Meyrowitz, R. Blidberg and R. C. Michelson, "Autonomous Vehicles," in *Proceedings of the IEEE,* 1996.

[9] J. E. Manley, "Unmanned surface vehicle, 15 years of development," in *OCEANS*, Quebec City, QC, Canada, 2008.

[10] P. C. Abreu, J. Botelho, P. Góis, A. Pascoal, J. Ribeiro, M. Rufino, L. Sebastião and H. Silva, "The MEDUSA class of autonomous marine vehicles and their role in EU projects," in *OCEANS 2016*, Shanghai, 2016.

[11] R. de Narbonne, "Les gyroplanes de Louis Breguet," *Le Fana de l'Aviation,* 1 Jun 2017.

[12] M. Sklar, "Boeing offers 2 of 5 development options in rotorcraft program," *Boeing Fronteirs,* pp. 44-45, December 2006.

[13] E. Frazzoli, M. A. Dahleh and E. Feron, “Trajectory tracking control design for autonomous helicopters using backstepping algorithm,” in *Proceedings of the American Control Conference*, Chicago, 2000.

[14] D. Kingston, R. Beard, T. McLain, M. Larsen and W. Ren, “Autonomous vehicle technologies for small fixed wing UAVs,” *Journal of Aerospace Computing, Information, and Communication,* vol. 2, no. 1, pp. 92-108, 2005.

[15] R. Mahony, P. Pounds, J. Roberts and P. Hynes, “Design of a Four-Rotor Aerial Robot,” in *Australasian Conference on Robotics and Automation*, Auckland, 2002.

[16] R. Mahoney, V. Kumar and P. Corke, “Multirotor Aerial Vehicles (Modelling, Estimation and Control of Quadrotor),” *IEEE Robotics & Automation Magazine,* pp. 20-32, September 2012.

[17] D. Cabecinhas, R. Cunha and C. Silvestre, “A nonlinear quadrotor trajectory tracking controller with disturbance rejection,” *Control Engineering Practice,* vol. 26, pp. 1-10, 2014.

[18] H. K. Khalil, “Backstepping,” in *Nonlinear Systems*, 3rd ed., East Lansing, Michigan, US, Prentice Hall, 2001, pp. 589-603.

[19] M. Breivik and T. I. Fossen, “Path following of straight lines and circles for marine surface vessels,” in *IFAC Conference on Computer Applications in Marine Systems*, Ancona, Italy, 2004.

[20] A. P. A. A. P. P. Maurya, “Marine Vehicle Path Following Using Inner-Outer Loop Control,” in *IFAC Proceedings Volume 42*, Lisbon, 2009.

[21] Z. Li, R. Bachmayer and A. Vardy, “Path-Following Control for Unmanned Surface Vehicles,” in *IEEE/RSJ International Conference on Intelligent Robots and Sytsmes (IROS)*, Vancouver, BC, Canada, 2017.

[22] J. P. Desai, J. Ostrowski and V. Kumar, “Controlling formations of multiple mobile robots,” in *International Conference on Robotics & Automation*, Leuven, Belgium, 1998.

[23] P. Abreu, M. Bayat, J. Botelho, P. Gois, A. Pascoal, J. Ribeiro, M. Ribeiro, M. Rufino, L. Sebastiao and H. Silva, “Cooperative Control and Navigation in the Scope of the EC CADDY Project,” in *OCEANS 2015*, Genova, 2015.

[24] R. C. Arkin, Behavior-Based Robotics, Cambridge, Massachusetts: MIT Press, 1998.

[25] F. J. Mendiburu, M. R. A. Morais and A. M. Lima, “Behavior coordination in multi-robot systems,” in *IEEE International Conference on Automatica (ICA-ACCA)*, Curicó, Chile, 2016.

[26] R. Ghabcheloo, A. P. Aguiar, A. Pascoal, C. Silvestre, I. Kaminer and J. Hespanha, “Coordinated path-following control of multiple underactuated autonomous vehicles in the presence of communication failures,” in *Proceedings of the 45th IEEE Conference on Decision & Control*, San Diego, CA, USA, 2006.

[27] V. Cichella, I. Kaminer, V. Dobrokhodov, E. Xargay, R. Choe, N. Hovakimyan, A. P. Aguiar and A. M. Pascoal, “Cooperative Path Following of Multiple Multirotors Over Time-Varying Networks,” *IEEE Transactions on Automation Science and Engineering,* vol. 12, no. 3, pp. 945-957, 2015.

[28] J. Ribeiro, “Motion Control of Single and Multiple Autonomous Marine Vehicles,” Instituto Superior Tecnico, Lisbon, 2011.

[29] R. Ghabcheloo, A. P. Aguiar, A. Pascoal and C. Silvestre, “Coordinated Path-Following of Multiple AUVs in the Presence of Communication Failures and Time Delays,” in *IEEE Conf. Decision Control,* Lisbon, 2007.

[30] G. Cai, B. M. Chen and T. H. Lee, “Chapter 2,” in *Unmanned Rotorcraft Systems,* Springer, 2011, pp. 23-34.

[31] R. Beard, “Quadrotor Dynamics and Control Rev 0.1,” Brigham Young University, Provo, Utah, 2008.

[32] A. Gibiansky, “Quadcopter Simulation and Dynamics,” Andrew Gibiansky, 2012.

[33] V. K. P. C. Robert Mahony, “Multirotor Aerial Vehicles - Modelling Estimation and Control of Quadrotor,” *IEEE Robotics & Automation Magazine,* pp. 20-32, September 2012.

[34] R. Tedrake, “Underactuated Robotics: Learning, Planning, and Control for Efficient and Agile Machines,” Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 2009.

[35] MathWorks, “Matlab Documentation - "atan2d",” [Online]. Available: https://www.mathworks.com/help/matlab/ref/atan2d.html#bu4zyzh. [Accessed 11 December 2017].

[36] J. Kalwa, “The GREX-Project: Coordination and control of cooperating heterogeneous unmanned systems in uncertain environments,” Atlas Elektronik, Bremen, 2009.

Table of Figures

List of Abbreviations

UAV: Unmanned Aerial Vehicle

WiMUST: Widely scalable Mobile Underwater Sonar Technology

ISR: Institute of Systems and Robotics

ASV: Autonomous Surface Vehicle

GPS: Global Positioning System

VTOL: Vertical Take-Off and Landing

SWAP: Size, Weight, and Power

TT: Trajectory Tracking

PF: Path Following

VF: Vector Field

LOS: Line Of Sight

LTS: Leader Tracking System

BC: Behaviour Control

NED: North, East Down

DoF: Degrees of Freedom

CB: Centre of Buoyancy

CM: Centre of Mass

TF: Transfer Function

RPM: Revolutions Per Minute

IMU: Inertial Measurement Unit

PID: Proportional, Integral, Derivative

RMS: Route Mean Squared

Appendix A – Source Code

Code examples and some documentation for this project are available on my GitHub profile (link below). This resource is not entirely comprehensive, it does not include results, nor versions from the early stages of my project.

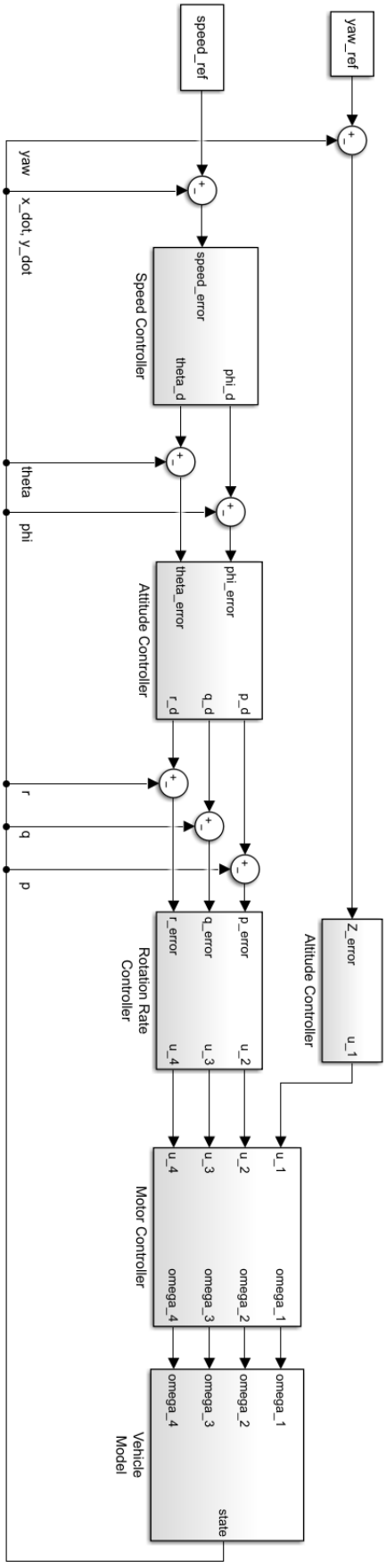
Information about the simulations and their methods is available in the files and their respective comments.

GitHub.com/LobotomyWeekend/Cooperative-Control

See the README for a basic outline of the file structure and installation.

g

Appendix B – Quadcopter Control Diagram



h

Appendix C – ASV Control Diagram