

Genetic Algorithm Optimisation of a Ship Navigation System

E. Alfaro-Cid, E. W. McGookin, D. J. Murray-Smith

The optimisation of the PID controllers' gains for separate propulsion and heading control systems of CyberShip I, a scale model of an oil platform supply ship, using Genetic Algorithms is considered. During the initial design process both PID controllers have been manually tuned to improve their performance. However this tuning approach is a tedious and time consuming process. A solution to this problem is the use of optimisation techniques based on Genetic Algorithms to optimise the controllers' gain values. This investigation has been carried out through computer-generated simulations based on a non-linear hydrodynamic model of CyberShip I.

Keywords: Ship control, PID controller, Genetic Algorithm optimisation.

1 Introduction

In order to ensure the safe navigation of surface vessels their motion has to be controlled accurately. This control can be provided through the application of control theory. In general, control theory provides design strategies that allow a better understanding of the system being controlled and a mechanism to regulate the way it operates. There are various control theories or methodologies that have their own structure. Unfortunately, not all these methods perform their controlling duties satisfactorily due to inherent limitations imposed by the controller structure. In addition, the performance of these controllers depends on the values of the controller's parameters. Conventionally, the designer manually tunes these parameters to find an acceptable solution. However, this relies on an ad hoc approach to tuning, which depends on the experience and qualitative judgement of the designer. This process can be very slow and there is no guarantee that the designed solution will perform satisfactorily.

A solution to this problem is to use optimisation techniques that tune such parameters automatically. The most powerful of these techniques is based on Genetic Algorithms (GAs) [1], [2], [3]. GAs are stochastic search methods that mimic the way species evolve in nature. They operate on a population of potential solutions applying the Darwinian principal of "survival of the fittest" to produce better and better possible solutions to a given problem. At each generation, a new set of candidate solutions is created by the process of selecting solutions that are better than others and breeding them together using operators borrowed from natural sexual reproduction in order to get populations of solutions that are better than the solutions they were created from, just as in natural adaptation.

This paper covers the optimisation of control systems for the propulsion and navigation of an oil platform supply ship using a GA. The particular vessel used in this study is a scale model called *CyberShip I* [4], [5], which is the test vehicle for the Guidance, Navigation and Control Laboratory at the Norwegian University of Science and Technology in Trondheim. Computer-generated simulations based on a non-linear hydrodynamic model of *CyberShip I* are used in the optimisation studies. These simulations have proven to be sufficiently representative of the full-scale manoeuvring dy-

namics of such a vessel. Hence the optimised controllers could be used to control the actual scale model without the need for further modification.

The investigation presented in the paper will represent part of a study into the optimisation of controller designs based on a number of different control methodologies. In this case the particular methodology considered is classical PID, a very simple and widespread controller. PID control is used to provide the structure for both propulsion and navigation controller. The goal of this study is to obtain controller solutions that satisfactorily track the desired heading and propulsion response while keeping actuator usage to a minimum.

The results obtained from this study illustrate the benefits of using GAs to optimise propulsion and navigation controllers for surface ships. The accuracy of the resulting simulations allows meaningful evaluation of the optimised controllers' performance.

2 Supply ship mathematical model

The dynamics of the vessel can be represented by the kinetics and kinematics equations. When kinetic and kinematic equations are combined together the following matrix equation is produced [4]

$$\begin{bmatrix} \dot{v} \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} -\mathbf{M}^{-1}(\mathbf{C}(v) + \mathbf{D}) & 0 \\ \mathbf{J}(\eta) & 0 \end{bmatrix} \cdot \begin{bmatrix} v \\ \eta \end{bmatrix} + \begin{bmatrix} \mathbf{M}^{-1} \\ 0 \end{bmatrix} \cdot \tau \quad (1)$$

which corresponds to a form of state space equation (i.e. $\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}) \cdot \mathbf{x} + \mathbf{B} \cdot \tau$).

Here \mathbf{M} , \mathbf{C} , \mathbf{D} and \mathbf{J} are the mass/inertia, Coriolis, damping and Euler matrices, $v = [u, v, r]^T$ is the body-fixed linear and angular velocity vector, $\eta = [x, y, \psi]^T$ denotes the position and orientation vector in the earth-fixed frame and $\tau = [\tau_1, \tau_2, \tau_3]^T$ is the input force vector, given that τ_1 is the thrust vector along the body fixed X-axis, τ_2 is the thrust vector along the body fixed Y-axis and τ_3 is the thrust vector along the Z-axis.

These three force vectors relative to the body-fixed axis that constitute the inputs to the vessel model are provided by four thrusters. Two of them are placed at the stern, symmetric with respect to the body-fixed X-axis, while the other two are placed at the bow, along the body-fixed X-axis [4]. Each

thruster is represented by the force (f_i) it produces and the angle (α_i) defining its direction. And the amplitude limit values for each of the thrusters' force and angle are $\pm 0.9\text{N}$ and $\pm\pi$ radians respectively [4]. The three input force components (τ) can be related to the individual forces produced by the thrusters (f_i) using trigonometric relationships [5].

Hence, the ship's model has 6 states (i.e. u , surge velocity; v , sway velocity; r , yaw rate; ψ , yaw angle; x_p , x-position on earth and y_p , y-position on earth) and 3 inputs (i.e. τ_1 , τ_2 and τ_3 , thrust forces along the body fixed X, Y and Z-axis respectively) [4].

Since Cybership I is a scale model of an actual type of supply vessel the simulations performed are scaled in order to generate results comparable to the response of full-scale ships.

3 Automatic control system

For most surface vessels there are two main sets of dynamics: propulsion and heading. Two classical PID controllers [6] have been used to provide the structure for the propulsion controller (for governing surge velocity) and the navigation controller (for governing heading). These controllers operate on the error signal, which is the difference between the desired output and the actual output, and generate the actuation signal (i.e. τ_1 and τ_3), which drives the plant (i.e. the vessel). They have three basic modes of operation: proportional action, in which the actuation signal is proportional to the error signal, integral action, when the actuation signal is proportional to the time integral of the error signal, and derivative action, where the actuation signal is proportional to the time derivative of the error signal, and their transfer function is given by:

$$D(s) = (K + K_I/s + K_D \cdot s) \cdot E(s). \quad (2)$$

These "three-term controllers" have been found to be so effective that PID is the standard control method used in the process industries. To design a particular control loop the three parameters (K , proportional gain, K_I , integral gain, and K_D , derivative gain) have to be "tuned" to arrive at acceptable performance.

4 Genetic Algorithm optimisation

GAs are optimisation techniques that mimic the way species evolve in nature. In natural evolution many organisms evolve by means of two mechanisms: natural selection and sexual reproduction. The concept of natural selection is described by the Darwinian theory of survival of the fittest. Sexual reproduction allows the offspring to inherit features from both its parents.

GAs emulate this process by evolving a population of parameter solutions through a number of generations. They initiate this process by randomly generating an initial population of possible parameters (suitably encoded). The performance of each solution is evaluated and a new generation is produced according to the three main operators of the GA: selection, crossover and mutation. Selection determines which solutions are chosen for mating according to the principle of survival of the fittest (i.e. the better the performance of the solution, the more likely it is to be chosen for mating and

the more offspring it produces). Crossover allows an improvement in the species in terms of the evolution of new solutions that are fitter than any seen before, and mutation reintroduces values that might have been lost through selection or crossover, or creates totally new features. The cycle is performed until a predetermined number of generations is met [1], [2].

In order to search the space of possible solutions the GA uses a string (called chromosome) of digits as a representation of the elements of the space of possible solutions (i.e. the possible solutions are suitably encoded).

In this study, the encoding of the parameters has been chosen to allow a range of possible solutions from $0.001 \cdot 10^{-2}$ to $9.999 \cdot 10^3$. Each controller parameter value is encoded as a string of five genes. As there are three parameters to optimise in the controller (i.e. K , proportional gain, K_I , integral gain, and K_D , derivative gain), each possible solution is represented by a chromosome that is a string of 30 genes (i.e. 15 genes represent the propulsion controller and 15 represent the heading controller). These genes, instead of being binary bits (as they used to be in the traditional GA), are integers included within the interval $[0, 9]$, in order to allow a wider range of possible solutions in smaller chromosomes [4].

$$\begin{aligned} \text{Parameter Encoding} = abcde &\Rightarrow \text{Parameter Value} = \\ &= (a + 0.1b + 0.01c + 0.001d) \cdot 10^{(e/2-2)}. \end{aligned} \quad (3)$$

Once an initial population of chromosomes is generated at random, the chromosomes are decoded to get the corresponding parameters and these are introduced in the controllers. A simulation is run and the result obtained for each set of controller's parameters of the population is evaluated, using the cost function to be minimized.

Based on this cost function the selection procedure, used to draw chromosomes from the evaluated population, takes place. There are three main types of algorithms for this (i.e. roulette wheel selection, tournament selection and rank-based selection [4]), but they all share the feature that the probability of selecting a chromosome for reproduction is a decreasing function of the chromosome's cost score.

Roulette wheel selection allows sub-optimal solutions to have a chance of being accepted. It ensures a good mix of good and bad solutions and prevents premature convergence to a local minimum, however it also leads to a very slow convergence rate. Therefore, to avoid this slow convergence, the selection scheme chosen for this study is a variation of roulette wheel selection. The population is sorted according to the actual cost and a new cost value is assigned to each chromosome depending only on its position in the chromosome's rank and not on the actual cost value.

Consider N_{chrom} the number of chromosomes in the population, Pos the position of a chromosome in this population and SP the selective pressure (i.e. the probability of the best chromosome being selected compared to the average probability of selection of all chromosomes). The new cost is calculated as [7]:

$$\text{Cost}(Pos) = 2 - SP + 2 \cdot (SP - 1) \cdot (Pos - 1) / (N_{chrom} - 1). \quad (4)$$

Then a biased roulette wheel is created where each chromosome in the population has a roulette wheel slot sized in proportion to its cost. To reproduce, the roulette wheel thus

defined is spinned as many times as chromosomes are in the population [3].

The fact of assigning this new cost value to each chromosome (instead of creating a biased roulette wheel straight from the actual cost values obtained after the simulation) allows a bigger differentiation among the chromosomes. In a even population where most chromosomes have approximately the same cost value, assigning new cost values depending only in the position of the chromosome once the population has been ranked, increases the chances of the fittest chromosomes being chosen for reproduction. This way the convergence to the minimum is faster [7].

Once the new population has been obtained, chromosomes are ready for mating and mutation. The crossover operator combines the features of two parents to create new solutions. One or several crossover points are selected at random and then complementary fractions from the two parents are spliced together to form a new chromosome [4]. In this study two-point crossover has been used because multi-point crossover encourages a greater amount of variation in the new chromosomes produced and two points do not greatly increase the complexity of the algorithm.

The mutation operator alters a copy of a chromosome. One or more locations are selected on the chromosome and replaced with new randomly generated values. Mutation is used to help insure that all areas of the search space remain reachable. The mutation probability chosen has been 5% [4].

5 Manual tuning results

This investigation has been carried out through simulation studies in MATLAB. The accuracy of computer-generated simulations based on this mathematical model permits the use of such simulations as a method for evaluating the performance of the controllers.

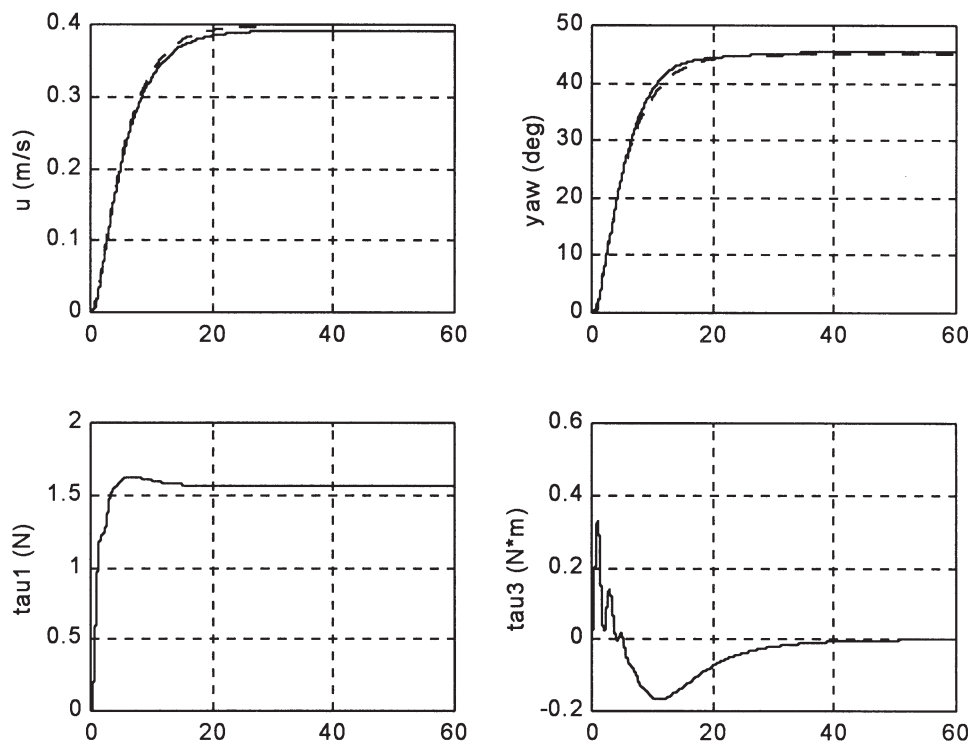


Fig. 1: Manually tuned controllers response

In the vessel simulation two PID controllers have been integrated: one for the heading subsystem (i.e. generates the input τ_3), and other for the propulsion subsystem (i.e. generates the input τ_1). The parameters of both controllers (i.e. K , proportional gain, K_I , integral gain, and K_D , derivative gain) need to be tuned in order to get a response of the system as accurate as possible. The manual tuning yields the parameters shown in Table 1.

Table 1: Manually tuned controllers parameters

	Heading controller	Propulsion controller
K	9	200
K_I	1.5	2
K_D	20	8

The resulting controller generates the simulated results shown in Figure 1 when a 45° overdamped second order step input is applied in the heading and a 0.4 m/s overdamped second order step input in the propulsion.

Although the results obtained in this case are considered to be quite good, manual tuning is a tedious job and not very reliable, since sometimes the values of the controller's gains can be particularly difficult to obtain. Therefore a GA has been used to optimise these gains.

6 Genetic Algorithm optimisation results

The results presented in this section are best cases obtained from several runs (i.e. 10 runs) of the GA. The rest of the runs converge to the same minimum area and the control-

lers obtained show similar characteristics, although they do not perform as well as the best case.

Since it is desirable that the population size is approximately equal to the number of genes in each chromosome and that the generation size doubles this [4], in every run the population consisted of 60 chromosomes and the number of generations the optimisation cycle performed was 120. Thus, the GA initialisation is carried out by generating an initial population matrix of uniformly distributed random integers, whose size is 60×30 (i.e. number of chromosomes \times number of genes in a chromosome).

The optimisation design criterion is defined by the cost function. In addition to this there is a desired response that the controller must track. The desired heading and propulsion manoeuvres used in the GA optimisation is the same than the one used in the manually tuned controller so that both studies can be compared (see Figure 1).

Since the objective of the controllers is to make the vessel track desired dynamic responses with the minimum actuator effort, the cost function will have two terms for each controller [4]:

$$C = \sum_{i=0}^{tot} [(\Delta\psi_i)^2 + \lambda_1(\tau_{3i})^2 + (\Delta u_i)^2 + \lambda_2(\tau_{1i})^2]. \quad (5)$$

Here $\Delta\psi_i$ is the i th heading angle error between the desired and obtained heading, τ_{3i} is the i th yaw thrust force, Δu_i is the i th surge velocity error between the desired and obtained surge velocity, τ_{1i} is the i th surge thrust force, tot is the total number of iterations and λ_1 and λ_2 are scaling factors. Since the optimisation processes attempt to minimise the value

of this function it is easy to see that $\Delta\psi$, Δu , τ_3 and τ_1 will be minimised too. Therefore, the quantities $\Delta\psi$ and Δu give an indication of how well the controllers are operating by showing the tracking between the actual and the desired heading and surge velocity and the input components τ_3 and τ_1 are used to keep the actuators movement to a minimum so that they can operate well within their operating limits.

As the input force and torque are always larger than the output errors near the optimum, they dominate the cost values in this area. This leads to solutions that provide very small thruster effort, but a very poor tracking of the desired responses. In order to avoid this, τ_3 and τ_1 are scaled by two constants, λ_1 and λ_2 , so that an equally balanced trade-off between the elements is obtained and the four terms of the cost function are equally optimised. Several scaling factors have been considered, resulting in choosing λ_1 and λ_2 equal to 0.001 as the most appropriate.

In Figure 2 the fast convergence of the GA for these scaling factors is shown. The best solution converges to a solution very close to the minimum in less than 20 generations. The total cost is shown to converge to a low cost region within 5 generations, although after that there are some peaks of higher cost due to the mutation operator.

In the final generation there are 52 chromosomes that have similar cost functions (i.e. 86.6 % of the population). This means that the amount of saturation is high and provides confidence in the final solution being near optimal.

The parameters shown in Table 2 are typical of those obtained as a solution by this optimisation.

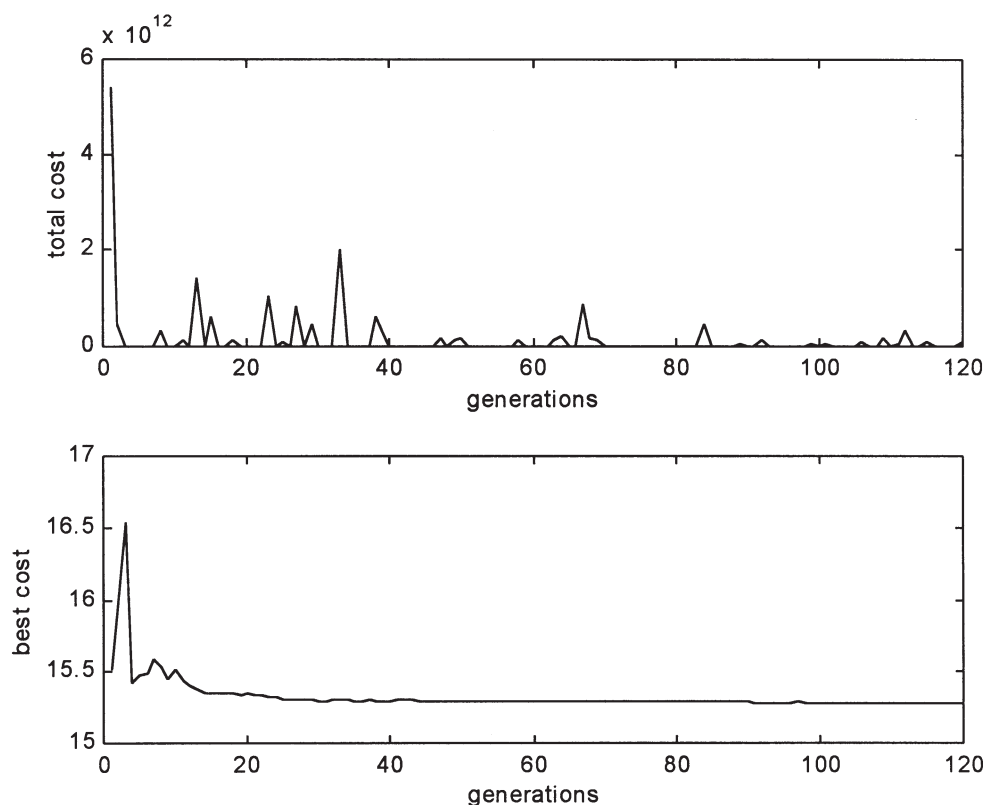


Fig. 2: GA optimisation convergence

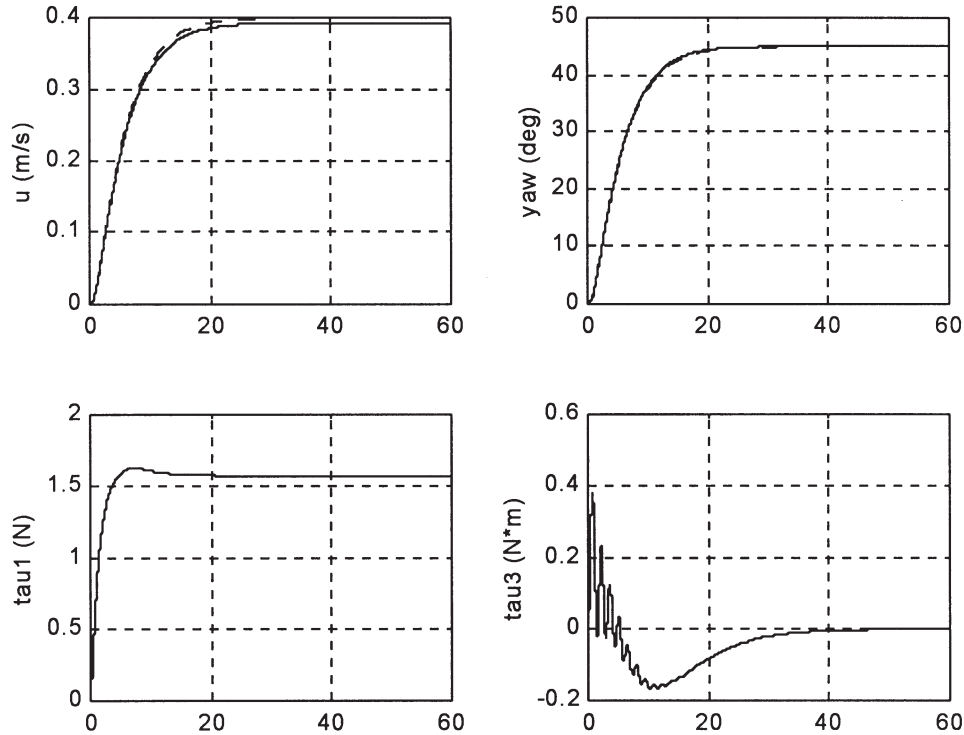


Fig. 3: GA controllers response using cost function (5)

Table 2: GA controllers parameters

	Heading controller	Propulsion controller
K	25.36	223
K _I	0.0025	0.0031
K _D	37.03	519.9

When these parameters are implemented the simulated responses shown in Figure 3 are obtained.

As can be seen from the figure, the tracking of the desired overdamped response in both subsystems, propulsion and heading, is excellent, although the surge velocity response shows a very small steady state error (0.01 m/s approximately). Obviously, τ_1 increases while the ship is accelerating and stays constant once the ship has reached the desired surge speed. On the other hand, τ_3 shows initially a positive peak, corresponding with the beginning of the manoeuvre and tends to zero as the ship completes the turn. Both input signals, τ_3 and τ_1 , are well kept within the operational limits, although τ_3 shows a very oscillating behaviour. This leads to an unnecessary wearing and tearing of the actuators that shortens their life. A new cost function has been considered to try to reduce the oscillations in the actuators:

$$C = \sum_{i=0}^{tot} \left[(\Delta \psi_i)^2 + \lambda_1 (\tau_{3i})^2 + \mu_1 \left(\frac{\tau_{3i} - \tau_{3i-1}}{\Delta t} \right)^2 + (\Delta u_i)^2 + \lambda_2 (\tau_{1i})^2 + \mu_2 \left(\frac{\tau_{1i} - \tau_{1i-1}}{\Delta t} \right)^2 \right]. \quad (6)$$

These two new terms introduce a measurement of the inputs increasing or decreasing rates. In the minimization process these two terms will also be minimized, leading to a smoother input response. Both terms are scaled by two coefficients, μ_1 and μ_2 .

After several trials, the coefficients μ_1 and μ_2 that seem to fulfil the role of reducing the wear and tear in the actuators without worsening the tracking are: μ_1 equal to 0.01 and μ_2 equal to 0. This value of μ_1 gives a good trade-off between good tracking of the desired heading and a significant reduction in the oscillations of the input τ_3 . Also, any value for μ_2 different to 0 leads to a very slow response, often with a big steady-state error, caused by the slowing down of τ_1 . This is to be expected since the previous optimised controller did not present oscillations in the input τ_1 . Hence, although the inclusion of the input rate term has provided a less oscillatory response for the input τ_3 keeping a good tracking, in the case of τ_1 it leads to a much worse tracking with slight improvements in the input response. It seems necessary to have a prior knowledge of the actuators behaviour in order to include the input rate term in the cost function. Introducing the input rate term in the cost function if there is no wearing and tearing of the actuator will just result in a slower input and less accurate tracking.

Considering the convergence of the GA method through the total and best cost, the total cost converges to a low cost region within 3 generations and the best cost converges to a solution close to the final cost value in the 10th generation. Hence, the convergence in this case has been faster than in Figure 2. Also, observing the amount of saturation in the final generation it can be verified that the final solution is near optimal due to the high number of similar individuals in the final solution (i.e. 45 individuals, 75 % of the population).

When the PID controllers are optimised using (6) the values shown in Table 3 are typical of those obtained.

Table 3: GA controllers parameters

	Heading controller	Propulsion controller
K	4.28	885
K_I	5.126	96.34
K_D	10.04	560.2

The results obtained for the PID controller gains using this latter cost function (6) differ quite a lot from the ones got with the previous one (5), especially K_I . As it has been included a new term in the cost function (6) to reduce the oscillations in the input signal generated by the heading controller (i.e. τ_3), the GA has done so by decreasing K and K_D to reduce the gain of the system and the noise amplification. Also, in order not to get a poorer tracking response after reducing K, K_I has been increased to improve the tracking. Although the terms of the cost function (6) concerning the propulsion controller have not changed, the reduction of oscillation in τ_3 affects τ_1 and this deals to a different propulsion controller.

Once implemented the simulated responses shown in Figure 4 are obtained.

As can be seen from these plots the tracking of the surge velocity desired response is again excellent, and even the slight steady state error has disappeared. The tracking of the heading desired response is a bit worse than it was with the previous controller, especially in the transient (i.e. the first 10 seconds), but thanks to this the goal of this new optimisation, which was to reduce the wearing and tearing in the yaw thrusters, has been accomplished.

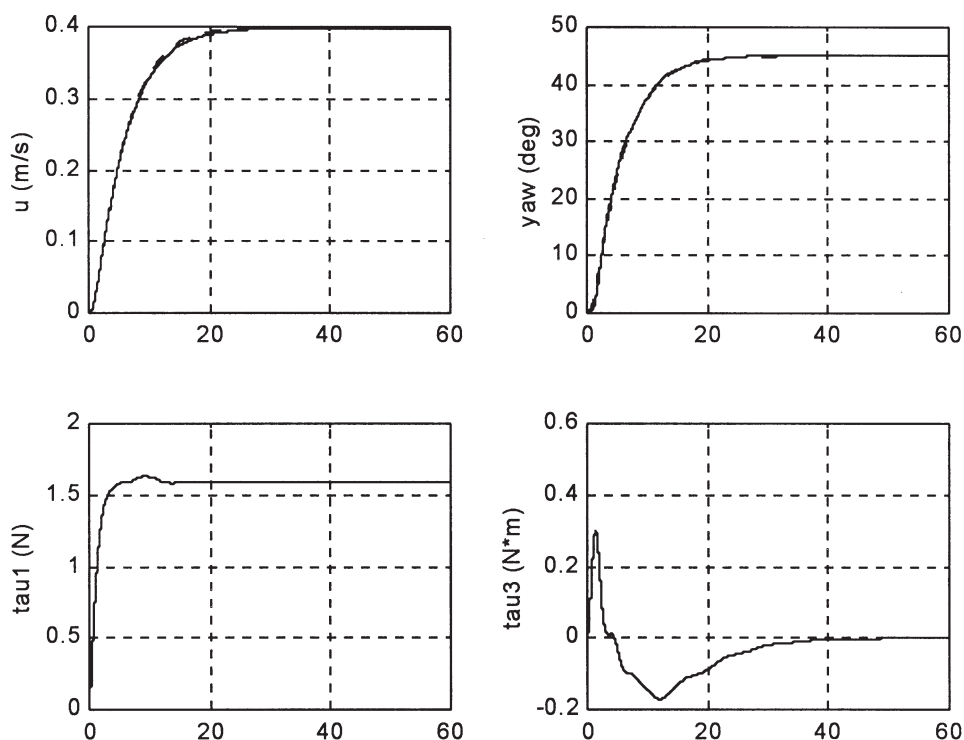


Fig. 4: GA controllers response using cost function (6)

7 Execution Time

Comparing the execution time for both methods, manual tuning and GA optimisation, it can be seen in the Table 4 below that the time spent tuning the controller was longer than the time it takes to run a GA simulation. It is quite difficult to reckon the time a designer will spend doing a manual tuning because it highly depends on his experience, engineering judgement and luck. Although it may seem that the simulation time is quite long and there is no real advantage in it, we must take into account that while the optimisation is running the designer does not need to be there.

Table 4: Execution time

Manual tuning	GA optimisation
15 h	8 h

8 Conclusions

This paper has presented the findings of a study of PID controller gains optimisation for an oil platform supply ship's propulsion and heading dynamics.

The GA process has performed well as optimisation technique. It has obtained optimal solutions for both propulsion and navigation PID controllers without a-priori knowledge of the optimal region. As this a-priori knowledge is not necessary, GA techniques are very suitable when the designer is unfamiliar with the plant to control.

The key factor in the GA optimisation is the choice of the cost function. As the objective of the optimised controller is to make the vessel follow as accurately as possible the desired track while avoiding excessive wear and tear in the actuators, these two terms (i.e. difference between desired response and

actual response and thrust forces) must be included in the cost function in order to be minimised. Numerous simulations considering different scaling factors show that there is a trade-off between the accuracy following the desired track and the actuators, usage and one of them cannot be improved without making the other worse. This leads to the necessity, of a balance so that the overall performance is good.

In addition, the inclusion in the cost function of a term considering the rate of variation of the thrust forces along the Z-axis leads to a significant reduction in the oscillation of this input keeping a good tracking of the desired heading. This is a considerable advantage since it avoids unnecessary wear and tear of the actuators, which could shorten their operational lifespan.

The results obtained illustrate the benefits of using GAs to optimise propulsion and navigation controllers for surface ships. The accuracy of the obtained simulations allows confidence in the good performance of these controllers once they are used to control the manoeuvring of the actual scale model.

References

- [1] Holland, J.: *Genetic Algorithms*. Scientific American, 1992
- [2] Ellis, C.: *A Bluffer Guide to Genetic Algorithms*. Engineering Design Newsletter, Science and Engineering Research Council Newsletter, 1993
- [3] Dutton, K., Thompson, S., Barraclough, B.: *The Art of Control Engineering*. Harlow, Addison-Wesley, 1997
- [4] McGookin, E. W.: *Optimisation of Sliding Mode Controllers for Marine Applications: A study of methods and implementation issues*. PhD Thesis, UK, 1997
- [5] Fossen, T. I.: *Guidance and Control of Ocean Vehicles*. Chichester, John Wiley & Sons Ltd., 1994
- [6] Goldberg, D.: *Genetic Algorithms in Searching Optimisation and Machine Learning*. Reading, Addison-Wesley, 1989
- [7] Polheim, H.: *Genetic and Evolutionary Algorithm Toolbox for Use with MATLAB*. <http://www.systemtechnik.tu-ilmenau.de/~pohlheim>

Eva Alfaro-Cid, MEng
 phone: +44 141 330 6031
 fax: +44 141 330 6004
 e-mail: alfaro@elec.gla.ac.uk

Euan W. McGookin, MEng, PhD

David J. Murray-Smith, BSc, MSc, PhD

Centre of Systems and Control & Dept.
 of Electronics and Electrical Engineering
 University of Glasgow
 Glasgow, G12 8LT, UK