Genetic Programming for the Automatic Design of Controllers for a Surface Ship

Eva Alfaro-Cid, Euan W. McGookin, *Member, IEEE*, David J. Murray-Smith, *Member, IEEE*, and Thor I. Fossen, *Senior Member, IEEE*

***Abstract*—In this paper, the implementation of genetic programming (GP) to design a controller structure is assessed. GP is used to evolve control strategies that, given the current and desired state of the propulsion and heading dynamics of a supply ship as inputs, generate the commanded forces required to maneuver the ship. The controllers created using GP are evaluated through computer simulations and real maneuverability tests in a laboratory water basin facility. The robustness of each controller is analyzed through the simulation of environmental disturbances. In addition, GP runs in the presence of disturbances are carried out so that the different controllers obtained can be compared. The particular vessel used in this paper is a scale model of a supply ship called CyberShip II. The results obtained illustrate the benefits of using GP for the automatic design of propulsion and navigation controllers for surface ships.**

***Index Terms*—Genetic programming, marine vehicle control.**

I. Introduction

NAVIGATION of ships has been a major concern for sailors and stakeholders since humans took to the waters. Approaches involving nautical sextants and the judgement of the navigator have been superseded by the introduction of advanced technology that assists with the sailing process. However, the size, number, and versatility of modern ships have highlighted the need for better navigational control to avoid accidents.

To ensure the safe navigation of surface vessels, their motion (i.e., navigation and propulsion capabilities) has to be accurately controlled. This can be achieved through the design and implementation of automatic control systems that can accurately govern the maneuvering capabilities of ships.

In general, control theory provides design strategies that allow a better understanding of the system being controlled (e.g., a vessel) and a mechanism to regulate the way in which the system operates. Various control methodologies have their own unique structure. Despite being fundamentally different in style, they perform the same task, i.e., to make the system behave in a desired manner.

Since the early 1970s, important research has been conducted on the subject of automatic control of marine surface vessels. A major text in this field is [1].

Genetic programming (GP) [2] is a powerful tool that allows the solution of problems without *a priori* specifying the size, shape, and structure of such a solution. Hence, GP is an evolutionary search technique based on the concept that, in nature, structures adapt to the environment, that is, the structure created over a period of time is the outcome of natural selection and sexual reproduction.

The application of evolutionary optimization techniques such as genetic algorithms (GAs) [3] has widely been used in the field of control engineering. The performance of the control techniques depends not only on the control structure chosen but on the values of the controller's parameters as well. GAs provide a way of optimizing the performance of the controllers by automatically tuning such parameters [4]–[6].

However, GAs are parameter optimizers and, in the majority of cases, do not vary the structure of the optimizing subject. In the context of controller optimization, they are presented with the structure of a particular control methodology and vary the associated parameters to obtain the desired performance for the system being controlled. Since these structures are usually rigid in form, the number of associated tuning parameters is set and, thus, involves a fixed-length string representation for the solutions.

On the other hand, in GP, the size and shape of solutions dynamically evolve. This flexibility in the representation of solutions allows taking the optimization problem a step forward. In the control problem being tackled here, GP will provide a control strategy; using the states of the system to control as an input, the outcome of the GP tree will be the actuator's commanded signal. Therefore, by using GP, not only will the controller's parameters be optimized, but the structure of the whole controller will be as well.

So far, GP has been applied to a small number of control problems. In [2], Koza included some of what he called cost-driven evolution problems (i.e., optimal control problems), such as the cart entering, broom balancing, and truck backer problems. After that, he has published a few more applications of GP to control problems [7], [8].

The use of GP to discover the control laws that allow the performance of attitude maneuvers for a satellite or spacecraft

Manuscript received April 4, 2007; revised September 26, 2007, November 27, 2007, and December 24, 2007. The Associate Editor for this paper was A. Hegyi.

E. Alfaro-Cid is with the Instituto Tecnológico de Informática, 46022 Valencia, Spain (e-mail: evalfaro@iti.upv.es).

E. W. McGookin and D. J. Murray-Smith are with the University of Glasgow, Glasgow G12 8QQ, U.K. (e-mail: e.mcgookin@eng.gla.ac.uk; djms@elec.gla.ac.uk).

T. I. Fossen is with the Norwegian University of Science and Technology, 7491 Trondheim, Norway (e-mail: fossen@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2008.922932

is shown in [9]. A stability proof for the GP-derived controller was included in that paper.

In [10], the inverted pendulum problem is solved, but instead of evolving equations to determine the direction of the bang-bang force, as done in [2], the authors also look for the magnitude of the force applied to the cart that can move it to a target position while keeping the pendulum standing.

The most popular control application using GP that is found in the literature is the control of mobile robots [11]–[13].

However, in the field of transportation, the references to GP are scarce; for instance, in [14], GP is applied to the control of vehicle systems: specifically an active suspension system.

In addition to GP, there are other intelligent control methods that have been employed in ship control, such as expert systems and fuzzy and neural networks [15]. As an example of expert systems applied to marine surface vessels, [16] presents an expert system for aided design of ship system automation. There are more contributions in the field of fuzzy logic and neural networks. For instance, [17] shows the design of a fuzzy autopilot for ship control. In [18], the use of an artificial neural network as a controller that incorporates the properties of a series of conventional controllers designed for different operating conditions is proposed as an alternative to adaptive control or gain scheduling for the automatic steering of a ship. The book of Harris *et al.* [19] includes the control of the steering of a ship as a practical example. Finally, [20] combines both methodologies by using a neural network to train a fuzzy ship autopilot.

One of the advantages of GP versus other artificial intelligence methods such as neural networks or support vector machines is that GP generates results in the shape of a function that can be analyzed, as opposed to "black box" type of results.

The particular application used in this paper is a scale model of an oil platform supply ship, called CyberShip II (CS2) [21]. CS2 is the test vehicle developed at the Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

In the control problem being tackled here, GP provides control structures to govern the heading and propulsion dynamics of CS2. Using the states of the system to control as an input, the outcome of the control structure is the actuator's commanded signal, i.e., rudder deflection for heading and propeller revolutions for propulsion.

The GP structural optimization of the control strategy for the navigation and propulsion dynamics of CS2 has been conducted through computer simulations in MATLAB using a mathematical model of CS2. The resulting optimized controllers have been implemented and tested on the actual physical model of CS2 at the Marine Cybernetics Laboratory (MCLab), NTNU.

II. CYBERSHIP II

A. MCLab

The model control subject used in this paper is CS2, which is a scale model (approximately 1:70) of an oil platform supply ship (see Fig. 1). This control test vessel has been developed at the Department of Engineering Cybernetics, NTNU, and employed in the MCLab (http://www.itk.ntnu.no/marinkyb/MCLab/) (see Fig. 2).

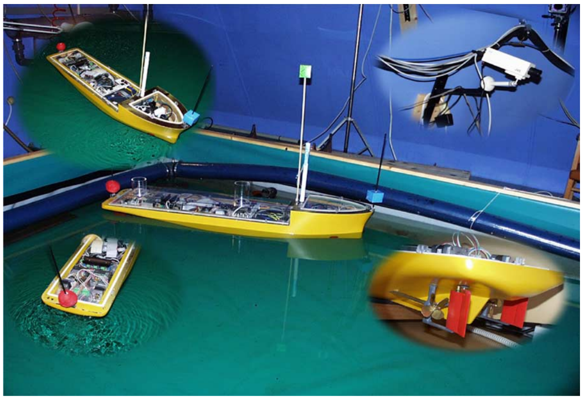

Fig. 1. Various aspects of CS2.

Fig. 2. Water basin in the MCLab.

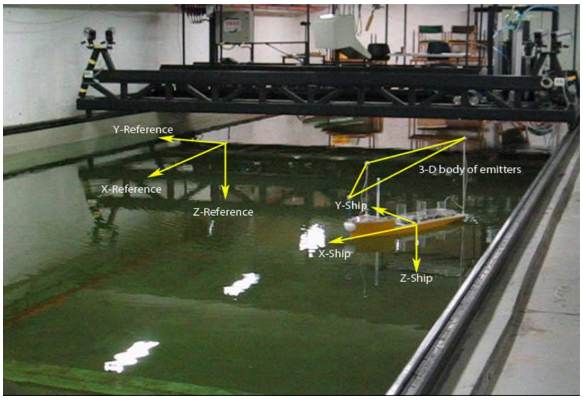

Fig. 3. Positioning system in the MCLab.

The MCLab is a purpose-built experimental laboratory for testing of ships and underwater vehicles. The testing of the optimized controllers obtained from this paper has been performed on the real CS2 model in the MCLab.

The length of CS2 is 1.25 m, and its mass is 23.8 kg. It is actuated by means of a tunnel thruster placed at the bow and two main propellers with rudders situated at the stern.

For position measurement purposes, CS2 is fitted with three 3-D emitters. The signals emitted are detected by four PC cameras that provide the measurements of the (x, y) coordinates plus the heading angle to the user (see Fig. 3).

CS2 is also equipped with an onboard PC running the QNX real-time operating system. The control is done in real time by an onshore PC. The connection between both PCs is made through a wireless Ethernet link and an automatic C-code generator. MATLAB, Simulink, and Real-Time Workshop are coupled with a graphical user interface in LabVIEW for real-time presentation of the results.

To simulate surface waves, the MCLab has a wave generator. Such a generator consists of a single flap controlled by a wave synthesizer and can produce regular and irregular waves with various spectra and wave height. The wave generator was used in the experimental trials in this paper.

For more accurate information about the MCLab and CS2, see [21]–[24].

B. CS2 Mathematical Model

Prior to the real testing, a nonlinear hydrodynamic model based on the kinetic and kinematic equations that represent the dynamics of the vessel was used for the simulations of the design stage. This model was developed at the Department of Engineering Cybernetics, NTNU [22].

When kinetic and kinematic equations are combined, the following matrix form is produced (assuming $\boldsymbol{M}$ to be invertible):

$$\begin{bmatrix} \dot{\boldsymbol{\nu}} \\ \dot{\boldsymbol{\mu}} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{M}^{-1}\left(C(\nu)+D\right) & 0 \\ \boldsymbol{J}(\mu) & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\mu} \end{bmatrix} + \begin{bmatrix} \boldsymbol{M}^{-1} \\ 0 \end{bmatrix} \boldsymbol{\tau}. \quad (1)$$

Here, $\boldsymbol{M}$ is the mass/inertia matrix, $\boldsymbol{C}$ is the Coriolis matrix, $\boldsymbol{D}$ is the damping matrix, and $\boldsymbol{J}$ is the Euler matrix. In addition, $\boldsymbol{\nu} = [u, v, r]^T$ is the body-fixed linear and angular velocity vector, $\boldsymbol{\mu} = [x, y, \psi]^T$ denotes the position and orientation vector with coordinates in the Earth-fixed frame, and $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3]^T$ is the input force vector, given that τ_1 is the body-fixed force along the X-axis, τ_2 is the body-fixed force along the Y-axis, and τ_3 is the body-fixed torque along the Z-axis. These are the inputs to CS2 that are used to manipulate the motion of the vessel. This expression corresponds to the nonlinear state-space equation $\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{x}) \cdot \boldsymbol{x} + \boldsymbol{B} \cdot \boldsymbol{\tau}$.

C. Environmental Disturbances

To evaluate the robustness against environmental disturbances of the controllers obtained through the GP optimization, simulations of maneuvers in the presence of environmental disturbances are used to evaluate each candidate solution obtained during the optimization.

There are three main types of environmental disturbances: 1) wind-generated waves; 2) ocean currents; and 3) wind. In this paper, the analysis has been restricted to the disturbance considered the most relevant for surface vessels, i.e., wind-generated waves. In addition, this type of disturbance can be reproduced in the MCLab for testing.

The model that has been used to simulate the wave's action on the vessel [25] derives the forces and moments induced by a regular sea on a block-shaped ship. It forms a vector called $\boldsymbol{\tau}_{\text{waves}}$ that is directly added to the input vector $\boldsymbol{\tau}$ in (1) using the principle of superposition. Thus

$$\begin{aligned} X_{\text{wave}}(t) &= \sum_{i=1}^{N} \rho g B L T \cos(\beta - \psi) s_i(t) \\ Y_{\text{wave}}(t) &= \sum_{i=1}^{N} -\rho g B L T \sin(\beta - \psi) s_i(t) \\ N_{\text{wave}}(t) &= \sum_{i=1}^{N} \frac{1}{24} \rho g B L (L^2 - B^2) \sin 2(\beta - \psi) s_i^2(t). \end{aligned} \quad (2)$$

Here, L, B, and T are the length, breadth, and draft of the wetted part of a ship, considering it as a parallelepiped. ρ is the density of water, $s_i(t)$ is the wave slope, and $(\beta - \psi)$ is the angle between the heading of the ship and the direction of the wave (in radians).

The wave slope s_i can be related to the wave spectral density function $S(\omega_i)$. To compute $S(\omega_i)$, different wave spectra can be considered. In this paper, a modified version of the Pierson–Moskowitz spectrum has been used [1]. Thus

$$S(\omega) = \frac{4\pi^3 H_s^2}{(0.710 T_o)^4 \omega^5} \exp\left(\frac{-16\pi^3}{(0.710 T_o)^4 \omega^4}\right). \quad (3)$$

Here, T_o is the modal period, and H_s is the significant wave height.

III. Genetic Programming

GP is an evolutionary methodology inspired by biological evolution to find solutions that perform a user-defined task [2]. Unlike the classical optimization methods, which are based on tracking a certain trajectory, GP works with a population of candidate solutions (called individuals).

As for any evolutionary optimization technique in GP, an initial population of individuals is created at random. Each individual in the population is evaluated using a fitness function. The individuals that performed better in the evaluation process have more possibilities of being selected as parents for the new population than the rest. A new population is created using the selection, crossover, and mutation operators. The individuals of this new population typically show better performance than those of the previous population, since the best individuals have a better chance of being selected for reproduction. The loop is run until a certain termination criterion is met, e.g., obtaining near-optimum solutions or a predetermined finite number of generations reached.

A. GP Tree Representation and Structure

The individuals being evolved in GP are hierarchical structures with no predefined size or shape. Traditionally, the structure being evolved in GP has a tree shape. This tree is formed by internal and external nodes. Prior to the optimization process, the user has to define which functions (i.e., mathematical operations that perform the calculations in the tree structure) and terminals (i.e., variables and parameters used in the calculations)

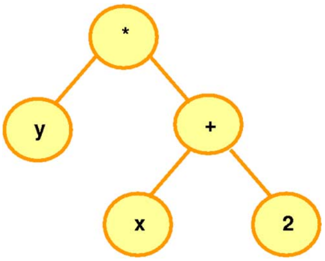

Fig. 4. Example of tree representation.

are relevant for the problem to solve. The internal nodes of the tree are occupied by functions, whereas the terminals take the external nodes.

For instance, the expression $y \cdot (x + 2)$ can be represented in tree form, as shown in Fig. 4.

An important issue to take into account when choosing the functions and terminals is the *closure property*. The closure property says that each function has to be able to take as an argument any value and data type returned from the evaluation of any function or from the terminal set. For example, the division function has to be implemented in a way that can take a zero as a denominator without returning an error. Usually, this is done by setting the division operator to return a fixed 0 or 1 value when the denominator is zero.

B. GP Operators

As with GAs, GP has three main operators that progress the evolutionary development of the population of candidate solutions. These are *selection*, *crossover*, and *mutation* [2].

1) Selection: The selection operator is used to determine which individuals are going to be parents of the individuals of the next generation. The selection procedure favors the selection of those individuals that exhibit better cost values. In this paper, tournament selection has been used [26].

2) Crossover: The most popular form of crossover is what has been called subtree crossover [2], [27]. This method involves selecting two individuals from the population using the selection operator. These are called the parents. A crossover node is chosen at random in each parent, and the whole of the subtrees rooted at those nodes are swapped, as shown in Fig. 5.

Typically, the probability of crossover is between 80% and 95%. In this paper, it has been chosen to be 80% due to the satisfactory results obtained with this probability in a study comparing the performance of various crossover probabilities and mutation probabilities presented in [6].

3) Mutation: The tree structure of GP solutions allows a variety of mutation operators. In this paper, a combination of two methods is employed, i.e., subtree mutation and point mutation.

The standard mutation operator is subtree mutation [27]. It selects a mutation point at random, removes the subtree at that point, and inserts a randomly generated subtree (see Fig. 6).

The point mutation operator [27] replaces the chosen node with another function of the function set with the same number of arguments, if the chosen node is an internal node, or with a terminal of the terminal set otherwise (see Fig. 7).

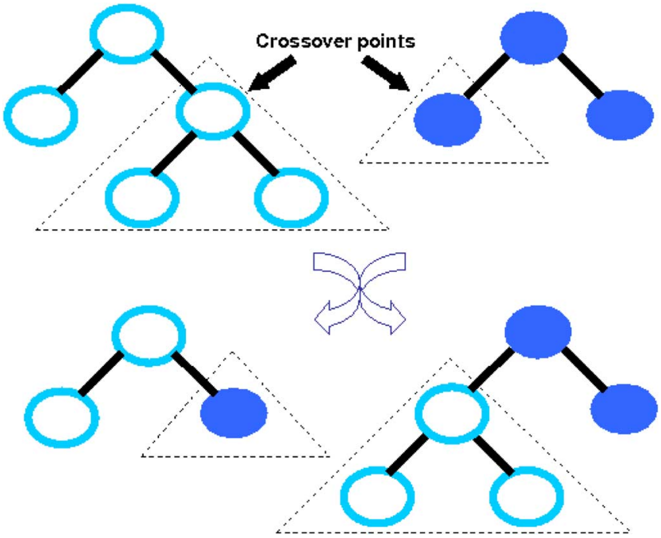

Fig. 5. Subtree crossover.

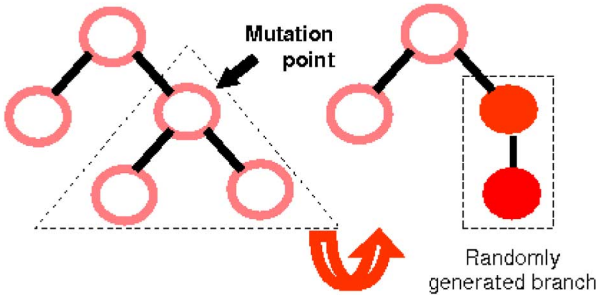

Fig. 6. Subtree mutation.

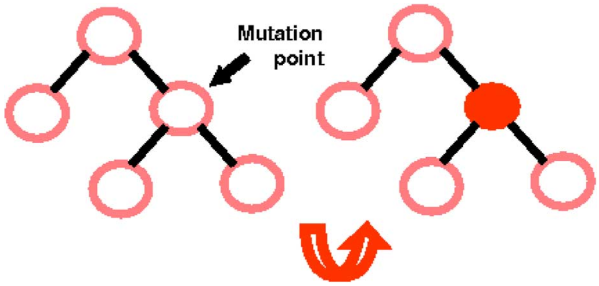

Fig. 7. Point mutation.

In this paper, mutation occurs with a probability of 0.1 (due to the results obtained in the previously mentioned comparison study [6]). Once a tree has been chosen for mutation, the probability of undergoing subtree or point mutation is 0.5.

C. Random Generation of the Numerical Constants

Our GP algorithm uses Koza's suggestion of simply introducing a random constant $\mathcal{R}$ in the set of terminals, so that every single time this terminal is chosen, a random number is generated and associated with that terminal node [2]. The GP should be able to generate other constants needed by using arithmetic operations to create them. This is a very simple approach, but it may raise questions about the accuracy of the result.

TABLE I
TERMINAL SETS FOR PROPULSION AND HEADING

Propulsion	Heading
surge error (ε_p)	heading error (ε_h)
u	ψ
u_d	ψ_d
$\mathcal{R}$	$\mathcal{R}$

TABLE II
FUNCTION SET

2-argument functions			
$arg_1 \cdot arg_2$	$arg_1 + arg_2$	$arg_1 - arg_2$	arg_1 / arg_2
$arg_1 \cdot \tanh\left(\frac{h'(x-x_d)}{arg_2}\right)$		$\text{place}(0, -arg_1, -arg_2)$	
1-argument functions			
$\int arg\,\mathrm{d}t$	$\frac{\mathrm{d}arg}{\mathrm{d}t}$	$\sin(arg)$	$\exp(arg)$
$arg\,\text{sign}(h'(x - x_d))$		$\text{place}(-arg)$	

However, as opposed to Koza's GP that does not use mutation, in this paper, point mutation has been included as an operator. This enables the GP to modify the terminal values. Thus, a numerical constant can change its value, and a terminal occupied by a variable can be mutated into a numerical constant [28].

D. GP Application to the Control Problem

In this paper, GP has been applied to the search of a controller structure for the control of the heading and propulsion dynamics of CS2. The objective of the control system is to provide good tracking of the desired response and minimize the use of actuators. GP does it by evolving tree structures. Therefore, there is no need to *a priori* choose the size or shape of the solutions. They are evolved along the generations.

Every solution to the stated control problem consists of two independent trees: one for heading control and the other for propulsion control (i.e., decoupled controllers). In order for these controller structures to satisfy these design objectives, the library of terminal and function sets has to be tailored to suit the problem at hand.

1) Function and Terminal Sets: For this application, the terminal set consists of four terms: error, state, reference, and one numerical constant. The error term has explicitly been used (although it is possible to obtain this term using the state and reference terms plus a minus operator) due to preliminary results that showed an improvement in the performance of the GP if this term was included. Thus, the propulsion terminal set consists of the surge error ε_p $(= u_d - u)$, the surge u, the desired surge u_d, and the numerical constant $\mathcal{R}$. The heading terminal set consists of the heading error ε_h $(= \psi_d - \psi)$, the heading ψ, the desired heading ψ_d, and the numerical constant $\mathcal{R}$. The terminal sets are shown in Table I.

The probability of generating a numerical constant is three times bigger than the probability of choosing any of the other terminals (i.e., probability of 0.5) since the number of numerical constants required to create a control structure is larger than the number of variables.

The function set is formed by 11 functions that are related to the following control techniques: *PID control*, *sliding-mode control* [29], and *pole placement* [30]. Five of those 11 functions are two-argument functions, five of them are one-argument functions, and one of the functions has two arguments when used for heading and only one argument when used for propulsion (Table II).

The four basic arithmetic operations (i.e., $+$, $-$, $*$, and $/$) are routinely included in most GP algorithms. The integral and derivative functions are included to account for a PID type of structure. The hyperbolic tangent and sign functions allow the construction of switching term equivalents, which are similar to sliding-mode control. The *place* command from MATLAB is included as a pole-placement technique. In addition, the sine and exponential functions give more versatility to the algorithm. Although many more functions could have been added, it is not advisable, since the performance of the GP algorithm degrades with the addition of numerous functions.

From the hyperbolic tangent and sign formulas in Table II, $x = [u]$ and $x_d = [u_d]$ for the propulsion control tree, whereas $x = [v, r, \psi]^T$ and $x_d = [v_d, r_d, \psi_d]^T$ for heading. The $\mathbf{h}$ matrix is the right eigenvector associated with a zero pole for the desired closed-loop system matrix. The desired closed-loop performance of a system is characterized by the position of the closed-loop poles of the system (i.e., the eigenvalues of the closed-loop system matrix). In a previous study [31], GAs have been used to optimize the parameters of a decoupled sliding-mode controller for CS2. Some of the parameters optimized were the closed-loop poles of the system. Therefore, the $\mathbf{h}$ matrices have been calculated based on the best solution found in the GA optimization of a decoupled sliding-mode controller for CS2. An alternative would be to consider the closed-loop poles of the system as arguments of the function (in this case, the hyperbolic tangent and sign functions) and leave the GP to find the more suitable values. However, this would considerably increase the complexity of the search.

The place command returns the value $-\mathbf{k} \cdot \mathbf{x}$, where $\mathbf{x}$ is as previously defined, and $\mathbf{k}$ is the feedback matrix obtained by executing $\text{place}(0, -arg_1, -arg_2)$ in the heading control tree or $\text{place}(-arg)$ in the propulsion control. Since the other functions in the function set only produce real values, the poles to be assigned are always real numbers.

Moreover, to ensure that the closure property is met, some of the functions have a *protection mechanism* to avoid situations where the solution is not defined (for example, division by zero). Thus, the division function is encoded so that if the denominator is 0, the result of the division is set to 1. In addition, the hyperbolic tangent function returns arg_1 when arg_2 is 0. The most likely function to cause problems is the *place* command. It has been set to return 0 if there is any error message activated (for example, if the poles are too close).

2) Evaluation of Solutions: Once an initial population of tree structures is generated at random, the structures are implemented in the model simulation as a newly formed control strategy. A simulation is run, and the controller's performance is evaluated. This is achieved by applying an optimization design criterion to the simulated responses obtained. For minimization problems, the optimization design criterion is usually called the cost function.

The optimization design criterion used in this paper is defined by the cost function in (4). Since the objective of the controllers is to make the vessel track the desired heading and propulsion trajectories with the minimum actuator effort, the components of the cost function must reflect these design objectives. In this case, there are three terms for each controller. Thus

$$C = \sum_{i=0}^{\mathrm{tot}} \left[(\Delta\psi_i)^2 + \lambda_1 (\tau_{3i})^2 + \mu_1 \left(\frac{\tau_{3i} - \tau_{3i-1}}{\Delta t} \right) \right] + \sum_{i=0}^{\mathrm{tot}} \left[(\Delta u_i)^2 + \lambda_2 (\tau_{1i})^2 + \mu_2 \left(\frac{\tau_{1i} - \tau_{1i-1}}{\Delta t} \right) \right]. \quad (4)$$

Here, i is the current iteration, $\Delta\psi_i$ is the ith heading angle error between the desired and obtained heading, τ_{3i} is the ith yaw thrust force, Δu_i is the ith surge velocity error between the desired and obtained surge velocities, and τ_{1i} is the ith surge thrust force. 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 desired heading and surge velocities, 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.

The third and sixth terms in (4) introduce a measurement of the inputs' increasing or decreasing rates. The minimization of these terms helps reduce the oscillations in the inputs, avoiding unnecessary wear and tear on actuators, which shortens their operational life spans. Consequently, the minimization of these two rate terms leads to a smoother input response, as there are fewer variations in the input responses.

In addition, in (4), tot is the total number of iterations (simulation time steps), and λ_1, λ_2, μ_1, and μ_2 are scaling factors. As the input force and torque are always larger than the output errors near the optimum, they dominate the cost values in this critical area of the search space, which leads to solutions that provide very small thruster effort but very poor tracking of the desired responses. To avoid this, these four weighting coefficients are introduced so that an equally balanced tradeoff between the six terms of the cost function is obtained, which is a single-objective multiaspect criterion. Each term in (4) represents a different aspect of the optimization problem. The weighted sum of the six terms results in a single-objective cost function.

IV. Results

A. Method

The GP algorithm has been run with and without the inclusion of realistic environmental disturbances (waves) in the optimization. The objective of including waves in the optimization process is to create a more realistic environment and to obtain control structures that are better suited to disturbance rejection (i.e., more robust controllers).

The maneuver used for the GP optimization in the evaluation of the candidate solutions has been a zigzag maneuver of 45° for heading while increasing the speed from rest to 0.7 m/s and back to rest. The GP algorithm has been run 20 times with and without disturbances. The disturbances used in the simulation are the same for each evaluation; therefore, two evaluations of the same controller would give the same cost value. The simulated waves have a significant height of 3 m, which corresponds to a sea state code of 5 (rough sea). The population size used is 120, and the number of generations is 30. These values have been chosen because the recommendation in the GP literature favors the use of various short runs, mainly to avoid the problem of the trees growing out of proportion [32].

The best results found in each optimization have been validated after the GP optimization. The reason for this validation test is to verify that the resulting tree is actually performing a control task and not merely generating a signal shaped in the right way for this maneuver but totally wrong for any other. Once the results have been validated with a totally different maneuver, we can conclude that the resulting trees are really controlling the navigation of the ship and not just creating a zigzag signal. In addition, the results obtained during the testing in the water tank will further confirm the adequacy of the resulting controllers.

The maneuver used in the validation test consists of two turning circle maneuvers linked together: first to port and then to starboard. The resulting trajectory is an ∞ shape. The reason for choosing this specific maneuver is that the Maritime Safety Committee, in its Resolution MSC.137(76) [33] on standards for ship maneuverability, recommends the turning point maneuver, together with the zigzag maneuver, as the most appropriate for ship performance testing.

The resulting best controllers have been tested on the real vessel in calm waters and while generating waves to study their effect. The wave synthesizer was set to generate irregular waves with a Pierson–Moscowicz spectrum [1], like the one used in the simulations. The significant wave height was chosen to be 5 mm (scale $1:70$) with a peak period of 0.80 s. The initial angle of attack of the waves is 0°. This is defined by the fixed position of the flap that generates the waves in the tank. This angle changes along the maneuver due to the turning of the boat.

B. Optimization Without Waves: Simulated and Real Results

Table III presents the best cost values obtained after each run of the GP optimization using the zigzag maneuver without waves and the posterior validation test using a double-turning-point maneuver.

The best overall result is obtained in run 1. It is not the best result achieved in the optimization, but the result in the validation test is extremely good.

In the final population of run 1, 32 out of 120 individuals have converged to a similar cost value, and the generation of convergence has been the tenth generation. The convergence of the cost value for run 1 is shown in Fig. 8. The upper subplot shows the overall convergence, and the bottom subplot shows in more detail the convergence in the final generations.

TABLE III
BEST GP RESULT OPTIMIZATION (WITHOUT WAVES)

Run No.	Zig-zag	∞-shape	Run No.	Zig-zag	∞-shape
1	2.44	91.80	2	4.14	243.60
3	28.63	314.54	4	200.73	552.00
5	3.04	1.40e4	6	13.20	3824.18
7	1.92	3431.07	8	152.54	3544.74
9	4.75	325.34	10	237.84	865.02
11	162.56	1230.30	12	367.86	867.61
13	421.25	5104.16	14	540.99	5.32e4
15	540.99	5.31e4	16	541.07	1.17e5
17	15.86	1.97e5	18	2.13	4.34e6
19	2.53	2.74e9	20	1.99	5.67e12

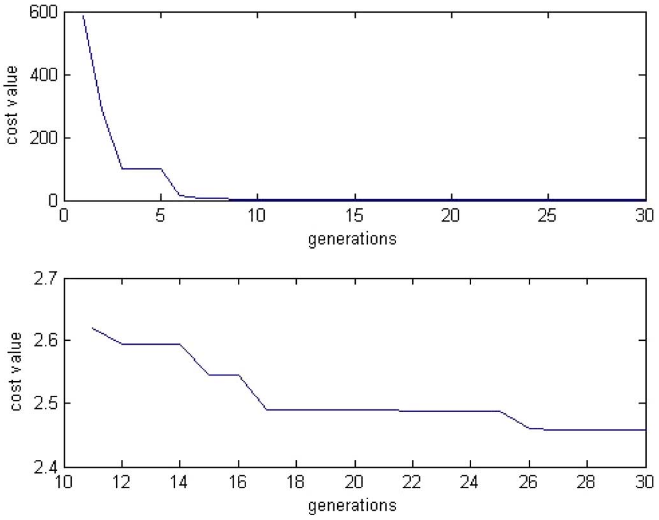

Fig. 8. Convergence of the cost value for run 1 (without waves).

TABLE IV
COST VALUES OBTAINED FOR THE ZIGZAG MANEUVER WITH VARIOUS CONTROL STRUCTURES (WITHOUT WAVES)

GP	PID	PP	SM+PI	SM	H_∞
2.44	1.2	0.96	4.5	4.5	1.6

Results from runs 2 and 9 have also consistently been good for both maneuvers. On the other hand, the results from runs 18, 19, and 20 are extremely bad. This corroborates the importance of the validation test since those three controllers converged to a very good cost value during the optimization.

In previous works [31], [34], [35], GAs have been used to optimize the parameters of a set of standard control methods such as PID, sliding mode, pole placement, and H_∞. The controller structures considered in those studies were as follows: two decoupled PID controllers, a multi-input–multi-output (MIMO) pole placement (PP) controller, a decoupled sliding-mode controller for heading plus a PI controller for propulsion (SM + PI), two decoupled sliding-mode controllers (SM), and a MIMO H_∞ controller.

Table IV compares the cost values obtained using the best results in the optimizations of the previous controllers with the cost value obtained using GP for structural optimization. As can be seen, the numerical result obtained with GP beats two of the five controller structures, and it is of the same order of magnitude as the other two.

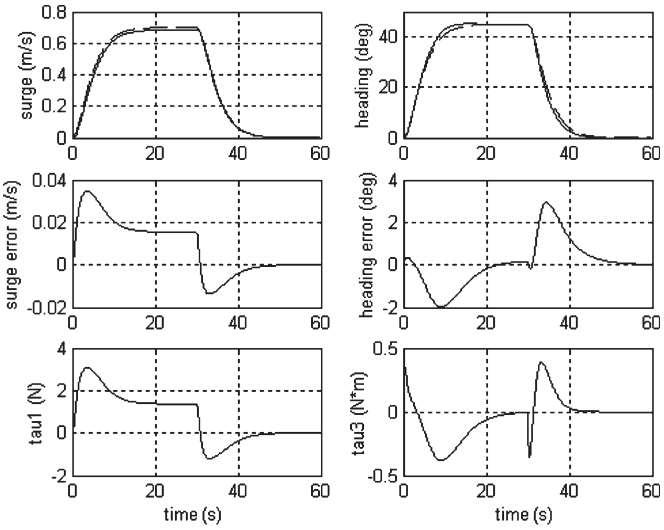

Fig. 9. Simulated results of the GP optimization without waves.

The structure of the best result obtained (run 1) consists of a proportional term for propulsion control and a hyperbolic tangent function for heading control, as follows:

$$\tau_1 = 88.4\varepsilon_p \tag{5}$$

$$\begin{aligned}\tau_3 = &-68.2\,\tanh\left(\frac{\mathbf{h}'_h(\mathbf{x}_h - \mathbf{x}_{hd})}{7.6}\right)\\ &- 8.87\,\tanh\left(\frac{\mathbf{h}'_h(\mathbf{x}_h - \mathbf{x}_{hd})}{-68.2}\right) - 0.13\,\sin(\psi)\\ &+ 11.7\,\tanh\left(\frac{\mathbf{h}'_h(\mathbf{x}_h - \mathbf{x}_{hd})}{\sin(91.5)}\right).\end{aligned} \tag{6}$$

The propulsion control expression (5) is merely a proportional action, whereas the heading control expression (6) appears to be very complicated. However, analyzing the range of values used as arguments of the hyperbolic tangent, it is easy to see that the function does not reach the saturation limits; thus, it acts as a proportional term. Therefore, it can be simplified by applying

$$\tanh\left(\frac{\mathbf{h}'_h(\mathbf{x}_h - \mathbf{x}_{hd})}{7.6}\right) = 0.132\mathbf{h}'_h(\mathbf{x}_h - \mathbf{x}_{hd}) \tag{7}$$

$$\tanh\left(\frac{\mathbf{h}'_h(\mathbf{x}_h - \mathbf{x}_{hd})}{-68.2}\right) = -0.015\mathbf{h}'_h(\mathbf{x}_h - \mathbf{x}_{hd}) \tag{8}$$

$$\tanh\left(\frac{\mathbf{h}'_h(\mathbf{x}_h - \mathbf{x}_{hd})}{\sin(91.5)}\right) = -2.63\mathbf{h}'_h(\mathbf{x}_h - \mathbf{x}_{hd}). \tag{9}$$

This results in a proportional controller plus a sine term, i.e.,

$$\tau_3 = -38.87\mathbf{h}'_h(\mathbf{x}_h - \mathbf{x}_{hd}) - 0.13\,\sin(\psi). \tag{10}$$

If the heading of the ship is smaller than 0.5 rad (approximately 30°), then the sine term can be approximated by the heading signal.

Fig. 9 shows that the performance of the controllers considered is best obtained from run 1 [see (5) and (10)] when tracking

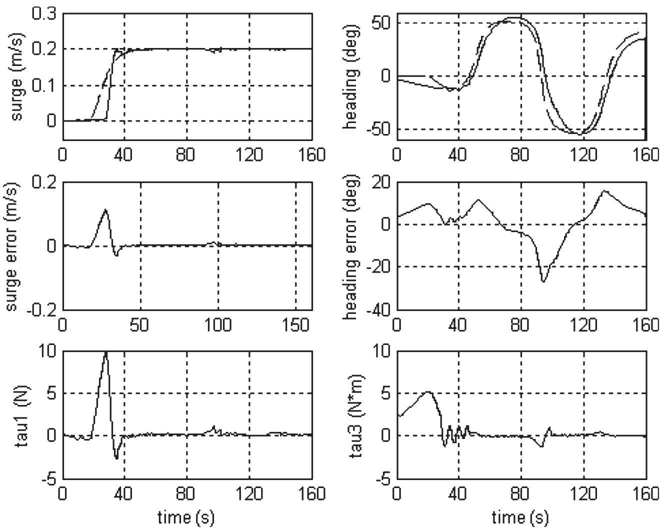

Fig. 10. Real results of the GP optimization without waves when maneuvering in calm waters.

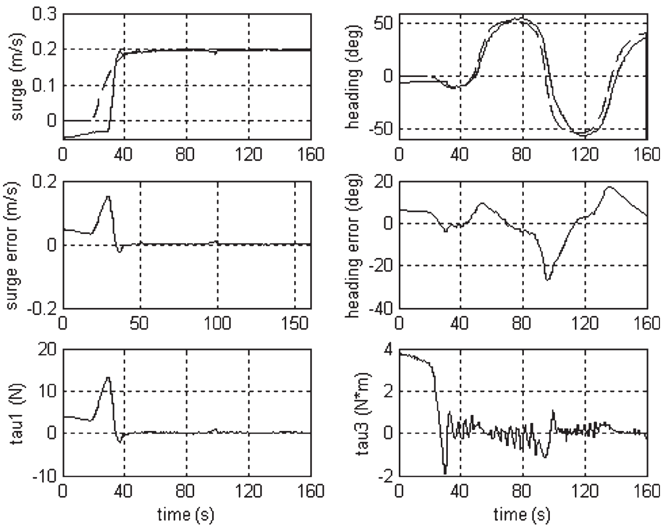

Fig. 11. Real results of the GP optimization without waves when maneuvering in the presence of waves.

the evaluation maneuver. This plot is the standard format that has been used in this paper for the presentation of the results. It is divided into six subplots. The results obtained for the propulsion subsystem are plotted on the left-hand side, whereas those for the heading subsystem are plotted on the right-hand side. The subplots at the top of the figure represent the desired and measured outputs, i.e., surge and heading. The desired outputs are plotted as a dashed line, whereas the actual outputs are represented as a solid line. The subplots in the middle of the figure represent the output errors, i.e., the surge and heading errors. Finally, the subplots at the bottom of the figure depict the control signals corresponding to the propulsion and heading subsystems, i.e., τ_1 and τ_3, respectively.

In the performance of the controllers for the zigzag maneuver, there is a slight steady-state error in the surge response, which is caused by the lack of an integral term in the propulsion control and the small proportional gain imposed by the lack of a derivative term [see (5)]. The heading control is also good, apart from a slight overshooting caused by the high gain.

Figs. 10 and 11 show the responses obtained when the controllers have been implemented on the real plant.

Fig. 10 illustrates the results obtained when maneuvering in calm waters. The responses obtained are quite good. The tracking is quite accurate, although the same overshooting that has been observed in the simulated responses for heading can also be appreciated in the real responses.

Fig. 11 shows the effect of real waves in the performance of the controllers obtained in the GP optimization without waves.

When comparing Figs. 10 and 11, it can be seen that the main difference is the presence of wave-induced high-frequency components in the heading control action. Apart from that, the tracking responses are very similar.

The values recorded at time 0 are caused by the difficulty of keeping the boat still while starting the maneuver. This was particularly difficult when waves were applied. It can also be observed that the system introduces a slight delay (i.e., it takes a few seconds for the boat to start) that causes the initial peaks in the surge error and in the propulsion force.

TABLE V
BEST GP RESULT OPTIMIZATION (WITH WAVES)

Run No.	Zig-zag	∞-shape	Run No.	Zig-zag	∞-shape
1	12.90	79.78	2	79.15	213.40
3	123.78	360.49	4	18.09	194.95
5	7.63	289.88	6	51.55	646.29
7	120.11	423.66	8	12.34	5.94e5
9	370.40	949.33	10	328.79	843.70
11	285.32	841.22	12	53.37	534.41
13	132.25	445.48	14	114.01	705.51
15	275.93	954.56	16	95.92	1.51e5
17	327.72	5.33e4	18	351.56	5.32e4
19	30.52	7.35e5	20	47.73	1.50e6

C. Optimization With Waves: Simulated and Real Results

Table V presents the cost functions obtained after the GP optimization using the zigzag maneuver in the presence of waves and the posterior validation test using a double-turning-point maneuver. Since the validation test has been run without disturbances, the numerical costs can be compared to those of Table III.

The best overall result is obtained from run 1. Again, it is not the best result from the GP optimization but, rather, the third best result.

The amount of convergence in the final population of run 1 has been 43 individuals, and the generation of convergence has been the 15th generation. The convergence of the cost value for run 1 is shown in Fig. 12. The upper subplot shows the overall convergence, whereas the bottom subplot shows in more detail the convergence in the final generations.

Run 5 provided the best result from the GP runs, and the validation result is reasonably good. Other controllers that have performed well in both the optimization and the validation test are those of runs 2 and 4. As with the optimization without disturbances, some of the good results from the GP optimization failed the validation test (e.g., results from runs 8, 19, and 20).

Table VI compares the cost values obtained by the best controllers in the GA optimizations described in the previous

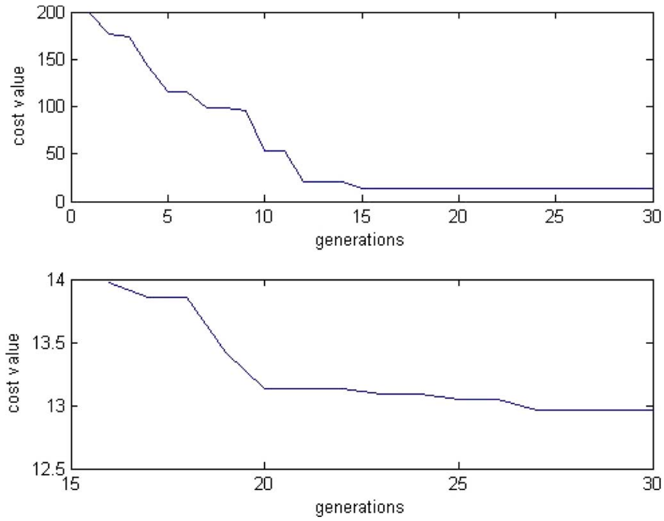

Fig. 12. Convergence of the cost value for run 1 (with waves).

TABLE VI
COST VALUES OBTAINED FOR THE ZIGZAG MANEUVER WITH VARIOUS CONTROL STRUCTURES (WITH WAVES)

GP	PID	PP	SM+PI	SM	H_∞
12.9	4.3	10.6	10.5	11.5	33.4

section with the cost value obtained by GP. As can be seen, the numerical result obtained with GP is significantly lower than the result obtained with H_∞, but it is comparable with the results provided by pole placement and sliding mode.

The heading and propulsion controllers of the best results are based on a hyperbolic tangent. Thus

$$\tau_1 = -203.54 \tanh\left(\frac{h'_p(u-u_d)}{11.8223}\right) + 2\sin(u_d) \qquad (11)$$

$$\tau_3 = 14.75 \tanh\left(\frac{\mathbf{h}'_h(\mathbf{x}_h - \mathbf{x}_{hd})}{-0.7931}\right). \qquad (12)$$

Again, the hyperbolic tangents are used as proportional controllers. Thus, the equations could be expressed as

$$\tau_1 \approx -17.5 h'_p(u-u_d) + 2\sin(u_d) \qquad (13)$$

$$\tau_3 \approx 19\mathbf{h}'_h(\mathbf{x}_h - \mathbf{x}_{hd}). \qquad (14)$$

The propulsion controller is similar to that obtained in the GP optimization without waves. The sine function of the desired surge speed increases the control effort by acting as a form of feedforward control (i.e., providing the same control effort regardless of the current state of the boat).

Fig. 13 illustrates the performance of the controllers from run 1 [see (13) and (14)] when tracking the zigzag maneuver used for GP optimization.

It can be seen that the inclusion of waves during the optimization process affects the propulsion control more than it affects the heading control, although both cope quite well. Comparing the results obtained using the GP optimization with and without waves, it can be observed that although the propulsion control in both controllers is proportional, the proportional gain for the controller optimized with waves is much smaller (17.5 versus 88.4), which explains the slow transient response. The heading control is not significantly affected by the disturbances, apart from the ripple in the actuator force caused by the disturbances. Again, when compared with the heading controller obtained in the GP optimization without waves, both are proportional controllers, but the inclusion of waves has led to a smaller gain (19 versus 38.9). This leads to the conclusion that the inclusion of disturbances in the optimization effectively reduces the use of actuators by reducing the controller gain.

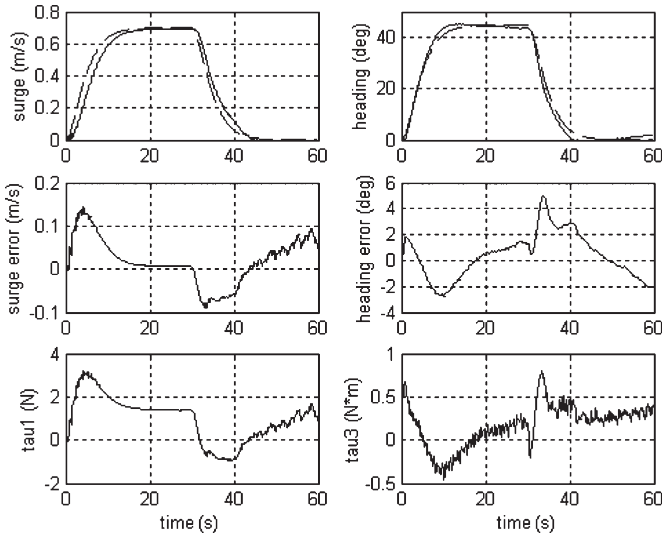

Fig. 13. Simulated results of the GP optimization with waves.

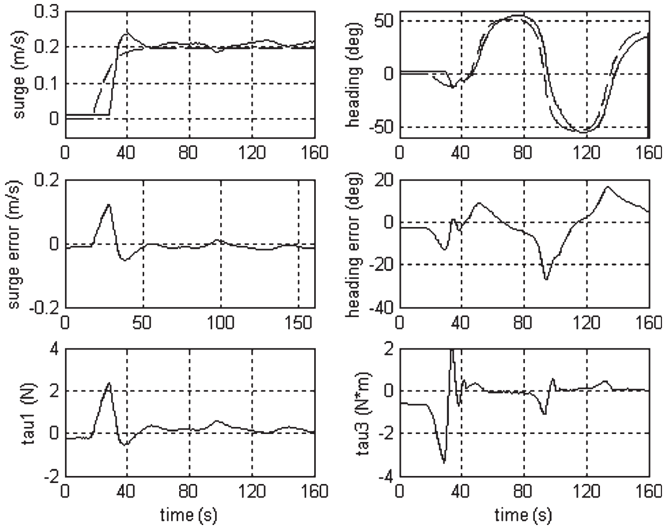

Fig. 14. Real results of the GP optimization with waves when maneuvering in calm waters.

Figs. 14 and 15 show the results obtained when the controllers are implemented in the real plant.

Fig. 14 shows the responses obtained when the maneuvering has been performed in calm waters. The real responses obtained are quite satisfactory. The tracking is quite good, particularly for the heading response, although it has a slight overshooting. When compared with the real responses obtained for the GP optimization without waves (see Fig. 10), it can be observed that, although the surge response is worse, the surge control action has been reduced.

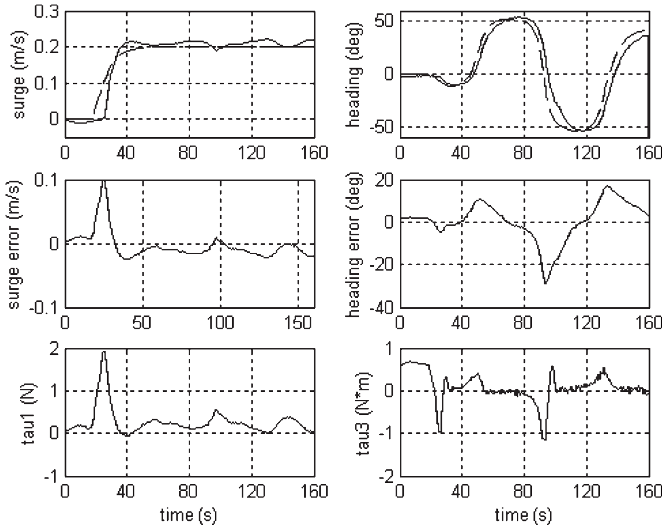

Fig. 15. Real results of the GP optimization with waves when maneuvering in the presence of waves.

Fig. 15 shows the responses obtained with the controllers when the same zigzag maneuver has been performed in the presence of waves in the water tank. As can be seen in the figure, the main effect of including waves can be observed in the disturbance-induced ripples in the heading control action. Apart from that, the responses show good tracking of the heading response and satisfactory surge tracking.

V. Conclusion and Future Work

The results obtained in the GP optimization are very satisfactory. The maneuvering performance of these controllers, as illustrated in the figures, proves their adequacy.

Both optimizations (with and without waves) have converged to trees that provide very similar control strategies. The best results obtained in all four sets of runs are based on a hyperbolic tangent function providing the heading control and a proportional term or another hyperbolic function providing the propulsion control.

The terminal values chosen by the search method as arguments of the hyperbolic functions for these best results make this function operate in its proportional range instead of in the switching area. Thus, in the case of the propulsion control, since the subsystem is of first order, the hyperbolic tangent provides an outcome that is proportional to the surge speed error (i.e., a proportional term). In the case of the heading control, the resulting commanded force is effectively of the form $\tau_3 \approx -\mathbf{Kh}'(\mathbf{x} - \mathbf{x}_d)$. This is, in fact, a full-state feedback control system with a feedback matrix equal to $\mathbf{Kh}'$ and a conditioning matrix multiplying the state reference equal to $\mathbf{Kh}'$.

With regard to the effect of the inclusion of waves in the evaluation of the candidate solutions, the results from the GP show that the inclusion of waves leads to controllers that reduce the control gain more than they improve the robustness against external disturbances. The GP concentrates on reducing the ripple in τ_1 (the main effect of the waves) to improve the cost function value, which degrades the performance of the propulsion controllers. On the other hand, the navigation subsystem is less sensitive to the effect of waves. Consequently, the resulting heading controller parameters in the optimization with waves are similar to those obtained in the GA optimization without waves.

As a result, the advantage of including noise in the simulation lies in obtaining controller solutions with smoother control signals that are able to reduce the wear and tear on the actuators. Unfortunately, the inclusion of simulated waves during the optimization process does not help provide solutions that exhibit better disturbance rejection properties.

In the future, we plan to consider multiobjective GP for the evaluation of the controller solution in the GP optimization. The cost function considered in this paper had six terms, reflecting the objectives of the control strategy: good tracking, minimization of actuator effort, and minimization of oscillations in the actuators for both propulsion and heading control. Using multiobjective GP would lead to a better tradeoff between these objectives.

In addition, although the decoupled controllers have proven to be very efficient for the control of this particular plant, the GP implementation could be upgraded to a MIMO controller having a single tree with three roots for τ_1, τ_2, and τ_3. This would provide a higher level of system integration.

References

[1] T. I. Fossen, *Guidance and Control of Ocean Vehicles*. Chichester, U.K.: Wiley, 1994.

[2] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.

[3] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.

[4] A. Chipperfield and P. Fleming, "Genetic algorithms in control systems engineering," *Control Comput.*, vol. 23, pp. 88–94, 1995.

[5] P. Wang and D. P. Kwok, "Optimal design of PID process controllers based on genetic algorithms," *Control Eng. Pract.*, vol. 2, no. 4, pp. 641–648, Aug. 1994.

[6] E. Alfaro-Cid, "Optimisation of time domain controllers for supply ships using genetic algorithms and genetic programming," Ph.D. dissertation, Univ. Glasgow, Glasgow, U.K., 2003.

[7] J. R. Koza, M. A. Keane, F. H. Bennet, III, J. Yu, W. Mydlowec, and O. Stiffelman, "Automatic creation of both the topology and parameters for a robust controller by means of genetic programming," in *Proc. IEEE Int. Symp. Intell. Control, Intell. Syst. Semiotics*, 1999, pp. 344–352.

[8] J. R. Koza, M. A. Keane, J. Yu, W. Mydlowec, and F. H. Bennet, III, "Automatic synthesis of both the topology and parameters for a controller for a three-lag plant with a five-second delay using genetic programming," in *Proc. EvoWorkshops*, 2000, vol. 1803, pp. 168–177.

[9] D. C. Dracopoulos and S. Kent, "Genetic programming for prediction and control," *Neural Comp. Appl.*, vol. 6, no. 4, pp. 214–228, Dec. 1997.

[10] H. Shimooka and Y. Fujimoto, "Generating equations with genetic programming for control of a movable inverted pendulum," in *Proc. SEAL*, 1998, vol. 1585, pp. 179–186.

[11] J. Busch, J. Ziegler, C. Aue, A. Ross, D. Sawitzki, and W. Banzhaf, "Automatic generation of control programs for walking robots using genetic programming," in *Proc. EuroGP*, 2002, vol. 2278, pp. 258–267.

[12] P. Nordin and W. Banzhaf, "Real time control of a Khepera robot using genetic programming," *Cybern. Control*, vol. 26, no. 3, pp. 533–561, 1997.

[13] S. G. Handley, "The automatic generation of plans for a mobile robot via genetic programming with automatically defined functions," in *Advances in Genetic Programming*. Cambridge, MA: MIT Press, 1994, pp. 391–407.

[14] K. A. Marko and R. J. Hampo, "Application of genetic programming to control of vehicle systems," in *Proc. Intell. Veh. Symp.*, Detroit, MI, 1992, pp. 191–195.

[15] N. A. J. Witt, R. Sutton, and K. M. Miller, "Recent technological advances in the control and guidance of ships," *J. Navigat.*, vol. 47, no. 2, pp. 236–258, May 1994.

[16] Z. Kowalski, R. Arendta, M. Meler-Kapciab, and S. Zielskib, “An expert system for aided design of ship systems automation,” *Expert Sys. Appl.*, vol. 20, no. 3, pp. 261–266, 2001.

[17] R. Sutton and I. M. Jess, “Design study of a self-organizing fuzzy autopilot for ship control,” *Proc. Inst. Mech. Eng., Pt. I. J. Syst. Control Eng.*, vol. 205, no. I1, pp. 35–47, 1991.

[18] M. A. Unar and D. J. Murray-Smith, “Automatic steering of ships using neural networks,” *Int. J. Adapt. Control Signal Process.*, vol. 13, no. 4, pp. 203–218, 1999.

[19] C. J. Harris, C. G. Moore, and M. Brown, *Intelligent Control: Aspects of Fuzzy Logic and Neural Networks*. Singapore: World Scientific, 1993.

[20] R. Sutton, G. N. Roberts, and S. D. H. Taylor, “Tuning fuzzy ship autopilots using artificial neural networks,” *Trans. Inst. Meas. Control*, vol. 19, no. 2, pp. 94–106, 1997.

[21] K. P. Lindegaard and T. I. Fossen, “Fuel efficient rudder and propeller control allocation for marine craft: Experiments with a model ship,” *IEEE Trans. Control Syst. Tech.*, vol. 11, no. 6, pp. 850–862, Nov. 2003.

[22] K. P. Lindegaard, “Acceleration feedback in dynamic positioning,” Ph.D. dissertation, Norwegian Univ. Sci. Technol., Trondheim, Norway, 2003.

[23] J. Corneliussen, “Implementation of a guidance system for CyberShip II,” M.S. thesis, Norwegian Univ. Sci. Technol., Trondheim, Norway, 2003.

[24] D. A. Sveen, “Robust and adaptive tracking control of surface vessel for synchronization with an ROV: Practical implementation on CyberShip II,” M.S. thesis, Norwegian Univ. Sci. Technol., Trondheim, Norway, 2003.

[25] J. K. Zuidweg, “Automatic guidance of ships as a control problem,” Ph.D. dissertation, Delft Univ. Technol., Delft, The Netherlands, 1970.

[26] D. E. Goldberg and K. Deb, “A comparative analysis of selection schemes used in genetic algorithms,” *Found. Genet. Algorithms*, vol. 1, pp. 69–93, 1991.

[27] U. O’Reilly, *An Analysis of Genetic Programming*. Ottawa, ON, Canada: Carleton Univ., 1995.

[28] E. Alfaro-Cid, E. W. McGookin, and D. J. Murray-Smith, “Evolution of a strategy for ship guidance using two implementations of genetic programming,” in *Proc. EuroGP*, 2005, vol. 3447, pp. 250–260.

[29] J. J. E. Slotine, “Sliding controller design for nonlinear systems,” *Int. J. Control*, vol. 40, no. 2, pp. 421–434, 1984.

[30] J. Kaustky, N. K. Nichols, and P. Van Dooren, “Robust pole assignment in linear state feedback,” *Int. J. Control*, vol. 41, no. 5, pp. 1129–1155, May 1985.

[31] E. Alfaro-Cid, E. W. McGookin, D. J. Murray-Smith, and T. I. Fossen, “Optimisation of a decoupled sliding mode controller for a supply ship using genetic algorithms: Simulated and real results,” *Control Eng. Pract.*, vol. 13, no. 6, pp. 739–748, 2005.

[32] S. Luke, “When short runs beat long runs,” in *Proc. GECCO*, San Francisco, CA, 2001, pp. 74–80.

[33] “Standards for Ships Manoeuvrability,” *Resolution MSC.137*, vol. 76, 2002.

[34] E. Alfaro-Cid, E. W. McGookin, and D. J. Murray-Smith, “GA-optimised PID and pole placement real and simulated performance when controlling the dynamics of a supply ship,” *Proc. Inst. Electr. Eng.—Control Theory Applicat.*, vol. 153, no. 2, pp. 228–236, 2006.

[35] E. Alfaro-Cid, E. W. McGookin, and D. J. Murray-Smith, “Optimisation of the weighting functions of an H_∞ controller using genetic algorithms and structured genetic algorithms,” *Int. J. Syst. Sci.*, vol. 39, no. 4, pp. 335–347, Apr. 2008.

Eva Alfaro-Cid received the degree in industrial engineering from the Universidad de Zaragoza, Zaragoza, Spain, and the Ph.D. degree from the University of Glasgow, Glasgow, U.K.

She has more than five years of experience in the field of evolutionary computation. She is currently with the Complex Adaptive Systems Group, Instituto Tecnológico de Informática, Valencia, Spain. Her work has addressed optimization, classification, prediction, and regression problems in several areas, particularly automatic control and finances. She has published many articles in international journals and conferences and belongs to the program committee of several of them.

Dr. Alfaro-Cid was a recipient of the Juan de la Cierva Research Fellowship in 2005.

Euan W. McGookin (S’96–A’97–M’01) received the M.Eng. degree in avionics and the Ph.D. degree in control engineering from the University of Glasgow, Glasgow, U.K.

Until August 2006, he was a Lecturer with the Department of Electronics and Electrical Engineering, University of Glasgow, where he is currently a Senior Lecturer with the Department of Aerospace Engineering, leading research into biomimetic underwater vehicles. His current research interests and activities are concerned with dynamic modeling and intelligent control system design for autonomous vehicles. Primarily, these activities are focused on underwater vehicle applications through both simulation and physical trials.

David J. Murray-Smith (M’01) received the B.Sc., Eng., and M.Sc. degrees from the University of Aberdeen, Aberdeen, U.K., and the Ph.D. degree from the University of Glasgow, Glasgow, U.K.

Until October 2005, he was a Professor of engineering systems and control with the Department of Electronics and Electrical Engineering, University of Glasgow, where he is currently an Emeritus Professor. Much of his research over the past 40 years has been interdisciplinary in nature and has involved collaboration with colleagues in departments outside the electrical engineering field. He has also been involved in joint projects with other universities and with research establishments and industry. His current research interests and activities are principally concerned with system modeling and simulation for control engineering and intelligent systems applications. Over the past 20 years, his main applications areas have been in helicopter flight mechanics modeling, helicopter flight control, and the control of surface ships and underwater vehicles. He has also been involved in biomedical projects concerned with modeling of elements of the neuromuscular control system and the development of models for the respiratory system for conditions of exercise. These modeling and simulation activities involve system identification from measured test data, as well as the development of improved methods for the optimization and external validation of models derived from physical principles.

Thor I. Fossen (S’88–M’91–SM’00) received the M.Sc. degree in naval architecture and the Ph.D. degree in control systems from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 1987 and 1991, respectively. In the period 1989–1990, he pursued postgraduate studies as a Fulbright Scholar in aerodynamics and flight control with the Department of Aeronautics and Astronautics, University of Washington, Seattle.

In 1993, he was appointed as a Professor of guidance and control with NTNU. He is currently the Head of Automatic Control, Centre for Ships and Ocean Structures (CESOS)–Norwegian Centre of Excellence, NTNU. He is the author of the books *Guidance and Control of Ocean Vehicles* (Wiley, 1994) and *Marine Control Systems* (Marine Cybernetics, 2002) and a coauthor of the book *New Directions in Nonlinear Observer Design* (Springer-Verlag, 1999). He has served as Principal Supervisor to 17 Ph.D. students. He has been instrumental in the development of several industrial autopilot and dynamic positioning systems. He also founded the company Marine Cybernetics, which offers service for hardware-in-the-loop testing of marine control systems. He has also been involved in the design of the SeaLaunch trim and heel correction systems.

Prof. Fossen was the International Program Committee Chair of the 1995 International Federation of Automatic Control (IFAC) Conference on Control Applications in Marine Systems (CAMS’95), the Network Operations Committee Chair of CAMS’98, the Chair of the IEEE Oceanic Engineering/Control Systems Joint Society Chapter of the Norway Section from 1996 to 2000, and the Vice Chair of the IFAC Technical Committee on Marine Systems from 1995 to 1999. He was the recipient of the Automatica Prize Paper Award in 2002 for his work on weather optimal positioning control for marine vessels.