Experimental Results of Supply Ship Autopilot Optimisation Using Genetic Algorithms

Euan W. McGookin[†], David J. Murray-Smith[†], Yun Li[†] and Thor I. Fossen[‡]

[†] Centre for Systems and Control & Department of Electronics and Electrical Engineering, University of Glasgow, UK

[‡] Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway

Abstract: The optimisation by Genetic Algorithm (GA) of a non-linear autopilot for ship course changing is studied in this paper. The theory of the optimisation technique is presented from its basis in Darwinian evolution to the development of the form of optimisation algorithm used in the application. This application involves a Sliding Mode (SM) controller for use as a course changing autopilot for a scale model of a supply ship. The GA optimised simulation results are presented and post-optimisation evaluation is carried out through further simulations and trials in a model ship testing tank. From the evaluation of the optimised autopilot the GA and SM control law are shown to be effective design methodologies for control implementation.

Index Terms- optimisation, genetic algorithms, sliding mode control, ship steering autopilot, model ship testing tank.

1. Introduction

Issues associated with the design of autopilots for marine surface vessels have already been extensively researched [1-5]. Many different control methodologies have been presented and their advantages and disadvantages discussed at length. However, very few studies have investigated the potential of non-linear controllers and have addressed the intricacies of tuning such autopilots during simulation and the additional tuning required when applying these to the actual vessel. Although some forms of non-linear control law are known to have excellent properties in terms of performance robustness the process of manually tuning a non-linear controller can be a difficult, time consuming and tedious task for the designer due to the numerical interaction between the parameters which are being tuned. This is especially true when the designer is unfamiliar with the controller structure and the design method. In recent years the application of numerous *optimisation techniques* for automatically tuning such controller parameters have been reported [8].
The most widely used method is one called *Genetic Algorithms* [5,6,8-12] which investigates the problem search space by emulating the evolution process of natural species. This approach applies the Darwinian theory of *survival of the fittest* to evolve parameter solutions for the problem under investigation. These parameter are evolved by the reproduction of encoded chromosomes made up of integers. They are then evaluated through simulation and the candidates with the best time histories are encouraged to evolve. As the algorithm progresses the best parameter solutions improve their performance and finally reach an evolved optimum solution.

In this paper a complete evaluation of an automatically tuned autopilot is presented. The type of control law used to define the autopilot is called *Sliding Mode* (SM) control which is a non-linear methodology known for its robust properties [1,5-7,10-13]. This autopilot is applied to a scale model ship and the optimal solution obtained through simulations subsequently evaluated by practical implementation. The ship considered here is a scale model of a surface vessel which is used in the Guidance, Navigation and Control (GNC) Laboratory at the Norwegian University of Science and Technology. This laboratory provides an excellent environment for implementing the autopilot without the cost of actual sea trials. This has enabled a study to be performed on the potential of non-linear control system optimisation techniques for the design and implementation of autopilot systems.

This paper consists of the following six sections. In Section 2 the GNC lab facilities and the mathematical representation of the model ship are presented. Section 3 provides the derivation for the sliding mode control law used in this investigation and Section 4 outlines the genetic algorithm (GA) theory for parameter optimisation. The penultimate section presents typical optimised results for a simulated 45^{O} manoeuvre which was carried out at the University of Glasgow. Also shown in this section are the evaluation results for simulated and practical trials which were carried out the Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim. Finally, section 6 concludes this paper by discussing the overall findings of this investigation.

2. Model Ship

2.1 GNC Lab

The Guidance, Navigation and Control (GNC) Laboratory is a watertank facility which is primarily used as a test-bed for the implementation of control systems for governing the motion of a scale model of a supply ship. This model is called *CyberShip I* and is 1.17m in length (one seventieth the size of the actual vessel). The constituent parts of the GNC lab are illustrated in Figure 1.

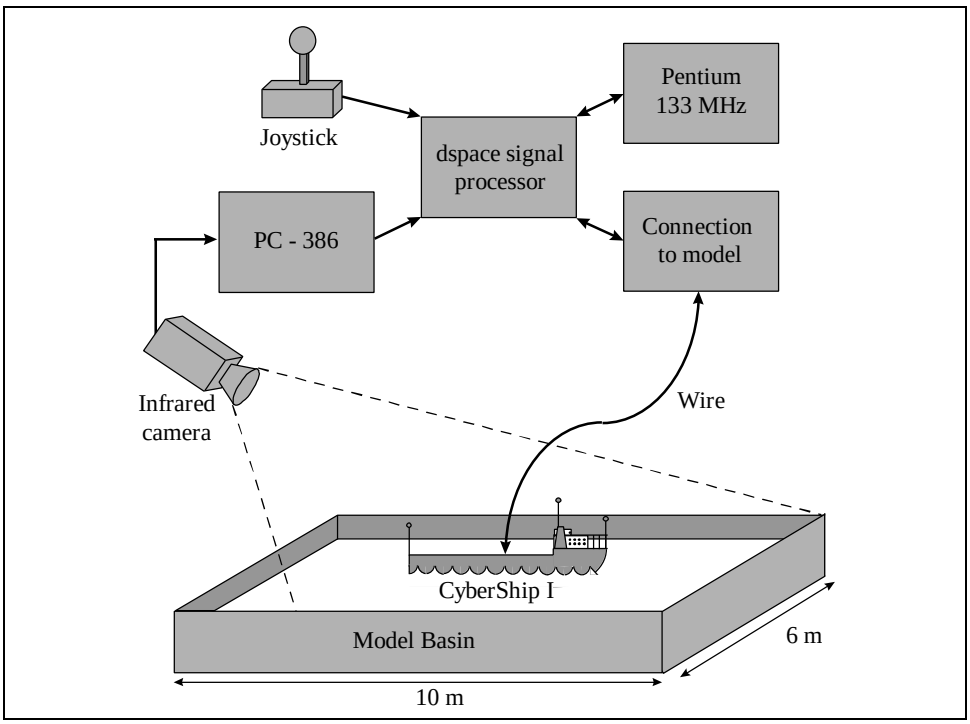

Figure 1: GNC Lab Schematic

Figure 1 shows that the GNC lab is centred around the *model basin* which provides the sailing environment for the ship. The other pieces of hardware provide telemetry and control data for the ship as outlined below.

CyberShip I (see Figure 2) is a 1:70 scale model of a supply ship for oil platform support. Not only does this call for the ship to carry out course changing manoeuvres it also needs to keep position well during loading and unloading. Therefore instead of a rudder/propeller configuration which is common to most ships, this vessel has four thrusters (i.e. two at the bow and two at the stern). The direction and force of these thrusters can be changed in order to manoeuvre the vessel in the desired manner.

The *Infrared Camera* provides positional information for the ship in the basin. It achieves this by locating the position of the three masts on the model.

The *PC-386* computer receives the positional data from the camera and calculates the velocities and heading of the ship as well as its position in terms of tank co-ordinates.

The *Pentium 133MHz* computer is used to implement the feedback control system in software. It does this by running Matlab Simulink programs which uses the telemetry data to provide commands that are intended to manoeuvre the ship. The *Joystick* allows the operator to input commands to the system and hence manoeuvre the ship. It can also be used to operate the ship in open loop without the used of a control system.

The *Wire Connection to the Model* carries the commands to the ship thruster servos and carries the actual thruster motion back to the computer system. Although at present this is a wire connection, in future this will be carried out by a radio transmitter system.

The *dspace signal processor* is the hub of the whole system. It processes the data from the other hardware elements and sends out the control system signals to the ship. Hence it is a crucial part of the system. The dspace system can be matched well to the Matlab Simulink software to provide an easy to use block diagram environment for the operator.

The complete system gives an ideal test facility for trials of marine control systems that could execute various operation roles for this ship (e.g. course changing, dynamic position keeping).

2.2 Ship Model Dynamics

The dynamics of the ship model are derived from the hydrodynamic equations of motion. These are obtained from considering the motion of the ship in body fixed and earth fixed axis systems. The motion is defined by linear and angular velocities for the body fixed reference frame of Figure 2 [1,5]. Also represented in this model is the co-ordinate position of the vessel in the earth fixed reference frame (i.e. x_p ,y_p) [1,5].

These reference frames provide two sets of equations, the *kinetic* and *kinematic* equations. The kinetic equations define the hydrodynamics of the vessel in the body-fixed reference frame and is represented by the following matrix equation (see Appendix A) [1].

$$\mathbf{M}\dot{\boldsymbol{n}} + \mathbf{C}(\boldsymbol{n})\boldsymbol{n} + \mathbf{D}\boldsymbol{n} = \boldsymbol{t} \tag{1}$$

Here **M** is the mass and inertia matrix, **C** contains the Coriolis terms and **D** is the damping matrix of the vessel. The vector $\boldsymbol{n}$ represents the body-fixed velocities (i.e. $\boldsymbol{n} = [u, v, r]^T$ and $\boldsymbol{t}$ is the input force vector provided by the thrusters (i.e. $\boldsymbol{t} = [t_1, t_2, t_3]^T$ where t_1 is the thrust vector along the body fixed x-axis, t_2 is the thrust vector along the body fixed y-axis and t_3 is the thrust about the body fixed z axis).

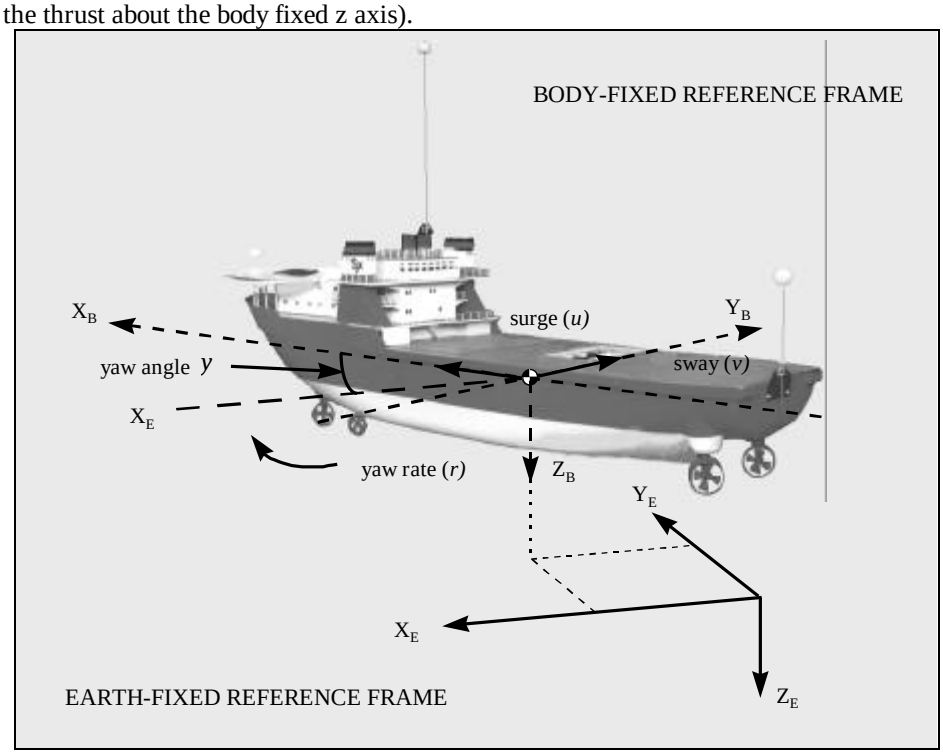

Figure 2: Ship Co-ordinate Frame

When equation (1) is rearranged the following differential equation is obtained.

$$\dot{\boldsymbol{n}} = -\mathbf{M}^{-1}\big(\mathbf{C}(\boldsymbol{n}) + \mathbf{D}\big)\boldsymbol{n} + \mathbf{M}^{-1}\boldsymbol{t} \tag{2}$$

The second set is the kinematic equations which define the geometrical relationship of the motion of the vessel relative to the earth-fixed frame of reference. These are represented by the following equation (see Appendix B):

$$\dot{\boldsymbol{h}} = \mathbf{J}(\boldsymbol{h})\boldsymbol{n} \tag{3}$$

Here **J** is the Euler equations relating the two reference frames and $\boldsymbol{h}$ represents the earth-fixed states (i.e. $\boldsymbol{h} = [x_p, y_p, y]^T$).

When equations (2) and (3) are combined the following matrix form is produced

$$\begin{bmatrix}\dot{\boldsymbol{n}}\\ \dot{\boldsymbol{h}}\end{bmatrix} = \begin{bmatrix}-\mathbf{M}^{-1}\big(\mathbf{C}(\boldsymbol{n}) + \mathbf{D}\big) & \mathbf{0}\\ \mathbf{J}\big(\boldsymbol{h}\big) & \mathbf{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{n}\\ \boldsymbol{h}\end{bmatrix} + \begin{bmatrix}\mathbf{M}^{-1}\\ \mathbf{0}\end{bmatrix}\boldsymbol{t} \tag{4}$$

This can be represented by the following state space equation.

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}\boldsymbol{t} \tag{5}$$

Here **x** is the state vector and **A(x)** is a non-linear system matrix which depends on the system's states and **B** is the input matrix.

2.3 Thruster Dynamics

The thrusters provide the driving forces that propel the vessel. *CyberShip I* has four thrusters in the configuration shown in Figure 3. Here it can be seen that the position of the thrusters is given relative to the centre of gravity which is also the origin for the body fixed reference frame.

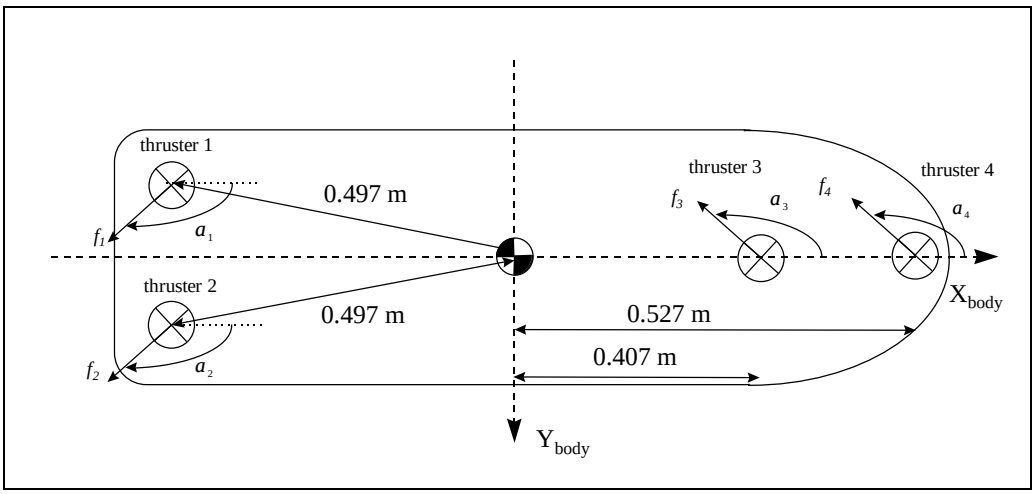

Figure 3: Thruster Configuration

In this model α_1 and α_2 can be set independently of each other whereas α_3 and α_4 are always equal showing that their corresponding thrusters operate in the same direction.

For the thrusters used in this model the following magnitude limits are defined. The maximum force values that can be provided by each of the thrusters (i.e. $f_{1..4}$) are estimated at ± 1.2 N. Their corresponding azimuth angle limits are estimated at $\pm \pi$ radians. These estimates provide physical limitations for the thrusters considered in this study.

The thruster forces and angles are related to the thruster inputs of equation (5) through the following trigonometric relationship [1].

$$\boldsymbol{t} = \mathbf{T}(\boldsymbol{a})\mathbf{f} \tag{6}$$

where $\mathbf{f} = [f_1, f_2, f_3, f_4]^T$, $\mathbf{a} = [a_1, a_2, a_3, a_4]^T$ and

$$\mathbf{T}(a) = \begin{bmatrix} \cos a_1 & \cos a_2 & \cos a_3 & \cos a_3 \\ \sin a_1 & \sin a_2 & \sin a_3 & \sin a_3 \\ 0.497\sin(a_1 - q_1) & 0.497\sin(a_2 - q_2) & 0.407\sin a_3 & 0.527\sin a_3 \end{bmatrix} \tag{7}$$

where q_1 and q_2 are the phase shift angles of the thrusters relative to the centre of gravity.

For this study the thruster angles are fixed at the following values : $a_1 = a_2 = \pi$ radians, $a_3 = a_4 = \pi/2$ radians. With these values and neglecting very small components, matrix $\mathbf{T}(\mathbf{a})$ becomes

$$\mathbf{T}(a) = \begin{bmatrix} -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0.407 & 0.527 \end{bmatrix} \tag{8}$$

This indicates that the surge motion is governed by thrusters 1 and 2 at the stern and the sway and yaw motion is governed by thrusters 3 and 4. Although a more complex configuration could have been implemented it is felt that this one was adequate for this investigation.

3. Sliding Mode Autopilot

In the context of this study the autopilot is a course changing control system. The particular control law which is used here is the non-linear *sliding mode* (SM) control which is derived from a linear state space representation of the ship [5,6] i.e.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{t} \tag{9}$$

As in equation (5) $\mathbf{x}$ is the state vector, $\mathbf{A}$ is the linearised system matrix, $\mathbf{B}$ is the input matrix and $\boldsymbol{t}$ represents the inputs of the system. This linear system is obtained by linearising the system matrix (particularly the Coriolis cross coupling components) about an equilibrium operating position for the ship. This results in a linear Multi-Input, Multi-State (MIMS) system which describes the dynamics of the ship about equilibrium [5,6]. Some of the states variables do not affect the heading motion significantly and need not be incorporated within the control system. Therefore not all of

the variables are required for a course changing autopilot and only the yaw rate thrust input (t_3) is used to control the course [5-7]. This reduces the order of the system and input matrices to give $\mathbf{A}_h \subset \mathbf{A}$ and $\mathbf{b}_h \subset \mathbf{B}$ as the subsystem equivalent forms of the system and input matrices. The resulting subsystem has the following standard state space form.

$$\dot{\mathbf{x}}_h = \mathbf{A}_h \mathbf{x}_h + \mathbf{b}_h t_3 \tag{10}$$

where $\mathbf{x}_h$ is the subsystem state and t_3 is the single input that governs the subsystem motion. This is the representation of the system which is used in the derivation of the control law.

For this type of controller the input consists of two distinct parts i.e. the linear equivalent control effort, t_{eq}, and the non-linear switching action t_{sw}. These are combined to form the input [1,5-7,10-13]:-

$$t_3 = t_{eq} + t_{sw} \tag{11}$$

Here the nominal equivalent control part is selected to be a state feedback gain controller of the following form.

$$t_{eq} = -\mathbf{k}_h{}^T \mathbf{x}_h \tag{12}$$

where $\mathbf{k}_h$ is a feedback gain obtained from robust pole placement theory [14]. This theory is regarded as robust since it avoids the numerical uncertainties associated with Ackermann's work [15]. Here the feedback gains are obtained by minimising the sensitivity of the assigned poles to disturbances [14]. Since this feedback control law is designed around a nominal linear plant it would not necessarily work well for all operating conditions of the vessel. Therefore it is not adequate for the control purposes of this study. This is overcome by adding the switching term to the control effort which compensates for any deviation from this nominal operating condition.

The non-linear switching term is derived from the intrinsic building block of SM control i.e. the *sliding surface* [1,5-7]. This defines the tracking dynamics of the switching action of the controller used. In this study the following sliding surface is used.

$$s_h(\Delta\mathbf{x}) = \mathbf{h}_h{}^T \Delta\mathbf{x}_h = \mathbf{h}_h{}^T \left(\mathbf{x}_h - \mathbf{x}_{hd}\right) \tag{13}$$

Here $\mathbf{h}_h$ is defined as the right eigenvector of the closed loop system matrix and $\Delta\mathbf{x}_h$ is the state error vector. On differentiating this with respect to time the following is obtained.

$$\dot{s}_h(\Delta\mathbf{x}) = \mathbf{h}_h{}^T \Delta\dot{\mathbf{x}}_h = \mathbf{h}_h{}^T \left(\dot{\mathbf{x}}_h - \dot{\mathbf{x}}_{hd}\right) \tag{14}$$

Substituting equation (12) for $\dot{\mathbf{x}}_h$ in the above equation gives

$$\dot{s}_h(\Delta\mathbf{x}) = \mathbf{h}_h{}^T \left(\mathbf{A}_h \mathbf{x}_h + \mathbf{b}_h u_h - \dot{\mathbf{x}}_{hd}\right) \tag{15}$$

and substituting for t_3 results in the following equation.

$$\dot{s}_h(\Delta\mathbf{x}) = \mathbf{h}_h{}^T \left(\mathbf{A}_h \mathbf{x}_h + \mathbf{b}_h t_{eq} + \mathbf{b}_h u_{sw} - \dot{\mathbf{x}}_{hd}\right) \tag{16}$$

If equation (12) is used to replace t_{eq} then the above equation becomes

$$\begin{aligned} \dot{s}_h(\Delta\mathbf{x}) &= \mathbf{h}_h{}^T \left(\mathbf{A}_h \mathbf{x}_h - \mathbf{b}_h \mathbf{k}_h{}^T \mathbf{x}_h + \mathbf{b}_h t_{sw} - \dot{\mathbf{x}}_{hd}\right) \\ &= \mathbf{h}_h{}^T \left(\mathbf{A}_{ch} \mathbf{x}_h + \mathbf{b}_h t_{sw} - \dot{\mathbf{x}}_{hd}\right) \end{aligned} \tag{17}$$

where

$$\mathbf{A}_{ch} = \mathbf{A}_h - \mathbf{b}_h \mathbf{k}_h{}^T \tag{18}$$

is the closed loop system matrix created by the feedback gain. Hence the eigenvectors of $\mathbf{A}_{ch}$ can be defined in terms of the feedback gain vector. Rearranging (19) yields

$$t_{sw} = \left(\mathbf{h}_h{}^T \mathbf{b}_h\right)^{-1} \left(\mathbf{h}_h{}^T \dot{\mathbf{x}}_{hd} - \mathbf{h}_h{}^T \mathbf{A}_{ch} \mathbf{x}_h - \dot{s}_h(\Delta\mathbf{x})\right) \tag{19}$$

This assumes that $\mathbf{h}_h{}^T \mathbf{b}_h$ is nonzero. Since $\mathbf{h}_h{}^T$ is chosen as the right eigenvector of $\mathbf{A}_{ch}$ it therefore corresponds to an zero eigenvalue of this matrix. Hence it provides the following relationship

$$\mathbf{h}_h{}^T \mathbf{A}_{ch} = \left(\mathbf{A}_{ch}{}^T \mathbf{h}_h\right)^T = 0 \tag{20}$$

Therefore (19) becomes

$$t_{sw} = \left(\mathbf{h}_h{}^{\mathbf{T}}\mathbf{b}_h\right)^{-1}\left(\mathbf{h}_h{}^{\mathbf{T}}\dot{\mathbf{x}}_{hd} - \dot{s}_h(\Delta\mathbf{x})\right) \tag{21}$$

From [5] $\dot{s}_h$ is defined as

$$\dot{s}\left(\Delta\mathbf{x}_h\right) = -h_h \operatorname{sgn}\left(s\left(\Delta\mathbf{x}_h\right)\right) \tag{22}$$

Here h_h is the switching gain which determines the amount of switching control action which characterises this kind of control. The switching action itself is provided by the signum function which simply indicates the sign of the sliding surface i.e.

$$\operatorname{sgn}\left(s\left(\Delta\mathbf{x}_h\right)\right) = \begin{cases} 1 & \text{if } s \geq 0 \\ 0 & \text{if } s = 0 \\ -1 & \text{if } s < 0 \end{cases} \tag{23}$$

This determines which way the control effort should be applied in order to drive s and ultimately $\Delta\mathbf{x}_h$ to zero. The size of this switching action depends on the magnitude of the switching gain.

When (22) is applied to (21) the following is obtained

$$t_{sw} = \left(\mathbf{h}_h{}^{\mathbf{T}}\mathbf{b}_h\right)^{-1}\left(\mathbf{h}_h{}^{\mathbf{T}}\dot{\mathbf{x}}_{hd} - h_h \operatorname{sgn}\left(s\left(\Delta\mathbf{x}_h\right)\right)\right) \tag{24}$$

By combining (12) and (24) into (11) the total controller equation is obtained.

$$t_3 = -\mathbf{k}_h{}^{\mathbf{T}}\mathbf{x}_h + \left(\mathbf{h}_h{}^{\mathbf{T}}\mathbf{b}_h\right)^{-1}\left(\mathbf{h}_h{}^{\mathbf{T}}\dot{\mathbf{x}}_{hd} - h_h \operatorname{sgn}\left(s\left(\Delta\mathbf{x}_h\right)\right)\right) \tag{25}$$

The only reported drawback with this controller is that the signum function provides a hard switching action which can cause high frequency oscillations in the control signal when the control tracking approaches the zero sliding surface [1,5-7,13]. In order to remove this *chattering* (as it is called) the signum may be replaced by a function that exhibits some sort of gradual transition as the tracking approaches the zero sliding surface. A suitable candidate has been found in the continuous tanh function. This provides a gradual decrease as the sliding surface approaches zero and results in the following controller equation.

$$t_3 = -\mathbf{k}_h{}^{\mathbf{T}}\mathbf{x}_h + \left(\mathbf{h}_h{}^{\mathbf{T}}\mathbf{b}_h\right)^{-1}\left(\mathbf{h}_h{}^{\mathbf{T}}\dot{\mathbf{x}}_{hd} - h_h \tanh\left(s\left(\Delta\mathbf{x}_h\right)/f_h\right)\right) \tag{26}$$

Here the size of transition zone or *boundary layer* is defined by f_h in the above equation. This parameter is called the *boundary layer thickness* [1,5-7]. Hence this is the structure of the controller which is used in this study to provide course changing input commands for the thrusters. The implementation of this controller is in Simulink code run on the Pentium described in the previous section.

4. Genetic Algorithm Theory

Genetic Algorithms (GAs) provide a basis for an optimisation method which is thought to be one of the most powerful available at present [5,6,8-12]. They are based on the natural selection process outlined in the Darwinian theory of *survival of the fittest*. This theory states that species evolve through their fittest genetic variation. In this context, fittest means the strongest, healthiest and most intelligent genus. Therefore, in order for a species to survive it must adapt to its surroundings by utilising and improving its abilities. As time goes on, the strongest become stronger and the weakest fade out until the species reaches its evolutionary optimum [5,6,8-12].

4.1 The Genetic Algorithm

A GA explores the problem search space by using strings of integers as a representation of the parameters to be optimised. These strings are called *chromosomes* and their individual integer components are called *genes* (which have a value range (called *alleles*) from 0 to 9 in this work). A number of these chromosomes are initially generated at random and are called the *population*. The number of chromosomes is the *population size*. The initial population is the first *generation* and is evaluated by the following stages (see Figure 4) [5,6,9].

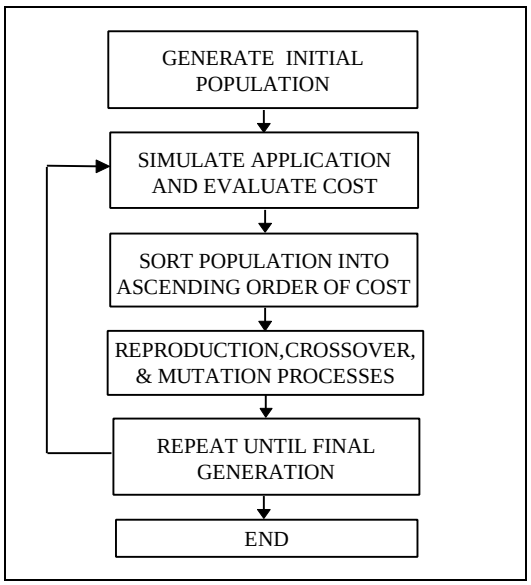

Figure 4: GA Flow Diagram

Firstly, the chromosome is decoded from its integer representation into the form used in the optimisation problem (usually real numbers). Then the decoded form is applied to the problem in question (i.e. ship control in this case) which is then simulated. The simulation data is then used to evaluate the chromosome by obtaining a value of the *cost*. This cost value is used to determine how well the present solution and corresponding chromosome is performing in terms of a predetermined set of guidelines. In this study the smaller the cost value is, the better the response. This evaluation process is carried out for every chromosome in the population. After all the cost values are obtained they are subjected to a selection process which arranges the chromosomes into descending cost order. Then the operations of *Reproduction*, *Crossover* and *Mutation* are executed in order to change the chromosomes and search in different areas of the search space.

Reproduction is where the best chromosomes of the present population (typically the top 10 - 20%) are kept for the next population. The remainder are replaced by new chromosomes which are formed through the crossover and mutation of the present population. This reproduction is called *rank based selection*. Since only the elite chromosomes remain this type of optimisation technique is called an *Elite Genetic Algorithm* [5,6,8].

Crossover takes any two chromosomes from the present generation (these are called the *parents*), selects a number of one of the parents genes and exchanges them with the same number and positioned genes in the other parent [5,6,8-12]. This forms two new chromosomes called the *children.* This is repeated until there are enough children to replace the 80 - 90 % of the present population which have the poorest cost values .

Mutation is simply the random selection of a percentage of the new population's genes and the random change of these genes' values [5,6,8-12] (i.e. random change of genes in the range between 0 and 9). The elite chromosomes are unaffected by this operation.

After the chromosomes have been altered to form the new population, they are evaluated in the same way as the first population (see above). Then the processes of cost evaluation, ranking, reproduction, crossover and mutation are carried out for a set number of iterations. This number is called the *generation size* and when it is reached, the GA should be near the optimum. Usually the final population will have a number of similar chromosomes which add validity to the optimal region and give more confidence to the final result.

4.2 Cost Function

The cost function used as the design criterion in the course changing sections of this investigation is defined by equation (27) [2,3,5,6]. This function is fundamentally a discrete version of the integral least squares criterion.

$$C = \sum_{i=0}^{m} \left[(\Delta y_i)^2 + (t_{3i})^2 \right] \tag{27}$$

Here m is the total number of iterations, Δy_i is the ith heading angle error between the desired and obtained heading and t_{3i} is the ith thruster force value in yaw [1,2,3,5,6]. Since the GA is trying to minimise the value of this function it is easy to see that both Δy_i and t_{3i} will be minimised too. The reasoning behind this selection of elements for the cost

function is as follows. The quantity Δy_i gives an indication of how close the actual heading is to the desired heading, therefore showing how well the controller is operating. However, t_{3i} is used to keep the yaw thruster force value to a minimum so that it can operate well within its operational limits. This is of particular importance with SM controllers which have a tendency to chatter if the switching gain and boundary layer values are not chosen properly. Therefore there is a trade off between heading tracking accuracy and thrust magnitude. Another advantage of minimising the thruster force is the savings in terms of fuel consumption since the resistance to the forward motion is minimised [3,5]. Since the thruster force is reduced, the hull produces less drag and hence more of the forward force goes to producing a larger surge velocity. However this results in a slower turning rate.

5. Results

5.1 Controller Parameter Optimisation

Key design parameters have been chosen to be obtained by the GA in order to optimise the performance of the autopilot. Four parameters are used here (see Table 1) which alter the operational ability of the autopilot to execute course changing manoeuvres. The first two parameters (*ph1*, *ph2*) are two poles of the decoupled closed system (the third is zero and corresponds to the yaw dynamics). These poles are used to define the feedback gain vector $\mathbf{k}_h$ that constitutes the equivalent controller in equation (12). This is then used to calculate $\mathbf{A}_{ch}$ from equation (18) and subsequently $\mathbf{h}_h$ which is a major component of this autopilot. The final two design parameters are h_h and f_h which are related to the switching action of this autopilot. As mentioned previously the switching gain h_h determines the magnitude of the switch and f_h defines the size of the boundary layer.

Table 1: Parameters to be optimised

Autopilot Parameters	
1st Heading Closed loop pole	p_{h1}
2nd Heading Closed loop pole	p_{h2}
Heading switching gain	h_h
Heading Boundary Layer Thickness	f_h

Although the GA can provide a suitable autopilot by optimising these four parameters, a desired heading response is required so that the same manoeuvre is used throughout the optimisation. For this application a 45^O turn is chosen which gives the critically damped step shown in Figure 5.

This provides the desired states for the controller and thus enables the state error vector to be evaluated. From these the value of the heading error can be obtained and used to calculate the cost (equation (27)) for each solution that the GA generates. Then the costs are used to evaluate each solution in the way described above.

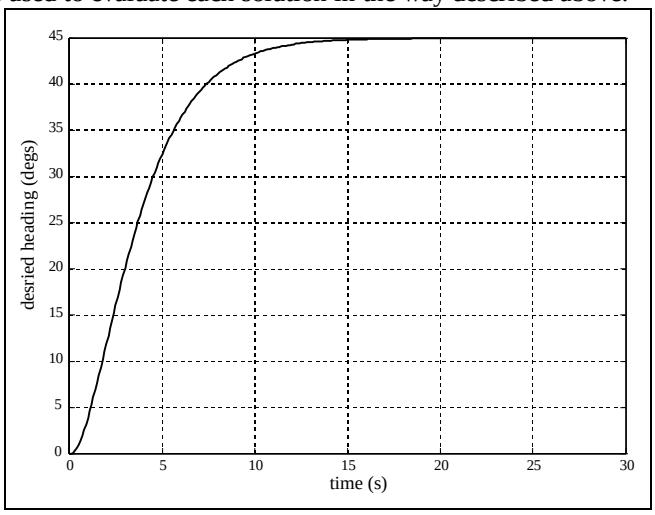

Figure 5: Desired Heading Response (45^O manoeuvre)

In this investigation a GA with a population size of 50 and generation size of 100 is used. Through repeated application of this optimisation method the set of parameter values shown in Table 2 are obtained which are typical of the values obtained for this particular application.

Table 2: Optimised Parameter Values

Autopilot Parameter Values	
1st Heading Closed loop pole (*ph1*)	-2.2092
2nd Heading Closed loop pole (*ph2*)	-0.2059
Heading switching gain (h_h)	8.1620
Heading Boundary Layer Thickness (f_h)	1.2851

These give the simulated responses shown in Figure 6 which show the heading (y), the heading error (Δy) and the thrust commanded for this manoeuvre (t_3). It can be seen that the results appear consistent with the design criteria that constitute the cost function since the peak error magnitude is very small (i.e. 0.1^O which 0.2 % of the desired heading) and tracks to zero in the steady state. It is also apparent that the thrust magnitude is small and that the vessel executes this manoeuvre well within its operational envelope.

From these results it can be deduced that the GA has optimised the autopilot parameters to satisfy the design criteria. However these are simulated responses for a single manoeuvre and may not operate well for different simulated manoeuvres or when it is applied to the actual system. Therefore further evaluation of this result is required.

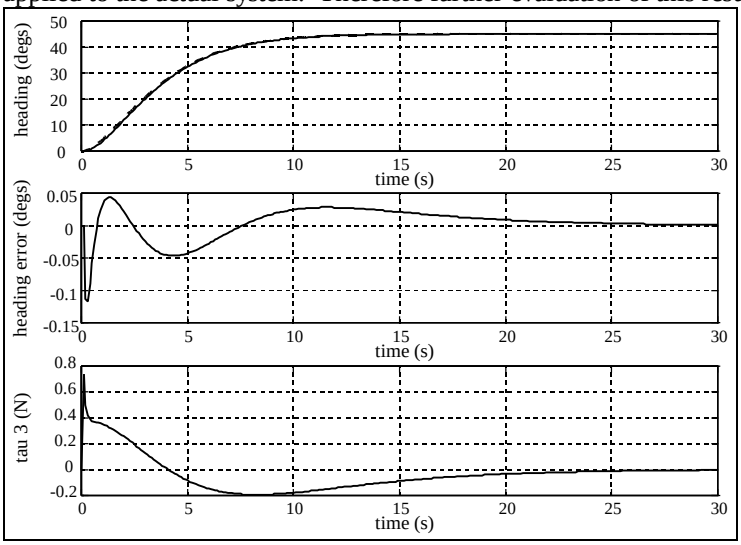

Figure 6: Optimised Controller Simulation Response (45^O manoeuvre)

5.2 Optimised Controller Evaluation

5.2.1 Simulation Results

In order to test the optimised controller further it is simulated using another manoeuvre. A 20^O/-20^O manoeuvre is chosen which gives the desired heading response shown in Figure 7 [1].

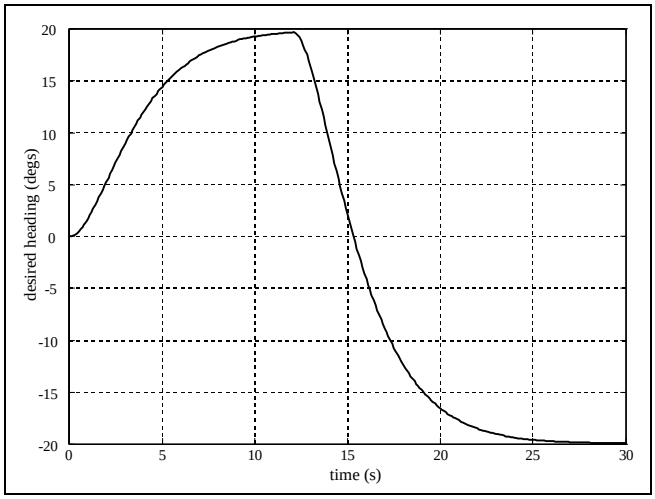

Figure 7: Desired Heading Response (20^O/-20^O manoeuvre)

The above response provides a manoeuvre which will assess any asymmetry with in the motion of the vessel since it commands positive and negative turns. When the autopilot is simulated for this desired heading the set of responses shown in Figure 8 is obtained. As with the optimised responses of the heading, heading error and thruster time histories are shown. Again the error and thruster force are small and the heading tracks the desired response well. This indicates that the autopilot can handle numerous course changing commands and is not designed specifically for a single manoeuvre. Therefore it can be deduced that the autopilot solution provided by the GA can be used for general course changing operations.

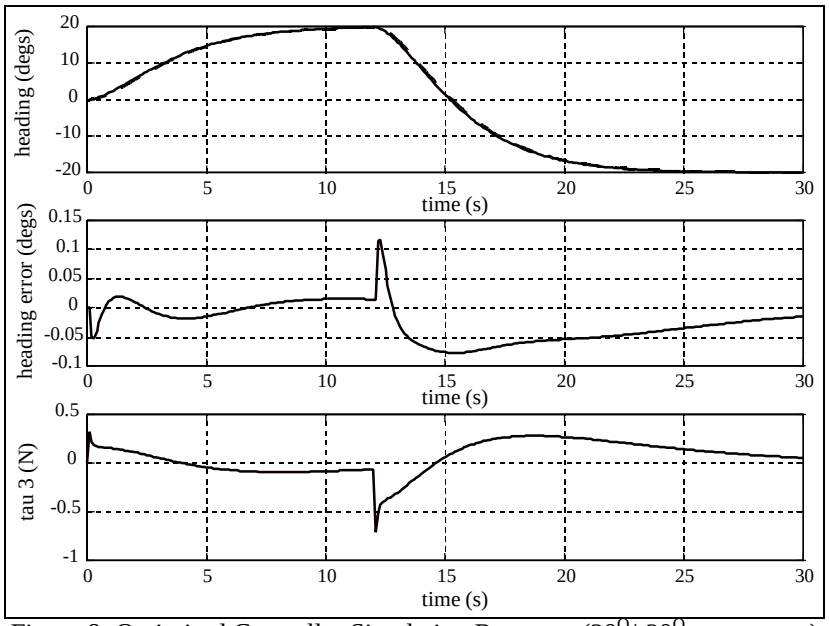

Figure 8: Optimised Controller Simulation Response (20^O/-20^O manoeuvre)

5.2.2 Experimental Results

The final test is the implementation of this autopilot for course changing on the GNC scale model. So that a comparison with the simulation results can be carried out, the responses for a 20^O/-20^O manoeuvre are given in Figure 9.

These show a slight difference from the simulated responses for this manoeuvre. It can be clearly seen that both the peak heading error and thruster commands are larger than in the simulated case. There are three main reasons for this difference. Firstly it has been found that the mathematical representation of the model is not accurate and therefore gives slightly unrealistic responses in the simulation study. The second aspect is that very little consideration is given to the drag caused by the position of the thrusters. Since the bow thrusters are perpendicular to the flow over the hull they cause maximum drag and this affects the motion of the model. This could account for the slight offset in the thruster plot. Finally the roll motion and water disturbance effects are also not considered and can alter the motion of the model in the lab's model basin.

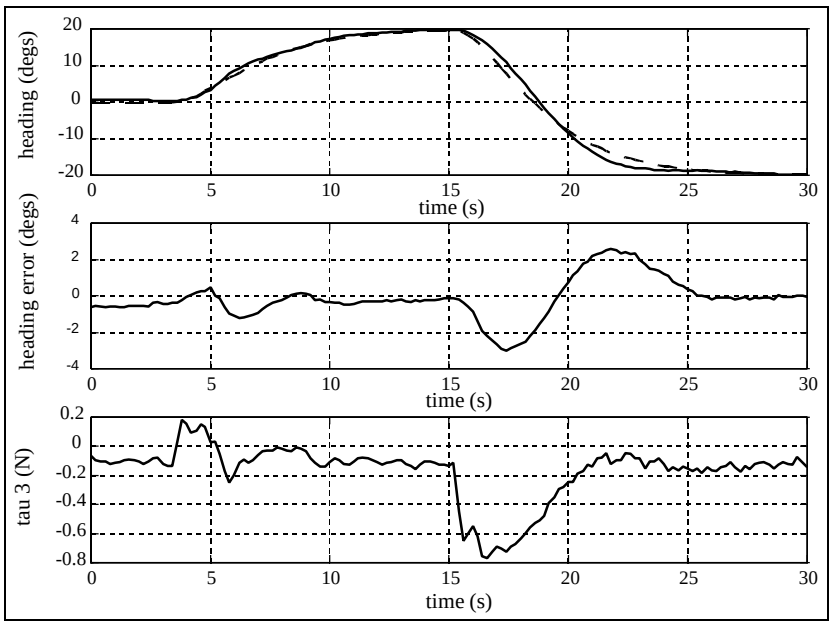

Figure 9: Optimised Controller Model Basin Responses (20^O/-20^O manoeuvre)

Although these aspects could alter the motion considerably, their influence does not substantially degrade the performance of the autopilot. It still operates satisfactorily in that the errors are relatively small and the thruster forces are within their physical limits. From these responses it can be seen that the autopilot is still able to track the desired response adequately and a minor amount of further tuning could yield a slightly better solution. However the solution provided by the GA from simulated data has been shown to operate adequately when applied to the actual scale model.

6. Conclusions

In this paper a study of autopilot parameter optimisation and evaluation has been presented. The successful optimisation of a sliding mode (SM) autopilot for simulated course changing manoeuvres has been shown for a 45^O heading change. The resulting autopilot was then evaluated for a simulated 20^O/-20^O manoeuvre and shown to work equally well when the operating conditions for the ship are changed. Thus the SM autopilot is shown through simulation to robust performance.

Finally the successful implementation of this autopilot in the GNC lab allowed it to be evaluated in terms of a physical application. Although the actual performance in this case was slightly poorer during a 20^O/-20^O manoeuvre the autopilot still operated well in terms of the criteria for which it was optimised. The resulting autopilot solution could be further improved if the mathematical model of the ship was more accurate.
From a practical view point it can be seen that the genetic algorithm optimisation technique can be used as a powerful design tool for the control engineer. It is able to provide key parameter values for controllers that with very little further tuning (or none at all) can be implemented successfully on a real system.

Acknowledgements

The authors wish to thank The Scottish International Education Trust and The Institute of Electrical Engineers' Hudswell Bequest Travelling Fellowship for funding this research.

References

1. Fossen, T. I., "Guidance and Control of Ocean Vehicles", John Wiley & Sons Ltd, Chichester, 1994.
2. Kallstrom, C.G., Astrom, K.J., Thorell, N.E., Eriksson, J. and Sten, L., "Adaptive Autopilots for Tankers", Automatica, Vol. 15, pp 241-254, 1979.
3. Dove, M. J. and Wright, C. B., "Development of Marine Autopilots", Computer Methods in Marine and Offshore Engineering, Computational Mechanics Publications, Southampton, pp 259-272, 1991,
4. Vik, B. and Fossen, T.I., "Semiglobal Exponential Output Feedback Control of Ships", IEEE Transactions on Control Systems Technology, Vol.5, No.3, pp 360-370, 1997.
5. Healey, A. J. and Lienard, D., "Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles", IEEE Journal of Oceanic Engineering, Vol. 18, No. 3, pp 327-339, 1993.
6. Goldberg, D., "Genetic Algorithms in Searching. Optimisation and Machine Learning", Addison Wesley, Reading, MA, 1989.

7. M^{c}Gookin, E. W., Murray-Smith, D. J. and Li, Y., "Submarine Sliding Mode Controller Optimisation using Genetic Algorithms", International Conference on Control '96, Exeter, UK, Vol.1, pp 424-429, 1996.
8. M^{c}Gookin, E. W., Murray-Smith, D. J., Li, Y. and Fossen, T.I., "Parameter Optimisation of a Non-Linear Control System using Genetic Algorithms", To be published in the Proceedings of Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, Glasgow, UK, 1997.
9. Li, Y., Ng, K. C., Tan, K. C., Gray, G. J., McGookin, E. W., Murray-Smith, D.J., and Sharman, K.C., "Automation of linear and non-linear control systems design by evolutionary computation", Proceedings of the International Federation of Automatic Control Youth Automation Conference, Beijing, China, pp 53-58, 1995.
10. Li, Y., Ng, K. C., Murray-Smith, D. J., Gray, G. J. and Sharman, K. C., "Genetic Algorithm Automated approach to design of Sliding Mode Controller systems", International Journal of Control, 63(4), pp 721-739. 1996.
11. Ng, K. C., Li, Y., Murray-Smith, D. J. and Sharman, K. C., "Genetic Algorithm applied to Fuzzy Sliding Mode Controller design", First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, pp 220-225, Sheffield, UK, 1995.
12. Brooks, R. R., Iyengar, S. S. and Chen, J., "Automatic correlation and calibration of noisy sensor readings using elite genetic algorithms", Artificial Intelligence 84, pp 339-354, 1996.
13. Burton, J.A. and Zinober, A.S.I., 'Continuous self-adaptive control using a smoothed variable controller', International Journal of Systems Science, Vol.19, No.8, pp 1515-1528, 1988.
14. Kautsky, J., Nichols, N. K. and van Doorens, P.,"Robust pole assignment in linear state feedback", International Journal of Control, Vol.41, No.5, pp 1129-1155, 1985.
15. Franklin, G.F., Powell, J.D. and Emami-Naeini, A., "Feedback Control of Dynamic Systems", 2nd Edition, Addison Wesley, New York, 1991.

Appendix A

The kinetics of *CyberShip I* are represented by the following equation

$$\mathbf{M}\dot{\boldsymbol{n}} + \mathbf{C}(\boldsymbol{n})\boldsymbol{n} + \mathbf{D}\boldsymbol{n} = \boldsymbol{t}$$

where $\boldsymbol{n} = [u, v, r]^{\mathrm{T}}$, $\boldsymbol{t} = [t_1, t_2, t_3]^{\mathrm{T}}$,

$$\mathbf{M} = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & m_{23} \\ 0 & m_{32}\left(=m_{23}\right) & m_{33} \end{bmatrix} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_G - Y_{\dot{r}} \\ 0 & mx_G - N_{\dot{v}} & I_z - N_{\dot{r}} \end{bmatrix} = \begin{bmatrix} 19.0 & 0 & 0 \\ 0 & 35.2 & 0 \\ 0 & 0 & 2.0 \end{bmatrix}$$

$$\mathbf{C}(n) = \begin{bmatrix} 0 & 0 & -m_{22}v - m_{23}r \\ 0 & 0 & m_{11}u \\ m_{22}v + m_{23}r & -m_{11}u & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -35.2v \\ 0 & 0 & 19.0u \\ 35.2v & -19.0u & 0 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & 0 \\ 0 & 0 & -N_r \end{bmatrix} = \begin{bmatrix} m_{11}/T_1 & 0 & 0 \\ 0 & m_{22}/T_2 & 0 \\ 0 & 0 & m_{33}/T_3 \end{bmatrix} = \begin{bmatrix} 6.3 & 0 & 0 \\ 0 & 7.0 & 0 \\ 0 & 0 & 2.0 \end{bmatrix}$$

given that the time constants for surge, sway and yaw are T_1 = 3.0s, T_2 = 5.0s and T_3 = 1.0s respectively.

Appendix B

The kinetics of *CyberShip I* are represented by the following equation

$$\dot{\boldsymbol{h}} = \mathbf{J}(\boldsymbol{h})\boldsymbol{n}$$

where $\boldsymbol{n} = [u, v, r]^{\mathrm{T}}$, $\boldsymbol{h} = [x_p, y_p, y]^{\mathrm{T}}$ and

$$\mathbf{J}(h) = \begin{bmatrix} \cos y & -\sin y & 0 \\ \sin y & \cos y & 0 \\ 0 & 0 & r \end{bmatrix}$$