Multi-objective design of finite word-length controllers

R.S.H. Istepanian
Dept of Electrical and Computer Engineering
Ryerson Polytechnic University
350 Victoria St
Toronto M5B 2K3, Canada.

J.F. Whidborne
Division of Engineering
Kings College London
Strand, London WC2R 2LS, UK
james.whidborne@kcl.ac.uk

Abstract- Feedback controller implementations with fixed-point arithmetic offer the advantages of speed, memory space, cost and simplicity compared to floating-point arithmetic. Thus, to date, fixed-point processors dedicated for digital controller implementations are still the dominant architecture in many modern digital control engineering applications, particularly for automotive, consumer, military and medical applications. However, most control design methodologies assume that the designed controller will be implemented with infinite precision. Thus, when the controller is implemented, rounding errors in the implementation can cause a performance degradation and even instability. However, to directly design the controller with finite-precision control parameters can result in lower performance and stability robustness. Hence, there is a trade-off between the cost of the controller (in terms of word-length and memory requirements) and the performance of the system. This paper will present an approach to designing finite word-length feedback controllers by means of a multi-objective genetic algorithm. The approach provides a set of solutions that are (near) Pareto optimal with respect to minimal word-length and memory requirements and to closed loop performance. The method is applied to the design of a finite word-length controller for two examples, a PID controller for a rolling strip mill, and a controller the IFAC93 benchmark problem.

1 Introduction

It is well known that controller implementations with fixed-point arithmetic offer advantages of speed, memory space, cost and simplicity when compared to floating-point arithmetic (Masten & Panahi 1997). Thus, to date, fixed-point processors are still the dominant architecture in many modern digital control engineering applications found in automotive, consumer, military and medical applications. 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 in fact strongly dependent upon the parameterization of the controller. Thus, over the years, many results have been reported in the literature dealing with FWL implementation and their relevant parameterization issues, for example, Knowles & Edwards (1965), Morony (1983), Williamson (1991) and Gevers & Li (1993).

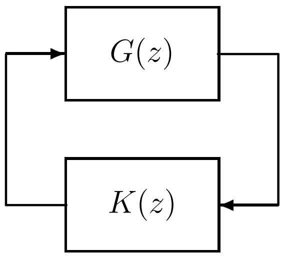

Figure 1: Discrete time feedback system

Consider the discrete time system shown in Figure 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, $K(z) = C_k(zI - A_k)^{-1}B_k + D_k$. In this paper, (A_k, B_k, C_k, D_k) is also called a realization of $K(z)$. The realizations of $K(z)$ are not unique, if $(A_k^0, B_k^0, C_k^0, D_k^0)$ is a realization of $K(z)$, then $(T^{-1}A_k^0T, T^{-1}B_k^0, C_k^0T, D_k^0)$ is an equivalent realization for any non-singular similarity transformation T. A common approach (for example, (Gevers & Li 1993)) to the FWL problem for state space controllers is to to find equivalent controller realizations (or similarity transformations T) such that the closed loop system is insensitive to perturbations in the controller parameters.

A more direct Genetic Algorithm (GA) based approach is presented in this paper. GA's have been used to design FWL digital filters, for example, Xu & Daley (1995), but the authors are not aware of any work on using GA's for closed loop FWL controller design. Basically, the approach is to find an FWL controller that is near to the originally designed controller such that the robust stability and closed loop performance degradation and the FWL implementation cost are simultaneously minimized. 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) is then used to find sets of Pareto optimal near-equivalent FWL controllers. This allows the designer to trade-off FWL implementation stability and performance against the memory and other requirements of the implementation by enabling the designer to simply choose the most appropriate design from the set.

Two features of GA's make them very attractive for solving this problem. Firstly, GA's 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, GA's allow the optimal structure of the solution to be found (Man, Tang, Kwong & Halang 1997, Tang, Man & Gu 1996, Dakev, Whidborne, Chipperfield & Fleming 1997), 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 minimized.

The developed approach is applied to two problems. The first is the implementation of a PID controller for a steel rolling mill problem (Hori 1996) which was also solved using a hill-climbing approach by Istepanian, Wu, Whidborne & Chu (1998). The second problem is the implementation of a PID controller designed for the IFAC93 benchmark problem (Graebe 1992, Whidborne, Murad, Gu & Postlethwaite 1995). The method is entirely generic with respect to the objective cost functions, so any set of stability/performance and implementation measures could be used. For reasons of simplicity, for the first problem, the performance and stability are assessed simply by the maximal change in closed loop poles, and for the second, by the H_∞-norm of the difference between the FWL closed-loop transfer function and the original closed-loop transfer function. The implementation cost for both examples is taken as the parameter memory storage requirement.

The paper is organized as follows. In the next section, some of the concepts underpinning the method are introduced, namely, multi-objective optimization, multi-objective genetic algorithms, FWL representation. The method also requires a linear system similarity completion problem to be solved, this also presented in the next section. In Section 3, details of the developed method are presented. Thetwo example problems are presented in Section 4, and in the final section, some conclusions are drawn.

2 Preliminary concepts and theory

2.1 Multi-objective optimization

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 optimization problem:

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

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 optimization is known as a Pareto-optimal solution (Pareto 1906). 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.

A point $p^* \in \mathcal{P}$ is defined as being Pareto-optimal if and only if there exists no other point $p \in \mathcal{P}$ such that $\phi_i(p) \leq \phi_i(p^*) \forall i = 1, \dots, n$ and $\phi_j(p) < \phi_j(p^*)$ for at least one j.

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, cross-over 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 cross-over 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 cross-over, the fitness being a function of the function which is being minimized. By means of these operations, the population will evolve towards a near-optimal solution.

GA's are very-well suited for multi-objective optimization problems. The approach used here is the Multi-Objective Genetic Algorithm (MOGA) (Fonseca & Fleming 1993*a*, Fonseca & Fleming 1993*b*, Fonseca & Fleming 1995), which is an extension on an idea by Goldberg (1989). The idea behind the MOGA is to develop a population of Pareto-optimal or near Pareto-optimal solutions. This is achieved by finding a set of solutions which are non-dominated. An individual j with a set of objective functions $\phi^j = (\phi^j_1, \dots, \phi^j_n)$ is said to be non-dominated if for a population of N individuals, there are no other individuals $k = 1, \dots, N, k \neq j$ such that $\phi^k_i \leq \phi^j_i \forall i = 1, \dots, n$ and $\phi^k_i < \phi^j_i$ for at least one i. With the MOGA, non-dominated 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 non-dominated, near Pareto-optimal individuals. Details of this mechanism are given in Fonseca & Fleming (1995).

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 non-dominated individuals is fairly uniform. A common mechanism to ensure this is fitness sharing (Fonseca & Fleming 1995), 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, 2 individuals may have the same genotype, but different gene strings. Thus it difficult to measure the difference between 2 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.

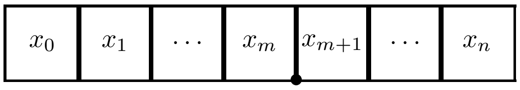

Figure 2: FWL fixed-point representation

A typical 2's complement FWL fixed-point representation of a number $q(x)$ is shown in Figure 2. The number $q(x)$ is represented by an $m+n+1$ binary string x where $x = [x_0, x_1 \dots, x_m, x_{m+1}, \dots, x_{m+n}]$, $x_i \in \{0,1\}$, $m \in \{0,1,2,\dots\}$, $n \in \{0,1,2,\dots\}$. The value $q(x)$ is given by

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

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

2.3 A linear system equivalence completion problem

The following theorem is required.

Theorem 1 *Given a 2 state SISO LTI system $F(z)$, where*

$$F \overset{s}{=} \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \qquad (3)$$

$$\overset{s}{=} \left[\begin{array}{cc|c} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ \hline c_1 & c_2 & d \end{array}\right], \qquad (4)$$

then (provided certain rank and singularity conditions are satisfied) 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

$$\tilde{F} \overset{s}{=} \left[\begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & \tilde{D} \end{array}\right] \qquad (5)$$

$$\overset{s}{=} \left[\begin{array}{cc|c} \tilde{a}_{11} & \tilde{a}_{12} & \tilde{b}_1 \\ \tilde{a}_{21} & \tilde{a}_{22} & \tilde{b}_2 \\ \hline \tilde{c}_1 & \tilde{c}_2 & \tilde{d} \end{array}\right], \qquad (6)$$

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

$$F(z) = C\left[zI - A\right]^{-1} B + D \qquad (7)$$

and

$$\tilde{F}(z) = CT\left[zI - TAT^{-1}\right]^{-1} T^{-1}B + D, \qquad (8)$$

where

$$T = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \qquad (9)$$

is non-singular.

Proof: The proof is by construction. From (8),

$$\tilde{A} = TAT^{-1}, \qquad (10)$$

so by a property of similar matrices (Nering 1990, p.115), $\operatorname{tr}\tilde{A} = \operatorname{tr}A$. Hence, given $\tilde{a}_{11}, \tilde{a}_{12}$, $\tilde{a}_{22}$ is given by $\tilde{a}_{22} = a_{11} + a_{22} - \tilde{a}_{11}$. Similarly, from the property of similar matrices that $\det\tilde{A} = \det A$ (by definition), given $\tilde{a}_{12} \neq 0$, $\tilde{a}_{21}$ is given by $\tilde{a}_{21} = (\tilde{a}_{11}\tilde{a}_{22} - \det A)/\tilde{a}_{12}$. From (10), $AT - T\tilde{A} = 0$, which gives (Horn & Johnson 1991, p. 255)

$$\left[I \otimes A - \tilde{A}^T \otimes I\right]\mathbf{t} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad (11)$$

where $\mathbf{t} = [t_{11}, t_{21}, t_{12}, t_{22}]^T$ and $[I \otimes A - \tilde{A}^T \otimes I]$ is rank 2. Now, from (8), $CT = \tilde{C}$. Hence, given $\tilde{c}_1, \tilde{c}_1$,

$$\begin{bmatrix} c_1 & c_2 & 0 & 0 \\ 0 & 0 & c_1 & c_2 \end{bmatrix}\mathbf{t} = \begin{bmatrix} \tilde{c}_1 \\ \tilde{c}_1 \end{bmatrix}. \qquad (12)$$

A $Y \in \mathbb{R}^{4\times4}$ can be constructed from (11) and (12) where $Y\mathbf{t} = z$, and where $z \in \mathbb{R}^4$ is a column vector with 2 elements of $\tilde{C}$ and two zero elements. If Y is non-singular rank 4, T and hence $\tilde{B}$ can be calculated. From (8), $\tilde{d} = d$. ■■

Note that by redefining the transformation matrix as T^{-1}, the problem given $\tilde{B}$ instead of $\tilde{C}$ can be solved.

3 MOGA for optimal FWL controller structures

3.1 Procedure outline

The problem can be summarized as follows. 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 parameterization such that the difference between the closed loop system and the original closed loop system and the cost of implementation are simultaneously minimized.

For this paper, two indices are defined for the MOGA. The first index, ϕ_1, is some closed loop measure of the implementation accuracy of the FWL controller compared to the original controller. The second index, ϕ_2, is some measure of the cost of implementing the FWL controller.

The main elements of the approach are now summarized.

i) Generate partially filled random FWL parameterizations of the 2 state controller i.e.

$$\tilde{K} \overset{s}{=} \left[\begin{array}{cc|c} q_1 & q_2 & ? \\ ? & ? & ? \\ \hline q_3 & q_4 & D_K \end{array}\right] \qquad (13)$$

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.

$$K_q \stackrel{s}{=} \left[\begin{array}{cc|c} q_1 & q_2 & q_7 \\ q_5 & q_6 & q_8 \\ \hline q_3 & q_4 & q_9 \end{array}\right] \tag{14}$$

where $q_j \in \mathcal{Q}_{n,m}, j = 1, \ldots, 9$.

iv) Calculate the implementation accuracy index, ϕ_1, and the implementation cost index, ϕ_2, for K_q.

v) Use the MOGA to evolve a set of near Pareto-optimal solutions.

3.2 Encoding of solution space

In order to generate the partially filled parameterizations of K given by (13), 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 $n = \max(\sum_{i=4}^{7} x_i 2^{i-4}, 15 - m)$. 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 (2).

4 Applications

4.1 Steel rolling mill system

The proposed approach is applied to a PID controller of steel rolling mill (Hori 1996). Note that data given in this section are shown to only 4 decimal places.

4.1.1 Performance indices

For this example, the measure of the difference between the FWL implemented closed loop system and the original closed loop system is based on the difference between the closed loop pole positions of the systems.

Consider the discrete time system shown in Figure 1. Let (A_g, B_g, C_g, D_g) be a state-space description of the SISO plant $G(z) = C_g(zI - A_g)^{-1}B_g + D_g$, where $A_g \in \mathbb{R}^{m\times m}$, $B_g \in \mathbb{R}^{m\times 1}$, $C_g \in \mathbb{R}^{1\times m}$ and $D_g \in \mathbb{R}$. Let (A_k, B_k, C_k, D_k) be a state-space description of the 2 state controller, $K(z) = C_k(zI - A_k)^{-1}B_k + D_k$, where $A_k \in \mathbb{R}^{2\times 2}$, $B_k \in \mathbb{R}^{2\times 1}$, $C_k \in \mathbb{R}^{1\times 2}$ and $D_k \in \mathbb{R}$. The transition matrix of the closed loop system is $\bar{A} =$

$$\begin{bmatrix} A_g + B_g D_k (I - D_g D_k)^{-1} C_g & B_g (I - D_k D_g)^{-1} C_k \\ B_k (I - D_g D_k)^{-1} C_g & A_k + B_k (I - D_k D_g)^{-1} D_g C_k \end{bmatrix}.$$

When (A_k, B_k, C_k, D_k) is implemented with a finite word-length device, the elements are rounded to the realization $K_q = (A_q, B_q, C_q, D_q)$, that is $K(z)$ is perturbed to $K_q(z)$ where $K_q(z) = C_q(zI - A_q)^{-1}B_q + D_q$ and where $A_q \in \mathcal{Q}_{n,m}^{2\times 2}$, $B_q \in \mathcal{Q}_{n,m}^{2\times 1}$, $C_q \in \mathcal{Q}_{n,m}^{1\times 2}$ and $D_q \in \mathcal{Q}_{n,m}$. The transition matrix of the FWL implemented closed loop system is thus $\bar{A}_q =$

$$\begin{bmatrix} A_g + B_g D_q (I - D_g D_q)^{-1} C_g & B_g (I - D_q D_g)^{-1} C_q \\ B_q (I - D_g D_q)^{-1} C_g & A_q + B_q (I - D_q D_g)^{-1} D_g C_q \end{bmatrix}.$$

A closed loop measure, η, of the accuracy of the FWL implementation is taken as

$$\eta(K_q) \triangleq \max_{i \in \{1, \ldots, m+2\}} \left| \lambda_i - \lambda_i^Q \right| \tag{15}$$

where $\{\lambda_i : i = 1, \ldots, m+2\}$ represents the set of all eigenvalues of $\bar{A}$ and $\{\lambda_i^Q : i = 1, \ldots, m+2\}$ the corresponding set of all eigenvalues of $\bar{A}_q$.

Two objective functions, $\phi_1(x), \phi_2(2)$ are defined. A measure of implementation accuracy (if the implemented controller and system is stable) is defined as

$$\phi_1 = \begin{cases} \eta(K_q) & \text{if } \left|\lambda_i^K\right| \le 1 \forall i \\ & \text{and } \left|\lambda_i^Q\right| < 1 \forall i, \\ \max_i \left|\lambda_i^K\right| & \text{if } \left|\lambda_i^K\right| > 1 \exists i, \\ \\ \max_i \left|\lambda_i^Q\right| & \text{if } \left|\lambda_i^K\right| \le 1 \forall i \\ & \text{and } \left|\lambda_i^Q\right| \ge 1 \exists i \end{cases} \tag{16}$$

where $\{\lambda_i^K : i = 1, 2\}$ represents the pair of eigenvalues of A_q and $\{\lambda_i^Q : i = 1, \ldots, m+2\}$ represents the set of all eigenvalues $\bar{A}_q$. 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 (2).

4.1.2 Model

The continuous time linearized ideal two-mass inertia model of the system plant, $G(s) = C(zI - A)^{-1}B$, is given as

$$A = \begin{bmatrix} 0 & -9763.7203 & 0 \\ 1 & 0 & -1 \\ 0 & 13424.9859 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 249.03 \\ 0 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

The plant, $G(s)$, is discretized with sampling period of 0.001 seconds to give $G(z) = C_g(zI - A_g)^{-1}B_g$ as

$$A_g = \begin{bmatrix} 0.9951 & -9.7260 & 0.0049 \\ 0.0010 & 0.9884 & -0.0010 \\ 0.0067 & 13.3732 & 0.9933 \end{bmatrix},$$

$$B_g = \begin{bmatrix} 0.2486 \\ 0.0001 \\ 0.0006 \end{bmatrix}, C_g = [\,1 \quad 0 \quad 0\,].$$

4.1.3 Controller

A PID vibration suppression and disturbance rejection controller is designed as

$$K(s) = \frac{0.00269s}{0.001s+1} - 0.435 - \frac{14.26}{s}.$$

The bilinear transform,

$$s = \frac{2}{h}\frac{1-z^{-1}}{1+z^{-1}},$$

gives the digital PID controller as

$$K(z) = -\frac{0.01426}{z-1} - \frac{1.1956}{z-0.3333} + 1.3512.$$

The initial realization K^0 is set to

$$A_k^0 = \begin{bmatrix} 1 & 0 \\ 0 & 0.3333 \end{bmatrix}, B_k^0 = \begin{bmatrix} -1 \\ -1 \end{bmatrix},$$

$$C_k^0 = [\,0.01426 \quad 1.1956\,], D_k^0 = 1.3512.$$

4.1.4 Design

The MOGA is implemented in MATLAB using the GA Toolbox (Chipperfield, Fleming, Pohlheim & Fonseca 1994). An elitism strategy is used whereby all non-dominated 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 cross-over 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 400 generations (which takes about 1 hour on a 450 MHz Pentium II), a set of non-dominated solutions is obtained. This set is shown in Figure 3. Note that the axis for ϕ_1 is shown to log scale for clarity. The figure clearly shows the trade-off between the memory requirements and the stability degradation.

4.2 IFAC93 benchmark design

The proposed approach is applied to a PID controller (Whidborne et al. 1995) designed for the IFAC93 benchmark problem (Graebe 1992).

4.2.1 Performance indices

For this example, the 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

$$\phi_1 = \|\, R - R_q \,\|_\infty \tag{17}$$

where

$$R = \frac{GK}{1+GK} \text{ and } R_q = \frac{GK_q}{1+GK_q}. \tag{18}$$

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 (2).

4.2.2 Model

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

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

The plant, $G(s)$, is discretized with sampling period of $t_s = 0.05$ seconds, which gives $G(z) = C_g(zI - A_g)^{-1}B_g$ as

$$A_g = \begin{bmatrix} 0.82342 & -1.1761 & -0.22856 \\ 0.045713 & 0.9697 & -0.0058983 \\ 0.0011797 & 0.049488 & 0.9999 \end{bmatrix}$$

$$B_g = \begin{bmatrix} 0.045713 \\ 0.0011797 \\ 1.9963e-005 \end{bmatrix}, C_g = [\,0 \quad -2 \quad 5\,].$$

4.2.3 Controller

A PID controller is designed as (Whidborne et al. 1995)

$$K(s) = 1.311 + 0.431/s + 1.048s/(1+12.92s),$$

and discretized using the bilinear transform. The initial realization K^0 is set to

$$A_k^0 = \begin{bmatrix} 1 & -0.99614 \\ 0 & 1.99614 \end{bmatrix}, B_k^0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$C_k^0 = [\,0.049578 \quad 0.049469\,], D_k^0 = 1.41693.$$

4.2.4 Design

The MOGA is implemented in MATLAB using the GA Toolbox (Chipperfield et al. 1994). An elitism strategy is used whereby all non-dominated individuals propagate through to the next population. Selection is performed using stochastic

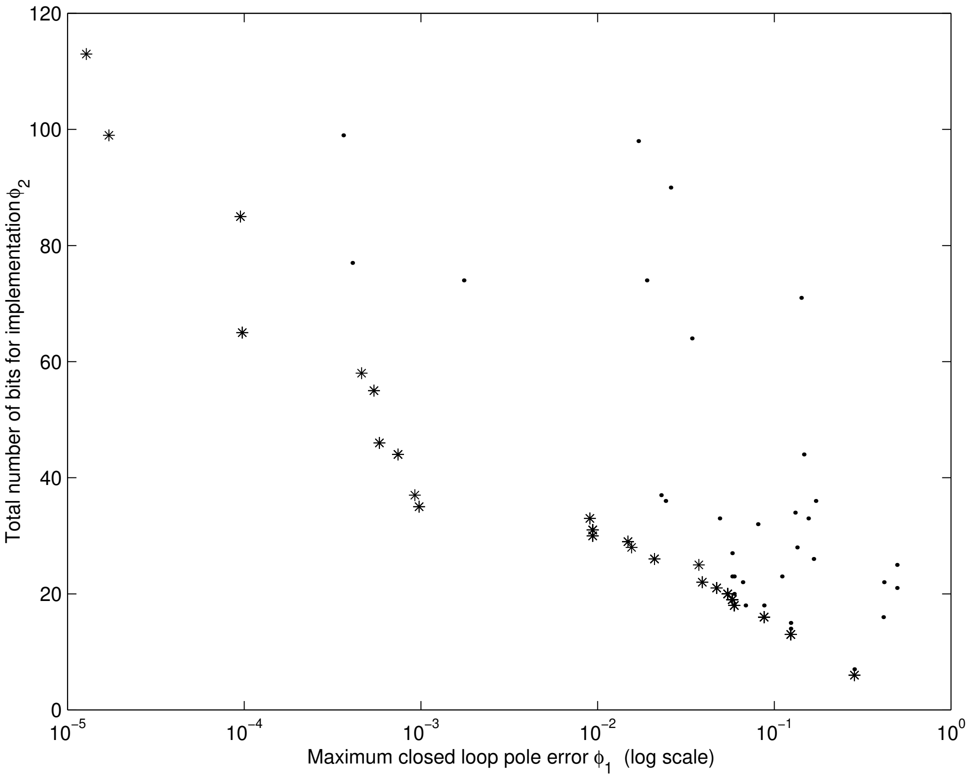

Figure 3: Solution set showing trade-off with MOGA non-dominated set ($*$) and MOGA dominated set ($\cdot$)

universal sampling with a fitness determined by the number of dominating individuals. Single point cross-over 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 non-dominated solutions is obtained. This set is shown in Figure 4. The figure also shows the stable dominated solutions along with FWL implementations of the initial realization, K_q^0, for various word-lengths. In addition, the diagonal equivalent balanced realization of $K(z)$ is calculated using the `ssbal` function from the MATLAB Control System Toolbox (Mat 1998, pp 9/200–201), and FWL implementations of the realization for various word-lengths are shown. Note that the axis for $\| R - R_q \|_\infty$ is shown to log scale for clarity. The figure clearly shows that improved FWL state-space realizations for the controller can be obtained using the approach.

The 45 bit solution labelled (1) in Figure 4 is selected and is the controller

$$K_q \stackrel{s}{=} \left[\begin{array}{cc|c} 0.99609375 & 2^{-7} & -1.5859375 \\ 0 & 1 & -0.3515625 \\ \hline -2^{-5} & 0 & 1.41796875 \end{array} \right],$$

which requires $m = 1$ and $n = 8$ for the FWL representation given by (2). In addition, one of the parameters is zero, another is 1, and 2 others are powers of 2, and can hence be implemented by register shifts. Figure 5 shows the frequency response of the original closed loop transfer function $\left|R(e^{j\omega t_s})\right|$ and the difference between the systems $\left|R(e^{j\omega t_s}) - R_q(e^{j\omega t_s})\right|$.

5 Conclusion

A GA-based method to determine optimal FWL structures for PID digital controllers has been developed. The method is illustrated by a practical 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 similarity completion problem. A solution to the linear system similarity completion problem for 2 state SISO systems has been presented. This solution would need to be extended to general linear systems for the methodology to be extended to higher order controllers. Otherwise, the method is entirely generic, and any computable set of stability and performance measures could be included. In addition, other implementation measures could be included, such as computation time, as well as measures of other important quantization effects, namely quantization noise and scaling requirements.

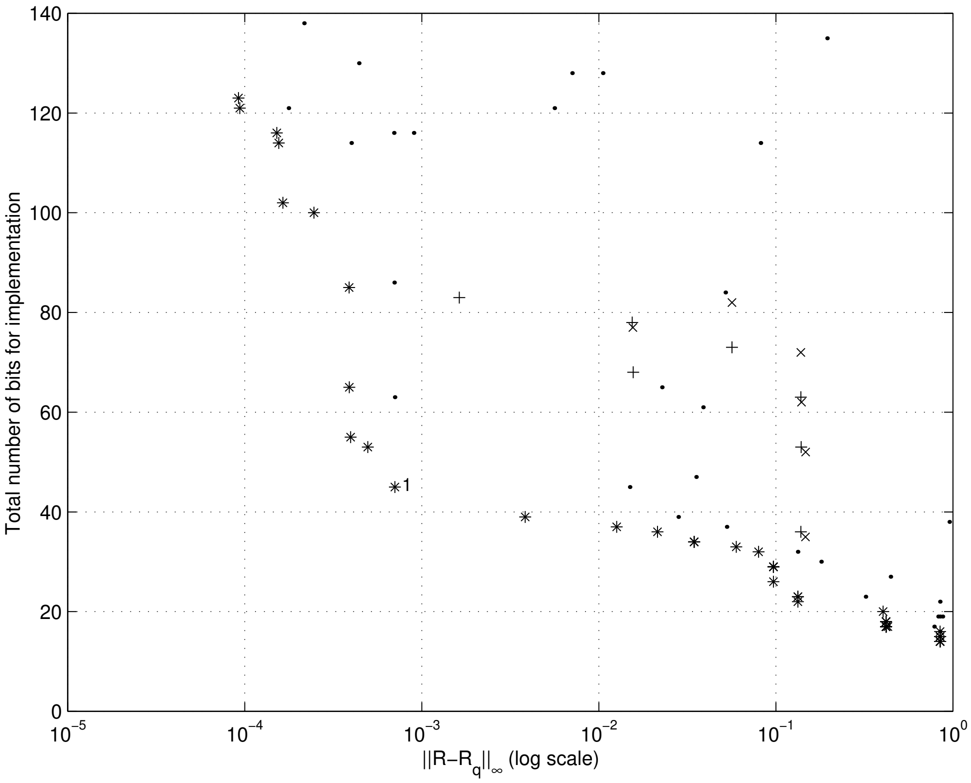

Figure 4: Solution set showing trade-off with MOGA non-dominated set ($*$), MOGA dominated set ($\cdot$), K_q^0 realizations ($\times$) and equivalent balanced realizations ($+$). The selected controller is marked 1.

Bibliography

Chipperfield, A., Fleming, P., Pohlheim, H. & Fonseca, C. (1994), *Genetic Algorithm Toolbox: User's Guide*, Dept. Automatic Control and Systems Engineering, University of Sheffield, U.K.

Dakev, N., Whidborne, J., Chipperfield, A. & Fleming, P. (1997), 'H_∞ design of an EMS control system for a maglev vehicle using evolutionary algorithms', *Proc. IMechE, Part I: J of Sys and Contr.* **311**(4), 345–355.

Fonseca, C. & Fleming, P. (1993*a*), Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, *in* 'Genetic Algorithms: Proceeding of the Fifth International Conference', San Mateo, CA, pp. 416–423.

Fonseca, C. & Fleming, P. (1993*b*), Multiobjective genetic algorithms, *in* 'IEE Colloquium on Genetic Algorithms for Control Systems Engineering', number 1993/130, London, England, pp. 6/1–6/5.

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

Gevers, M. & Li, G. (1993), *Parametrizations in Control, Estimations and Filtering Problems: Accuracy Aspects*, Springer-Verlag, Berlin.

Goldberg, D. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.

Graebe, S. (1992), 'Benchmark IFAC 93: Adaptive/robust control of unknown plant', Pre-conference communication. Centre for Industrial Control Science, University of Newcastle, Australia.

Hori, Y. (1996), A review of torsional vibration control methods and a proposal of disturbance observer based new techniques, *in* 'Proc. 13th IFAC World Congress', San Fransisco, CA, pp. 7–13.

Horn, R. & Johnson, C. (1991), *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, U.K.

Istepanian, R., Wu, J., Whidborne, J. & Chu, J. (1998), Maximizing lower bound stability measure of finite precision PID controller realizations by nonlinear programming, *in* 'Proc. 1998 Amer. Contr. Conf.', Philadelphia, PA, pp. 2596–2600.

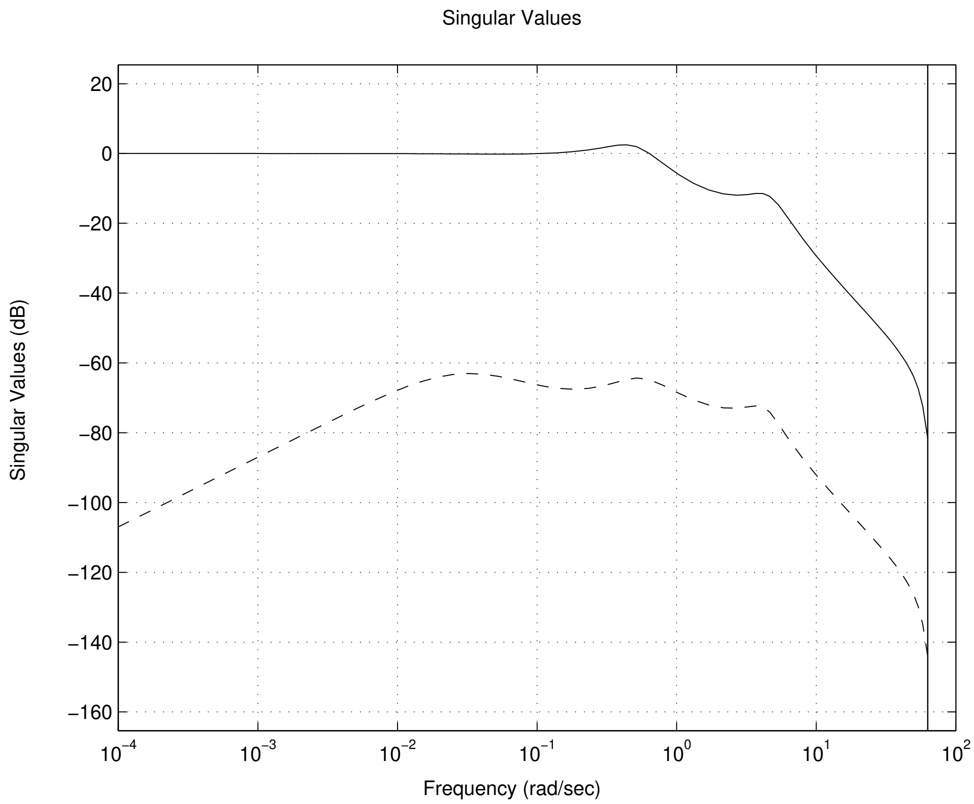

Figure 5: Singular values of R (—) and $R - R_q$ (- - -)

Knowles, J. & Edwards, R. (1965), 'Effect of a finite-word-length computer in a sampled-data feedback system', *Proc. IEE* **112**(6), 1197–1207.

Man, K., Tang, K., Kwong, S. & Halang, W. (1997), *Genetic Algorithms for Control and Signal Processing*, Advances in Industrial Control, Springer, London.

Masten, M. & Panahi, I. (1997), 'Digital signal processors for modern control systems', *Control Engg Practice* **5**(4), 449–458.

Mat (1998), *Control System Toolbox: User's Guide, version 4*.

Morony, P. (1983), *Issues in the Implementation of Digital Feedback Compensators*, number 5 *in* 'Signal Processing, Optimization, and Control Series', MIT Press, Cambridge, MA.

Nering, E. (1990), *Linear Algebra and Matrix Theory*, 2 edn, John Wiley, New York.

Pareto, V. (1906), *Manuale di Economia Politica*, Societa Editrice Libraria, Milan, Italy.

Tang, K., Man, K. & Gu, D.-W. (1996), 'Structured genetic algorithm for robust H^∞ control system design', *IEEE Trans. Industrial Electronics* **43**(15), 575–582.

Whidborne, J., Murad, G., Gu, D.-W. & Postlethwaite, I. (1995), 'Robust control of an unknown plant – the IFAC 93 benchmark', *Int. J. Control* **61**(3), 589–640.

Williamson, D. (1991), *Digital Control and Implementation: Finite Wordlength Considerations*, Systems and Control Engineering, Prentice Hall, Englewood Cliffs, NJ.

Xu, D. & Daley, M. (1995), 'Design of optimal digital filter using a parallel genetic algorithm', *IEEE Trans. Circuits Syst.: II* **42**(10), 673–675.