Genetic algorithm approach to designing finite-precision controller structures

J.F.Whidborne and R.S.H.Istepanian

Abstract: The parameters of a digital control design usually need to be rounded when the controller is implemented with finite precision arithmetic. This often results in degradation of the closed loop performance and reduced stability margins. This paper presents a multi-objective genetic algorithm based approach to designing the structure of a finite-precision second-order state space controller implementation, which can simultaneously minimise some set of performance degradation indices and implementation cost indices. The approach provides a set of solutions that are near Pareto-optimal, and so allows the designer to trade-off performance degradation against implementation cost. The method is illustrated by the design of the structure of a PID controller for the IFAC93 benchmark problem.

1 Introduction

It is well known that controller implementations with fixed-point arithmetic offer a number of advantages over floating-point implementations [1, 2]. Fixed-point processors contribute to lower overall system costs and require less on-chip hardware. They have a smaller package size, less weight, consume less power and are often faster than floating-point processors. In addition, the simplicity of the architecture contributes to safer implementations. Thus for high-volume, price-sensitive applications, safety critical applications and hand-held and aerospace applications where power consumption and weight are important considerations, fixed-point processors are preferred over floating point. However, a closed-loop control system will suffer a performance degradation and may even become unstable when the designed infinite-precision controller is implemented with a fixed-point digital processor due to the finite precision of the parameter representation resulting from the finite word length (FWL). This so-called FWL effect is strongly dependent upon the parameterisation of the controller. Thus, over the years, many results have been reported in the literature dealing with FWL implementation and their relevant parameterisation issues, for example [3–8].

Consider the discrete time system shown in Fig. 1 with the plant $G(z)$. Let (A_k, B_k, C_k, D_k) be a state–space description of the state space controller

$$K(z) := C_k(zI - A_k)^{-1}B_k + D_k \tag{1}$$

In this paper (A_k, B_k, C_k, D_k) is also called a realisation of $K(z)$. The realisations of $K(z)$ are not unique, if $(A_k^0, B_k^0, C_k^0, D_k^0)$ is a realisation of $K(z)$, $(T^{-1}A_k^0T, T^{-1}B_k^0, C_k^0T, D_k^0)$ is an equivalent realisation for any nonsingular similarity transformation T. A common approach, for example [6, 7, 9], to the FWL problem for state–space controllers is to find equivalent controller realisations (or similarity transformations T) such that the closed-loop system is, in some way, insensitive to perturbations in the controller parameters.

A more direct genetic algorithm (GA) based approach is presented in this paper. GAs have been used to design FWL digital filters, for example [10, 11], but the authors are not aware of any work on using GAs for closed-loop FWL controller or controller implementation design. Basically, the approach is to find an FWL controller that is near to the originally designed controller such that the closed loop performance and robustness degradation and the FWL implementation cost are simultaneously minimised. The approach is based on the generation of near-equivalent finite word-length controller representations by means of the solution to a linear system equivalence completion problem followed by a rounding operation. A multi-objective genetic algorithm (MOGA) [12–14], is then used to find sets of pareto-optimal near-equivalent FWL controllers. This allows the designer to trade-off FWL implementation performance and robustness against the cost and other requirements of the implementation by enabling the designer to simply choose the most appropriate design from the set.

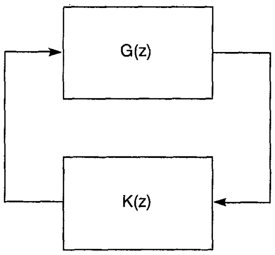

Fig. 1 *Discrete time feedback system*

IEE Proceedings online no. 20010604
DOI: 10.1049/ip-cta:20010604
Paper first received 24th June 1999 and in revised form 7th March 2000

J.F. Whidhorne is with the Division of Engineering, King's College London, Strand, London WC2R 2LS, UK
E-mail: james.whidborne@kcl.ac.uk

R.S.H. Istepanian is with the Department of Electronic and Computer Engineering, Brunel University, Uxbridge, Middlesex, UB8 3PH, UK
E-mail: robert.istepanian@brunel.ac.uk

Two features of GAs make them very attractive for solving this problem. Firstly, GAs require the solution space to be encoded in binary strings, and since controllers implemented with FWL are also coded in binary, a one-to-one relationship between the genotype and the phenotype within the GA can be defined. Secondly, GAs allow the optimal structure of the solution to be found [15–17]. This means that the implementation word length does not need to be defined *a priori*, but the GA will select the best from a predefined set, and so the implementation cost in the form of the memory requirement can also be minimised.

In this paper, a multi-objective genetic algorithm is used. This allows the designer to trade-off FWL implementation robust stability and performance measures against cost measures and other requirements of the implementation. The method is entirely generic, and any computable set of stability, performance, robustness, round-off noise and implementation cost measures could be used. The developed approach is illustrated by a simplified application to the problem of the implementation of a PID controller designed for the IFAC93 benchmark problem [18, 19].

2 Preliminary concepts and theory

2.1 Multi-objective optimisation

The majority of engineering design problems are multi-objective, in that there are several conflicting design aims which need to be simultaneously achieved. If these design aims are expressed quantitatively as a set of n design objective functions $\phi_i(p){:}i=1\dots n$, where p denotes the design parameters chosen by the designer, the design problem could be formulated as a multi-objective optimisation problem:

$$\min_{p\in\mathcal{P}}\{\phi_i(p),\ \text{for } i=1\dots n\} \qquad (2)$$

where $\mathcal{P}$ denotes the set of possible design parameters p. In most cases, the objective functions are in conflict, so the reduction of one objective function leads to the increase in another. Subsequently, the result of the multi-objective optimisation is known as a pareto-optimal solution. A pareto-optimal solution has the property that it is not possible to reduce any of the objective functions without increasing at least one of the other objective functions.

2.2 Genetic algorithms

Genetic algorithms are search procedures based on the evolutionary process in nature. The idea is that the GA operates on a population of individuals, each individual representing a potential solution to the problem, and applies the principle of survival of the fittest on the population, so that the individuals evolve towards better solutions to the problem.

The individuals are given a chromosomal representation, which corresponds to the genotype of an individual in nature. Three operations can be performed on individuals in the population, selection, crossover and mutation. These correspond to the selection of individuals in nature for breeding, where the fitter members of a population breed and so pass on their genetic material. The crossover corresponds to the combination of genes by mating, and mutation to genetic mutation in nature. The selection is weighted so that the 'fittest' individuals are more likely to be selected for crossover, the fitness being a function of the function which is being minimised. By means of these operations, the population will evolve towards a near-optimal solution.

GAs are very well suited for multi-objective optimisation problems. The approach used here is the multi-objective genetic algorithm (MOGA) [12–14], which is an extension on an idea by [20]. The idea behind the MOGA is to develop a population of Pareto-optimal or near pareto-optimal solutions, and so maintaining the genuine multi-objective nature of the problem. This is achieved by finding a set of solutions which are nondominated. An individual j with a set of objective functions $\phi^j=(\phi^j_1,\dots,\phi^j_n)$ is said to be nondominated if, for a population of N individuals, there are no other individuals $k=1,\dots,N,\ k\neq j$ such that $\phi_i^k\leq\phi_i^j\forall i=1,\dots,n$ and $\phi_i^k<\phi_i^j$ for at least one i. With the MOGA, nondominated individuals are given the greatest fitness, and individuals that are dominated by many other individuals are given a small fitness. Using this mechanism, the population evolves towards a set of nondominated, near pareto-optimal individuals. Details of this mechanism are given in [14].

In addition to finding a set of near pareto-optimal individuals, it is desirable that the sample of the whole Pareto-optimal set given by the set of nondominated individuals is fairly uniform. A common mechanism to ensure this is fitness sharing [14], which works by reducing the fitness of individuals that are genetically close to each other. However, as will be seen in Section 3.2, not all the bits of a candidate solution bit string are necessarily active. Thus, two individuals may have the same genotype, but different gene strings. Thus it is difficult to measure the difference between two genotypes in order to implement fitness sharing, so, for the sake of simplicity in this paper, multiple copies of genotypes are simply removed from the population.

2.3 FWL representation

A typical 2's complement FWL fixed-point representation of a number $q(x)$ is shown in Fig. 2. The number $q(x)$ is represented by a $m+n+1$ binary string x where

$$x=[x_0,x_1,\dots,x_m,x_{m+1},\dots,x_{m+n}] \qquad (3)$$

$x_i\in\{0,\ 1\}$, $m\in\{0,\ 1,\ 2,\dots\}$ and $n\in\{0,\ 1,\ 2,\dots\}$. The value $q(x)$ is given by

$$q(x)=-x_0 2^m+\sum_{i=1}^{m}x_i 2^{i-1}+\sum_{i=m+1}^{m+n}x_i 2^{m-i} \qquad (4)$$

The set of possible values which can be taken by an FWL variable represented by a $m+n+1$ binary string x is defined as $\mathcal{Q}_{n,m}$ given by

$$\mathcal{Q}_{n,m}:=\{q:q=q(x),x_i\in\{0,1\}\forall i\}. \qquad (5)$$

2.4 Linear system equivalence completion problem

In the proposed method, near-equivalent finite word-length controller representations are generated by means of the solution to a linear system equivalence completion problem; some preliminary results are first required. The definition below is from [21, p 154].

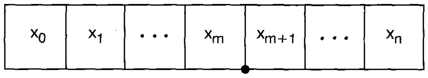

Fig. 2 *FWL fixed-point representation*

Definition 1: A matrix A is similar to a matrix $\tilde{A}$ if, and only if, there exists a nonsingular matrix T such that $\tilde{A} = T^{-1}AT$.

Lemma 1: Given a matrix $A \in \mathbb{R}^{2\times 2}$ such that $A \neq \alpha I \forall \alpha \in \mathbb{R}$, where

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{6}$$

and given the real pair $(\tilde{a}_{11}, \tilde{a}_{12} \neq 0)$ then A is similar to $\tilde{A}$ where

$$\tilde{A} = \begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} \\ \tilde{a}_{21} & \tilde{a}_{22} \end{bmatrix} \tag{7}$$

and where $\tilde{a}_{22} = a_{11} + a_{22} - \tilde{a}_{11}$ and $\tilde{a}_{21} = (\tilde{a}_{11}\tilde{a}_{22} - \det A)/\tilde{a}_{12}$.

Proof: To prove this Lemma, we use the property that if A and $\tilde{A}$ have the same minimal polynomial and the same characteristic polynomial, and their minimal polynomial is the same as their characteristic polynomial, then A and $\tilde{A}$ are similar [22, p 150]. The characteristic equations of A and $\tilde{A}$ are $p_A(t) = \det(tI - A) = t^2 - (a_{11} + a_{22})t + (a_{11}a_{22} - a_{21}a_{12})$ and $p_{\tilde{A}}(t) = \det(tI - \tilde{A}) = t^2 - (\tilde{a}_{11} + \tilde{a}_{22})t + (\tilde{a}_{11}\tilde{a}_{22} - \tilde{a}_{21}\tilde{a}_{12})$, respectively. By equating coefficients, we obtain the expressions for $\tilde{a}_{21}$ and $\tilde{a}_{22}$. It remains to be shown that the minimal polynomial of A is the same as the characteristic polynomial. For the case where the eigenvalues of A are unique, it is clear that the the minimal polynomial is the same as the characteristic polynomial. For the case where the eigenvalues of A are repeated and A is nonderogatory, the Jordan canonical form of A is

$$J_A = \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix} \tag{8}$$

where λ is the repeated eigenvalue and the minimal polynomial is the same as the characteristic polynomial [23, p. 6]. However, if A is diagonal with repeated eigenvalues, that is $A = \lambda I$, $A = T^{-1}AT$ for all nonsingular T. Thus the Jordan canonical form equals A and A is derogatory and so the minimal polynomial is of lower order than the characteristic polynomial. This form is explicitly excluded. □

The derogatory case excluded in the above lemma is shown to be impractical for controller implementation by the following lemma.

Lemma 2: For a two-state SISO LTI system $F(z)$, where

$$F \overset{s}{=} \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \tag{9}$$

$$\overset{s}{=} \left[\begin{array}{cc|c} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ \hline c_1 & c_2 & d \end{array}\right] \tag{10}$$

the system is unobservable if $A = \alpha I \forall \alpha \in \mathbb{R}$.

Proof: If $A = \alpha I$, the observability matrix:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \end{bmatrix} \tag{11}$$

is given by

$$\mathcal{O} = \begin{bmatrix} c_1 & c_2 \\ \alpha c_1 & \alpha c_2 \end{bmatrix} \tag{12}$$

which is rank deficient, hence the system is unobservable. □

The main theorem in which a linear system equivalence completion problem is solved can now be stated.

Theorem 1: Given an observable two-state SISO LTI system $F(z)$, where

$$F \overset{s}{=} \left[\begin{array}{c|c} A & B \\ \hline \tilde{C} & \tilde{D} \end{array}\right] \tag{13}$$

$$\overset{s}{=} \left[\begin{array}{cc|c} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ \hline c_1 & c_2 & \tilde{d} \end{array}\right] \tag{14}$$

given $(\tilde{a}_{11}, \tilde{a}_{12} \neq 0, \tilde{c}_1, \tilde{c}_2)$ there exists an equivalent state-space representation $\tilde{F}$ where

$$\tilde{F} \overset{s}{=} \left[\begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & \tilde{D} \end{array}\right] \tag{15}$$

$$\overset{s}{=} \left[\begin{array}{cc|c} \tilde{a}_{11} & \tilde{a}_{12} & \tilde{b}_1 \\ \tilde{a}_{21} & \tilde{a}_{22} & \tilde{b}_2 \\ \hline \tilde{c}_1 & \tilde{c}_2 & \tilde{d} \end{array}\right] \tag{16}$$

such that $\tilde{F}(z) = F(z)$ where

$$F(z) = C[zI - A]^{-1}B + D \tag{17}$$

and

$$\tilde{F}(z) = CT[zI - T^{-1}AT]^{-1}T^{-1}B + D \tag{18}$$

where

$$T = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \tag{19}$$

is nonsingular if $(\tilde{A}, \tilde{C})$ is observable.

Proof: The proof is by construction. From eqn. 18:

$$\tilde{A} = T^{-1}AT \tag{20}$$

From Definition 1, and from Lemma 2, since the system is observable, the case $A = \alpha I \forall \alpha \in \mathbb{R}$ can be excluded. Hence from Lemma 1, $\tilde{a}_{22}$ is given by $\tilde{a}_{22} = a_{11} + a_{22} - \tilde{a}_{11}$ and $\tilde{a}_{21}$ is given by $\tilde{a}_{21} = (\tilde{a}_{11}\tilde{a}_{22} - \det A)/\tilde{a}_{12}$. From eqn. 20, $AT - T\tilde{A} = 0$, which gives [24, p. 255]

$$[I \otimes A - \tilde{A}^T \otimes I]\mathrm{t} = 0 \tag{21}$$

where $\mathrm{t} = [t_{11}, t_{21}, t_{12}, t_{22}]^T$ and $[I \otimes A - \tilde{A}^T \otimes I]$ is rank 2. Now, from eqn. 18, $CT = \tilde{C}$. Hence, given $\tilde{c}_1, \tilde{c}_2$:

$$\begin{bmatrix} c_1 & c_2 & 0 & 0 \\ 0 & 0 & c_1 & c_2 \end{bmatrix} \mathrm{t} = \begin{bmatrix} \tilde{c}_1 \\ \tilde{c}_2 \end{bmatrix} \tag{22}$$

A $Y \in \mathbb{R}^{4\times 4}$ can be constructed from eqns. 21 and 22, where $Y\mathrm{t} = z$, where Y is nonsingular rank 4 and where $z \in \mathbb{R}^4$ is a column vector with two elements of $\tilde{C}$ and two zero elements. Hence t can be calculated and T obtained. Since F is observable, the observability matrix

$$\mathcal{O} = \begin{bmatrix} C \\ CA \end{bmatrix} \tag{23}$$

is rank 2. Since the pair $(\tilde{A}, \tilde{C})$ is required to be observable, the observability matrix:

$$\tilde{O} = \begin{bmatrix} \tilde{C} \\ \tilde{C}\tilde{A} \end{bmatrix} \tag{24}$$

is rank 2 and since

$$\begin{bmatrix} \tilde{C} \\ \tilde{C}\tilde{A} \end{bmatrix} = \begin{bmatrix} CT \\ CAT \end{bmatrix} \tag{25}$$

T must be nonsingular. Thus $\tilde{B}$ can be calculated. From eqn. 18, $\tilde{d} = d$. □

Note that, by redefining the transformation matrix as T^{-1}, the problem given $\tilde{B}$ instead of $\tilde{C}$ can be solved. Note also that the only restrictions on the problem are that the original system and the equivalent system are observable, and that certain canonical realisations (i.e. diagonal and lower triangular) are excluded by the constraint that a_{12} is not zero. These realisations can be separately programmed into the method if necessary.

3 MOGA for optimal FWL controller structures

3.1 FWL design procedure

The proposed approach is to use the MOGA to evolve a set of near pareto-optimal solutions to the following problem.

Problem: Given a discrete time nominal plant $G(z)$ and designed PID (or other second-order) controller $K(z)$, find an FWL controller $K_q(z)$ and state–space parameterisation such that the difference (in some sense) between the closed loop system and the original closed loop system and the implementation costs are simultaneously minimised.

For each individual in the population, the following procedure is used to generate a possible solution candidate.

(i) Generate a partially filled random FWL parameterisation of the two-state controller, i.e.

$$\tilde{K} \overset{s}{=} \left[\begin{array}{cc|c} q_1 & q_2 & ? \\ ? & ? & ? \\ \hline q_3 & q_4 & D_K \end{array}\right] \tag{26}$$

where $q_j \in \mathcal{Q}_{n,m}, j = 1, \ldots, 4$, i.e. each q_j is FWL.
(ii) By Theorem 1, solve a linear system equivalence completion problem such that $\tilde{K}(z) = K(z)$.
(iii) Obtain $K_q \approx \tilde{K}$ by rounding the non-FWL parameters in $\tilde{K}$ so that they are FWL, i.e.

$$K_q \overset{s}{=} \left[\begin{array}{cc|c} q_1 & q_2 & q_7 \\ q_5 & q_6 & q_8 \\ \hline q_3 & q_4 & q_9 \end{array}\right] \tag{27}$$

where $q_j \in \mathcal{Q}_{n,m}, j = 1, \ldots, 9$.
(iv) Calculate a set of robust stability/performance degradation indices and implementation cost indices $\{\phi_i : i = 1 \ldots n\}$ for K_q.

3.2 Encoding of solution space

The encoding of the solution space is described in this Section. Here, it is assumed there is a maximum possible word length of 16 bits, however, it is a trivial matter to change this to use an 8, 32 or 64 bit maximum word length.

To generate the partially filled parameterisations of K given by eqn. 26, the genotype of each individual consists of a 71 bit binary string $[x_1, \ldots, x_{71}]$, $x_i \in \{0,1\}$. The bit length of the integer part of the parameters' representation $m \in 0, \ldots, 7$ is represented by $[x_1, x_2, x_3]$, and is given by $m = \sum_{i=1}^{3} x_i 2^{i-1}$. The bit length of the fractional part of the parameters' representation $n \in 0, \ldots, 15$ is represented by $[x_4, \ldots, x_7]$, and is given by $n = \min(\sum_{i=4}^{7} x_i 2^{i-4}, 15 - m)$. The four $m + n + 1$ word-length parameters $q_j \in \mathcal{Q}_{n,m}$, $j = 1, \ldots, 4$, where $(q_1, q_2, q_3, q_4) = (\tilde{a}_{11}, \tilde{a}_{12}, \tilde{c}_1, \tilde{c}_2)$ respectively, are represented by $[x_{8+16(j-1)}, \ldots, x_{8+m+n+16(j-1)}]$. Thus not all the bits in x are necessarily active. The values of q_j are calculated by eqn. 4.

4 Application to an IFAC93 benchmark problem design

The proposed approach is illustrated by application to the 'stress level 2' PID controller [19] designed for the IFAC93 benchmark problem [18]. In this problem, the nominal plant is provided along with perturbation ranges for 5 of the plant parameters. The PID controller has been optimally tuned for a set of robust performance criteria. For further details, see [19].

To obtain the optimal digital controller structure for this example, two indices are defined for the MOGA. The first index ϕ_1 is the closed loop measure of the implementation accuracy of the FWL controller compared to the original controller and is taken as a robust stability/performance degradation measure in terms of an H_∞-norm. The second index ϕ_2 is a measure of the total number of bits required to store the FWL controller parameters. These indices are defined in Appendix 7.1.

The continuous time nominal plant, $G(s)$, is given as:

$$G(s) = \frac{25(1 - 0.4s)}{(s^2 + 3s + 25)(5s + 1)} \tag{28}$$

The plant, $G(s)$, is discretised with sampling period of $t_s = 0.05$ seconds. A PID controller is designed as [19]:

$$K(s) = 1.311 + 0.431/s + 1.048s/(s + 12.92) \tag{29}$$

and discretised using the bilinear transform. The initial realisation K^0 is set to:

$$K_q \overset{s}{=} \left[\begin{array}{cc|c} 0 & -0.51172 & 1 \\ 1 & 1.5117 & 0 \\ \hline -0.36524 & -0.17638 & 2.1139 \end{array}\right] \tag{30}$$

The MOGA is implemented in MATLAB using the GA Toolbox [25]. An elitism strategy is used whereby all nondominated individuals propagate through to the next population. Selection is performed using stochastic universal sampling with a fitness determined by the number of dominating individuals. Single point crossover is used with a probability of 0.7. Each bit has a probability of mutation of 0.00933.

The MOGA is run with a population of 120. After 800 generations (which takes about 3 hours on a 450 MHz Pentium II), a set of nondominated solutions is obtained. This set is shown in Fig. 3. The figure also shows the stable dominated solutions along with FWL implementations of the initial realisation, K_q^0, for various word lengths. In addition, the equivalent balanced realisation [26, pp. 72–78] of $K(z)$ is calculated and FWL implementations of the realisation for various word lengths are shown. Note that the axis for $\phi_2 = \|R - R_q\|_\infty$ is shown to log scale for clarity (the closed loop transfer functions R and R_q are defined in Appendix 7.1). The figure shows that

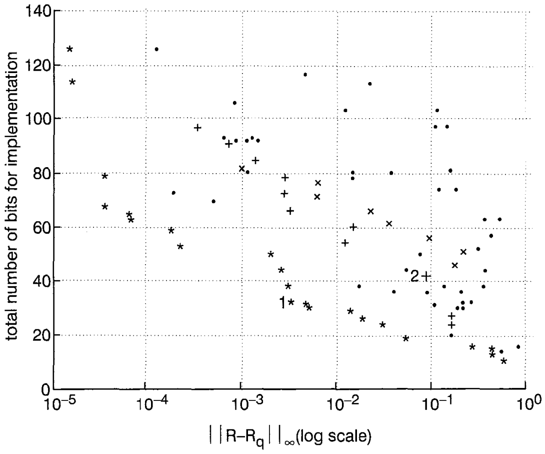

Fig. 3 *Solution set showing trade-off with MOGA non-dominated set (*), MOGA dominated set (·), K_q^0 realisations (×) and equivalent balanced realisations (+). The selected controllers are marked 1 and 2*

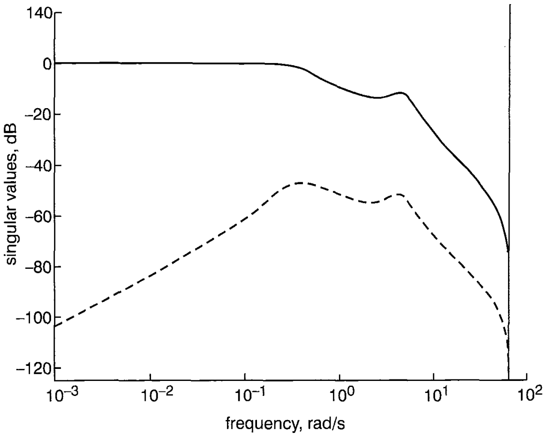

Fig. 4 *Singular values of R (—) and $R - R_q^{(1)}$ (- - -)*

improved FWL state–space realisations for the controller can be obtained using the approach.

The 32-bit solution labelled (1) in Fig. 3 is selected and is the controller:

$$K_q^{(1)} \overset{s}{=} \left[\begin{array}{cc|c} 2^{-1} & -2^{-1} & 2^{-4} \\ 0 & 1 & 0.34375 \\ \hline -1 & -0.9375 & 2.125 \end{array}\right] \qquad (31)$$

which requires $m = 2$ and $n = 5$ for the FWL representation given by eqn. 4. In addition, one of the parameters is zero, two are ±1, and three others are powers of 2, and can hence be implemented by register shifts. Fig. 4 shows the frequency response of the original closed loop transfer function $|R(e^{j\omega t_s})|$ and the difference between the systems $|R(e^{j\omega t_s}) - R_q^{(1)}(e^{j\omega t_s})|$. Fig. 5 shows the step response of the system with the original controller design $K(s)$ and the digital controller K_q for the nominal plant and the envelope provided by the 32 plants with the parameters at their extreme values. There is very little difference between the two sets of responses.

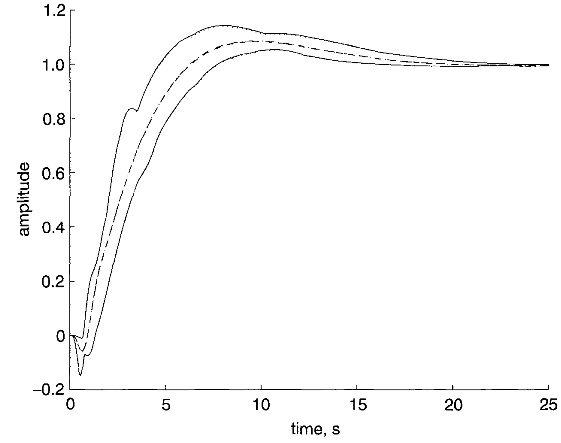

Fig. 5 *Step response for original design, K, with nominal plant (-·-·-) and envelope of extreme plants (···) and for controller, $K_q^{(1)}$, with nominal plant (- - -) and envelope of extreme plants (——)*

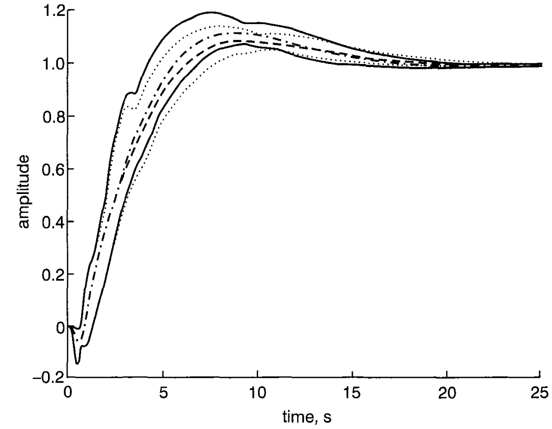

Fig. 6 *Step response for original design, K, with nominal plant (-·-·-) and envelope of extreme plants (···) and for controller, $K_q^{(2)}$, with nominal plant (- - -) and envelope of extreme plants (——)*

For comparison, the 43-bit balanced FWL controller parameterisation labelled (2) in Fig. 3 is selected and is the controller:

$$K_q^{(2)} \overset{s}{=} \left[\begin{array}{cc|c} 1 & 0 & -0.15625 \\ 0 & 2^{-1} & -0.625 \\ \hline -0.15625 & 0.625 & 2.125 \end{array}\right] \qquad (32)$$

which also requires $m = 2$ and $n = 5$ for the FWL representation given by eqn. 4. Fig. 6 shows the step response of the system with the original controller design $K(s)$ and the digital controller $K_q^{(2)}$ for the nominal plant and the envelope provided by the 32 plants with the parameters at their extreme values. The deterioration in the step response here is marked.

5 Conclusions

A GA-based method to determine optimal FWL structures for PID digital controllers has been presented. The method is illustrated by an example. The method exploits the fact that the implementation of FWL controllers is by means of binary numbers, as is the representation in genetic algorithms.

The method requires the solution of a linear system equivalence completion problem. A solution to the linear system equivalence completion problem for 2 state SISO systems has been presented. This solution needs to be extended to general linear systems for the methodology to be extended to higher order and multivariable controllers. Otherwise, the method is entirely generic, in that any set of computable stability and performance measures as well as implementation cost measures can be used. For the illustrative example, the only cost measurement used was the number of bits required for the implementation. For an actual implementation, measures of other important quantisation effects, namely quantisation noise and scaling requirements should also be included. The example also illustrates that robust control system designs are often far more sensitive to perturbations in the controller than in the plant; some extreme examples of this are presented in [27].

6 References

1 CLARKE, P.: 'MATRIXx 6.0: a fixed point development process for automotive applications', *in* 'IMechE seminar publication – automotive electronics (Autotech '97)', (Mechanical Engineering Publications, Birmingham, UK, 1997), pp. 59–67

2 MASTEN, M.K., and PANAHI, I.: 'Digital signal processors for modern control systems', *Contr. Eng. Pract.*, 1997, **5**, (4), pp. 449–458

3 KNOWLES, J.B., and EDWARDS, R.: 'Effect of a finite-word-length computer in a sampled-data feedback system', *Proc. IEE*, 1965, **112**, (6), pp. 1197–1207

4 MORONY, P.: 'Issues in the implementation of digital feedback compensators', *in* 'Signal processing, optimization, and control series', (MIT Press, Cambridge, MA, 1983) No. 5

5 WILLIAMSON, D.: 'Digital control and implementation: finite wordlength considerations', *in* 'Systems and control engineering', (Prentice Hall, Englewood Cliffs, NJ, 1991)

6 GEVERS, M., and LI, G.: 'Parametrizations in control, estimations and filtering problems: accuracy aspects' (Springer-Verlag, Berlin, 1993)

7 ISTEPANIAN, R.H., LI, G., WU, J., and CHU, J.: 'Analysis of sensitivity measures of finite-precision digital controller structures with closed-loop stability bounds', *IEE Proc. Contr. Theory Appl.*, 1998, **145**, (5), pp. 472–478

8 CHEN, S., WU, J., ISTEPANIAN, R.H., CHU, J., and WHIDBORNE, J.F.: 'Optimizing stability bounds of finite-precision controller structures for sampled-data systems in the δ-operator domain', *IEE Proc. Control Theory Appl.*, 1999, **146**, (6), pp. 517–526

9 WHIDBORNE, J.F., WU, J., and ISTEPANIAN, R.H.: 'Finite word length stability issues in an ℓ_1 framework', *Int. J. Contr.*, 2000, **73**, (2), pp. 166–176

10 XU, D.J., and DALEY, M.L.: 'Design of optimal digital filter using a parallel genetic algorithm', *IEEE Trans. Circuits Syst.: II*, 1995, **42**, (10), pp. 673–675

11 UESAKA, K., and KAWAMATA, M.: 'Synthesis of low coefficient sensitivity digital filters using genetic programming', *Proc. IEEE Int. Symp. Circuits Syst.*, 1999, **3**

12 FONSECA, C.M., and FLEMING, P.J.: 'Genetic algorithms for multiobjective optimization: formulation, discussion and generalization', *in* 'Genetic algorithms: proceedings of the fifth international conference' (San Mateo, CA, 1993), pp. 416–423

13 FONSECA, C.M., and FLEMING, P.J.: 'Multiobjective genetic algorithms'. IEE colloquium on genetic algorithms for control systems engineering (London, UK, 1993), No. 1993/130, pp. 6/1–6/5

14 FONSECA, C.M., and FLEMING, P.J.: 'Multiobjective optimization and multiple constraint handling with evolutionary algorithms – part I: a unified formulation', *IEEE Trans. Syst. Man & Cybernetics, A*, 1995, **28**, (1), pp. 26–37

15 MAN, K.F., TANG, K.S., KWONG, S., and HALANG, W.A.: 'Genetic algorithms for control and signal processing', *in* 'Advances in industrial control' (Springer, London, UK, 1997)

16 TANG, K.S., MAN, K.F., and GU, D.-W.: 'Structured genetic algorithm for robust H^{∞} control system design', *IEEE Trans. Ind. Electr.*, 1996, **43**, (15), pp. 575–582

17 DAKEV, N.V., WHIDBORNE, J.F., CHIPPERFIELD, A.J., and FLEMING, P.J.: 'H_{∞} design of an EMS control system for a magley vehicle using evolutionary algorithms', *Proc. I MechE, Part I: J. Syst. & Contr.*, 1997, **311**, (4), pp. 345–355

18 GRAEBE, S.F.: 'Benchmark IFAC 93: Adaptive/robust control of unknown plant'. Preconference communication, 1992. Centre for Industrial Control Science, University of Newcastle, Australia

19 WHIDBORNE, J.F., MURAD, G., GU, D.-W., and POSTLETHWAITE, I.: 'Robust control of an unknown plant – the IFAC 93 benchmark', *Int. J. Control*, 1995, **61**, (3), pp. 589–640

20 GOLDBERG, D.E.: 'Genetic algorithms in search, optimization and machine learning' (Addison-Wesley, Reading, MA, 1989)

21 HERNSTEIN, I.N., and WINTER, D.J.: 'Matrix theory and linear algebra' (Macmillan, New York, 1988)

22 HORN, R.A., and JOHNSON, C.R.: 'Matrix analysis' (Cambridge University Press, Cambridge, UK, 1985)

23 BARNETT, S.: 'Matrices in control theory – with applications to linear programming' (Van Nostrand Reinhold, London, UK, 1971)

24 HORN, R.A., and JOHNSON, C.R.: 'Topics in matrix analysis' (Cambridge University Press, Cambridge, UK, 1991)

25 CHIPPERFIELD, A., FLEMING, P.J., POHLHEIM, H., and FONSECA, C.M.: 'Genetic algorithm toolbox: user's guide' (Department Automatic Control and Systems Engineering, University of Sheffield, UK, 1994)

26 ZHOU, K., DOYLE, J.C., and GLOVER, K.: 'Robust and optimal control' (Prentice Hall, Upper Saddle River, NJ, 1996)

27 KEEL, L.H., and BHATTACHARRYYA, S.P.: 'Robust, fragile, or optimal?', *IEEE Trans.*, 1997, **Ac-42**, (8), pp. 1098–1105

7 Appendix

7.1 Performance indices

A measure of the difference between the FWL implemented closed loop system and the original closed loop system is taken as the H_{∞} norm of the difference between the closed loop pole transfer functions of the systems. If both the implemented controller and closed loop system are stable, ϕ_1 is defined as

$$\phi_1 := \| R - R_q \|_{\infty} \qquad (33)$$

where

$$R = \frac{GK}{1 + GK} \qquad (34)$$

and

$$R_q = \frac{GK_q}{1 + GK_q} \qquad (35)$$

This is also a measure of the robust stability/performance degradation.

An implementation memory function, ϕ_2, is defined as the total number of bits used to implement K_q. This function is calculated bearing in mind that parameters of K_q that are 1, 0, -1 or that are a power of 2 require less memory requirement than the $m + n + 1$ bits from eqn. 4.