International Journal of Automation and Computing, Vol. 3 No.3, July 2006 pg 314-322

Kolmogorov-Chaitin Complexity of Digital Controller Implementations

James F. Whidborne* John McKernan
Department of Aerospace Sciences, Cranfield University, Bedfordshire MK43 0AL, U.K.
Da-Wei Gu
Department of Engineering, University of Leicester, Leicester LE1 7RH, U.K.

Abstract

The complexity of linear, fixed-point arithmetic digital controllers, is investigated from a Kolmogorov-Chaitin perspective. Based on the idea of Kolmogorov-Chaitin complexity, practical measures of complexity are developed for state-space realizations, parallel and cascade realizations, and for a newly proposed generalized implicit state-space realization. The complexity of solutions to a restricted complexity controller benchmark problem is investigated using this measure. The results show that from a Kolmogorov-Chaitin viewpoint, higher-order controllers with a shorter word-length may have lower complexity and better performance, than lower-order controllers with longer word-length.

Keywords

Controller complexity, finite-precision arithmetic, finite word length, digital controller, Kolmogorov-Chaitin complexity

1 Introduction

Modern control design methods usually result in digital controllers that are high-order and hence regarded as complex, and thus costly and difficult to implement. Furthermore, rounding errors occur in digital controller implementations, and these errors become magnified with controller order. Designers then often move to more expensive longer word-length or floating point arithmetic to overcome these problems. Control system designers are thus motivated to produce low-order controllers which generally have lower complexity.

However controller order, whilst a good indicator of controller complexity, may not be an accurate measure of actual implemented controller complexity. Consider the following two digital controllers:

$$K_1(z) = \frac{1}{(z+1)^3} \text{ and } K_2(z) = \frac{1}{z+0.1} \qquad (1)$$

Controller K_1 has order 3, and controller K_2 has order 1. Hence K_2 would normally be regarded as having lower complexity than K_1. However, controller K_1 can be implemented exactly without any multiplication operations and just 3 addition operations. Controller K_2 requires a multiplication operation, and can only be implemented exactly with an infinite word-length. This is because the multiplicand 0.1 in base-10 cannot be represented exactly by a finite word-length binary string, since it recurs in base-2 as $0.000110011\cdots$. Clearly then, K_2 has higher complexity than K_1. Furthermore, for a practical implementation a perturbation is introduced into controller K_2 of a magnitude determined by the implementation word-length and controller realization. Note that in the case of a digital filter implementation, it has long been recognized that increasing filter order may decrease complexity [1, 2, 3, 4, 5].

Thus, it is not sufficient to consider just controller order, but required word-length and arithmetic operations as other factors that determine controller complexity. Williamson [6] has considered complexity in terms of the number of multiplications, and less importantly the number of additions required for an implementation. Another measure was introduced in [7] based on the number of bits required to store controller parameters. This measure seems related to the idea of Kolmogorov-Chaitin complexity [8, 9], whereby the complexity of a computable object is measured by the length in bits of the shortest program that computes the object. The measure of [7] is also considered in [10, 11], and applied to the EJC restricted complexity controller benchmark problem [12, 13].

Clearly, using floating-point instead of fixed-

*Corresponding Author. E-mail address j.f.whidborne@cranfield.ac.uk

point arithmetic, increases the complexity of controller implementation. Furthermore, fixed-point implementations are much cheaper and faster. Therefore in this paper, fixed-point implementations are considered. A number $q(x)$ in a two's complement fixed-point representation is typically given by a $m+n+1$ binary string x, where $x = |x_0, x_1 \ldots, x_m, x_{m+1}, \ldots, x_{m+n}|$, $x_i \in \{0,1\}$, $m \in \{0,1,2,\ldots\}$, $n \in \{0,1,2,\ldots\}$. The value $q(x)$ is given by:

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

Whilst there have been few attempts to quantify complexity in feedback controllers, the problem of designing controllers and filters for implementation using finite-word-length arithmetic has been extensively studied; particularly for (open-loop) filters. Early analysis of finite-word effects can be found in [14, 15, 16, 17]. What became evident early on, is that effects are very dependent upon the particular realization of a filter/controller [18]; and there has subsequently been much work on optimizing controller/filter realization so as to minimize word-length (e.g. [19, 20, 21, 22, 23, 24]). Comprehensive accounts include a text by Morony [25], and more recent work by Williamson [6], Gevers and Li [26], and Istepanian and Whidborne [27]. It is obvious that realizations which have a small number of parameters are preferred. If however, the realization is such that many of the parameters are trivial, i.e. 0 or ± 1, then this leads to a simplification of the implementation, and the so-called sparse optimal realization problem. Procedures to minimize both word-length and the number of non-trivial parameters have been developed. One noted procedure that has been used for controllers is based on Chan [28], and has undergone a number of developments for the sparse controller realization problem [29, 30, 26, 31, 32]. For filters, there has been a large amount of research over the years (e.g. [33, 34, 35, 36, 37]).

The objective of this paper is hence to consider the complexity of linear, fixed-point arithmetic digital controllers, from a Kolmogorov-Chaitin perspective. Note that practical measures based on the concept of Kolmogorov-Chaitin computational complexity have been developed in other application fields; notably for the complexity of identification models [38, 39], and for the complexity of neural network implementations [40]. In this paper, a practical measure of complexity is defined for linear time-invariant controllers, based on the idea of Kolmogorov-Chaitin complexity. The use of this measure is considered for state-space realizations, for parallel and cascade realizations, and for a generalized implicit state-space realization [41]. The complexity of designs for the restricted complexity controller benchmark problem is investigated. It is seen that from a Kolmogorov-Chaitin viewpoint, higher-order controllers with a shorter word-length may have lower complexity and better performance, than lower-order controllers with longer word-length. Note that a shortened version of this paper appeared in [42].

2 Measures of complexity

Digital controllers are implemented as a computer program, hence it is of interest to investigate whether complexity measures used for software are applicable for measuring digital controller complexity.

One traditional measure of computational complexity is time complexity, i.e. an estimate of the amount of time (or number of machine cycles) that is required to complete an algorithm. Digital controllers are time critical, and there is therefore usually a computational time constraint on the algorithm; however this has not been explicitly considered in this paper.

2.1 Kolmogorov-Chaitin complexity

A fundamental concept in information and computation theory is that of Kolmogorov-Chaitin complexity. Put simply, Kolmogorov-Chaitin computational complexity [8, 9] measures the complexity of a computable object by the length in bits of the shortest program that computes the object.

The following definition is from [43, p 147], and refers explicitly to the computational complexity of a binary string.

Definition 1. *The Kolmogorov complexity $\Gamma_{\mathcal{U}}(x)$ of a computable finite-length binary string x, with respect to a universal Turing machine $\mathcal{U}$ is defined as:*

$$\Gamma_{\mathcal{U}}(x) = \min_{p:\mathcal{U}(p)=x} \ell(p), \qquad (3)$$

where p denotes a program executed by $\mathcal{U}$, and $\ell(p)$ denotes the description length of p.

Of course, digital controllers are not implemented on universal Turing machines. The following theorem [43, p 147] relates Kolmogorov complexity to length complexity using some other computing machine.

Theorem 1. *If $\mathcal{U}$ is a universal computer, then for any other computer $\mathcal{A}$:*

$$\Gamma_{\mathcal{U}}(x) \leq \Gamma_{\mathcal{A}}(x) + c_{\mathcal{A}} \tag{4}$$

for all binary strings x where the constant $c_{\mathcal{A}}$ does not depend on x.

For a digital controller, at each sample period the digital computer calculates a binary string $u(v)$ (the control), which is dependent on another binary string v (a concatenation of binary strings representing the measurement, controller states, and other variables that are stored in the computer). What is actually needed is a measure of the complexity of the controller implementation for all possible measurements and relevant controller states, and hence all controls. The controller is a mapping from $\mathbf{V}$ (the set of all possible v), to $\mathbf{U}$ (the set of all possible u). A practical complexity measure consists of the sum of the minimum length of the stored program for all possible v, and the maximum length of v. This gives a measure of the total storage requirements for the program, input data, and stored data. Based on this, the following complexity measure is proposed.

Definition 2. *The complexity Γ of a controller implementation, with respect to a controller implementation machine $\mathcal{K}$, is defined as:*

$$\begin{aligned}\Gamma = &\max\{\ell(v) : v \in \mathbf{V}\} \\ &+ \min\{\ell(p) : \mathcal{K}(p) = u(v), \text{ for all } v \in \mathbf{V}\},\end{aligned} \tag{5}$$

where p denotes a controller program executed by $\mathcal{K}$, $\ell(p)$ is the length in bits of the executed program, $\ell(v)$ is the length in bits of the internal and measurement variables v, and $\mathbf{V}$ is the set of all possible v.

For a linear time-invariant controller K, the controller complexity definition can be simplified. Consider for example a state-space controller with n_x state variables, n_y inputs, and n_u outputs; where at the ith time step:

$$x(i+1) = Ax(i) + By(i), \tag{6}$$

$$u(i) = Cx(i) + Dy(i). \tag{7}$$

The string v has a constant length dependent on the chosen word-length, number of states n_x, and the number of measurement inputs n_y. The program is independent of u, y, and x, hence the program length $\ell(p)$ is independent of v and u. Let k be the concatenated binary string of the parameters of the controller. If the chosen word-length is w, then the length of k is given by $\ell(k) = (n_x + n_y)(n_x + n_u)w$, and the length of v given by $\ell(v) = (n_x + n_y)w$; therefore the required data storage of the state-space controller is $\ell(k) + \ell(v)$. However, this is not a satisfactory complexity measure for state-space controllers, because it does not consider the possibility of sparse realizations. Sparse realizations are where some controller parameters are zero or unity. These reduce program complexity because there is no need for a multiplication in the case of unity, or an addition in the zero case.

Therefore, in order to be able to use Definition 2 to measure controller complexity, the length of program $\ell(p)$ needs to be determined. This requires some practical consideration of the implementation machine $\mathcal{K}$. For example, consideration needs to be made whether $\mathcal{K}$ has hardware support for floating point calculations, or for multiplication operations, and if so, what length "overhead" is given for the use of more complex operations over simpler operations such as addition or barrel shift.

Very simple computing devices are often preferred for reasons of cost and reliability. Such simple devices use fixed point arithmetic and implement multiplication in software via a sequence of additions and shifts, i.e. one addition, and one barrel shift, for each unity bit of the constant multiplicand. Thus, a simple complexity measure is to count the number of unity bits in each multiplicand string. Consider for example $2.125 \times y$. The binary representation of 2.125 is 10.001_2. This is equivalent to $2\times y + 1/8\times y$, which is evaluated by $\overset{1}{\leftarrow} y + \overset{3}{\rightarrow} y$ where $\overset{n}{\rightarrow}$ denotes an n right shift operation, etc. Therefore, the following complexity measure is proposed.

Definition 3. *The controller complexity $\Gamma(K)$ of a linear time-invariant controller realization K is given by:*

$$\Gamma(K) = \ell(v) + \sum_{i=1}^{\ell(k)} k_i, \tag{8}$$

where the concatenated binary string of the parameters of K, is given by $k = |k_1, k_2, \ldots, k_{\ell(k)}|$, $k_i \in \{0, 1\}$.

This measure incorporates the measure of the length of the program, in terms of the number of operations, and in terms of data storage requirements.

2.2 State-space realizations

A common and fairly general representation of a system, is a state-space representation. This real-

ization is simple to program, and has a clear structure that is easy to analyze. Here, linear time-invariant controllers are considered with a state space representation $K_s = (A, B, C, D)$, where at the ith time step:

$$x(i+1) = Ax(i) + By(i) \tag{9}$$

$$u(i) = Cx(i) + Dy(i) \tag{10}$$

where $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_y}$, $C \in \mathbb{R}^{n_u \times n_x}$ and $D \in \mathbb{R}^{n_u \times n_y}$. Let k be the concatenated binary string of the parameters of K_s. If chosen word-length is w, then the length of k is given by $\ell(k) = (n_x + n_y)(n_x + n_u)w$. Let v be a binary string representation of (x^T, y^T), and the length of v be given by $\ell(v) = (n_x + n_y)w$.

Note that the time complexity of K_s, is independent of x and y, and hence of v. At each sampling instant, the time to evaluate (10) should be minimized, and (9) must just be evaluated before the next sample.

2.3 Parallel and cascade realizations

A state space realization is over-parameterized, in that the number of parameters of K_s is not minimal. However, the number of internal variables is minimal. In this section, two common minimal parameter realizations are considered, namely parallel and cascade forms.

Parallel and cascade realizations decompose a controller into a number of first and second-order direct-form filter blocks (see [44, pp 120-123]). The structure of a second-order direct form filter block is shown in Fig. 1, and a transfer function given by:

$$F(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}. \tag{11}$$

The transfer function of a first-order direct-form filter block is similarly given by:

$$F(z) = \frac{a_0 + a_1 z^{-1}}{1 + b_1 z^{-1}}. \tag{12}$$

The parallel structure used here is (for SISO controllers):

$$K_p(z) = \sum_{i=1}^{m} F_i(z) \tag{13}$$

where each F_i is a first or second-order transfer function of the form given by (11) or (12). The cascade structure is given by:

$$K_c(z) = \prod_{i=1}^{m} F_i(z). \tag{14}$$

Assume that there are m_1 first-order, and m_2 second-order direct-form filter blocks, and that $m_2 > 0$. Let the concatenated binary string representation of the filter block parameters a_0, a_1, a_2, b_1, and b_2 for all filter blocks, be k. The length of k is therefore $\ell(k) = (3m_1 + 5m_2)w$, where w is word-length. Each second-order direct-form filter block shown in Fig. 1, requires two previous outputs and two previous inputs to be stored. But for a parallel structure SISO controller, input is common to all blocks, so the input and two previous sample inputs need only be stored once. The length of v is hence given by $\ell(v) = (m_1 + 2m_2 + 3)w$.

For a cascade structure controller, when second-order blocks are connected, the two previous outputs of the first block are equal to the two previous inputs of the second block (and similarly for first-order blocks). The measurement must also be stored, hence the length of v is also given by $\ell(v) = (m_1 + 2m_2 + 3)w$.

2.4 A generalized implicit state-space realization

There are many more possible realizations apart from the standard state-space parallel and cascade realizations studied above. Implementation of δ and γ-operators [45] for example, requires the storage of intermediate variables; and these add to complexity. Other common realizations include the direct form (obtained directly from the difference equation), and the lattice form. Most common realizations can be formulated using a generalized implicit state-space realization that has recently been proposed [41].

The generalized implicit state-space realization is given by $K_g = (E, F)$, where:

$$\begin{bmatrix} E_1 & 0 & 0 \\ -E_2 & I & 0 \\ -E_3 & 0 & I \end{bmatrix} \begin{bmatrix} z(i) \\ u(i) \\ x(i+1) \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{21} \\ F_{31} & F_{23} \end{bmatrix} \begin{bmatrix} y(i) \\ x(i) \end{bmatrix}, \tag{15}$$

where $E \in \mathbb{R}^{(n_z+n_u+n_x)\times n_z}$ and $F \in \mathbb{R}^{(n_z+n_u+n_x)\times(n_x+n_y)}$ are appropriately partitioned, E_1 is lower-triangular with 1's on the main diagonal, and $z(i)$ is a vector of intermediate variables of dimension n_z. The realization is implemented in a row-ordered manner, that is at each time step i, $z_1(i)$ is evaluated first, followed by $z_2(i)$, and so on. Because E_1 is lower triangular, $z_j(i)$ is not dependent on $z_k(i)$ for $j < k$. Note that the control $u(i)$ is evaluated before $x(i+1)$ so as to reduce computation lag time, and $x(i+1)$ can be evaluated after $u(i)$ is applied to the actuators and before the next sample time.

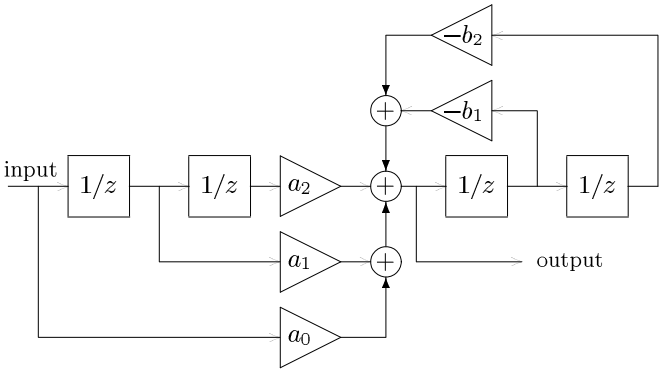


Figure 1: Second-order Direct-form Filter Block

Note that realization (15) is equivalent to a state-space realization (A, B, C, D), by:

$$A = E_3 E_1^{-1} F_{12} + F_{32} \tag{16}$$
$$B = E_3 E_1^{-1} F_{11} + F_{31} \tag{17}$$
$$C = E_2 E_1^{-1} F_{12} + F_{22} \tag{18}$$
$$D = E_2 E_1^{-1} F_{11} + F_{21}. \tag{19}$$

Examples of several common realizations in the form of (15), are give by Hilaire *et al.* [41].

Evaluating the complexity of (15) using Definition 3 is straightforward, noting that if the chosen word-length is w then $\ell(v) = (n_z + n_x + n_y)w$.

3 Restricted complexity controller benchmark

An active suspension system is the basis for the benchmark problem [12]. The objective is to design the simplest controller which meets some control specifications. The plant is a hydro-suspension system subjected to a disturbance. A block diagram of the active suspension system is shown in Fig. 2, where u_p denotes disturbance, u is the actuator input signal, y is system output (residual force), and p is the force resulting from the disturbance. Models for G_p (primary plant) and G (secondary plant) are provided. The sampling frequency is $F_s = 800$Hz.

The residual force y is measured, and fed back to the secondary plant via a digital controller K, as shown in Fig. 3. The reference signal r is generally zero. The objective of the control is to reduce the residual force y, at the first and second vibration modes of the primary plant G_p. There is additional constraint on actuator input u. In

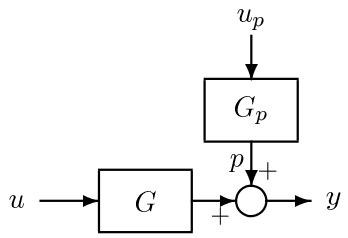


Figure 2: Suspension system

particular, controller gain should be zero at $F_s/2$, which is achieved by ensuring the controller includes the term $(z^{-1} + 1)$. Objectives can be achieved by ensuring that the output sensitivity function $S_{yp} = (1 + KG)^{-1}$, and the input sensitivity function $S_{up} = -K(1 + KG)^{-1}$, are constrained in the frequency domain, by the functions $F_{yp}(\omega)$ and $F_{up}(\omega)$, which are shown in Fig. 10. Two performance measures are therefore defined as:

$$\phi_y = \max_{\omega \in [0, \omega_s/2]} (|S_{yp}(j\omega)| - F_{yp}(\omega)) \tag{20}$$

and:

$$\phi_u = \max_{\omega \in [0, \omega_s/2]} (|S_{up}(j\omega)| - F_{up}(\omega)) \tag{21}$$

where $\omega_s = 2\pi F_s$ is the sampling frequency in rad/sec.

3.1 State-space realization controllers

The following definition is required.

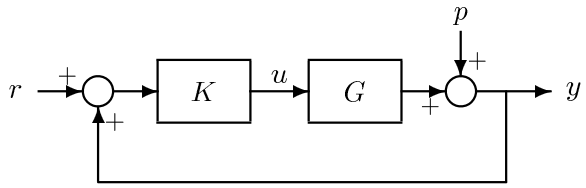


Figure 3: Closed loop suspension control system

Definition 4. *A solution j among a set of N solutions, with a set of objective function values $\phi^j = (\phi_1^j, \ldots, \phi_n^j)$, is said to be non-dominated if there is no other solution $k \in \{1, \ldots, N\}$, $k \neq j$, such that $\phi_i^k \leq \phi_i^j \; \forall \; i = 1, \ldots, n$ and $\phi_i^k < \phi_i^j$ for at least one i.*

A set of non-dominated state-space realization controllers that satisfy the criteria:

$$\phi_y(K_s) < 0 \quad \text{and} \quad \phi_u(K_s) < 0, \tag{22}$$

were obtained by means of a multi-objective genetic algorithm [46]. These solutions are an improvement over those given by [10].

The performance of the set is shown in Fig. 4, where the trade-off between ϕ_y and ϕ_u can be observed. The complexity of each solution is graded so that more complex solutions are shown larger, and each solution labelled with its word-length. Solutions marked ○ are 7th-order controllers, and there are 3 solutions that are 5th-order controllers, which are marked •. The superiority of the performance of the higher order controllers is clear. Figs 5 and 6 show ϕ_y against Γ, and ϕ_u against Γ respectively, for the set. The lower complexity of the lower-order controllers is less clear.

3.2 Parallel and cascade realization controllers

A set of parallel and cascade realization controllers that satisfy criteria (22), were obtained by [11], again using a multi-objective genetic algorithm [46].

The performance of this set is shown in Fig. 7, where the trade-off between ϕ_y and ϕ_u can be observed. The complexity of each solution is graded so that more complex solutions are shown larger, and each solution is labelled with its word-length. 5th-order controllers are marked •, the remaining solutions are 7th-order, and marked ○. The superiority of the performance of the higher-order controllers is again clear. Figs 8 and 9 show ϕ_y against Γ, and ϕ_u against Γ respectively, for the set. These figures show that lower complexity controllers are not necessarily lower-order controllers.

The lowest complexity controller that satisfied criteria (22) is selected for illustration. It is a 7th-order[1] cascade controller with an 8-bit word length, and is given exactly by:

$$\begin{aligned} K(z) = (1 + z^{-1}) &\left(\frac{1.5z^{-1} + 1.0625z^{-2}}{1 - 0.5703125z^{-1}} \right) \\ &\times \left(\frac{1 - 0.46875z^{-1} + 0.25z^{-2}}{1 - 0.3125z^{-1} - 0.6484375z^{-2}} \right) \\ &\times \left(\frac{0.0078125 + 0.0078125z^{-1}}{1 - 21875z^{-1} + 0.484375z^{-2}} \right). \end{aligned}$$

The closed loop system has $\phi_y = -0.0204$, $\phi_u = -0.6183$, and complexity $\Gamma = 110$. The output and input sensitivity functions are shown in Fig. 10, along with the performance constraints $F_{yp}(\omega)$ and $F_{up}(\omega)$.

4 Conclusions

This paper considers the complexity of controllers based on a fundamental concept of computational complexity, namely Kolmogorov-Chaitin complexity. From this notion, a practical measure of complexity is developed for both state-space realizations, and for parallel and cascade realizations. These measures are extended for a new generalized realization description. Controllers that are low-complexity in this sense will have low word-length, a low number of arithmetic operations, and simple realizations. These are the most important considerations for practical implementations. The measure is related to controller order, but since other factors are involved the lowest complexity controller does not necessarily have the lowest order. A higher-order controller provides an extra degree of freedom to help obtain controller realizations that have a shorter word-length and a simpler realization structure, and therefore a lower complexity. The limitation of the measure is that the designer needs to have decided upon the possible realizations and arithmetic before it can be used. Therefore, it is much easier for a designer to use just controller order as an indication of complexity, and consider implementation later. Details of methods to obtain realizations that have good numerical conditioning (hence low

[1] If fact, there is a zero parameter in one of the denominator coefficients, so it is actually 6th order. Inspection of the transfer function this reveals a common factor in one of the numerator terms, so the complexity could be reduced even further.

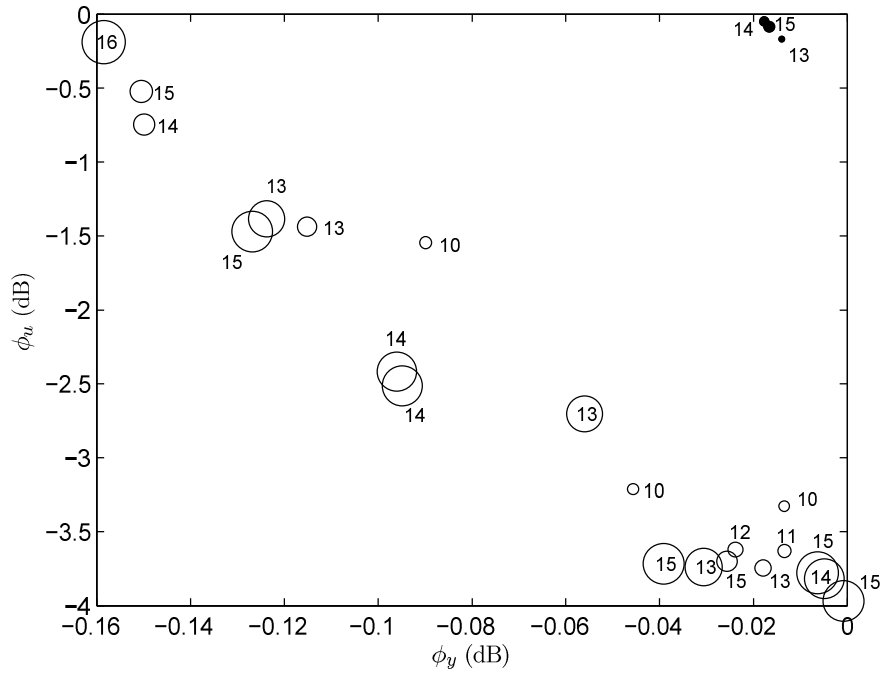


Figure 4: State space realization solution set showing trade-off between ϕ_y and ϕ_u. Each solution is labelled with its word-length. More complex solutions are larger. 5th-order controllers are marked • and 7th-order controllers are marked ○.

word-length) and sparse parameterizations, can be found in 27 and 26 .

There is an increasing interest in the use of Field-Programmable Gate Arrays (FPGA) for the implementation of DSP and control algorithms [47]. These devices provide extremely high speed processing with comparatively low cost. It is important to keep gate counts low for such systems, and hence there has been considerable work recently in designing low-complexity, low word-length filters, e.g. [48, 49]. For FPGA implementations, a suitable complexity measure would be a gate-count. The proposed complexity measures would provide a good estimate of the magnitude of the gate count.

References

[1] D. Kodek and K. Steiglitz. Filter-length word-length tradeoffs in FIR digital filter design. *IEEE Trans. Acoustics, Speech & Sig. Proc.*, vol.28, no.6, pp. 739 744, 1980

[2] D. Kodek. Design of optimal finite wordlength FIR digital filters using integer programming techniques. *IEEE Trans. Acoustics, Speech & Sig. Proc.*, vol.28, no.3, pp. 304 308, 1980

[3] K. Nakayama. Permuted difference coefficient realisation of FIR digital filters. *IEEE Trans. Acoustics, Speech & Sig. Proc.*, vol.30, no.2, pp. 269 278, 1982

[4] A.G. Dempster and M.D. Macleod. Variation of FIR filter complexity with order. *Proc. 38th Midwest Symp. Circuits and Systems*, pp. 342–345, Rio de Janeiro, Aug. 1995,

[5] A.G. Dempster. Reducing complexity by increasing order of IIR digital filters. *Proc. 3rd IEEE Int. Conf. Electr., Circuits and Syst. (ICECS '96)*, pp. 522 – 525, Greece, Oct. 1996

[6] D. Williamson. *Digital Control and Implementation: Finite Wordlength Considerations*, Prentice Hall, Englewood Cliffs, NJ, 1991

[7] J.F. Whidborne and R.H. Istepanian. A genetic algorithm approach to designing finite-precision controller structures. *IEE Proc. Control Theory and Appl.*, vol. 148, no.5, pp. 377 382, 2001

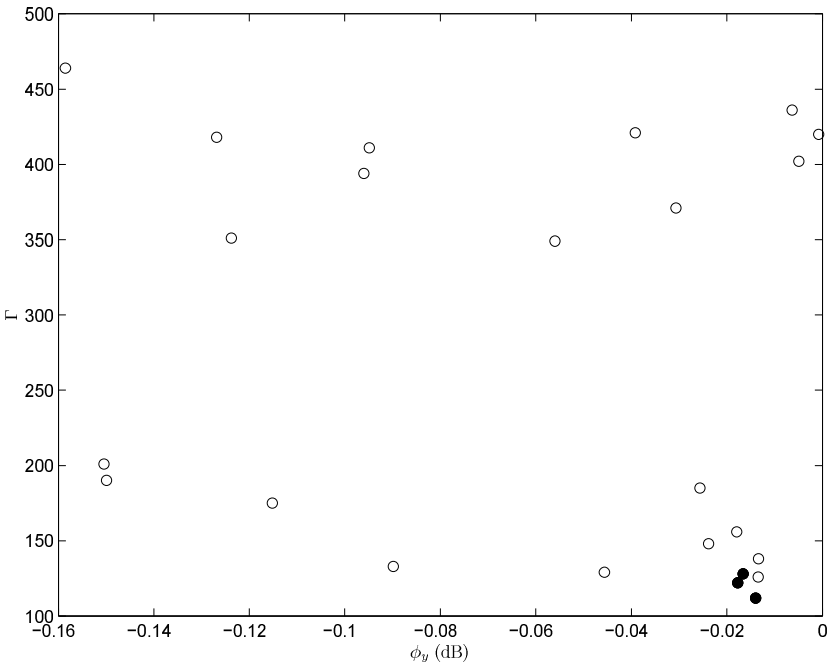


Figure 5: State space realization solution set showing ϕ_y against Γ. 5th-order controllers are marked $\bullet$, and 7th-order controllers are marked $\circ$.

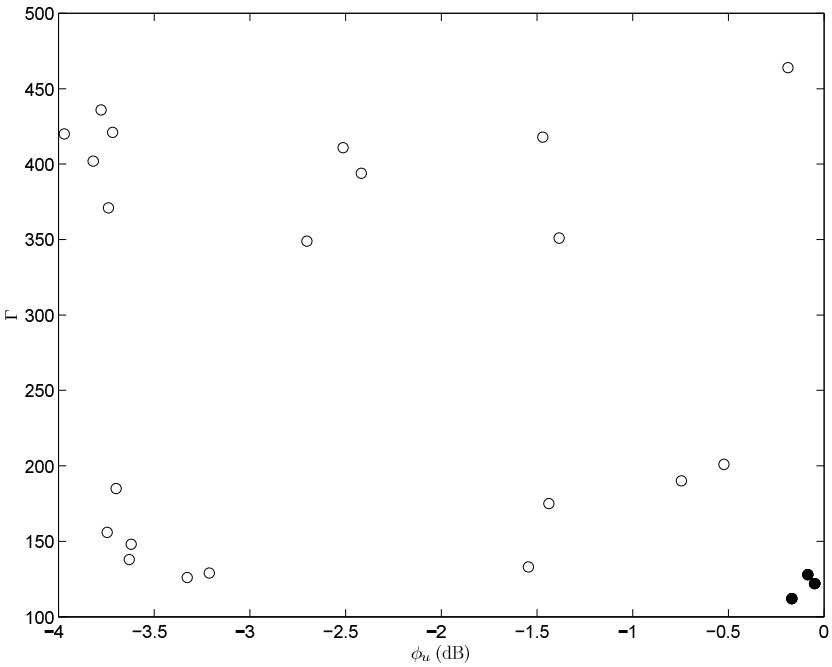


Figure 6: State space realization solution set showing ϕ_u against Γ. 5th-order controllers are marked $\bullet$, and 7th-order controllers are marked $\circ$.

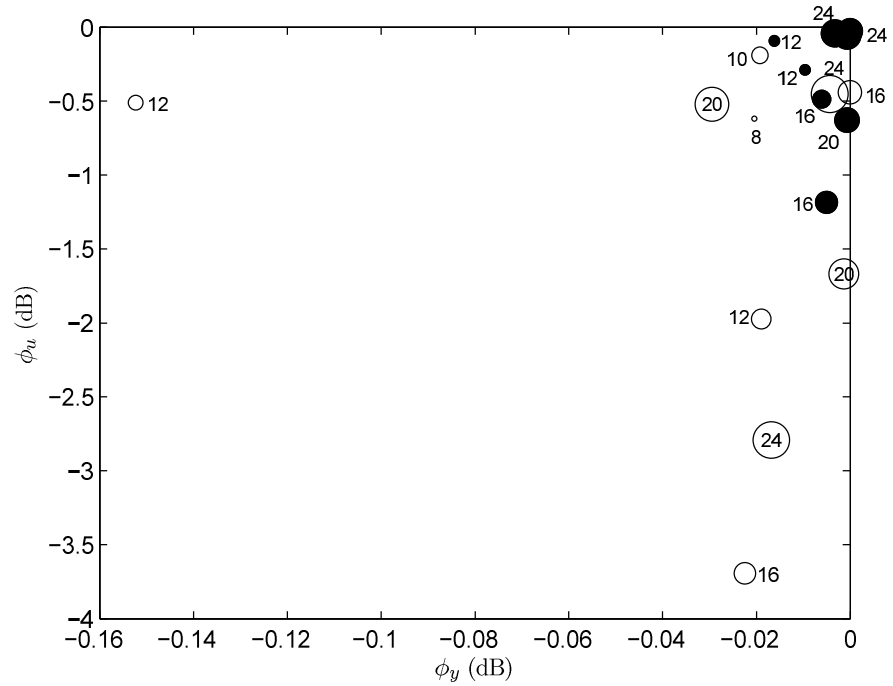


Figure 7: Parallel and cascade realization solution set showing trade-off between ϕ_y and ϕ_u. Each solution is labelled with its word-length. More complex solutions are shown larger. 5th-order controllers are marked •, and 7th-order controllers are marked ○.

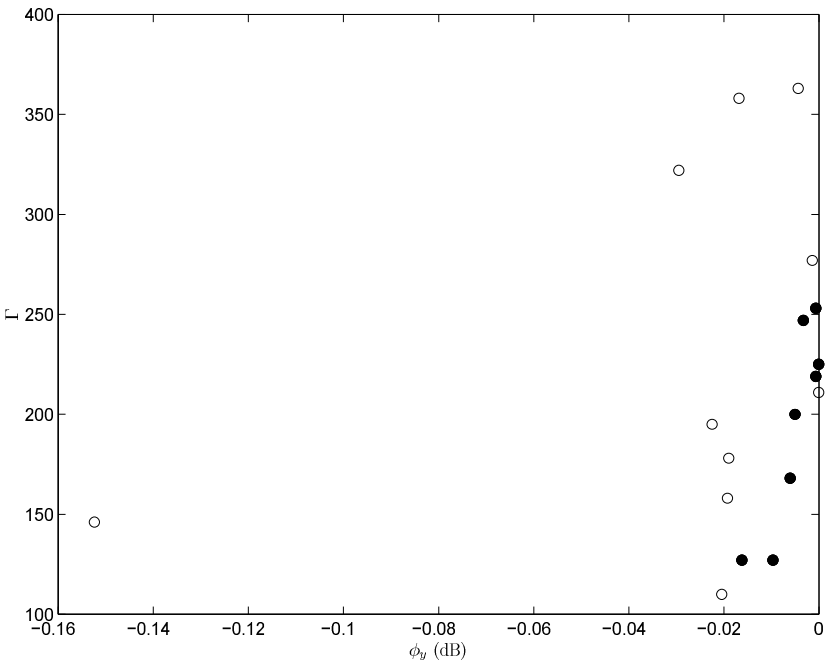


Figure 8: Parallel and cascade realization solution set showing ϕ_y against Γ. 5th-order controllers are marked •, and 7th-order controllers are marked ○.

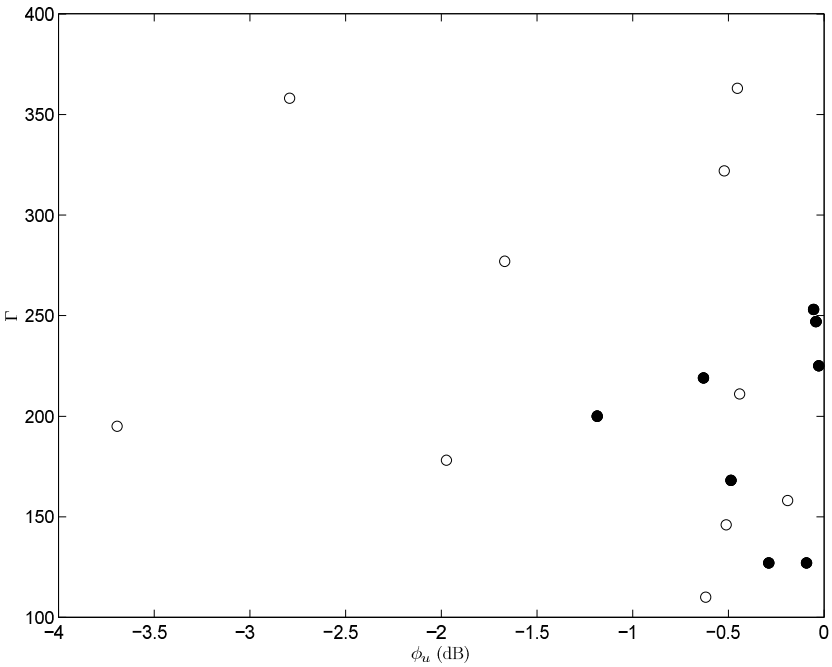


Figure 9: Parallel and cascade realization solution set showing ϕ_u against Γ. 5th-order controllers are marked •, and 7th-order controllers are marked ○.

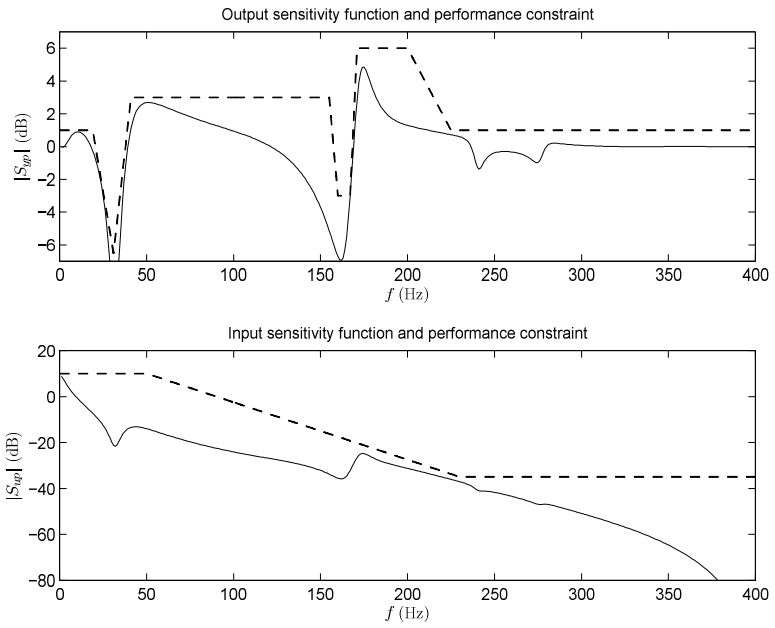


Figure 10: Output and input sensitivity functions (—) for the lowest complexity controller, and performance constraints (- - -).

[8] A.N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Inform. Transmission*, vol.1, pp. 1 7, 1965

[9] G.J. Chaitin. On the length of programs for computing finite binary sequences. *J. ACM*, vol.13, no.4, pp. 547 569, 1965

[10] J.F. Whidborne and J.-B. Yang. Multiobjective design of low complexity digital controllers. in *Proc. IEEE Symp. on Comp. Aided Contr. Syst. Design (CACSD 2002)*, pp. 27–32, Glasgow, UK, Sept. 2002

[11] J. McKernan. Low complexity controller design using genetic algorithms. MSc Thesis, Dept. Mech. Engineering, King's College London, UK, 2002

[12] I.D. Landau, A. Karimi, L. Miskovic, and H. Prochazka. Control of an active suspension system as a benchmark for design and optimisation of restricted complexity controllers. *Eur. J. Control*, vol.9, no.1, pp. 3 12, 2003

[13] Y. Cao and W. Yan. Multi-objective optimisation of restricted complexity controllers. *Eur. J. Control*, vol.9, no.1, pp. 61 65, 2003

[14] J.E. Bertram. The effects of quantization in sampled-feedback systems. *Trans. AIEE*, vol.77, pp. 177 182, 1958

[15] J.B. Slaughter. Quantization errors in digital control systems. *IEEE Trans. Autom. Control*, vol.9, pp. 70 74, 1964

[16] J.B. Knowles and R. Edwards. Effect of a finite-word-length computer in a sampled-data feedback system. *Proc. IEE*, vol. 112, no.6, pp. 1197 1207, 1965

[17] E.E. Curry. The analysis of round-off and truncation errors in a hybrid control system. *IEEE Trans. Autom. Control*, vol.12, pp. 601 604, 1967

[18] P.E. Mantey. Eigenvalue sensitivity and state-variable selection. *IEEE Trans. Autom. Control*, vol.13, no.3, pp. 263 269, 1968

[19] C.T. Mullis and R.A. Roberts. Synthesis of minimum round off noise fixed-point digital filters. *IEEE Trans. Circuits & Syst.*, vol.23, pp. 551 562, 1976

[20] S.Y. Hwang. Minimum uncorrelated unit noise in state-space digital filtering. *IEEE Trans. Acoustics, Speech & Sig. Proc.*, vol.25, pp. 273 281, 1977

[21] P. Morony, A.S. Willsky, and P.K. Houpt. Roundoff noise and scaling in the digital implementation of control compensators. *IEEE Trans. Acoustics, Speech & Sig. Proc.*, vol.31, no.6, pp. 1464 1474, 1983

[22] M.E. Ahmed and P.R. Belanger. Scaling and roundoff in fixed-point implementation of control algorithms. *IEEE Trans. Ind. Electr.*, vol.31, no.3, pp. 228 234, 1984

[23] L. Thiele. Design of sensitivity and round-off noise optimal state-space discrete systems. *Int. J. Circuit Theory Appl.*, vol.12, pp. 39 46, 1984

[24] L. Thiele. On the sensitivity of linear state space systems. *IEEE Trans. Circuits & Syst.*, vol.33, no.5, pp. 502 510, 1986

[25] P. Morony. *Issues in the Implementation of Digital Feedback Compensators*, MIT Press, Cambridge, MA, 1983

[26] M. Gevers and G. Li. *Parametrizations in Control, Estimations and Filtering Problems: Accuracy Aspects*, Springer-Verlag, Berlin, 1993

[27] R.S.H. Istepanian and J.F. Whidborne, Eds. *Digital Controller Implementation and Fragility: A Modern Perspective*, Springer-Verlag, London, UK, 2001

[28] D.S.K. Chan. Constrained minimization of roundoff noise in fixed-point digital filters. in *Proc. IEEE Int. Conf. On Acoust., Speech, and Signal Processing (ICASSP '79)*, pp. 335–339, Washington, DC, Apr. 1979

[29] B.W. Bomar and J.C. Hung. Minimum roundoff noise digital filters with some power-of-two coefficients. *IEEE Trans. Circuits & Syst.*, vol.31, pp. 833 840, 1984

[30] G. Amit and U. Shaked. Minimization of roundoff errors in digital realizations of Kalman filter. *IEEE Trans. Acoustics, Speech & Sig. Proc.*, vol.37, pp. 1980 1982, 1989

[31] G. Li, B.D.O. Anderson, M. Gevers, and J.E. Perkins. Optimal FWL design of state space digital systems with weighted sensitivity minimization and sparseness consideration. *IEEE Trans. Circuits & Syst. II*, vol.39, no.5, pp. 365 377, 1992

[32] G. Li. On the structure of digital controllers in sampled-data systems with stability consideration. In *Digital Controller Implementation and Fragility: A Modern Perspective*, R.S.H. Istepanian and J.F. Whidborne, Eds., pp. 123–142, Springer-Verlag, London, UK, 2001

[33] D.S.K. Chan and L.R. Rabiner. Analysis of quantization errors in the direct form for finite impulse response digital filters. *IEEE Trans. Audio Electroaccoust.*, vol.21, no.4, pp. 354 366, 1973

[34] P.R. Cappello and K. Steiglitz. Some complexity issues in digital signal processing. *IEEE Trans. Acoustics, Speech & Sig. Proc.*, vol.32, no.5, pp. 1037 1041, 1984

[35] D.R. Bull and D.H. Horrocks. Primitive operator digital filters. *IEE Proc.-G*, vol. 138, no.3, pp. 401 412, 1991

[36] A.G. Dempster and M.D. Macleod. IIR digital filter design using minimum adder multiplier blocks. *IEEE Trans. Circuits & Syst. II*, vol.45, no.6, pp. 761 763, 1998

[37] D.E. Quevedo and Goodwin G.C. Moving horizon design of discrete coefficient FIR filters. *IEEE Trans. Sig. Proc.*, vol.53, no.6, pp. 2262 2267, 2005

[38] S. Beheshti and M.A. Dahleh. A new minimum description length. *Proc. 2003 Amer. Contr. Conf.*, pp. 1602–1607, Denver, June 2003

[39] S. Beheshti and M.A. Dahleh. A new information-theoretic approach to signal denoising and best basis selection. *IEEE Trans. Sig. Proc.*, vol.53, no.10, pp. 3613 3625, 2005

[40] J. Schmidhuber. Discovering neural nets with low Kolmogorov complexity and high generalization capability. *Neural Networks*, vol.10, no.5, pp. 857 873, 1997

[41] T. Hilaire, P. Chevrel, and Y. Trinquet. Implicit state-space representation: a unifying framework for FWL implementation of LTI systems. *Proc. 16th IFAC World Congress*, Prague, 2005

[42] J.F. Whidborne, J. McKernan, and D.-W. Gu. Kolmogorov-Chaitin complexity of linear digital controllers implemented using fixed-point arithmetic. *8th Int. Conf. Control, Automation, Robotics and Vision (ICARCV 2004)*, pp. 1587–1592, Kunming, China, Dec. 2004

[43] T.M. Cover and J.A. Thomas. *Elements of Information Theory*, John Wiley, New York, NY, 1991

[44] W. Forsythe and R.M. Goodall. *Digital Control — Fundamentals, Theory and Practice*, Macmillan, Basingstoke, UK, 1991

[45] J. Wu, S. Chen, G. Li, R.H. Istepanian, and J. Chu. Shift and delta operator realisations for digital controllers with finite word length considerations. *IEE Proc.-Control Theory Appl.*, vol. 147, no.6, pp. 664 672, 2000

[46] C.M. Fonseca and P.J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms – part I: a unified formulation. *IEEE Trans. Syst. Man & Cybernetics A*, vol.28, no.1, pp. 26 37, 1998

[47] Z.W. Fang, J.E. Carletta, and R.J. Veillette. A methodology for FPGA-based control implementation. *IEEE Trans. Control Syst. Technology*, vol.13, no.6, pp. 977 987, 2005

[48] G.A. Constantinides, P.Y.K. Cheung, and W. Luk. Wordlength optimization for linear digital signal processing. *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol.22, no.10, pp. 1432 1442, 2003

[49] K.-I. Kum and W. Sung. Combined wordlength optimization and high-level synthesis of digital signal processing systems. *IEEE Trans. Computer Aided Design*, vol.20, no.8, pp. 921–930, 2001

Author Biographies

James Whidborne was born in Zimbabwe in 1960. He received a B.A. degree in Engineering from Cambridge University in 1982, and an M.Sc. and a Ph.D. in Systems and Control from UMIST, Manchester, in 1987 and 1992 respectively. From 1982 to 1985 he worked in industry, and from 1986 to 1991, he was with the Control Systems Centre, UMIST. From 1991 to 1994, he held a position of Research Associate with the Department of Engineering, at the University of Leicester. From 1994 to 2003, he was a Lecturer, then Senior Lecturer with the Department of Mechanical Engineering, King's College London. James Whidborne is currently a Senior Lecturer in the Department of Aerospace Sciences at Cranfield University, UK.

He is a Chartered Engineer, and a Member of the IEE and IEEE. He has over 100 research publications, including three books. His research interests include optimal finite-precision controller implementations, multi-objective robust control design, fluid flow control, and control of UAVs.

John McKernan received a B.A. degree in Engineering from Cambridge University in 1980. From 1980 to 2001 he worked in industry. In 2002 he received an M.Sc. in Computer Aided Mechanical Engineering from Kings College, London. John McKernan is currently completing his Ph.D. at Cranfield University, UK. His research covers the control of transition to turbulence of Poiseuille flow by the application of linear control. He is an Associate Member of the Institute of Mechanical Engineers.

Da-Wei Gu graduated from the Department of Mathematics, Fudan University, Shanghai, China, in 1979, and received an M.Sc. degree in applied mathematics from Shanghai Jiao Tong University, China, in 1981, and a Ph.D. degree in control system theory from the Department of Electrical Engineering, Imperial College of Science and Technology, London, UK, in 1985. Between 1981-1982, he was a lecturer at Shanghai Jiao Tong University. He was a postdoctoral research assistant in the Department of Engineering Science, Oxford University, UK, from 1985 to 1989. In 1989, he was appointed a University Lectureship in the Department of Engineering at Leicester University, UK, and has since been promoted to a Senior Lecturer and a Reader at Leicester. He is a Chartered Engineer, and a Member of the Institution of Electrical Engineers. He has over 230 research publications, including two books. His current research interests include robust and optimal control, optimization algorithms, control system computer-aided design, and numerical perturbation analysis, with applications in aerospace and the manufacturing industry.



https://dspace.org/

Kolmogorov-Chaitin complexity of digital controller implementations.

Whidborne, James F.; McKernan, John; Gu, Da-Wei

2006-07-01T00:00:00Z
Springer Science Business Media
http://dx.doi.org/10.1007/s11633-006-0314-3
James F. Whidborne, John McKernan, Da-Wei Gu. Kolmogorov-Chaitin Complexity of Digital Controller Implementations. International Journal of Automation and Computing, Vol. 3 No.3, July 2006 pg 314-322

Downloaded from a DSpace repository. https://dspace.org/