MULTI-OBJECTIVE GENETIC ALGORITHM BASED TIME AND FREQUENCY DOMAIN DESIGN UNIFICATION OF LINEAR CONTROL SYSTEMS

Kay Chen Tan and Yun Li

Centre for Systems and Control, and
Department of Electronics and Electrical Engineering
University of Glasgow, Glasgow G12 8LT, United Kingdom.
E-mail: K.Tan@elec.gla.ac.uk, Y.Li@elec.gla.ac.uk

Keywords: Linear control systems, CACSD, multi-objective optimisation, genetic algorithms.

Abstract: *This paper presents a methodology that unifies linear control system designs using a multi-objective genetic algorithm (MOGA). The MOGA incorporates the concept of Pareto domination and applies niching and mating restriction techniques to evolve individuals along the Pareto optimal trade-off surface. Unlike existing design methods, the proposed technique integrates designs in both the time and the frequency domains under performance satisfaction. This evolution based design automation methodology avoids the need for pre-selection of a particular control scheme and enables the integration of different control laws in the design. By incorporating a priority vector, the MOGA allows different preference of the objective components and makes the design transparent, suitable for 'minimum-commitment' at the CAD stage. Results obtained show that the unified design is powerful and can include practical constraints such as actuator saturation while satisfying some conflicting design objectives.*

I. INTRODUCTION

With rapid development in linear control system theories and algorithms, many linear time-invariant (LTI) control schemes have been available to control engineers. However, each scheme has its individual control characteristics or strategies, which imposes an increasing challenge to engineers in selecting an appropriate control law before detailed designs can be attempted. This has prompted the desire of unifying LTI control laws. The unification could be based on performance satisfactions, to avoid pre-fixing a specific control scheme [1], [2]. Unlike existing mutually independent individual LTI control schemes, uniform LTI control (ULTIC) would allow engineers to simplify their design and implementation tasks, to integrate and interplay with multiple design objectives and to examine design trade-offs.

Developing ULTIC control systems usually involves simultaneously determining multiple coefficients for multiple objectives in a multi-modal multi-dimensional design space. Such tasks can hardly be performed by conventional analytical and numerical means, since the performance index of design may not be "well-behaved" and may include hard-constrained conditions [1,2,6-12]. In addition, conventional gradient-guided search cannot handle multiple objectives in a multi-modal search space and may only offer "local optima" [4], [5].

Emulating the Darwinian-Wallace principle of "survival-of-the-fittest" in natural selection and genetics, multi-objective genetic algorithms (MOGAs) [6]-[8] have been found to be very effective in searching poorly understood and complex space for optimisation in engineering applications [8]-[11]. Unlike conventional methods that linearly combine multiple attributes to form a composite scalar objective function, a MOGA incorporates the concept of Pereto's domination to evolve a family of non-dominated solutions along the Pareto optimal frontier. Each of the individual components in the cost function can have different priorities or preferences to guide the optimisation from individual design specifications rather than pre-weighting the cost function.

This paper presents a niching evolutionary MOGA and its application to integrating linear control system design methodology by unifying LTI approaches under performance satisfaction. The problem formulation of the ULTIC strategy is presented in Section II. Implementation of the MOGA with Pareto optimal ranking is illustrated in Section III. An ULTIC design example using the MOGA is given in Section IV. Finally, results and further work are summarised in Section V.

II. UNIFICATION OF LTI CONTROL LAWS

Almost all types of LTI controllers are in the form of a transfer function or a transfer function matrix equivalent to a state-space representation when the design is eventually complete. The order and the coefficients of the transfer function, however, vary with the control law or a single design objective. For example, a controller designed from the LQR scheme tends to offer a minimised quadratic error with some minimal control effort, while an H_∞ controller offers robust performance with a minimal mixed sensitivity function. Although the obtained coefficients or orders of these two types of controllers may be different, the common purpose of both control laws is to devise an LTI controller that could guarantee a closed-loop performance to meet

certain customer specifications in either the time or the frequency domain. Therefore, a step towards unification of LTI controllers is to coin the design by meeting practical performance requirements, instead of by a specific scheme [1], [2] or in a particular domain as shown in Fig. 1. An universal LTI controller can thus be described by

$$H(s) = \frac{U(s)}{E(s)} = \frac{a_n s^n + \cdots + a_1 s + a_0}{b_n s^n + \cdots + b_1 s + b_0} \tag{1}$$

where $\mathcal{L}^{-1}[U(s)] = u(t)$ is the controller output voltage with usually a hard-constraint saturation range such as the limited drive voltage or current, $\mathcal{L}^{-1}[E(s)] = e(t)$ is the error input to the controller, the amplitude of which may also be restricted by an A/D converter, and $a_i, b_i \in \Re \; \forall \; i \in \{0, 1, ..., n\}$ are the coefficients to be determined in the design. The disturbance attenuation and the robustness to the plant uncertainty in Fig. 1 can be quantified using a sensitivity function $S(s)$ and a complementary sensitivity function $T(s)$, respectively. Here, $S(s)$ and $T(s)$ are defined as

$$S(s) \overset{def}{=} \{I + G(s)H(s)\}^{-1} \tag{2.1}$$

$$T(s) \overset{def}{=} G(s)H(s)\{I + G(s)H(s)\}^{-1} = I - S(s) \tag{2.2}$$

Singular values of S determines the disturbance attenuation since S is in fact the closed-loop transfer from the disturbance d to plant output y, while singular values of T can be used to measure stability margins of the feedback system in terms of multiplicative plant uncertainty.

Small Gain Theorem: Suppose the nominal plant in Fig. 1 is stable with Δ_M being zero. Then the size of the smallest stable $\Delta_M(s)$ for which the system becomes unstable is [3]

$$\bar{\sigma}(\Delta_M(j\omega)) = \frac{1}{\bar{\sigma}(T(j\omega))} \tag{3}$$

where $\bar{\sigma}$ defines the largest singular value. Thus, the smaller $\bar{\sigma}(T(j\omega))$ is, the greater the size of the smallest destabilising multiplicative perturbation will be and, hence, the greater the stability margins of the system.

For control engineering applications, the performance requirements should reflect the usual design specifications such as:

Spec. 1: A good relative stability of the closed-loop system;
Spec. 2: An excellent steady-state accuracy in terms of small steady-state errors;
Spec. 3: An excellent transient response in terms of small rise-time, settling-time, overshoots and undershoots;
Spec. 4: Robustness in terms of disturbance rejection; and
Spec. 5: Robustness in terms of parameter uncertainty or sensitivity attenuation.

Design of an LTI control system involves optimising multiple coefficients of the controller. This leads to a multi-dimensional and multi-modal design space characterised by the above often non-commensurable and competing multi-objective performance indices. These and practical system constraints usually make it almost impossible to use conventional analytical or numerical optimisation techniques to solve the design problem for either a composite design objective or multi-objectives. Partly because of this, the unification had not yet been realised until the use of evolutionary methods [1], [2].

III. NICHING PARETO MULTI-OBJECTIVE GENETIC ALGORITHM

Multi-objective (MO) optimisation seeks to optimise a vector of non-commensurable and often competing objective or cost function that reflects the practical design specifications and constraints, viz. it tends to find a design parameter set for

$$\min_{\boldsymbol{P} \in \Phi} \boldsymbol{F}(\boldsymbol{P}) \tag{4}$$

where $\boldsymbol{P} = \{p_1, p_2, ..., p_n\}$ and Φ defines the set of candidate parameter variables and $\boldsymbol{F}(\boldsymbol{P}) = \{f_1(\boldsymbol{P}), f_2(\boldsymbol{P}), ..., f_n(\boldsymbol{P})\}$ are the objectives function to be minimised. The solution to the problem is a family of points known as Pareto optimal set, where each objective components of any point along the Pareto front can only be improved by degrading at least one of its other objective components. Conventional MO methods include the methods of inequalities, goal attainment or weighted sum. These approaches require a set of precise settings of weights or goals, which are usually not well manageable or understood [4], [5].

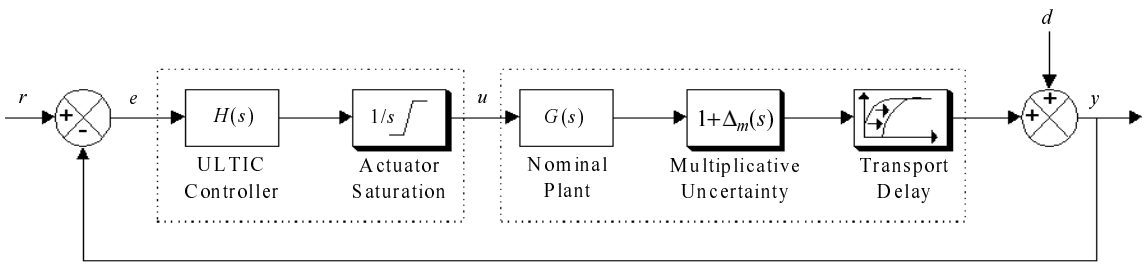

Fig. 1 A practical unity negative feedback control system

Pareto optimal ranking scheme for a MOGA is shown in Fig. 2a with two objectives f_1 and f_2 to be minimised. It assigns the same smallest cost for all non-dominated individuals, while the dominated individuals are inverse ranked according to how many individuals in the population dominating them [7]. So, an individual's rank in a population can be given by $rank(p_i) = 1 + k_i$, where k_i is the number of individuals that dominate the individual p_i in the current population. All the non-dominated individuals will be assigned rank 1. In order to further visualise or interplay with the objectives vector, a priority vector $\boldsymbol{G}_i = \{g_1, g_2, ..., g_n\} \in \aleph$ can be incorporated to reflect the preference of each component in the objective function. Where $\aleph$ is the natural numbers with a larger number represents a higher priority. Fig. 2.b illustrates the MO Pareto ranking with a higher preference upon objective f_2.

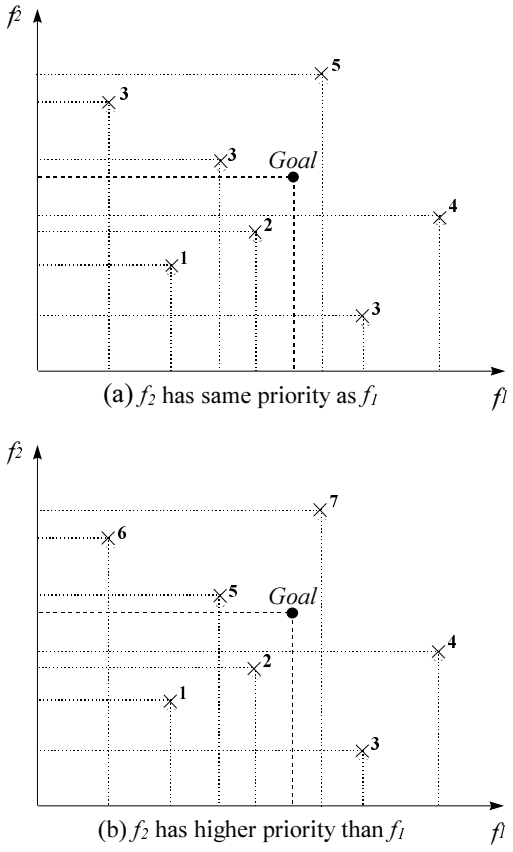

Fig. 2 Multi-objective Pareto optimal ranking in a MOGA

While maintaining a population of candidate solutions, MOGA searches the non-dominated solutions at multiple points and evolve along the Pareto frontier simultaneously and efficiently. For a problem with multiple, equal or unequal solutions, a simple GA will however converge to one of the global optimal solutions [6]. Thus, in order to spread equally the individuals along the Pareto surface, 'niche induction' technique is used and is implemented by a sharing function which creates subdivisions of the environment by degrading an individual's fitness proportional to the number of other individuals in its neighbourhood [6]. Mating restriction [7] is applied here, as arbitrary mating of individuals may result in a large number of unfit offspring and may subsequently highly decrease the non-dominated individuals along the Pareto surface.

Variation of the niching Pareto MOGA from a simple GA is shown in Fig. 3. Deterministic tournament selection is used in which individuals are divided into subgroups and the individuals with the smallest cost among the subgroup is selected. This approach avoids the need of chromosome sorting, interpolating and fitness averaging in every generation as required by the algorithms from [7] and thus saves the overhead computational time. Unlike other GA toolboxes for Matlab, this MOGA can be easily compiled into either stand-alone C++ source codes or "glue" MEX-files using Matcom [13] to significantly reduce the execution time. Here, a simple graphical user interface (GUI) as shown in Fig. 4 has been written in M-files to make use of the Matlab graphical capabilities.

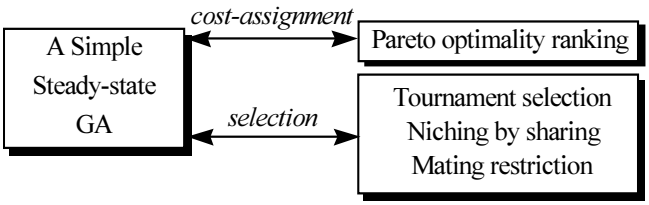

Fig. 3 A multi-objective genetic algorithm

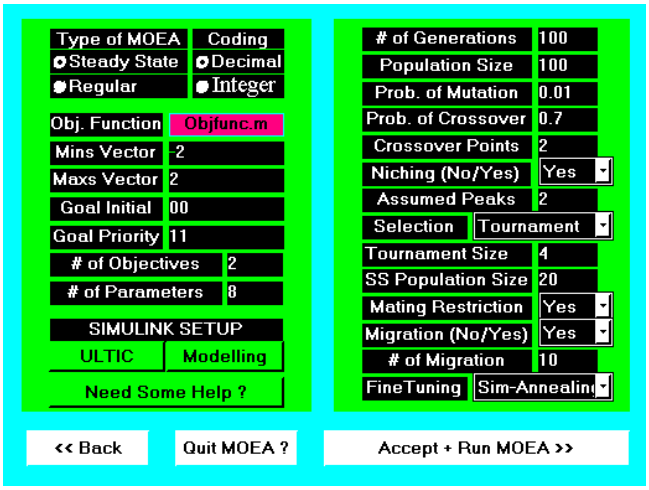

Fig. 4 A tabular graphical user interface for the MOEA

To study the performance of the MOGA, consider the minimisation of the following two objectives [14]:

$$f_1(x_1,...,x_8) = 1 - exp\left(-\sum_{i=1}^{8}\left(x_i - \frac{1}{\sqrt{8}}\right)^2\right)$$
$$f_2(x_1,...,x_8) = 1 - exp\left(-\sum_{i=1}^{8}\left(x_i + \frac{1}{\sqrt{8}}\right)^2\right) \quad (5)$$

Due to the symmetry of the two functions f_1 and f_2, the Pareto-optimal frontier are the points on the line defined by

$$x_1 = x_2 = \ldots = x_8 \wedge \frac{-1}{\sqrt{8}} \le x_1 \le \frac{1}{\sqrt{8}} \qquad (6)$$

Fig. 5 shows a trade-off surface corresponding to Equation 6. The MOGA with a population size of 100 was run for 50 generations. The parameters are decimal coded in the interval [-2,2) and concatenated to form the chromosomes. Multi-objective cost-assignment was performed as described earlier. It can be seen from Fig. 6 that the population is able to remain distributed and evolve across the entire trade-off surface. Goal information can also be accommodated to drive the population to sample a given region of the trade-off surface. As shown in Fig. 7, the MOGA was run for another 50 generations with a goal setting of $\{g_1, g_2\} = \{0.6, 0.8\}$. Obviously, the population is seen to concentrate on the preferred region of the trade-off surface, as desired.

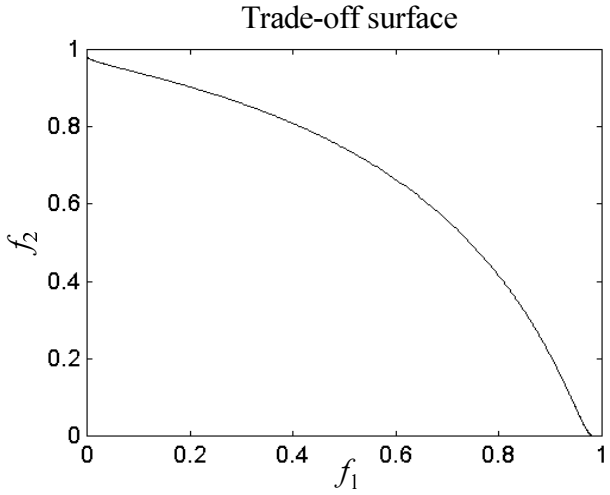

Fig. 5 Trade-off surface of the multi-objective optimisation

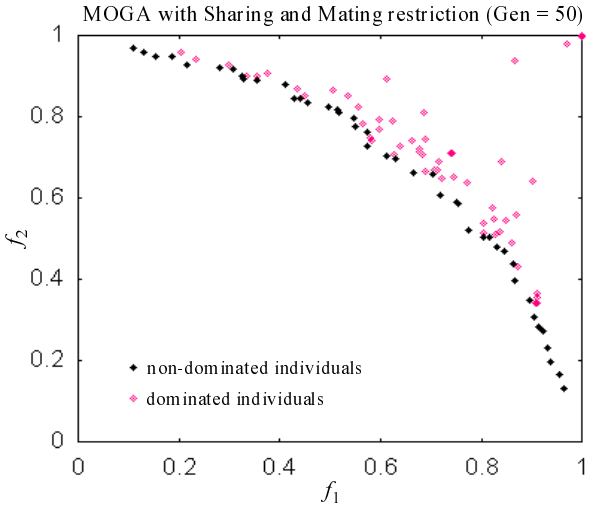

Fig. 6 MOGA with sharing and mating restriction

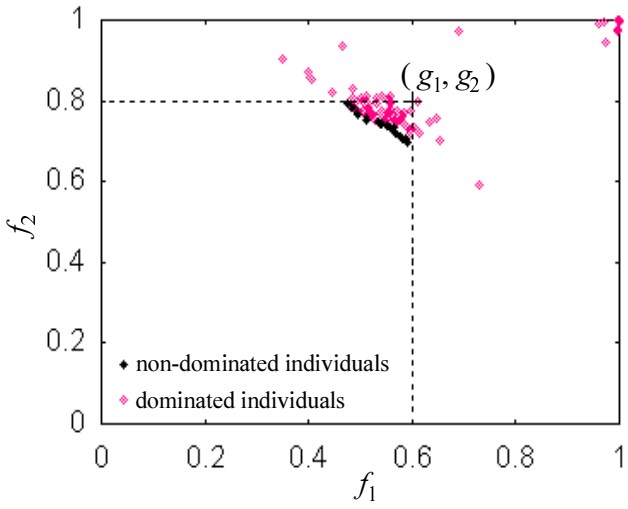

Fig. 7 Zooming into a region of the trade-off surface by setting goals $\{g_1, g_2\} = \{0.6, 0.8\}$

IV. ULTIC SYSTEM DESIGN EXAMPLE

To show how the MOGA can benefit the ULTIC system design, consider the following non-minimal phase plant [15]:

$$G(s) = \frac{-6.475s^2 + 4.0302s + 175.77}{5s^4 + 3.5682s^3 + 139.5021s^2 + 0.0929s} \qquad (7)$$

The design problem is to find an ULTIC controller that satisfies a number of time domain and frequency domain specifications as highlighted in Fig. 8, where W_1 and W_2 are the weighting functions used for the $S(s)$ and $T(s)$, respectively. The include of W_1 and W_2 allows the mixed sensitivity design objectives to be normalised. The gains of W_1 and W_2 are shown in Fig. 9. Table 1 details the nine design objectives for a step input command. The underlying aim of setting the priority vector in the third column of Table 1 is for obtaining a controller that first stabilises the system within the actuator saturation limit. Then, the system must also be robust to unstructured plant uncertainty and disturbance attenuation under certain level of tolerances defined by the weighting functions of W_1 and W_2, respectively. Having fulfilled these requirements, the system must also satisfy time domain specifications defined by the transient and steady-state responses. These specification can be conveniently made by a practising engineer using graphical means, such as the boundaries of the clear area shown in Fig. 10.

Although determination of the objectives and the priorities vector may be a subjective matter and depends on the performances requirement, ranking the priorities may be unnecessary and can be ignored for a 'minimum-commitment' design [16]. If, however, an engineer commits himself to prioritising the objectives, it is a much easier task than weighting the objectives. It is obvious that other design specifications such as gain margins, phase margins and noise rejection (can be quantified by distinctive *LQG* or H_2 norms) may also be added to the design if necessary.

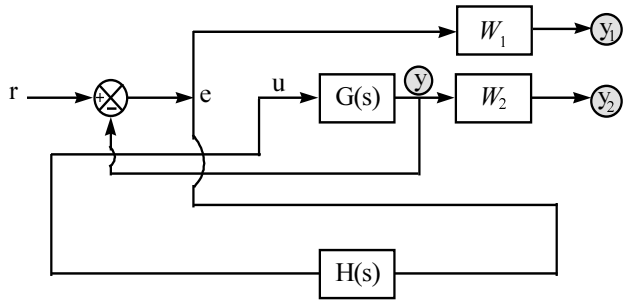

Fig. 8 Block diagram of an MO ULTIC design

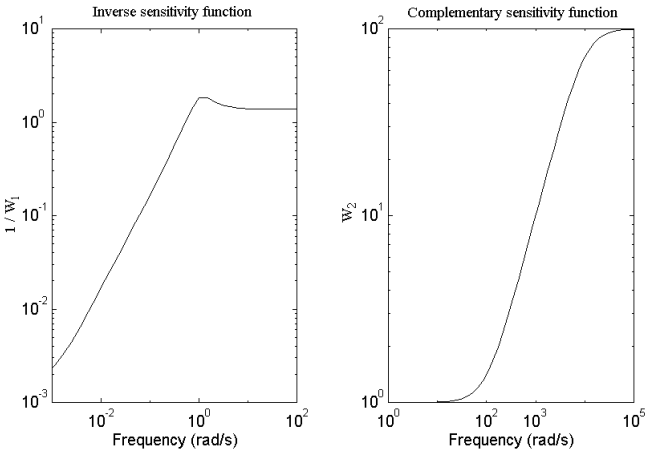

Fig. 9 Frequency responses of W_1 and W_2

A decimal-coding scheme [12] with two-point crossover were used in the MOGA applied to this example. The order of candidate controllers are not fixed, while allowing its maximum to be the third order. It took less than 1 hour for the MOGA to run for 100 generations with a population size of 100 on a Pentium 100MHz processor. All the nine design objectives in Table 1 have been met. The closed-loop responses of all the evolved non-dominated ULTIC controller that met the design specifications are shown in Fig. 10. An additional merit of using the MOGA for ULTIC design is that the priorities or goals can be changed at any time during the evolution process. After the evolution, the control engineer can preferably examine trade-offs between the nine design objectives if so desired, including other constraints or objectives or even zoom into the region of interested points before selecting one final controller for on-line implementations.

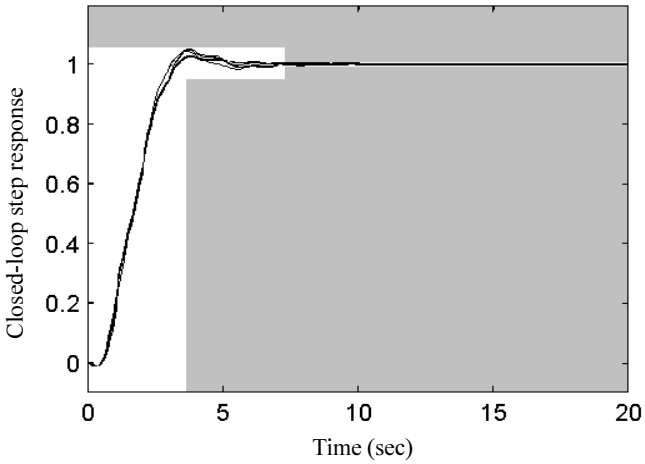

Fig. 10 The MOGA optimised output responses

V. CONCLUSIONS AND FURTHER WORK

This paper has developed an evolutionary MOGA technique to integrate linear control system design. The MOGA implements Pareto optimal domination with niche induction technique to evolve a set of non-dominated solutions along the Pareto optimal frontier. In addition, incorporation of the priority vector offers the MO optimisation to be guided on the basis of design specifications rather than the properties or the weights of the objective functions.

The methodology unifies LTI design approaches under performance satisfaction, eliminating the need of pre-selecting a specific control scheme. In principle, any number or combinations of design specifications or constraints can be added to the ULTIC system. The MOGA also allows engineers to interplay with the design objectives and zoom into the region of interest before the final commitment. In addition, the resulting ULTIC systems are easy to implement with only minor storage and computational overheads. This evolutionary MOGA approach has allowed the MO ULTIC design to be transparent and to satisfy the 'minimum-commitment' principle [16] at the CAD stage.

Currently, this evolution based design automation methodology has been extended to multi-input and multi-output control systems and to integrate other design objectives such as mixed-norm, controller structures and economical costing considerations. Progress and results will be reported in due course.

Table 1. Time domain and frequency domain design specifications

Customer specifications		Objective	Goal	Priority
Frequency domain	**1. Stability (Closed-loop poles)**	$Nr([eig(A_clp)] > 0)$	0	4
	2. Closed-loop sensitivity **3. Disturbance rejection**	$\overline{\sigma}[S(j\omega)]$	1	2
	4. Plant uncertainty	$\overline{\sigma}[T(j\omega)]$	1	2
Time domain	**5. Actuator saturation**	$\mathrm{Max}(u)$	0.5	3
	6. Rise time	T_{rise}	4	1
	7. Overshoots	O_{shoot}	0.05	1
	8. Settling time	T_{settling}	7	1
	9. Steady-state error	SS_{error}	0.01	1

ACKNOWLEDGEMENT

Mr. K.C. Tan is grateful to the University of Glasgow and CVCP for financial support in the form of a Postgraduate Scholarship and an Overseas Research Scheme Award.

REFERENCES

[1] Y. Li, K.C. Tan, K.C. Ng and D.J. Murray-Smith, (1995). Performance based linear control system design by genetic algorithm with simulated annealing, *Proc. 34th IEEE CDC,* New Orleans, 731-736.

[2] Y. Li, K.C. Tan and C. Marionneau, (1996). Direct design of linear control systems from plant I/O data using parallel evolutionary algorithms, *Int. Conf. on Control'96, Special Session on Evolutionary Algorithms for Control Engineering*, University of Exeter, UK, 680-686.

[3] G. Zames, (1966). On the input-output stability of time-varying non-linear feedback systems, Parts I and II, *IEEE Trans. Auto. Control*, AC-11, 2 & 3, 228-238 & 465-476.

[4] A. Grace, (1992). *Optimisation Toolbox User's Guide*, The MathWorks, Inc.

[5] A. Osyczka, (1984). *Multicriterion Optimisation in Engineering*, Ellis Horwood, Chichester.

[6] D.E. Goldberg, (1989). *Genetic Algorithms in Search, Optimisation and Machine Learning,* Addision-Wesley, Reading, Masachusetts.

[7] C.M. Fonseca and P.J. Fleming, (1993). Genetic algorithms for multiobjective optimisation: Formulation, Discussion and Generalisation, *Proc. of fifth Int. Conf. on Genetic Algorithms,* Morgan-Kauffman, 416-423.

[8] J. Horn, N. Nafpliotis and D.E. Goldberg, (1994). A niched pareto genetic algorithm for multiobjective optimisation, *Proc. First IEEE Conf. on Evolutionary Computation, IEEE World Cong. on Computational Intelligence,* Vol. 1, 82-87.

[9] A.J. Chipperfield and P.J. Fleming, (1995). Gas turbine engine controller design using mutliobjective genetic algorithms, *Proc. First IEE/IEEE Int. Conf. on GAs in Eng. Syst.: Innovations and Appl.,* Sheffield, UK, 214-219.

[10] N.V. Dakev, J.F. Whidborne and A.J. Chipperfield, (1995). H_∞ design of an EMS control system for a maglev vehicle using evolutionary algorithms, *Proc. First IEE/IEEE Int. Conf. on GAs in Eng. Syst.: Innovations and Appl.,* Sheffield, UK, 226-231.

[11] S.J. Goh, D.W. Gu and K.F. Man, (1996). Multi-layer genetic algorithms in robust control system design, *UKACC/IEE/InstMC/IMechE/IChemE Int. Conf. on Control'96, Special Session on Evolutionary Algorithms for Control Engineering*, University of Exeter, UK, 699-704.

[12] K.C. Ng, (1995). *Switching Control Systems and Their Design Automation via Genetic Algorithms,* Ph.D. Thesis, Dept. of Electronics and Electrical Engineering, University of Glasgow.

[13] Mathtools, (1996). *Matcom V2.01- A Matlab to C++ Compiler*, The MathTools.

[14] C.M. Fonseca and P.J. Fleming, (1995). Multiobjective genetic algorithms made easy: Selection, sharing and mating restriction, *Proc. First IEE/IEEE Int. Conf. on GAs in Eng. Syst.: Innovations and Appl.,* Sheffield, UK, 45-52.

[15] J.C. Doyle, B. Francis and A. Tannenbaum, (1992). *Feedback Control Theory,* Macmillan Publishing Company, New York.

[16] K. X. Guan and K. J. MacCallum, (1996), Adopting a minimum commitment principle for computer aided geometric design systems, *in Artificial Intelligence in Design '96 (Gero, J. S. and Sudweeks, F., eds)*, Kluwer Academic Publishers, 623-639.