Multiobjective fuzzy genetic algorithm optimisation approach to nonlinear control system design

A.Trebi-Ollennu
B.A.White

Indexing terms: Multiobjective optimisation, Fuzzy sets, Genetic algorithm, Marine systems, Robustness, Multivariable control

Abstract: Owing to the large number of free control parameters for modern nonlinear robust controllers, it is almost impossible to heuristically tune these parameters. The multiobjective fuzzy genetic algorithm optimisation is shown to provide an effective, efficient and intuitive framework for selecting these parameters. The control structure and specifications are assumed to be given. Using the concept of fuzzy sets and convex fuzzy decision making, a multiobjective fuzzy optimisation problem is formulated and solved using a genetic algorithm. The relative importance of the objective functions is assessed by using a new membership weighting strategy. The technique is applied to the selection of free control parameters for an input-output linearising controller with sliding mode control, in a remotely-operated underwater vehicle depth control system.

1 Introduction

The number of nonlinear control techniques has greatly increased in the last decade. The theory of feedback linearisation has provided the necessary tools for synthesis of nonlinear systems and solution of the MIMO decoupling problem. However, current industrial practice does not generally include the use of modern nonlinear robust controllers. This may be attributed to several factors, one of which is the lack of a systematic and intuitive approach in selecting the large number of free control parameters to obtain an optimal controller.

In general, the design of a control system for a nonlinear multivariable system with many degrees of freedom involves a number of constraints and competing objectives. To obtain an optimal solution (i.e. a trade-off between the stated objectives) a formal mathematical method for decision-making is required. Multiobjective optimisation techniques [1–3] have been proposed as a possible solution. These provide a platform for incorporating the relative importance of each objective, a fundamental requirement when dealing with competing objectives. The only difficulty with this technique is the lack of a unique optimal solution, although the concept of a Pareto-optimum solution is introduced. This establishes a trade-off pattern such that no improvement may be made in one objective without adversely affecting another.

However, the emphasis on mathematical rigour conflicts with the imprecision that arises in the description of constraints, objective functions and the evaluation of the relative importance of objectives. In a typical control system design the constraints are usually mixed, with equality, inequality and fuzzy constraints such as settling time and thruster limit.

In order to generate flexible and robust solutions for multiobjective optimisation, the concept of fuzzy sets must be employed to represent the vast amount of vagueness that exists in both the objective and constraint functions. The concept of multiobjective fuzzy optimisation has been used extensively in strutural optimisation literature [4–7].

A new framework is presented for selecting free control parameters of an input-output linearising controller with sliding mode control (SMC) for the depth control system of a remotely operated underwater vehicle (ROV). This uses the concept of multiobjective fuzzy genetic algorithm (GA) optimisation and a new membership weighting strategy.

The study was inspired by contract work on the Autosub [Note 1] project for an ROV to be used for mine countermeasure operations and remote minehunting missions. In this role, the ROV has a sidescan sonar for mine hunting and clearance purposes. The operational requirements dictate precise control of depth and pitch in water depths of between 20m and 300m, to ensure a constant sonar footprint over an undulating seabed. Changes in depth should involve minimal pitching motion to avoid gaps in the sonar coverage. For a detailed description of the vehicle the reader is referred to [8].

Due to the highly nonlinear nature of the vehicle, it is natural for nonlinear control techniques such as input–output linearisation with SMC to be proposed as a possible solution to the depth control system. The concept of input–output linearisation guarantees the required decoupling between heave (depth) and pitch, while SMC provides performance and stability robustness to modelling errors. The simulation results for the above control structure are reported in [9]. The major problem encountered in this work was the lack of a systematic approach for selecting the free control parameters to ensure that the actuators did not saturate.

IEE Proceedings online no. 19971031

Paper first received 28th March and in revised form 8th October 1996

The authors are with Cranfield University, Royal Military College of Science, School of Engineering and Applied Science, Shrivenham, Swindon, Wiltshire SN6 8LA, UK

Note 1. Vehicle data from NERC Autosub, data source DRA Haslar.

2 ROV model

The equations of motion supplied by the DRA [Note 2] for the pitch plane of the ROV are

$$E\dot{x} = Fx + Gu \tag{1}$$

$$\begin{bmatrix} m - p3Z_{\dot{w}} & -p4Z_{\dot{q}} & 0 & 0 \\ -p4M_{\dot{w}} & I_y - p5M_{\dot{q}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} p2Z_{uw}U & (p3Z_{uq} + m)U & mu & 0 \\ p3M_{uw}U & p4M_{uq} & 0 & mgBG \\ 1 & 0 & 0 & -U \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} w \\ q \\ z \\ \theta \end{bmatrix} + \begin{bmatrix} p2Z_{uu\delta p}U^2 & p2Z_{uu\delta s}U^2 & 1 & 1 \\ p2M_{uu\delta p}U^2 & p3M_{uu\delta s}U^2 & -l_p & l_p \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_p \\ \delta_s \\ T_{stern} \\ T_{bow} \end{bmatrix} \tag{2}$$

where $p2 = l^2\rho/2$, $p3 = l^3\rho/2$, $p4 = l^4\rho/2$, $p5 = l^5\rho/2$, $l_p = 1.05$ is the thruster moment arm in the pitch plane, w = heave speed, q = pitch rate, z = depth (heave), θ = pitch, δ_p = port hydroplane deflection, δ_s = starboard hydroplane deflection, T_{stern} = stern thruster and T_{bow} = bow thruster. These equations are defined in body coordinates (x, y, z) and the ROV hydrodynamic data and notations may be found in [8].

The following control specifications are given with respect to a 10m step change in depth:

Peak overshoot	$O_s \le 0.3\,\mathrm{m}$
Rise time	$17 \le t_r \le 25\,\mathrm{s}$
Settling Time	$25 \le t_s \le 45\,\mathrm{s}$
Maximum pitch angle	$\theta_{max} \le 7.5°$
Hydroplane limit	$\delta_{hdp} \le 25°$
Thruster limit	$T_{bow} = T_{stern} \le 120\,\mathrm{N}$

where t_r is the time required for the depth (heave) to rise from 10% to 90% of its final value, and t_s is the time required for the depth to rise to ±2% of the final value (10m).

The initial specifications omitted the maximum allowable pitch angle for a 10m step change in depth. This is an important requirement since the initial pitch angle has a direct effect on the settling time. The maximum allowable pitch angle will also depend on the type of sidescanner employed if a constant footprint is to be ensured. Since no details of the sonar sidescanners are available, the maximum allowable pitch angle is set to 7.5°.

3 Robust controller design for ROV

Eqn. 1 can be transformed to

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \tag{3}$$

$$y(t) = h(x(t)) \tag{4}$$

where $f(x(t))$ and $g(x(t))$ are defined in [10], $f(x(t)) = E^{-1}F$, $g(x(t)) = E^{-1}G$, $u(t) = [u_q, u_h]^T$, and $y(t) = [z, \theta]^T$. The control objective is to make the output y track a desired trajectory y_d. The detailed derivation of the control law can be found in [10, 11].

Performing input–output feedback linearisation on eqns. 3 and 4 and assuming a (vector) relative degree $[r_1, \ldots, r_m]$ yields the following input–output mapping

$$\begin{bmatrix} y_1^{(r_1)} \\ y_2^{(r_2)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix} = B(x(t)) + A(x(t)) \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_m(t) \end{bmatrix} \tag{5}$$

It then follows that the state feedback control law of the form

$$u(t) = A^{-1}(x(t))[-B(x(t)) + v(t)] \tag{6}$$

where $v(t) = [v_1(t), \ldots, v_m(t)]^T \in R^m$ yields the linear system

$$y_i^{r_i}(t) = v_i(t) \quad i = 1, \ldots, m \tag{7}$$

The control law of eqn. 6 was formulated with the assumption that an accurate description of the system is available. However, in practice an accurate model is not readily available, so the presence of uncertainties will cause loss of I/O decoupling, steady-state tracking errors and a deterioration in transient responses. To increase the robustness of the control to parameter uncertainty, sliding mode control can be employed.

If $e_1 = z - z_d$ and $e_2 = \theta - \theta_d$ a sliding surface can be specified as

$$\dot{s} - Y^r = \begin{bmatrix} \dot{s}_1 - \ddot{z}_d \\ \dot{s}_2 - \ddot{\theta}_d \end{bmatrix} = \begin{bmatrix} k_1(\dot{z} - \dot{z}_d) - \ddot{z}_d \\ k_2(\dot{\theta} - \dot{\theta}_d) - \ddot{\theta}_d \end{bmatrix} \tag{8}$$

where

$$s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} \dot{e}_1 - k_1 e_1 \\ \dot{e}_2 - k_2 e_2 \end{bmatrix} \tag{9}$$

The overall expression for the control law u is:

$$\begin{bmatrix} u_p \\ u_q \end{bmatrix} = A^{-1} \begin{bmatrix} -B_1 - k_{11}(\dot{z} - \dot{z}_d) + \ddot{z}_d - \eta_1 \mathrm{sat}\left(\frac{s_1}{\Phi}\right) \\ -B_2 - k_{21}(\dot{\theta} - \dot{\theta}_d) + \ddot{\theta}_d - \eta_2 \mathrm{sat}\left(\frac{s_2}{\Phi}\right) \end{bmatrix} \tag{10}$$

The desired reference trajectory is generated using a command generator of the form

$$\ddot{y}_d = -q_i y_{di} - c_i \dot{y}_{di} + q_i r_i \tag{11}$$

where $i = 1, 2$; q_i, $c_i > 0$ and r_i is an external input.

4 Multiobjective fuzzy optimisation

In this Section, the multiobjective fuzzy optimisation problem is stated and fuzzy convex decision-making principles are outlined. The general multiobjective fuzzy optimisation problem can be stated as follows:

Find X which

$$\text{Minimizes} \quad f(X)$$
$$\text{such that} \quad g_j \in \underset{\sim}{b}{}_{\sim j}$$

where $f(X) = [f_1(X), f_2(X), \ldots, f_k(X)]$ is a vector objective function and $g_j(X)$ are constraints, with the tilde symbol indicating that the constraints contain fuzzy information.

The first stage is to fuzzify the objective functions and the fuzzy constraints. The membership function for the fuzzy objective function is

$$\mu_{f_i}(X) = \begin{cases} 0 & \text{if } f_i(X) > f_i^{max} \\ \frac{-f_i(X) + f_i^{max}(X)}{f_i^{max} - f_i^{min}} & \text{if } f_i^{min} < f_i(X) \le f_i^{max} \\ 1 & \text{if } f_i(x) \le f_i^{min} \end{cases} \tag{12}$$

where $\mu_{f_i}(X)$: $\mathcal{R}^n \rightarrow [0, 1]$ and $\mu_{f_i}(X)$ is a mapping from the real number set $\mathcal{R}^n$ to the closed interval [0, 1],

---

Note 2. Defence Research Agency, Sea Systems Sector, Winfrith.

which is a measure of the degree of satisfaction for any $X \in \mathcal{R}^n$ in the ith fuzzy objective function. f_i^{min} and f_i^{max} represent the minimum and maximum values for the objective function, respectively and are defined as

$$f_i^{min} = \min_i f_i(X^*) \quad \text{and} \quad f_i^{max} = \max_i f_i(X^*) \tag{13}$$

where X^* is the solution for each of the objective functions in the crisp domain.

The fuzzy constraints membership function is defined as

$$\mu_{g_j}(X) = \begin{cases} 0 & \text{if } g_j(X) > b_j + d_j \\ 1 - \left(\frac{g_j(X) - b_j}{d_j}\right) & \text{if } b_j \leq g_j(X) \leq b_j + d_j \\ 1 & \text{if } g_j(X) \leq b_j \end{cases} \tag{14}$$

where $\mu_{g_j}(X)$: $\mathcal{R}^n \rightarrow [0, 1]$ and $\mu_{g_j}(X)$ is the mapping from the real number set $\mathcal{R}^n$ to the closed interval [0, 1], which is an indication of the degree of satisfaction for any $X \in \mathcal{R}^n$ in the jth fuzzy constraint. $\mu_{g_j}(X) = 1$ represents complete satisfaction, $\mu_{g_j}(X) = 0$ is not satisfied and values between 0 and 1 represent the degree of satisfaction of the jth constraint. The allowable tolerances for each fuzzy constraint are given by d_j.

4.1 Fuzzy decision-making

The objective functions and constraints have been defined as fuzzy subsets in the space of alternatives using linear membership functions $\mu_{f_i}(X)$ and $\mu_{g_j}(X)$, respectively. The optimal decision is made by selecting the best alternative from the fuzzy decision space D characterised by the membership function μ_D. In other words, find the optimum X^* which maximises μ_D. This can be expressed mathematically as

$$\mu_D(X^*) = \max \mu_D(X) \tag{15}$$

where $\mu_D \in [0, 1]$.

The fuzzy decision can be made by employing one of the three generalised fuzzy decisions: intersection decision, convex decision and product decision. Convex decision-making principles are used in the present study.

The convex decision [12] uses the concept of arithmetic mean and provides a framework to incorporate the relative importance of all the objectives and constraints. This can be expressed mathematically as follows

$$D = \alpha f(X) + \beta g(X) \tag{16}$$

where α and β are weighting factors, which satisfy

$$\alpha + \beta = 1 \quad \alpha \geq 0 \quad \beta \geq 0 \tag{17}$$

For any fuzzy optimum set points, the weights α_i and β_j are given so that a linear weighted sum can be obtained. Thus the membership function for the convex decision can be expressed as follows

$$\mu_D(X) = \sum_{i=1}^{k} \alpha_i \mu_{f_i} + \sum_{j=1}^{m} \beta_j \mu_{g_j} \tag{18}$$

where α_i and β_j satisfy

$$\begin{aligned} &\sum_{i=1}^{k} \alpha_i + \sum_{j=1}^{m} \beta_j = 1 \\ &\alpha_i \geq 0 \quad i = 1, 2, \ldots, k \\ &\beta_j \geq 0 \quad j = 1, 2, \ldots, m \end{aligned} \tag{19}$$

From eqn. 19 the original multiobjective fuzzy optimisation problem can be transformed into the following single-objective nonfuzzy optimisation problem:

$$\max \mu_D(X) = \sum_{i=1}^{k} \alpha_i \mu_{f_i}(X) + \sum_{j=1}^{m} \beta_j \mu_{g_j}(X) \tag{20}$$

$$g_i(X) \leq b_j + d_j \quad j = 1, 2, \ldots, m \tag{21}$$

This problem can be solved using any standard optimisation technique.

4.2 Membership weighting strategy

In this Section, a new membership weighting strategy is outlined for a convex-decision multiobjective fuzzy optimisation problem. The membership weighting factor is formulated as follows:

$$\begin{aligned} w_{f_i} &= 1 - \mu_{f_i}(X) \quad \text{and} \quad w_{g_j} = 1 - \mu_{g_j}(X) \\ \alpha_i &= \frac{w_{f_i}}{\sum_{i=1}^{k} w_{f_i}} \quad \text{and} \quad \beta_j = \frac{w_{g_j}}{\sum_{j=1}^{m} w_{g_j}} \end{aligned} \tag{22}$$

such that

$$\begin{aligned} &\sum_{i=1}^{k} \alpha_i + \sum_{j=1}^{m} \beta_j = 1 \\ &\alpha_i \geq 0 \quad i = 1, 2, \ldots, k \\ &\beta_j \geq 0 \quad j = 1, 2, \ldots, m \end{aligned} \tag{23}$$

This strategy addresses the problem in which the designer is not certain how to quantify the relative importance of each objective. It can be easily extended to cases where a particular objective or subset of objectives are more important, although the designer has no means of quantifying their relative importance. Using convex decision, the multiobjective fuzzy optimisation problem for selecting the free control parameters k_1, k_2, η_1, and η_2 for eqn. 10 can be formulated in the following manner.

From the above specifications, rise time and settling time are defined as fuzzy objectives and peak overshoot, hydroplane, and thruster limits as constraints. Note that all the constraints are nonfuzzy. The membership functions corresponding to the fuzzy objective functions are defined as

$$\mu_{t_s} = \begin{cases} 0 & \text{if } t_s \geq 45 \\ 1 - \frac{t_s - 25}{20} & \text{if } 25 < t_s < 45 \\ 1 & \text{if } t_s \leq 25 \end{cases} \tag{24}$$

and

$$\mu_{t_r} = \begin{cases} 0 & \text{if } t_s \geq 25 \\ 1 - \frac{t_r - 17}{8} & \text{if } 17 < t_r < 25 \\ 1 & \text{if } t_s \leq 17 \end{cases} \tag{25}$$

The convex decision for multiobjective fuzzy optimisation of eqn. 20 will be used, since there are no fuzzy constraints ($\beta_i = 0$). To obtain the highest degree of membership to the fuzzy convex decision set, the multiobjective fuzzy optimisation is formulated as; find k_1, k_2, η_1, and η_2

$$\begin{aligned} \text{which maximize} \quad & \mu_D = \alpha_1 \mu_{t_s} + \alpha_2 \mu_{t_r} \\ \text{subject to} \quad & O_s \leq 0.3\,\text{cm} \\ & \theta_{max} \leq 7.5^\circ \\ & \delta_{hdp} \leq 25^\circ \\ & T_{bow} \leq 120\,\text{N} \\ & T_{stern} \leq 120\,\text{N} \end{aligned}$$

where α_i are defined in eqns. 22 and 23.

Genetic algorithms (GAs) will be used as the preferred optimisation technique to solve this problem. The choice of GAs is based on the fact that they have been shown to produce optimum solutions for parameterised nonlinear controllers in multimodal search spaces. Because GAs evaluate several points simultaneously they also have the potential to accomplish highly adaptable parallel processors, since each point is evaluated iteration by iteration. This is an important requirement in multiobjective optimisation problems. There is no need to assume that the search space is differentiable or continuous and GAs only require a knowledge of the quality of the solution produced by each point thus making them very flexible. The interested reader may find a brief introduction to GAs in [13–15].

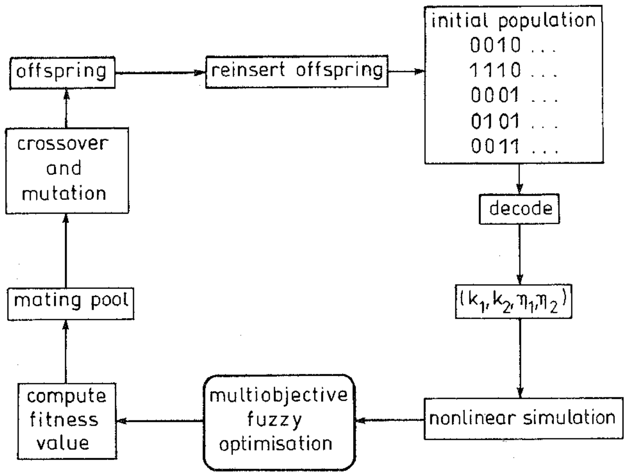

Fig. 1 *Flow chart of the design procedure for multiobjective fuzzy GA optimisation*

5 Multiobjective fuzzy GA optimisation

Fig. 1 shows a flow chart of the design procedure for the multiobjective fuzzy GA optimisation technique. A brief description of each step is given below:

Step 1: Initial population.

The basic execution cycle starts by randomly generating a population $P(0)$ of N individuals (strings), each of length l. It is worth noting that each individual represents a possible solution to the problem. These are then decoded into the decision variable space over the interval, e.g. $\{0, 500\}$, and are used in the nonlinear simulation.

Step 2: Evaluation of multiobjective fuzzy optimisation function.

The results of the simulation are passed to the multiobjective fuzzy optimisation to evaluate the degree of satisfaction (degree of membership, μ_D) and to check whether any of the constraints have been exceeded. The number of constraints exceeded and the degree of satisfaction values for each individual are passed to the fitness function.

Step 3: Fitness

Fitness is a metric used to assess the performance of individual members of a population relative to the rest of the population. The fitness of each individual in this example is calculated using the relation

$$\text{Fitness}_i = \mu_{D_i} - 0.2 C_N \mu_{D_i} \tag{26}$$

where C_N is the number of constraints exceeded and μ_{D_i} is the ith individual degree of satisfaction. The fitness values are then ranked using a linear ranking algorithm giving the fittest individual a fitness value of 2 and the least fit individual a fitness value of 0.

Step 4: Reproduction

Reproduction is the process whereby a mating pool is generated. Although the initial population $P(0)$ can be generated randomly, the next generation is chosen from the previous generation members using a probabilistic selection process. This ensures that individuals with large fitness values have a greater probability of contributing offsprings to the new population. The concept of generation gap is employed in this work.

The generation gap, G represents the percentage of the population to be replaced during each generation. For each new generation, $N * G$ individuals of the population $P(t)$ are selected to be replaced in $P(t + 1)$ generation; e.g. $G = 1.0$ implies that the whole population must be replaced for each generation. Using the function Select in [15], $N * G$ individuals are copied into a mating pool for possible use in the reproduction of the next generation.

Step 5: Crossover

Crossover utilises probabilistic decisions to exchange systematic information between two randomly selected individuals from the mating pool, to produce new individuals. The process involves the uniform random selection of a crossover point between the two individuals, followed by the exchange of all characters either to the right or left of this point. Two new individuals are generated after crossover. In this example, the single point crossover routine in [15] is used to perform crossover with probability, $P(x) = 0.7$.

Step 6: Mutation

Mutation generates new individuals by simply modifying one or more of the gene values (or bit values in the case of a binary representation) of an individual offspring after crossover. Mutation therefore provides a framework to ensure that a critical piece of information can always be reinstated or removed from a population. In this example, the default value and mutation function in [15] is employed.

Step 7: Reinsert offspring

Since a generation gap is used, the number of offspring is less than the size of the initial population. The offspring population is then inserted into the initial population to generate a new population.

Finally, steps 1 to 7 are repeated until $P(maxgen)$ = maximum number of generations ($maxgen$).

6 Simulation

Previous studies [9, 10, 16] have shown that the control specification for the ROV can only be met at the expense of thruster saturation. The multiobjective fuzzy GA optimisation will be applied to the ROV problem in an effort to eliminate the actuator saturation, while meeting the control specification.

Using the Genetic Algorithm Toolbox for use with MATLAB™ [15] the following GA parameters were set as $maxgen = 1000$, $N = 20$, $l = 32$, and $G = 0.9$. Binary chromosome structure and Gray scale coding was employed to represent each individual. For this study, $q_i = 0.9$, $c_i = 5$ and $r = [10, 0]$. The initial population is shown in Fig. 2 with corresponding fuzzy objective and fitness values for the respective individuals in this population. The fittest individual is marked with a circle.

The ROV responses are shown in Figs. 3 and 4 for the fittest individual. It is quite clear that the responses are not satisfactory, and so a new population is generated using the reproduction mechanism. This process is repeated several times.

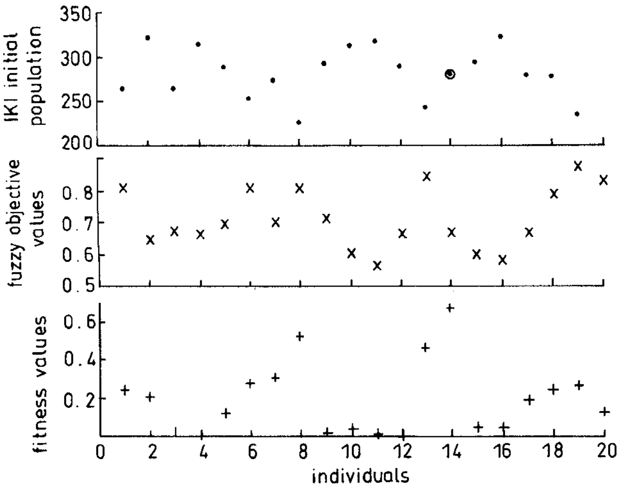

Fig. 2 *Initial population and corresponding fuzzy objective and fitness values*
○ fittest individual

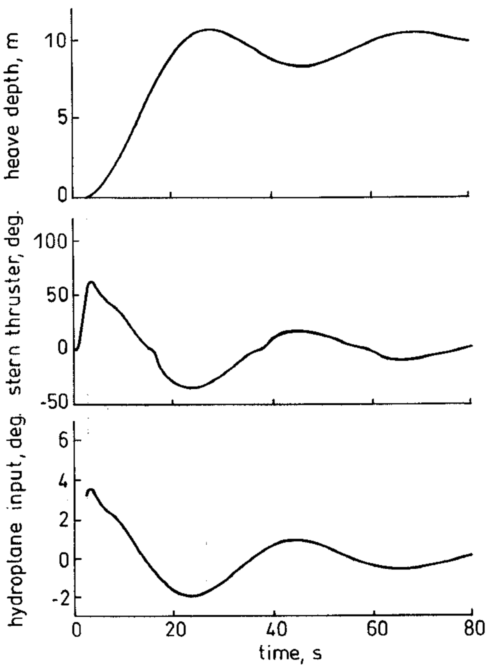

Fig. 3 *Nonlinear ROV responses for fittest individual in the initial population*

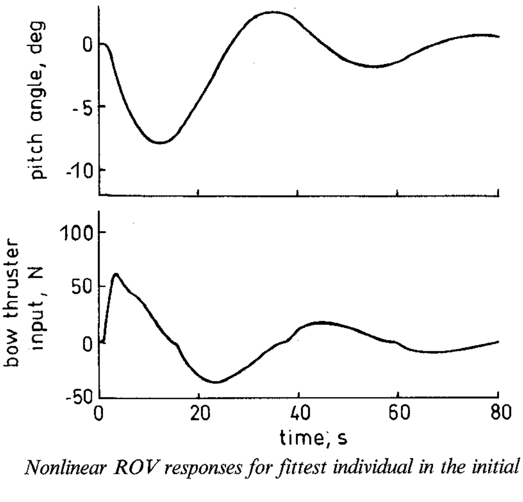

Fig. 4 *Nonlinear ROV responses for fittest individual in the initial population*

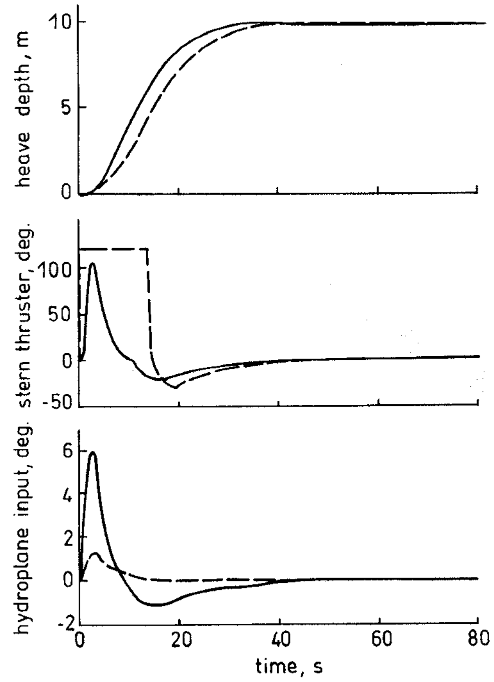

Fig. 5 *Nonlinear ROV reponses for multiobjective fuzzy GA optimisation and nonoptimisation*
——— optimisation
– – – – previous result

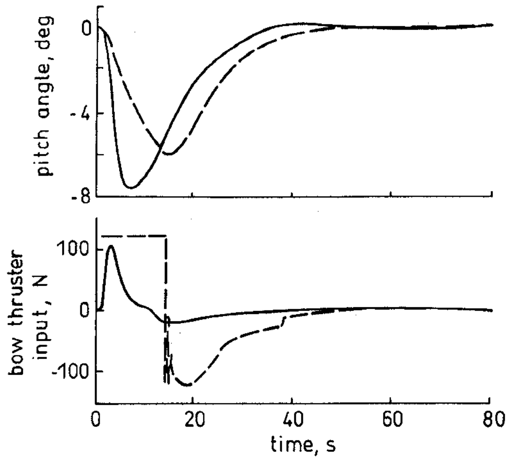

Fig. 6 *Nonlinear ROV reponses for multiobjective fuzzy GA optimisation and nonoptimisation*
——— optimisation
– – – – previous result

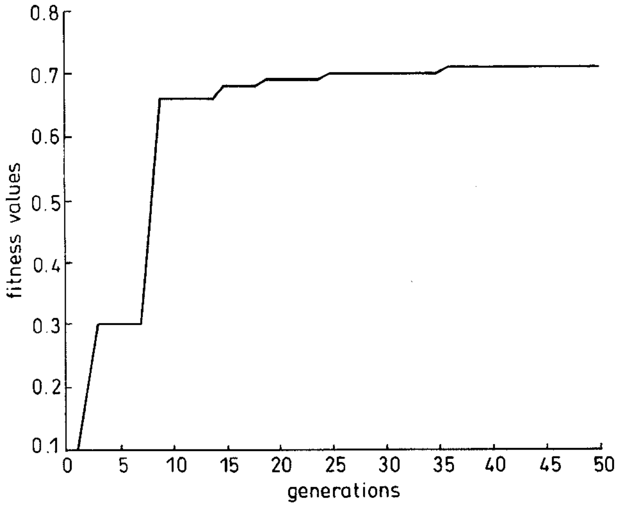

Fig. 7 *Generation history and convergence of fitness values*

After 50 generations, the optimal control parameters are $k_1 = 288.81$, $k_2 = 13.76$, $\eta_1 = 5.0$, and $\eta_2 = 38.0$. Figs. 4, 5 and 6 depicts the results of the ROV responses for multiobjective fuzzy GA optimisation and the results reported in [9]. The highest achievable degree of satisfaction (degree of membership) for the given constraints and objectives is 0.71, i.e. the best compromise solution due to the competing objectives. This corresponds to settling time $t_s = 31.62\,\text{s}$, rise time $t_r = 20.30\,\text{s}$, $O_s = 0.23\,\text{cm}$, $\theta_{max} = 7.14°$, maximum thrust demanded $T_{stern} = 104.60\,\text{N}$ and $T_{bow} = 104.04\,\text{N}$ and maximum hydroplane action demanded $\delta_{hdp} = 5.88°$. From Fig. 4 it is quite clear that thruster saturation was eliminated by employing more hydroplane action and consequently a larger pitch angle. The plot of the convergence of the fitness values is shown in Fig. 5.

The optimisation scheme was simulated in a SIMULINK/MATLAB environment using a 486 PC. The control system consisted of the nonlinear ROV model and control structure expressed in a SIMULINK block diagram. The execution time for each generation was 10min, i.e. 30s for each individual. It took eight and half hours to obtain the optimal controller. Future work will concentrate on reducing the execution time by employing workstations and parallel computers.

7 Conclusion

Multiobjective fuzzy GA optimisation has been shown via simulation to provide an intuitive process for selecting free control parameters for nonlinear controllers with competing and fuzzy specifications. The technique allows for the relative importance of each objective by employing a new membership weighting strategy.

The multiobjective fuzzy GA optimisation was successfully applied to an ROV depth control problem to eliminate actuator saturation, while maintaining performance. Future work will consider application of the proposed technique to the full six degree of freedom ROV and development of a metric objective function for performance robustness analysis.

8 References

1 HAJELA, P., and SHIH, C.J.: 'Multiobjective optimum design in mixed integer and discrete design variable problems', *AIAA J.*, 1990, pp. 670–675

2 FLEMING, P.J.: 'Computer aided design of regulators using multiobjective optimization'. Proceedings of IFAC conference on *Control applications of nonlinear programming and optimization*, Capri, Italy, 1985

3 FLEMING, P.J., and PASHKEVICH, A.P.: 'Application of multi-objective optimization to compensator design for SISO control systems', *Electron. Lett.*, 1986, pp. 258–259

4 RAO, S.S.: 'Multi-objective optimization of fuzzy structural systems', *Int. J. Numer. Methods Eng.*, 1987, pp. 1157–1171

5 XU, C.: 'Fuzzy optimization of structures by the two-phase method', *Computers and Structures*, 1989, pp. 575–580

6 SHIH, C.J., and LAI, T.K.: 'Fuzzy weighting optimization with several objective functions in structural design', *Computers and Structures*, 1994, pp. 917–924

7 SUZUKI, S., and YOSHIZAWA, T.: 'Multi-objective trajectory optimization by goal programming with fuzzy decisions', *J. Guidance Control and Dynamics*, 1994, pp. 297–303

8 WHITE, B.A., STACEY, B.A., PATEL, Y., and BINGHAM, C.: 'Robust control design of an ROV'. Contract no. DRA/PD/102/94, Cranfield University (RMCS), School of Electrical Engineering and Science, Shrivenham, April 1994

9 TREBI-OLLENNU, A., KING, J., and WHITE, B.A.: 'A study of robust multivariable control designs for remotely operated vehicle depth control systems', *IEE Digest*, 1995, **124**

10 TREBI-OLLENNU, A., and WHITE, B.A.: 'Nonlinear robust control designs for a remotely operated underwater vehicle depth control system', *IMechE Part I: J. Syst. Control Eng.*, 1996, **210**, pp. 85–97

11 TREBI-OLLENNU, A.: 'Robust nonlinear control designs using adaptive fuzzy systems'. PhD thesis, Cranfield University, 1996

12 YU, M., and XU, C.: 'Multi-objective fuzzy optimization of structures based on generalized fuzzy decision-making', *Computers and Structures*, 1994, pp. 411–417

13 AUSTIN, S.: 'An introduction to genetic algorithms', *AI Expert*, March 1990,

14 KRISHNAKUMAR, K.: 'Genetic algorithms: an introduction and an overview of their capabilities'. AIAA-92-4462-CP, 1992

15 CHIPPERFIELD, A., FLEMING, P., POHLHEIM, H., and FONSECA, C.: 'Genetic algorithm toolbax, for use with MATLAB'. Department of Automatic Control and Systems Engineering, University of Sheffield, 1994

16 TREBI-OLLENNU, A., KING, J., and WHITE, B.A.: 'A study of robust multivariable control designs for remotely operated vehicle depth control systems'. IFAC World Congress, San Francisco, 1996