Genetic algorithm automated approach to design of sliding mode control systems

Y. Li† , K. C. Ng†, D. J. Murray-Smith†, G. J. Gray† and K. C. Sharman‡

Although various nonlinear control theories, such as sliding mode control, have proved sound and successful, there is a serious lack of effective or tractable design methodologies due to difficulties encountered in the application of traditional analytical and numerical methods. This paper develops a reusable computing paradigm based on genetic algorithms to transform the "unsolvable problem" of optimal designs to a practically solvable "nondeterministic polynomial problem", which results in computer automated designs directly from nonlinear plants. The design methodology takes into account practical system constraints and extends the solution space, allowing new control terms to be included in the controller structure. In addition, the practical implementations using laboratory-scale systems demonstrate that such "off-the-computer" designs offer a superior performance to manual designs in terms of transient and steady-state responses and of robustness. Various contributions to the genetic algorithm technique involving the construction of fitness functions, coding, initial population formation and reproduction are also presented.

1. Introduction

Classical and modern control theory has been successful for systems that are well defined, both in terms of deterministic and stochastic descriptions. In many engineering applications, however, it is impossible or very difficult to obtain an accurate model of the plant to be controlled, due to the lack of detailed *a priori* information, complex dynamics, nonlinearity and time-varying characteristics of the plant. Modelling difficulties like these have forced control engineers to use simplified or linearised models, which are often inaccurate and vulnerable to parameter inaccuracy (or structured uncertainty) and unmodelled dynamics (or unstructured uncertainty), with degraded controlled system performance.

Currently, one widely adopted approach in control engineering to combating (or limiting the effect of) these modelling difficulties involves robust control, such as the nonlinear *sliding mode control* (SMC) which is also known as *variable structure system* (VSS) control. This control strategy makes use of singular arcs in the state-space, producing control settings based mainly on the observed plant input/output behaviour (as opposed to model predictions) and on considerations concerning some characterisation of the modelling uncertainties (Itkis 1976, Utkin 1993). Such an approach to robust control enables efficient control of high-order nonlinear plants and is easy to implement. Although existence and convergence conditions for designing such control systems have been studied extensively and successful applications to control of various industrial processes, power systems, electric drives, robots, aircraft, spacecraft and space structures, ships, and guidance systems have been widely reported (Hung et al 1993, Utkin 1993), there is a serious lack of effective or tractable design methods. Reported designs have been based on trial-and-error simulations, starting from a nominal model and theoretical sufficient conditions, with subsequent on-line manual adjustments.

The problems in nonlinear control system design arise from difficulties in applying calculus-based analytical methods to parameter optimisation under constraint conditions, when the design criteria or performance index may not be differentiable. Existing numerical optimisation methods used with computer-aided control system design (CACSD) are also inadequate, since they are based on derivative or gradient guidance and thus have difficulties in finding the global optimum in the multimodal design space. Further, the objective functions needed in these numerical methods must be "well-behaved" (Goldberg 1989, Holland 1992, Srinivas and Patnaik 1994) and would not, therefore, reflect practical system constraints (Ng and Li 1994, Li 1995a, Gray *et al* 1995). In addition, a modern paradigm of CACSD should also provide an environment that accepts the following challenges (Barker 1995, Li 1995a):

- Complexity of practical systems;

---

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

‡ Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow G12 8LT, U.K.

- Required high quality and accuracy of design;
- Speed of design;
- Competition with available design tools (in terms of ease of use, for example); and
- Robustness, reliability and safety arising from the design.

This paper develops a systematic and reusable evolutionary computing paradigm for solving generic design problems, with a focus on the design automation of sliding mode control systems by a genetic algorithm (GA). A sliding mode control scheme that is easy for practising engineers to use is developed in Section 2, also setting the scene for the discussions. To avoid the design optimisation difficulties mentioned above, it is proposed in Section 3 to transform the design problem into an analysis problem initially. A genetic algorithm is then developed to intelligently explore the analysed candidate controllers, leading to the globally optimised control system. It is illustrated that, by trading off precision slightly for improved tractability, robustness and ease of design (Herrera and Verdegay 1996), such computing paradigm meets the above challenges. Section 4 subsequently illustrates this methodology through two examples, one being a nonlinear system with relatively slow dynamics and the other a linear time-varying system with relatively fast dynamics. Conclusions and areas for future work are outlined in Section 5.

2. The Sliding mode control law and design difficulties

2.1. The control law

Suppose a nonlinear system to be controlled is given by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), t\right) + \boldsymbol{b}\,\boldsymbol{u}(t) \tag{2.1}$$

where $\boldsymbol{x} \in \Re^{n}$ is the state vector, $\boldsymbol{u} \in \Re^{m}$ the input vector, n the order of the system, m the number of inputs and $\boldsymbol{f}$ and $\boldsymbol{b}$ are properly dimensioned. Then the *sliding surface* hyperplane, $S(\boldsymbol{e})$, for n-dimensional tracking is defined by (Itkis 1976, Utkin and Yang 1978, Hung et al 1993, Utkin 1993)

$$S(\boldsymbol{e}) = \{\ \boldsymbol{e} \mid\ s(\boldsymbol{e}, t) = \boldsymbol{H}^{\mathrm{T}}\ \boldsymbol{e}(t) = 0\ \} \tag{2.2}$$

where $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{x}_d$ is the (negative) tracking error for a desired trajectory $\boldsymbol{x}_d(t)$ and $\boldsymbol{H} \in \Re^{n}$ represents the coefficients, or *slope*, of the sliding surface. It is shown that (Slotine and Li 1991), using SMC, this n-dimensional tracking problem is in effect a first-order stabilisation problem that keeps the error states on $S(\boldsymbol{e})$ for all $t > t_s$, where t_s is the time at which the sliding mode starts, and the dynamics while in sliding mode is given by $\dot{s}(\boldsymbol{e}, t) = 0$. Note that it is common to use $\boldsymbol{e}$ as the state vector $\boldsymbol{x}$ in SMC schemes.

Without loss of generality, consider a common second-order single-input and single-output (SISO) system with nonlinearity given by $f(\boldsymbol{x}, t)$. In SMC, a feedback linearisation term can be included in the control signal to cancel the nonlinear dynamics and thus to simplify the design. An estimate of this is given by a nominal model, $\hat{f}(\boldsymbol{x}, t)$, and is used together with a linear combination of error states to form an equivalent control signal $\hat{u}_{eq}(\boldsymbol{x}, \boldsymbol{x}_d, t)$ that keeps the states in the sliding mode (Slotine and Li 1991). To satisfy the robust stability condition derived from a Lyapunov function and given by

$$s(\boldsymbol{e})\ \dot{s}(\boldsymbol{e}) \le -\eta\left|s\right|, \quad \forall s \tag{2.3}$$

where $\eta > 0$, a switching term must be added in the control signal. This yields the sliding mode control strategy given by

$$u = -k\,\mathrm{sgn}(s) + \hat{u}_{eq} \tag{2.4}$$

where

$$\mathrm{sgn}(s)=\begin{cases}+1, & s>0\\ 0, & s=0\\ -1, & s<0\end{cases}$$

Here conservatively choosing $k \geq \eta + \left\| f - \hat{f} \right\|_{\infty}$ will satisfy the robustness condition given by (2.3) (Slotine and Li 1991). In the presence of model imprecision, this switching control law attempts to drive the system states from an arbitrary initial point onto (or close to) the sliding surface and to keep them in the sliding mode afterwards.

To simplify the equivalent control and ease the design task for practising engineers, Nandam and Sen (1992) have proposed an equivalent control action based on the proportional plus derivative control law. A potential advantage of this strategy, which the authors did not address, is its ability of using the undistorted nonlinear model of the physical system in a simulation based design process. This paper further extends this control strategy by incorporating an integration term to form a generic controller structure given by

$$u = -\phi - \varphi_P e - \varphi_I \int e\,dt - \varphi_D\, de/dt \tag{2.5}$$

with generalised "hard-switching" parameters being

$$\phi=\begin{cases}\alpha_1, & s<0\\ \alpha_2, & s>0\end{cases},\quad \varphi_P=\begin{cases}\beta_1, & es<0\\ \beta_2, & es>0\end{cases},\quad \varphi_I=\begin{cases}\gamma_1, & s<0\\ \gamma_2, & s>0\end{cases},\quad \varphi_D=\begin{cases}\delta_1, & \dot{e}s<0\\ \delta_2, & \dot{e}s>0\end{cases} \tag{2.6}$$

Note again that e is the *negative* tracking error. The first term in (2.5) represents an asymmetric switching "voltage" for asymmetric systems, the second a *proportional* (P) term of the error state, the third an *integral* (I) term, and the last a *derivative* (D) term. The use of the PID terms for equivalent control in (2.4) gives practising engineers a more tractable implementation. The design of this simplified SMC system involves the choice of the eight switching parameters in (2.6) and the switching logic associated with the sliding surface of (2.2) determined by $\boldsymbol{H}^{\mathrm{T}} = [h\ 1]$. The design task is thus to find a parameter vector

$$\boldsymbol{P}_i = \left[h\ \alpha_1\ \alpha_2\ \beta_1\ \beta_2\ \gamma_1\ \gamma_2\ \delta_1\ \delta_2\right]^{\mathrm{T}} \tag{2.7a}$$

in the solution space

$$\boldsymbol{P} = \left\{\boldsymbol{P}_i \;\middle|\; \left\|\boldsymbol{P}_i\right\|_{L_2} < \infty,\ \forall i\right\} \subseteq \Re^{9} \tag{2.7b}$$

such that the parameter set yields a sliding mode controller that best meets the design criterion.

2.2. Difficulties in existing design approaches

Although the method proposed by Nandam and Sen (1992) is reported useful and the minimum switching gain of k in (2.4) may be derived from the robustness condition (2.3), there is no direct design methods reported for selecting the controller parameters such that they best meet performance requirements such as a short rise-time, low oscillations and low steady-state errors. A useful design approach based upon a linearised ideal nominal model, however, exists and has been commonly studied (Itkis 1976, Utkin and Yang 1978, Dorling 1985, Hung 1983, Utkin 1993). This approach starts with design considerations in solving the existence and reachability problems using eigenvalue assignment for equivalent control, and then a calculus based optimisation method is applied to differentiate the performance index in order to solve for optimised SMC parameters analytically. A cost function for this *quadratic minimisation* has been proposed by Utkin and Yang (1978) and used by Dorling (1985) in the form of

$$J = \frac{1}{2}\boldsymbol{x}^T(\infty)\boldsymbol{F}\,\boldsymbol{x}(\infty) + \frac{1}{2}\int_{t_s}^{\infty}\left\{\boldsymbol{e}^T(t)\boldsymbol{Q}(t)\boldsymbol{e}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}(t)\boldsymbol{u}(t)\right\}dt \tag{2.8}$$

where $\boldsymbol{F} \in \Re^{n\times n}$ and $\boldsymbol{Q}(t) \in \Re^{n\times n}$ should usually be positive semi-definite and $\boldsymbol{R}(t) \in \Re^{m\times m}$ positive definite. This cost function is similar to that used in a *linear quadratic regulator* or an *optimal control* problem. The requirement of a linearised model will, however, result in loss of useful information of the physical system. The scope of this approach is thus limited in a similar way to linear robust methods, such as the H_∞ and μ-synthesis based methods.

In addition, this approach is too difficult or impossible to apply to practical problems, since the plant dynamics could be too complex or too nonlinear for the analytical approach. The use of "*gradient-guidance*" based numerical optimisations, such as the Newton and the Fibonacci methods, would not usually result in the *global* optimum in the multimodal design space and would require the performance index to be "well-behaved" in practice (Goldberg 1989, Holland 1992, Srinivas and Patnaik 1994). Further, since the SMC controller parameters and the control signals that an actuator can provide are usually bounded in practice, it becomes more difficult to apply such traditional optimisation techniques under these practical constraints. This type of optimal design problem thus forms an *unsolvable problem* known in computer science.

Due to the lack of effective optimisation tools, existing CACSD packages have largely been developed to carry out passive simulation tasks, giving few direct or automated design facilities. Using such a package for design, a design engineer first needs to input some *a priori* controller parameters, such as those obtained from some preliminary analysis, and should then undertake simulations and evaluations. If the simulated performance of the "designed" control system does not meet the specification, the designer would then modify the values of the parameters manually and run the simulations repeatedly until a "*satisfactory*" design emerges. Clearly, such a design process is neither automated nor easily carried out, since mutual interactions among parameters are hard to predict. In addition, the resulting "satisfactory" system may still not necessarily offer the best or near-best performance. These deficiencies have contributed to the failure of SMC and many other nonlinear control schemes to be widely accepted by practising engineers.

An alternative approach does, however, exist. A bounded parameter set in (2.7) could be represented or *reasonably* "*encoded*" in some way, for example, by a binary string of length proportional to the logarithm of the total range. *Exhaustive search* (or *enumerations*) for all possible parameter values could then allow evaluation of every candidate controller by simulating it directly with the nonlinear system model. Thus the simulated results obtained from an existing CACSD package could reveal the best candidate. This, in effect, transforms the unsolvable problem into a *solvable problem*. If it worked in practice, it would solve almost all design problems. Suppose, however, that in the context of SMC each simulation of a candidate design on a 120 MHz Pentium processor takes 0.1 second, and that each of the 9 parameters has only 8 candidate values to be studied within their individual ranges. On the basis of these figures, this method would take 0.1×8^9 seconds = 5 months to evaluate all possible designs, which is not practically acceptable. This problem becomes even more severe if the plant is multivariable in form, since the total simulation time increases exponentially with the dimension or the number of parameters to be selected, i.e., this is a *nonpolynomial problem*. In the next section, a genetic algorithm is developed to transform this problem into a *nondeterministic polynomial (NP) problem* (or, precisely, an *NP-complete* problem), which reduces the exponential search time to polynomial time and thus makes design automation possible in practice.

3. Automated design using genetic algorithms

3.1 The genetic algorithm

Based on the *survival-of-the-fittest* Darwinian principle in the natural process for biological reproduction and mutation, the *genetic algorithm* (Goldberg 1989, Holland 1992, Srinivas and Patnaik 1994) has been in development for three decades (Li 1995f). This approach has proved to be particularly effective in searching through poorly understood and irregular spaces. A GA generally uses coded strings (*chromosomes*) of binary numbers (*genes*) in the search process. Such an algorithm is based on an analogy with the *genetic code* in our own DNA structure, where the coded chromosome is composed of many genes having 64 values. This analogy inspired the use of Base-7 coding (Ng and Li 1994) and other integer coding mechanisms, such as decimal coding (Li and Ng 1994) presented in this paper. Compared with natural evolution, this emulated process is more efficient, controllable and yet more flexible for artificial optimisation.

The *Schema Theory* (Goldberg 1989, Holland 1992, Srinivas and Patnaik 1994) implies that a genetic algorithm is a (*nondeterministic*) *polynomial algorithm* and thus requires an exponentially reduced search time, compared with the *exponential algorithm* of exhaustive search. This also means such a GA can be used to transform the nonpolynomial (exponential) problem discussed in Section 2.2 into an *NP-complete problem*. The experimental studies on an artificial problem by Keane (1995) have also shown that a GA, particularly when it is fine-tuned by *simulated annealing* (SA) (Tan *et al* 1995, Li *et al* 1995e), provides a much higher convergence robustness and rate than conventional optimisation means, including the linear approximation and the heuristic search algorithms. This technique and the related *genetic programming* (GP), *evolutionary programming* and *evolution strategies* are referred to as "*evolutionary computing*" techniques. The GA based search technique has successfully been applied to the general area of modern computing and, in particular, control system analysis and design, including parameter estimation (Sharman and McClurkin 1989), system identification (Kristinsson and Dumont 1992, Sharman and Esparcia-Alcázar 1993, Li and Tan 1995d), linearisation (Tan *et al* 1995), controller order reduction (Caponetto *et al* 1994), auto-tuning PID (Wang and Kwok 1992), multiobjective optimisation (Chipperfield *et al* 1992), optimal control (Kwok *et al* 1991, Hunt 1992, Kwok and Sheng 1994), uniform linear controller design (Li *et al* 1995e) and its parallel realisation (Li 1995a), robust control (Murdock *et al* 1991, Patton and Liu 1994), fuzzy logic control system design (Karr 1991, Linkens and Nyongesa 1993, Ng and Li 1994) and its rule base reduction (Li and Ng 1995c), and neural network modelling (Li *et al* 1995b) and control (Harp and Samad 1992, Häußler *et al* 1995).

A schematic of the genetic evolution for a coded design problem is shown in Fig. 1. In the process of evolution, a *population* of chromosomes are updated according to the relative individual *fitness* that reflects the evaluated performance index. The GA uses four operators, namely, *selection*, *crossover*, *mutation* and *inversion*. The crossover operation exchanges information between the parent pair (i.e., between two search points) and mutation changes the value of a gene (i.e., self-adjustment) (see Goldberg 1989 for detail). The fourth operator can be derived from crossover and mutation, and is thus not commonly used in a GA. More detail on using the GA will be given in the following sections.

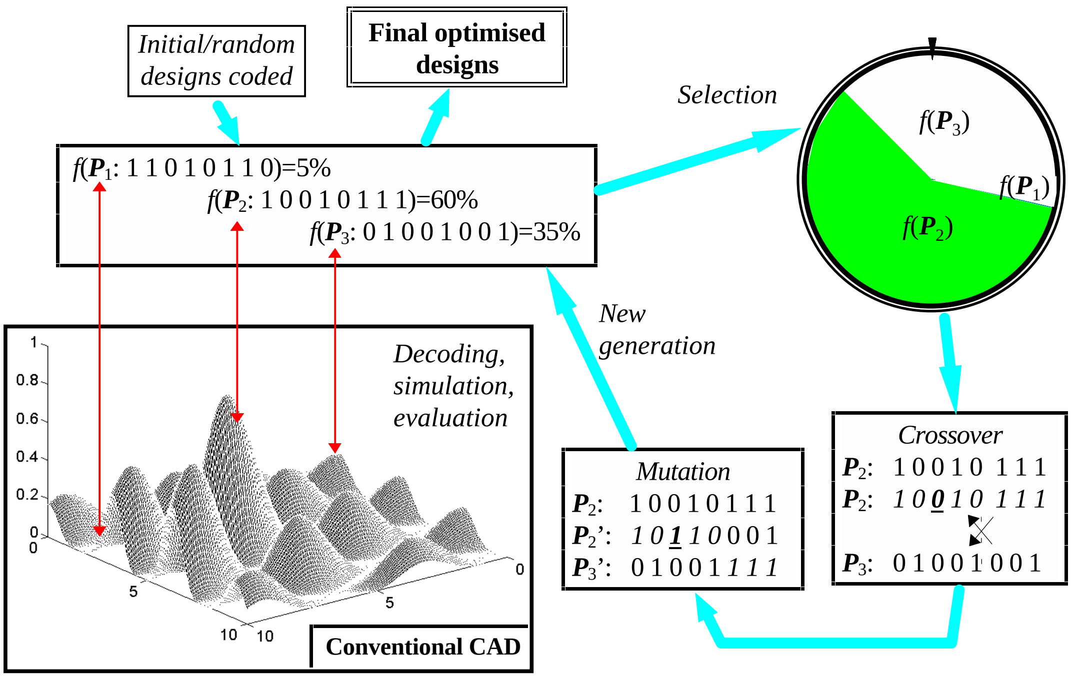

Figure 1. Evolution of coded designs by fitness evaluations and genetic operations.

3.2 Design automation

Owing to the exponentially reduced search time compared with exhaustive search, a GA makes an automated design of control systems possible. Before a GA is employed for the design purposes, it is necessary to carry out coding, choose an initial population and prepare a fitness function.

Coding

In the preparation phase, all the parameters in (2.7) need to be encoded by an integer string to form a chromosome. The decimal value of each parameter is mapped as

$$C = C_{\min} + \frac{a_{p-1}\, b^{p-1} + ... + a_0\, b^0}{b^p - 1}\left(C_{\max} - C_{\min}\right) \tag{3.1}$$

where $C \in [C_{\min}, C_{\max}]$ is the real value being coded, b=10 the base value of decimal coding presented in this paper, $a_i \in \{0, ..., \text{b-1}\}$ an unsigned integer code and p the number of digits that dictates the compromise between accuracy and speed in the search. The decimal and Base-7 coding will reduce the "Hamming Cliff" effect (Ng and Li 1994) associated with the commonly used binary coding shown in Fig. 1, and will reduce the convergence time. It should be noted that coding in a GA is not limited to integer strings and, logarithmic real coding, for example, can also be used.

In this paper, each controller parameter is coded by 2 digits. Different values of $C_{\max}$ and $C_{\min}$ can be used for different parameters, if their ranges are known to be different. If the expected parameter range is 10 times larger, or a 10 times finer resolution is preferred, 3 digits can be used. In the paper, a third digit is added to the 2-digit parameter to indicate 5 (or less than 11) discrete choices of the range for the one hundred quantities coded. These ranges are [0, 0.99], [0, 9.9], [0, 49.5], [-50, 49] or [50, 99.5], coded by the third gene. Alternatively, the ranges of a parameter can be adjusted adaptively to fit the ±30% range of the fittest parameter found after a large number of generations, since at this stage all parameters usually converge to a narrow range. Then the search can be conducted by a fine-tuning mechanism hybrid with, for example, the traditional "hill-climbing" or SA techniques (Keane 1995, Li *et al* 1995e, Tan *et al* 1995, Li 1995f).

Initial population

In the preparation, the second step is to form an initial population of candidate designs as shown in Fig. 1, where there are three chromosomes, P_1, P_2 and P_3. A typical population size for the SMC design is chosen as 50 in view of the size and complexity of the design problem under study. The existing designs (e.g., P_3) and the designs that a control engineer would like to start with can first be coded to form some chromosome strings in the initial population, and the rest of the initial population can be filled by randomly generated chromosomes (e.g., P_1 and P_2). This ensures a direct starting point and is in contrast to the situation in artificial neural network based learning techniques, where the starting point involves no direct mapping of the SMC parameters. Starting with existing knowledge in this way will usually result in a more rapid convergence, although it is possible to start from completely random chromosomes. Evolving an entire population of candidate designs reduces the probability of landing at a local optimum.

Fitness function

The third step in the preparation is to establish for each candidate controller in the population a global measure of fitness, which is similar to the inverse of a cost-function in optimisation. A simple fitness function that reflects small steady-state errors, a short rise-time, low oscillations, low overshoots and a good relative stability is given by

$$f\left(\boldsymbol{P}_i\right) = \begin{cases} 0 & \text{if } \left|e_j\right| > 10\left|r_j\right| \\ \dfrac{N}{\sum_{j=1}^{N}\left\{\left|e_j\right| + w\left|\Delta e_j\right|\right\} j} & \text{otherwise} \end{cases} \tag{3.2}$$

where N is the duration of the simulation for evaluating the design, j the time index in simulation, e_j the error at simulation step j and Δe_j the change in error, w (being fixed at 1 in this paper) a bias weighting between e_j and Δe_j, and r_j the reference input. Here the time-weighted L_1 norm of errors is used as inverse fitness (Häußler *et al* 1995). The Δe_i term can be distinctively weighted to further suppress oscillations. An L_2 or its square (Sharman *et al* 1995) that weights the error, change of error and control energy in a similar way to that given by (2.8) or with an exponential amplification (Ng and Li 1994, Ng *et al* 1995, Tan *et al* 1995) can also be used. It is worth noting that use of either norm will be adequate in terms of fitness formation, as both norms are related by Kakutani's *metric equivalence theorem* (Rolewicz 1987). In the time-domain design, however, the L_∞ norm alone is often unsuitable to this application, since it only emphasises one value of error.

In evaluating the performance of a practical controller, an *implicit constraint* on the parameters resulting from the voltage limit (e.g., between ±5V) of the control signal can be applied conveniently in the simulations. This reduces wear of actuators and excessively large signals. The first part of (3.2) reflects a severe penalty on the fitness if the resulting error is too large. Other terms that reflect *explicit constraints* or specifications can also be included (Gray *et al* 1995). For example, if robustness against delays or hysteresis needs to be promoted, they can be included in the fitness directly or by a penalty. Search under these constraints by a GA is much easier than minimising a cost function by a conventional optimisation technique. When needed, the criteria of reliability and safety of the design can also be included explicitly in the fitness. Note also that this form of performance evaluation eliminates the need for inclusion of asymptotic stability, existence, reachability or convergence conditions, since controllers that do not satisfy these conditions would automatically result in very poor fitness and would not survive in the evolutionary design process.

The simple *roulette-wheel selection* scheme is employed in the reproduction process for demonstration of the GA methodology in this paper. The relative fitness represents a value that is the probability of the individual in question being selected to replicate itself for generating offspring. Suppose that Candidate *i* has a fitness of f_i. Then the *reproduction rate* is calculated by $f_i / \sum_j f_j$ as shown in Fig. 1. There are other selection mechanisms, such as the *proportionate, rank-based* (Ng *et al* 1995), *Boltzmann* (Li 1995f), *tournament*, *elitist strategies* and *steady-state* selection schemes. Robust roulette-wheel pre-mapped from arithmetic or geometric sequences can also be used to provide an invariant reproduction ratio and avoids linear and nonlinear scaling of fitness (Li 1995f).

Applicability

From the above discussions, it can be inferred that a control system can always be designed by a GA under the following conditions

- The solution space for design is finite (which is unnecessary if GP is used instead) and can be represented by finite quantisation;
- The system is simulatable, i.e., the performance of candidate designs can be evaluated; and
- There exists a performance index that has values with more information than a simple *True-or-False* answer, as otherwise the reproduction will be completely polarised and the GA will become a multi-point random search.

4. Illustrated design examples

4.1 A nonlinear and relatively slow system

A laboratory-scale, second order, nonlinear, asymmetric, liquid level control system shown in Fig. 2 is used to demonstrate the design using the genetic algorithm. In this simplified demonstration, only is the input to Tank 1 used as the input flow, *u*, in cm^3/sec, which maps to an actuator voltage in the implementations that follow. It is used to control the liquid level of Tank 2, $h_2 = x + r = e + r$ (cm). Here r is the desired level at Tank 2 and $[x\ \dot{x}]^T$ the state vector in the form of the (negative) tracking error and the (negative) change of error, respectively. Another input, d (cm^3/sec), to Tank 2 is used as a disturbance for the robustness test. A simplified nonlinear equation of this system is given by

$$\begin{cases} A\dot{h}_1 = u - a_1 c_1 \sqrt{2g(h_1 - x - r)} \\ A\dot{x} = a_1 c_1 \sqrt{2g(h_1 - x - r)} - a_2 c_2 \sqrt{2g(x + r - h_0)} + d \end{cases} \tag{4.1}$$

where A = 100 cm^2, is the cross-section area of both tanks; a_1 = 0.386 cm^2 and a_2 = 0.976 cm^2, the orifice areas of Tank 1 and Tank 2, respectively; $c_1 = c_2 = 0.58$, the discharge constants; h_1 (cm), the liquid level of Tank 1, being an intermediate variable; h_0 = 3cm, the height of the orifices; and g = gravitational constant = 981 cm/sec^2.

The objective of this laboratory control system is to input to Tank 1 and drive the liquid-level at Tank 2 towards the desired level of 10 cm as fast as possible with minimum overshoots and steady-state error. If a linearised model is acceptable, the eigenstructure assignment approach reported by Patton and Liu (1994) may be used. Clearly, using the GA automated method to design a sliding mode controller of (2.5), it is unnecessary to rewrite the nonlinear model of the physical system given by (4.1) into the nominal form of (2.1), provided the dimension of the sliding hyperplane to be designed equals the order of the system. This avoids another difficulty in applying the SMC theory.

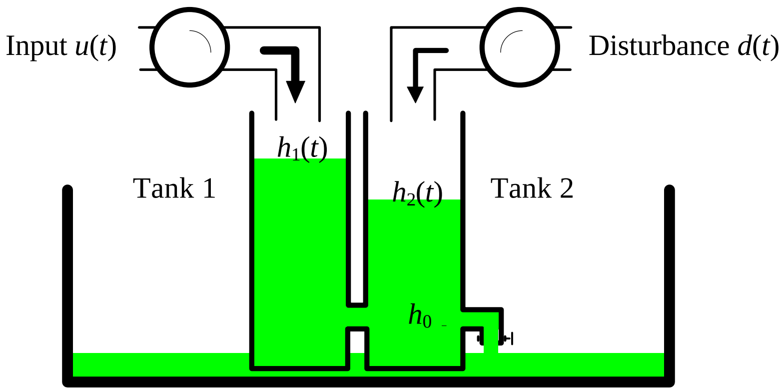

Figure 2. Laboratory-scale liquid tank test system.

In this example, no *a priori* knowledge on the design is assumed and all the initial 50 chromosomes are generated randomly. The *crossover rate* is fixed at 60% and the *mutation rate* at 0.5%, which are typical values used in a GA. *Two-point crossover* is adopted here, due to the long chromosome length. Here the *adjacent mutation scheme* (Ng 1995) is adopted for smaller disruptions in the evolution. The *Adaptive mutation* scheme (Ng 1995) is also used, the rate of which is calculated in proportion to the similarity of genes of two chromosomes in terms of the number of identical digits. This prevents similarities of the parent pair and also needs no prior determination of the mutation rate.

In this example, 100 generations have been evolved. In average, the Pascal program took a total of 1.5 hours to complete, using an Intel 80486 DX2 processor running at a clock rate of 50 MHz. In contrast, exhaustive search on 50×100 candidate designs has also been tested, which took about 1.4 hours. This implies that it would take 32 billion years if all the 100^9 candidates are exhaustively evaluated. At the end of this GA process, the highest fitness and the average fitness of every generation are captured and a typical example of these is shown in Fig. 3. It can be seen that the highest fitness of each generation converges quickly and stays at the "best" value after the 72nd generation. It is however appropriate to assess the global convergence by studying the average fitness instead. This also converges very well after 50 generations and indicates that the parameters then merge towards a narrow band. The optimal parameters found by the GA at the end of the evolutionary process are

$$h = 0.1,\quad \phi = \begin{cases} 0.192, & s < 0 \\ 0.161, & s > 0 \end{cases},\quad \varphi_P = \begin{cases} 9.51, & xs < 0 \\ 10.0, & xs > 0 \end{cases},\quad \varphi_I = \begin{cases} 0.111, & s < 0 \\ 0.201, & s > 0 \end{cases},\quad \varphi_D = \begin{cases} 0.101, & \dot{x}s < 0 \\ 0.918, & \dot{x}s > 0 \end{cases} \tag{4.2}$$

For comparison, a sliding mode controller has been designed for the model of (4.1) by repeated manual selections, using trial-and-error techniques starting from a PID controller. In this design, the experience reported in Nandam and Sen (1992) and Lua (1993) has been taken into consideration to obtain the best transient and steady-state performance that could be found by manual tuning. This is, however, a very difficult process, as multiple parameters must be tuned simultaneously and each tuning must start from repeating the entire input-output process. The final parameters obtained this way are

$$h = 0.5, \quad \phi = \begin{cases} 1, & s < 0 \\ 0.025, & s > 0 \end{cases}, \quad \varphi_P = \begin{cases} 8.5, & xs < 0 \\ 10.2, & xs > 0 \end{cases}, \quad \varphi_I = \begin{cases} 0.05, & s < 0 \\ 0.2, & s > 0 \end{cases}, \quad \varphi_D = \begin{cases} 0.4, & \dot{x}s < 0 \\ 0.5, & \dot{x}s > 0 \end{cases} \tag{4.3}$$

Simulated results for these two controllers using the model given by (4.1) are shown in Fig. 4. The command was set at a level of 10.0 cm for the Tank 2 for 500 seconds and was then switched to 5.0 cm for another 500 seconds. Note here that a step-down command is used to test the performance of the asymmetric system. An unmodelled disturbance input of 500cm^3/min is directly injected into Tank 2 250 seconds after the change of the input level for testing the sensitivity of the system. It is clear from the simulation results that the performance of the GA designed controller is superior to that of the manually designed system, in terms of transient response, steady-state error and robustness in rejecting the disturbance.

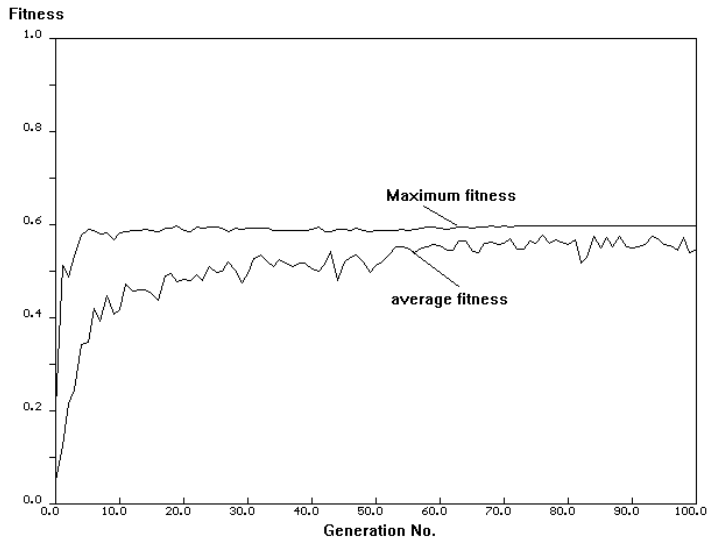

Figure 3. The highest and the average fitness functions in every generation.

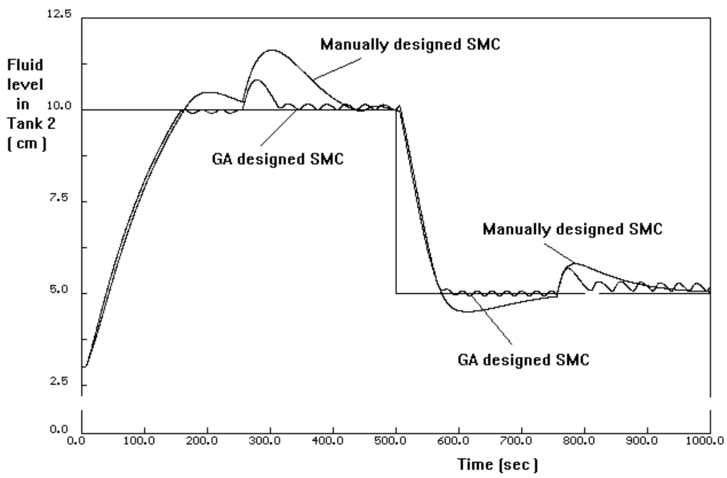

Figure 4. Simulated responses of the SMC systems optimised by the GA and designed manually.

Following the simulation studies, both controllers were implemented using the laboratory-scale coupled liquid-tank system described by Fig. 2. The implemented performance of the GA designed controller is captured and shown in Fig. 5. Again, a disturbance inflow is injected by the two-input digital control system at 250 second and at 750 second. It can be seen that the performance of the GA designed system is superior to that of the manually designed system.

In order to further verify the performance of the GA designed control system in terms of robustness to model uncertainties, additional simulations have been carried out. Firstly, this controller and the manually designed controller were applied to the model given by (4.1) with a time delay of 3 seconds added. The simulated results are shown in Fig. 6. Secondly, a white noise of 20% of the reference amplitude was added to the feedback signal and the captured responses are shown in Fig. 7. As can be seen, the GA designed system performs considerably better. Note that these implemented and simulated results are all based on the controller designed for the non-nominal model given by (4.1) using the fitness function given by (3.2), without considerations of parameter variations, disturbances or noise in the design simulations. It can be seen that, however, the SMC scheme is inherently robust.

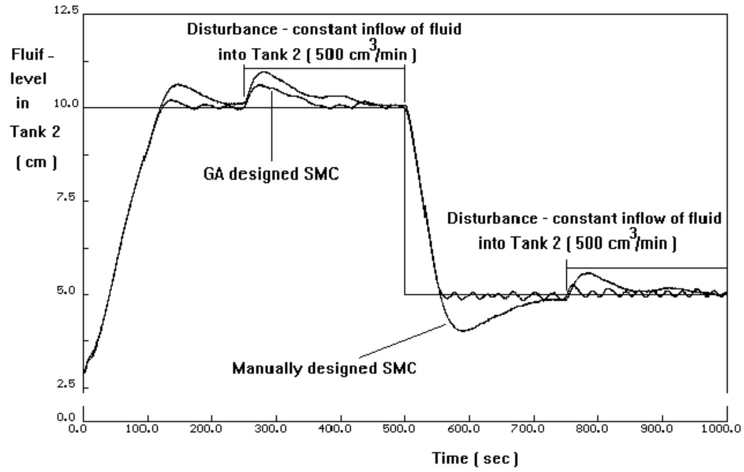

Figure 5. The implemented performance of the GA optimised and manually designed controllers.

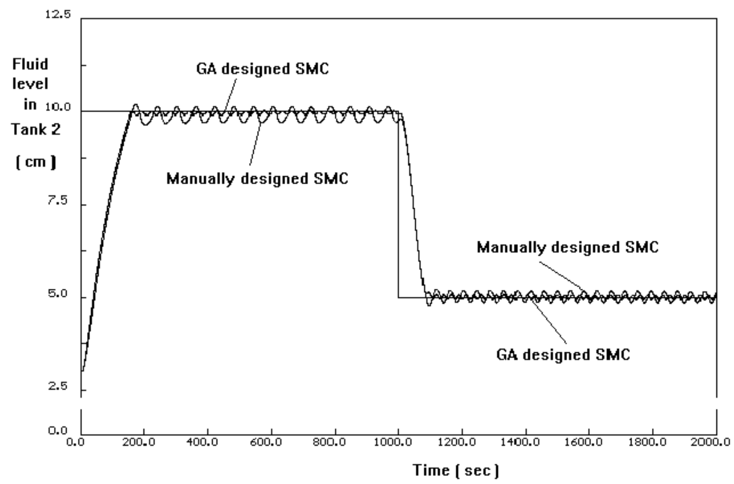

Figure 6. Simulated responses with artificially added time delay.

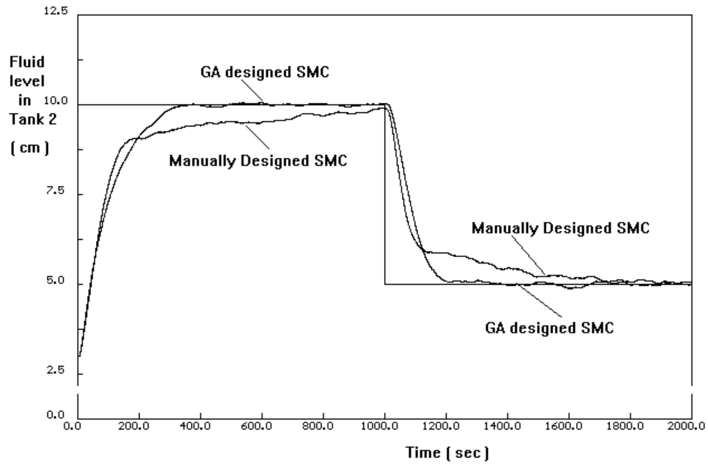

Figure 7. Responses to a step input with noisy feedback.

4.2 Equivalent surface control

The GA based design method for (2.6) can also be used in piece-wise SMC schemes. A simple switching equivalent control replacing (2.6) is given by

$$\phi = \begin{cases} \alpha_1, & s<0 \\ 0, & s=0 \\ \alpha_2, & s>0 \end{cases}, \quad \varphi_P = \begin{cases} \beta_1, & xs<0 \\ K_P, & xs=0 \\ \beta_2, & xs>0 \end{cases}, \quad \varphi_I = \begin{cases} \gamma_1, & s<0 \\ K_I, & s=0 \\ \gamma_2, & s>0 \end{cases}, \quad \varphi_D = \begin{cases} \delta_1, & \dot{x}s<0 \\ K_D, & \dot{x}s=0 \\ \delta_2, & \dot{x}s>0 \end{cases} \tag{4.4}$$

Here three more parameters need to be found by the GA. With the addition of these parameters, it took nearly the same amount of time as that in the previous section to evolve 100 generations. The corresponding parameters obtained by this NP algorithm are

$$h = 0.117, \quad \phi = \begin{cases} 0.021, & s<0 \\ 0, & s=0 \\ 0.021, & s>0 \end{cases}, \quad \varphi_P = \begin{cases} 2.2, & xs<0 \\ 19.1, & xs=0 \\ 83.33, & xs>0 \end{cases}, \quad \varphi_I = \begin{cases} 0.15, & s<0 \\ 0.133, & s=0 \\ 0.11, & s>0 \end{cases}, \quad \varphi_D = \begin{cases} 48.0, & \dot{x}s<0 \\ 36.0, & \dot{x}s=0 \\ 0.0, & \dot{x}s>0 \end{cases} \tag{4.5}$$

The simulated and implemented performances of the GA optimised controller is shown in Fig. 8 and Fig. 9, respectively. It can be seen that they are both superior to those obtained by the controller of (2.6) with parameters given by (4.2), as shown in Fig. 4 and Fig. 5. More importantly, this example demonstrates the ability of applying the GA computing paradigm to the design of other nonlinear control systems, which can be uniformly described by the parameter space given by (2.7).

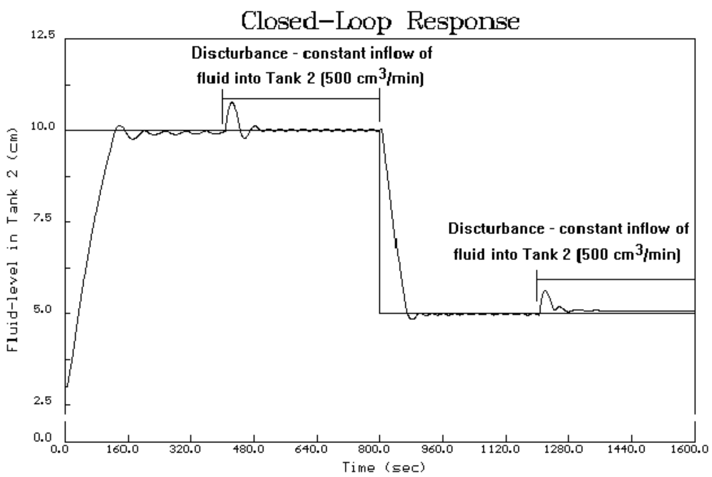

Figure 8. Simulated response of the GA optimised SMC with PID equivalent control.

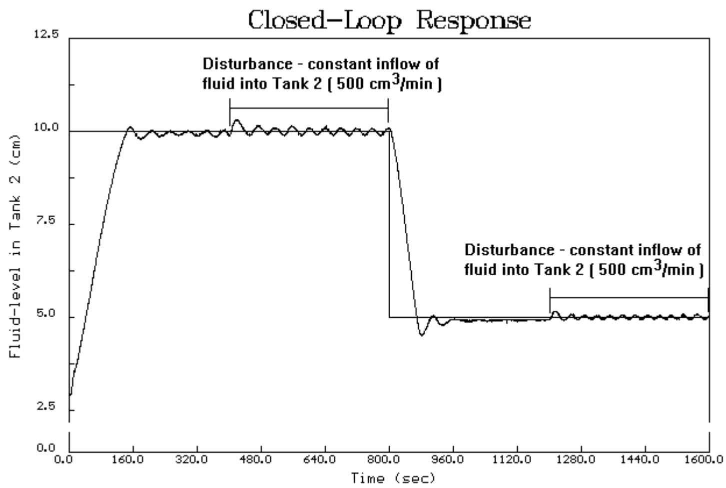

Figure 9. Implemented response of the GA optimised SMC with PID equivalent control.

4.3 A linear time-varying and relatively fast system

In this example, the objective is the real-time speed control of a laboratory-scale DC servomechanism whose dynamics are significantly faster than the coupled liquid-tank system. The typical differential equation defining the open-loop servomotor with field control is given by

$$\frac{d^2\omega}{dt} + \left(\frac{JR + LB}{LJ}\right)\frac{d\omega}{dt} + \left(\frac{RB}{LJ}\right)\omega = \left(\frac{K_T}{LJ}\right)v_{in} \tag{4.6}$$

where $v_{in} \in$ [-5V, 5V] is the input control voltage with implicit constraint; K_T = 13.5 Nm/A the torque constant; R = 9.2 Ω the resistance of the motor winding; L = 0.25 H the inductance; B = 2.342x10^{-3} Nms the friction coefficient of the shaft with an eddy current brake applied to the inertia disk; and J = 1.0x10^{-3} kgm^2 the moment of inertia of the load and the machine. Here the friction coefficient B is subject to a significant change when the brake is removed. The transient dynamics are dominated by the mechanical time constant, J/B, which is still significantly smaller than that of the liquid-tank system.

To track a step command, let the error states be e = ω - r and $\dot{x} = \dot{\omega}$, where r is the reference signal. The structure of the controller is also governed by equation (2.6). The same design procedure as that of Section 4.1 is carried out, with a total design time of 40 minutes (for 100 generations). Fig. 10 shows the convergence curves of the design that started from completely random candidates.

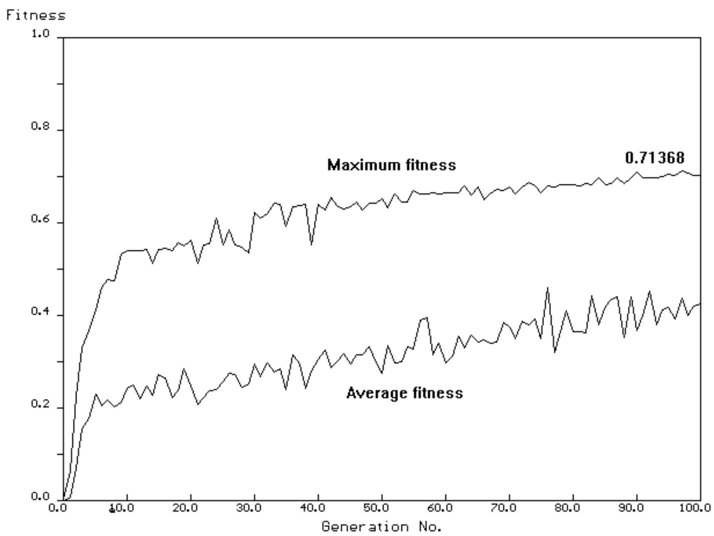

Figure 10. Highest and average fitness values in each generation without prior knowledge incorporated.

Before another GA automated design process was carried out, however, an existing “satisfactory” manual design (Lua 1993) was investigated and its parameters were further fine-tuned heuristically. The best set of parameters finally achieved by careful adjustment is given by

$$h = 0.5,\quad \phi = \begin{cases} 0.00, & s < 0 \\ 0.03, & s > 0 \end{cases},\quad \varphi_P = \begin{cases} 2.6, & xs < 0 \\ 3.2, & xs > 0 \end{cases},\quad \varphi_I = \begin{cases} 2.1, & s < 0 \\ 2.6, & s > 0 \end{cases},\quad \varphi_D = \begin{cases} 0.07, & \dot{x}s < 0 \\ 0.009, & \dot{x}s > 0 \end{cases} \tag{4.7}$$

This prior knowledge in designs was therefore incorporated in the initial population, with the remaining 49 chromosomes generated randomly. The GA was then run again, which yielded the highest and average fitness as shown in Fig. 11. Compared with Fig. 10, it can be seen that the convergence rates improve substantially. It should also be noted that, although 100 generations are used in the program, a quarter of the evolution would in fact be adequate, i.e., the design time could be reduced to 10 minutes. The optimised parameters by the genetic algorithm are

$$h = 0.476,\quad \phi = \begin{cases} 0.0400, & s < 0 \\ 0.0496, & s > 0 \end{cases},\quad \varphi_P = \begin{cases} 3.6, & xs < 0 \\ 3.2, & xs > 0 \end{cases},\quad \varphi_I = \begin{cases} 1.2, & s < 0 \\ 1.2, & s > 0 \end{cases},\quad \varphi_D = \begin{cases} 0.0000, & \dot{x}s < 0 \\ 0.0929, & \dot{x}s > 0 \end{cases} \tag{4.8}$$

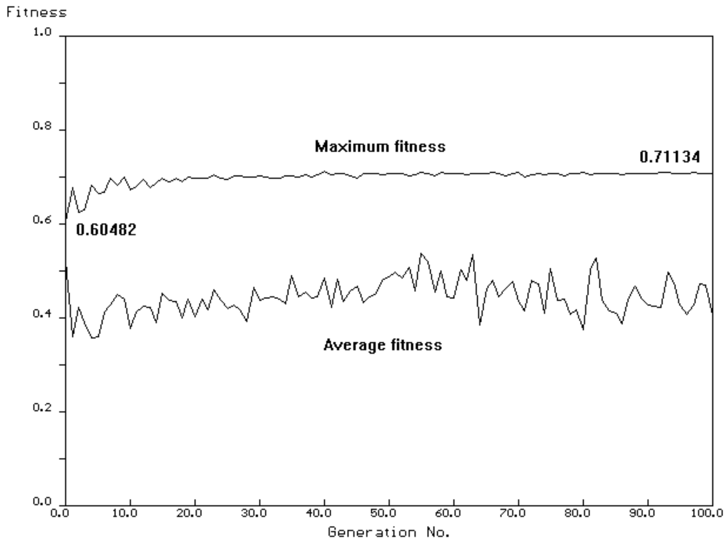

Figure 11. Highest and average fitness values in each generation with prior knowledge incorporated.

The real-time implemented response of this GA optimised SMC system to a step command at 18 r.p.s. for 5 seconds and then at 9 r.p.s. for another 5 seconds was captured and is shown in Fig. 12. The overshoot and undershoot during the steady-state period occurred at the moment when the brake torque was manually removed and applied, respectively, to vary the friction parameter and thus test for robustness. To allow comparisons, the implemented performance of the manually designed SMC under the same operating conditions is also shown in this figure. It is clear from the captured response that, in addition to the tremendously reduced design time, the performance of the GA designed controller is superior to that of the manually designed system, in terms of transient response, steady-state error and robustness.

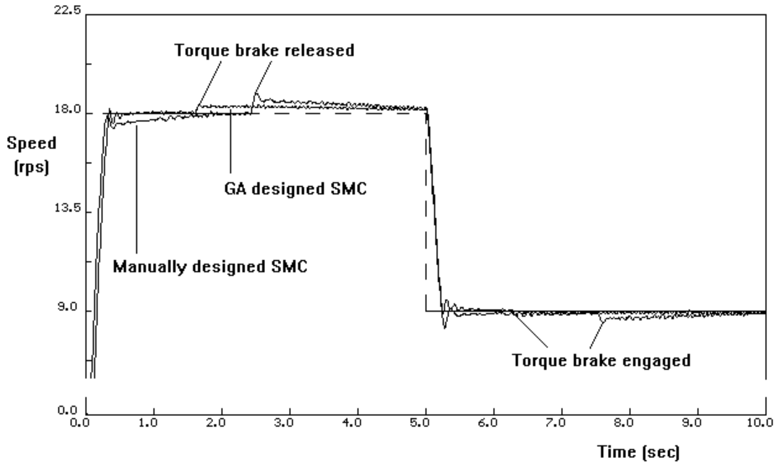

Figure 12. The captured responses of the GA optimised and manually designed controllers.

5. Conclusion and further work

This paper has reported the development of a genetic algorithm automated approach to the design of nonlinear control systems. Owing to the exponentially reduced search time obtained from the evolutionary NP program, this paradigm provides an "off-the-computer" design for control systems. The use of such an approach not only avoids the tedious manual trial-and-error process arising from the lack of analytical and numerical design approaches, but also yields control systems that give a better performance than manual designs in terms of transient and steady-state performance and of robustness in suppressing uncertainties, noise and parameter variations. The methodology is particularly useful in engineering systems, since a practical system always has constraints imposed by physical limitations.

A fast hybrid micro-evolution program (Li 1995f) is being developed for on-line tuning of the sliding mode controllers. Since coding is required by a GA, the development of an efficient "byte-coding" technique is also involved. This is not, however, necessary for other types of evolution algorithms. Further work has also been directed towards multivariable designs and towards finding useful novel SMC structures. This is, however, better achieved by GP techniques, which are the extension of the genetic model of learning into a space of both predictable and unpredictable solutions. Such a computing paradigm has currently been applied to the design of signal processors at Glasgow (Sharman *et al* 1995).

ACKNOWLEDGEMENTS

The support in part by the EPSRC under grant GR/K24987 ("Evolutionary Programming for Nonlinear Control") is gratefully acknowledged. Mr. K. C. Ng would like to thank the University and the CVCP for a Postgraduate Scholarship and an ORS Award. The authors are grateful to colleagues in their Evolutionary Computing and Control group and Dr. J. J. Gribble for useful comments and to Dr. G. R. Moores of Institute of Biomedical and Life Sciences, University of Glasgow, for his *Bioelectronics IV* lecture notes on genetics and replication.

REFERENCES

BARKER, H. A., 1995, Open environments and object-oriented methods for computer-aided control system design, *Control Eng. Practice*, **3**, (3), 347-356.

CAPPONETTO, R., FORTUNA, L., MUSCATO, G., and XIBILIA, M.G., 1994, Genetic algorithms for controller order reduction, *Proc. 1st IEEE Conf. Evolutionary Computation*, IEEE World Congress on Computational Intelligence, Orlando, FL, **2**, 724-729.

DORLING, C. M., 1985, *The Design of Variable Structure Control Systems: Manual for the VASSYD CAD Package*, SERC Report (Project GR/C/03119, Grant-holders: A.S.I. Zinober and S.A. Billings), Department of Applied and Computational Mathematics, University of Sheffield, Sheffield S10 2TN.

CHIPPERFIELD, A. J., FONSECA, C. M., and FLEMING, P. J., 1992, Development of genetic optimisation tools for multi-objective optimisation problems in CACSD, *IEE Colloq. on Genetic Algorithms for Control Systems Eng. Digest*, London WC2R 0BL, **106**, 3/1-3/6.

FLEXIBLE INTELLIGENCE GROUP, L. L. C., 1995, *FlexTool(GA) (Matlab Toolbox) User's Manual*, M1.1, Tuscaloosa, AL, Jan.

GOLDBERG, D., 1989, *Genetic algorithms in searching, optimisation and machine learning* (Reading, MA: Addison-Wesley).

GRAY, G. J., LI, Y., MURRAY-SMITH, D. J., and SHARMAN, K. C., 1995, Specification of a control system fitness function using constraints for genetic algorithm based design methods, *Proc. First IEE/IEEE Int. Conf. on GA in Eng. Syst.: Innovations and Appl.*, Sheffield, 530-535.

HÄUßLER, A., LI, Y., MURRAY-SMITH, D. J., and SHARMAN, K. C., Neurocontrollers designed by a genetic algorithm, *Proc. First IEE/IEEE Int. Conf. on Genetic Algorithms in Eng. Syst.: Innovations and Appl.*, Sheffield, 536-542.

HARP, S. A., and SAMAD, T., 1992, Optimizing neural networks with genetic algorithms, *Proc. American Power Conf.*, Chicago IL, **54**, (263) 1138-1143.

HERRERA, F., and VERDEGAY, J. L. (eds.), 1996, *Genetic Algorithms and Soft Computing*, (Heidelberg: Physica-Verlag, in press).

HOLLAND, J. H., 1992, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (Cambridge, MA: The MIT Press).

HUNG, J. Y., GAO, W., and HUNG, J. C., 1993, Variable structure control: A survey, *IEEE Trans. Ind. Electronics*, **40**, (1), 2-22.

HUNT, K. J., 1992, Polynomial LQG and H_∞ controller synthesis - A genetic algorithm solution, *Proc. 31st IEEE Conf. Decision and Control*, Tucson, AZ, **4,** (865), 3604-3609.

ITKIS, U., 1976, *Control Systems of Variable Structure*, (New York: Wiley).

KARR, C. L., MEREDITH, D. L., and STANLEY, D. A., 1990, Fuzzy process-control with a genetic algorithm, *Proc. Control '90 - Mineral and Metallurgical Processing*, Salt Lake City, UT, (32), 53-60.

KEANE, A. J., 1995, Genetic algorithm optimisation of multi-peak problems: studies in convergence and robustness, *Artificial Intelligence in Engineering*, **9**, 75-83.

KRISTINSSON, K., and DUMONT, G. A., 1992, System identification and control using genetic algorithms, *IEEE Trans. Syst., Man and Cyber.*, **22**, (5), 1033-1046.

KWOK, D. P., TAM, P., SUN, Z. Q., and WANG, P., 1991, Design of optimal linear regulators with steady-state trajectory insensitivity, *Proc. IECON 91*, **3**, (416), 2183-2187.

KWOK, D. P., and SHENG, F., 1994, Genetic algorithm and simulated annealing for optimal robot arm PID control, *Proc. 1st IEEE Conf. Evol. Comp.*, IEEE World Cong. Comp. Intel., Orlando, FL, **2**, 708-713.

LI, Y., and NG, K. C., 1994, *Design of SISO Sliding Mode Control Systems Using Genetic Algorithms*, Technical Report CSC-94002, Centre for Systems and Control, University of Glasgow, 14 June. (Available on Internet with URL: http://www.mech.gla.ac.uk/Control/reports.html).

LI, Y., 1995a, Modern information technology for control systems design and implementation, *Proc. 2nd Asia-Pacific Conf. Control and Measurement*, ChongQing, China, 17-22.

LI, Y., NG, K. C., HÄUßLER, A., CHOW, V. C. W., and MUSCATELLI, V. A., 1995b, Macroeconomics modelling on UK GDP growth by neural computing, *Pre-prints of IFAC/IFIP/IFORS/SEDC Symp. Modelling and Control of National and Regional Economies*, Australia.

LI, Y. and NG, K. C., 1995c, Genetic algorithm based techniques for design automation of three-term fuzzy systems, *Proc. 6th Int. Fuzzy Sys. Asso. World Cong.*, São Paulo, Brazil.

LI, Y., and TAN, K. C., 1995d, *Physical Parametric Modelling of Nonlinear Systems by Evolution*, Technical Report CSC-95010, Centre for Systems and Control, University of Glasgow, 9 June. (Available on Internet with URL: http://www.mech.gla.ac.uk/Control/reports.html).

LI, Y., TAN, K. C., NG, K.C., and MURRAY-SMITH, D. J., 1995e, Performance based linear control system design by genetic evolution with simulated annealing, *Proc. 34th IEEE Conf. Decision and Control*, New Orleans, LA.

LI, Y., 1995f, *Applied Neural Networks IV*, Part I - Evolutionary Computing, Lecture Notes (1JYX), Department of Electronics and Electrical Engineering, University of Glasgow.

LINKENS, D. A., and NYONGESA, H. O., 1993, Real-time acquisition of fuzzy rules using genetic algorithms, *Annual Review in Automatic Programming*, VERBRUGGEN, H. B. and RODD, M. G., Eds., (Oxford: Pergamon Press Ltd), **17**, (82), 335-339

LUA, H. K., 1993, *Sliding mode control for DC motors*, Project Report, Department of Electronics and Electrical Engineering, University of Glasgow.

MURDOCK, T. M., SCHMITENDORF, W. E., and FOREST, S., 1991, Use of genetic algorithm to analyze robust stability problems, *Proc. American Control Conf.*, Evanston, IL, **1**, 886-889.

NANDAM, P. K., and SEN, P. C., 1992, Control laws for sliding mode speed control of variable speed drives, *Int. J. Control*, **56,** (5), 1167-1186.

NG, K. C., and LI, Y., 1994, Design of sophisticated fuzzy logic controllers using genetic algorithms, *Proc. 3rd IEEE Int. Conf. Fuzzy Syst.*, IEEE World Cong. Comp. Intel., Orlando, FL, **3**, 1708-1712.

NG, K. C., LI, Y., MURRAY-SMITH, D. J., and SHARMAN, K. C., 1995, Genetic algorithms applied to fuzzy sliding mode controller design, *Proc. First IEE/IEEE Int. Conf. on GA in Eng. Syst.: Innovations and Appl.*, Sheffield, 220-225.

NG, K. C., 1995, *Switching Control Systems and Their Design Automation via Genetic Algorithms*, Ph.D. Thesis, Department of Electronics and Electrical Engineering, University of Glasgow, Oct.

PATTON, R. J., and LIU, G. P., 1994, Robust control design via eigenstructure assignment, genetic algorithms and gradient-based optimisation, *IEE Proc., Pt. D - Control Theory Appl.*, **141**, 202-208.

ROLEWICZ, S, 1987, *Functional analysis and control theory* (Dordrecht, Holland: D. Reidel Publishing Co.)

SHARMAN, K. C., and MCCLURKIN, G. D., 1989, Genetic algorithms for maximum likelihood parameter estimation, *Proc. IEEE Conf. Acoustics, Speech and Signal Processing*, Glasgow, 2716-2719.

SHARMAN, K. C., and ESPARCIA-ALCÁZAR, A. I., 1993, Genetic evolution of symbolic signal models, *Proc. IEE/IEEE Workshop on Natural Algorithms in Signal Processing*, University of Essex.

SHARMAN, K. C., ESPARCIA-ALCÁZAR, A. I., and LI Y., 1995, Evolving digital signal processing algorithms by genetic programming, *Proc. First IEE/IEEE Int. Conf. on GA in Eng. Syst.: Innovations and Appl.*, Sheffield, 473-480.

SLOTINE, J. E., and LI, W., 1991, *Applied Nonlinear Control* (Englewood Cliffs, NJ: Pretice-Hall).

SRINIVAS, M., and PATNAIK, L. M., 1994, Genetic algorithms: A survey, *IEEE Computer*, **27**, 17-26.

TAN, K. C., LI, Y., MURRAY-SMITH, D. J., and SHARMAN, K. C., 1995, System identification and linearisation using genetic algorithms with simulated annealing, *Proc. First IEE/IEEE Int. Conf. on GA in Eng. Syst.: Innovations and Appl.*, Sheffield, 164-169.

UTKIN, V. I. and YANG, K. D., 1978, Methods for constructing discontinuity planes in multidimensional variable structure systems, *Automatic Remote Control*, Guest Editor, **39**, (10), 1466-1470.

UTKIN, V. I. (Guest Editor), 1993, Special Issue on Sliding Mode Control, *Int. J. Control*, **57**, (5).

WANG, P., and KWOK, D. P., 1992, Auto-tuning of classical PID controllers using an advanced genetic algorithm, *Proc. 1992 Int. Conf. Ind. Elect., Contr., Instrum. and Automation*, **3**, 1224-1229.