REAL TIME CONTROL OF ELECTROMECHANICAL SYSTEMS

6–7 March 1984
Cavendish Conference Centre

IERE Publication No. 58

SCIENTIA PRO HOMINIBUS

The Institution of Electronic and Radio Engineers

REAL TIME CONTROL OF ELECTROMECHANICAL SYSTEMS

Cavendish Conference Centre
6–7 March 1984

PUBLICATION No. 58
INSTITUTION OF ELECTRONIC AND RADIO ENGINEERS

Conference on

REAL TIME CONTROL OF ELECTROMECHANICAL SYSTEMS

MEMBERS OF THE ORGANISING COMMITTEE

Chairman: Professor D. R. Wilson, B.Sc., M.Sc., Ph.D., C.Eng., F.I.E.R.E.
(Polytechnic of Central London)

Members: M. J. Ashworth, PhD., C.Eng., M.I.E.R.E.
(Sunderland Polytechnic)

P. Atkinson, B.Sc., C.Eng., M.I.E.E., M.I.E.R.E
(Reading University)

D. J. Bainton, C.Eng., M.I.E.R.E.
(Ministry of Defence)

R. F. C. Butler, M.A.,C.Eng., M.I.E.R.E.
(Vickers Design and Projects Division)

R. L. Davey, B.Sc., Ph.D., C.Eng., M.I.E.R.E.
(Marconi Space and Development Systems)

W. E. Davies, C.Eng., M.I.E.R.E.
(Brighton College of Technology)

W. A. Evans, B.Sc., M.Sc., C.Eng., F.I.E.R.E.
(University College, Swansea)

C. Foxwell, B.Sc., M.I.E.R.E.
(Honeywell Control Systems Ltd)

D. J. Kenner, B.Sc., M.Sc., M.I.E.R.E.
(Brighton College of Technology)

M. Kinsey, C.Eng., M.I.E.R.E.
(The Railway Technical Centre)

M. A. Perry, C.Eng., F.I.E.R.E.
(Consultant)

J. R. Roberts, B.Sc., Ph.D., C.Eng., M.I.E.R.E.
(St Bartholomew's Hospital)

J. K. Stevenson, B.Sc., Ph,D., C.Eng., M.I.E.R.E.
(Wembley, Middlesex)

Professor D. R. Towill, M.Sc., C.Eng., M.I.Mech.E., M.I.Prod.E., F.I.E.R.E.
(U.W.I.S.T.)

Secretary: R. Larry, F.I.E.R.E., M.I.E.E.,
(I.E.R.E.)

Published by
The Institution of Electronic and Radio Engineers
99, Gower Street, London

ISBN 0 903748 53 3

Printed in Great Britain by MULTIPLEX techniques ltd, Orpington, Kent

Session 1
FUNDAMENTAL DESIGN PROBLEMS

CONTENTS

Papers are listed as closely as possible in their order of presentation at the conference

Page No.

Session 1. FUNDAMENTAL DESIGN PROBLEMS

Session 2. COMPUTER CONTROL OF MOTORS

Session 3. SOFTWARE ENGINEERING

Session 4. MICROPROCESSOR CONTROL SYSTEMS

Page No.

Session 5. HANDLING SYSTEMS

Session 6. CONTROL OF MECHANICAL SYSTEMS

INDEX OF AUTHORS

CHOOSE THE BITS AS WELL AS THE PIECES

J. B. KNOWLES

1. INTRODUCTION

An electronic digital computer operates on information in the form of binary numbers. In fixed-point two's complement arithmetic, a binary number $\delta_0 \delta_1 \delta_2 \dots \delta_{M-1}$ has the significance:

$$\delta_0 \delta_1 \delta_2 \dots \delta_{M-1} = -\delta_0 + \sum_{i=1}^{M-1} \delta_i 2^{-i} \qquad (1)$$

where M defines the word-length of the machine and

$$q = 2^{-(M-1)} \qquad (2)$$

its width of quantization. It is evident from equations (1) and (2) that all numbers in the machine lie in the interval $[-1; 1 - q]$. As a result of the finite word-length of the computer, errors are inherently present in its calculations. Firstly, when continuous data is read into the computer a quantization error is incurred. Secondly, each arithmetic product in a computation is subject to a round-off error which lies in the interval $[-q/2 ; +q/2]$ when the more preferable rounded trunc ation is applied. It should be noted that the processes of arithmetic addition and subtraction in the machine are not subject to error.

When a digital computer acts as the compensating element of a sampled-data feedback system, its inherent computational errors may cause the system's performance to deviate considerably from the ideal; even to the point of instability. An early paper by Bertram (1) examined the effect of such errors in a closed-loop sampled-data system. Using the state vector concept, an upper bound to the increase in system error was evaluated in terms of a matrix product. Using Bertram's matrix analysis, a later paper by Slaughter (2) determined an upper-bound to the steady-state increase in system error. Tsypin (3) investigated a closed-loop sampled-data system in which the input and output data are quantized but the multiplicative round-off errors in the computer are not considered. Knowles and Edwards (4, 5) determine the increase in the steady-state mean square error of the system due to both quantization and computer round-off errors. By exploiting the inherent statistical nature of these errors, a more realistic measure of the degeneration in system performance is obtained for any programming technique. Moreover, the necessary calculations are simple and involve the pulse transfer function block diagram of the system, rather than matrix equivalents. The transient mean square error for digital filters realised with floating point arithmetic has since been analysed by Liu (6) and others. However, the calculations are more complex, and in the author's opinion not relevant to microcomputers in control applications. Firstly, the relatively large transient control errors are in practice due very largely to system dynamics rather than computer word-length. Secondly contemporary microcomputers are much more easily programmed for fixed point multiplications.

This paper presents a simple 'back-of-envelope' calculation for determining a reasonable upper bound to the increase of control error due to arithmetic rounding in a digital controller (7). It is particularly useful in the normally iterative design procedure (10, 11), as a means of rapidly eliminating compensators whose realisation would require an excessive wordlength. If a more accurate assessment of wordlength effects appears necessary for the finalised design, then one short calculation on a mini or mainframe machine based on the analyses in references (4) and (5) gives the loss of control accuracy as a function of wordlength.

The coefficients in a pulse transfer function almost invariably require rounding for accommodation in machine format. In consequence, the frequency response of an actual controller differs from the ideal evolved in the compensator design procedure. Because generous stability margins are always allowed, coefficient rounding is normally unimportant as regards the practical realisation of phase-lead or phase-retard compensator elements. However, it is well-known (12, 13) that increasingly severe digital filter specifications (eg cut-off rate, bandwidth etc) accentuate coefficient rounding effects. Hence it is generally prudent, for example, to examine the wordlength required for a digital notch filter to be effective. These elements are employed in countering torsional resonances which have an intrinsically high Q-value in control systems due to very stiff couplings and low friction. My paper concludes with a simple analysis of this case; the more general situation is analysed in reference (14).

2. DATA QUANTIZATION ERROR

An ideally linear, sampled-data system having a digital compensator is shown in Figure 1. The true error sequence of the system is given by:-

$$e_k = x_k - y_k \qquad (3)$$

However, the computed error sequence is:-

$$e'_k = x'_k - y'_k \qquad (4)$$

where

$$\begin{aligned} x'_k &= x_k + r_x(k) \\ y'_k &= y_k + r_y(k) \end{aligned} \qquad (5)$$

and $r_x(k)$ and $r_y(k)$ represent the error incurred in reading into the machine the actual input and output of the system. From equations (3), (4) and (5) one obtains:-

$$e'_k = e_k + r_q(k) \qquad (6)$$

where

$$r_q(k) = r_x(k) - r_y(k) \qquad (7)$$

and

$$\mathrm{Max}\,|r_q(k)| \leqslant q \qquad (8)$$

3. ROUND-OFF ERROR ANALYSIS

The digital compensator of the sampled-data feedback system shown in Figure 1 ideally possesses a Pulsed Transfer Function:

$$\lambda D(z) = \lambda \frac{\sum_{n=1}^{N} b_n z^{-n}}{\sum_{n=0}^{N} a_n z^{-n}} \qquad (9)$$

where λ is a scalar gain constant and the rational function of z represents the frequency dependent part of the compensation. For physical realizability a_0 is non-zero and, without loss of generality, it may be taken as unity. In the direct method of programming (8, 9). the realisation of equation (9) is attempted using the recursion formulae:-

$$f_k = \lambda e'_k \qquad (10)$$

$$h_k = \sum_{n=1}^{N} b_n f_{k-n} - \sum_{n=1}^{N} a_n h_{k-n} \qquad (11)$$

However, due to the finite word-length of the computer, the computed values are:

$$f'_k = \lambda e'_k + r_\lambda(k) \quad \text{with } r_\lambda(k) \leqslant \frac{q}{2} \qquad (12)$$

$$h'_k = \sum_{n=1}^{N} b_n f'_{k-n} - \sum_{n=1}^{N} a_n h'_{k-n} + r_D(k) \qquad (13)$$

where $r_\lambda(k)$ and $r_D(k)$ are the total round-off errors incurred in evaluating $\lambda e'_k$ and:

$$\sum_{n=1}^{N} b_n f'_{k-n} - \sum_{n=1}^{N} a_n h'_{k-n} \qquad (14)$$

respectively. Substituting equations (6) and (12) into equation (13) yields:

$$h'_k = \sum_{n=1}^{N} b_n \left[\lambda e_{k-n} + \lambda r_q(k-n) + r_\lambda(k-n)\right] - \sum_{n=1}^{N} a_n h'_{k-n} + r_D(k) \qquad (15)$$

It is now assumed that the feedback system shown in Figure 1 is in steady-state equilibrium with its input and output set at zero displacement. If the control loop is now broken at the point A and a step function of height ε applied to the input, the actual steady-state computer output is given by:-

$$h'_k = \sum_{n=1}^{N} b_n \left[\lambda \varepsilon + \lambda r_q(k-n) + r_\lambda(k-n)\right] - \sum_{n=1}^{N} a_n h'_{k-n} + r_D(k) \qquad (16)$$

In cases of interest:-

$$\varepsilon \gg q \qquad (17)$$

so that equation (16) reduces to:-

$$h'_k = \sum_{n=1}^{N} \lambda \varepsilon b_n - \sum_{n=1}^{N} a_n h'_{k-n} + r_D(k) \qquad (18)$$

With a sufficiently large width of quantization, the magnitude of the error term in equation (18) can reduce the steady-state computer output to zero. For this condition:-

$$0 = \sum_{n=1}^{N} \lambda \varepsilon b_n + r_D(k) \qquad (19)$$

Hence, the condition for zero steady-state output from the computer is

$$\left|\sum_{n=1}^{N} \lambda \varepsilon b_n\right| = |r_D(k)| \qquad (20)$$

If equation (13) is implemented with a total of μ rounded multiplications not involving zero or a positive integral power of 2, then evidently:-

$$|r_D(k)| \leqslant \mu q/2 \qquad (21)$$

so that from equation (21) one obtains the inequality:

$$\left|\sum_{n=1}^{N} \lambda \varepsilon b_n\right| \leqslant \mu q/2 \qquad (22)$$

If the control loop were reconnected under the conditions specified by equation (22) no corrective action would be applied to the plant and a steady-state error of ε would exist in the closed loop system. Thus, the steady state degeneration in feedback system performance due to the finite word-length of the computer is given for rounded operations and direct programming by:-

$$|\varepsilon| \leqslant \left(\frac{\mu}{2\lambda\left|\sum_{n=1}^{N} b_n\right|}\right) \qquad (23)$$

Observe that a smaller width of quantization in equation (16) would produce a non-zero steady-state computer output giving rise to a corrective action to reduce the error. That is to say, for a given width of quantization, the right-hand side of equations (23) specifies an upper bound for the degeneration in system performance. For some second-order compensators, the inbuilt pessimism of this result was found experimentally (7) to be equivalent to some 2 or 3 bits.

To appreciate the significance of equation (23) consider the pulse transfer function of the phase-lead compensator:-

$$D(z) = \left(\frac{1 - b z^{-1}}{1 - a z^{-1}}\right) \quad \text{with } 0 \leqslant a < b < 1 \qquad (24)$$

for which equation (25) reads as:

$$|\varepsilon| \leqslant |1 - b|^{-1} \tag{25}$$

As more phase-lead is achieved by moving the zero (b) closer to the point $1\angle 0$ (the zero-frequency point), control system performance is evidently degenerated. This result parallels the well-known behaviour of R-C phase-lead networks, in which more phase-lead produces a greater amplification of circuit noise.

As shown in Reference (7), the previous analysis can be extended to the parallel and cascade programming techniques for realising pulse transfer functions. However, in the context of control engineering as opposed to signal processing, first- and second-order directly programmed transfer functions are satisfactory for many practical purposes, and equation (23) relates specifically to this situation. For more complex transfer functions, parallel and cascade programming techniques usually achieve a smaller loss of control accuracy. Although the above equations are not applicable to these techniques, they can still be used to establish the best of two algorithms having equally suitable frequency-response functions. If considered necessary, a more accurate check on the performance degradation for a final design can be easily implemented by the machine calculation described in references 4 and 5; even for very complicated controllers.

4. COEFFICIENT ROUNDING ANALYSIS

A simple digital notch-filter with centre-frequency ω_o and rad/s and unity gain at zero frequency has the pulse transfer function:-

$$N(z) = K_N\left(\frac{z^2 - az + 1}{z^2 - rbz + r^2}\right) \tag{26}$$

where

$$a = b = 2\cos(\omega_o T) \; ; \; 0 < r < 1$$
$$K_N = \left|[1 - r\exp(-j\omega_o T)]\Big/[1 - \exp(-j\omega_o T)]\right|^2 \tag{27}$$

Its frequency response $N^*(j\omega)$ is derived by setting:-

$$z = \exp(j\omega T) \tag{28}$$

but by defining the frequency displacement variable:-

$$\delta = \omega - \omega_o \tag{29}$$

and effecting a truncated Maclaurin series expansion of the exponential functions, its notch-type behaviour is more clearly revealed as (10, 11):

$$|N^*(j\delta)| \doteqdot \frac{K_{NO}}{\sqrt{1 + \left(\frac{1-r}{T\delta}\right)^2}} \tag{30}$$

where

$$K_{NO} = K_N\left|\frac{1 - \exp(-j2\omega_o T)}{1 - r\exp(-j2\omega_o T)}\right| \tag{31}$$

The bandwidth (B) of a notch-filter is the frequency increment about its centre-frequency (ω_o) for which the attenuation is greater than 3dB, so that from equation (30)

$$B = (1 - r)/T \tag{32}$$

Rounding the coefficients $\{a \, ; (rb) \, ; r^2\}$ in equation (26) to $\{a' ; rb' ; (r^2)'\}$ is seen from equations (27) and (32) to modify both the centre-frequency and bandwidth. Indeed, they may be altered so much that the network is no longer effective in blocking a resonant mechanical mode of the system. To quantify the effect of coefficient rounding on these important design parameters set:

$$a' = 2\cos(\omega_1 T) \quad ; \quad b' = 2\cos(\omega_2 T) \tag{33}$$
$$\delta = \omega - \omega_1 \quad ; \quad \varepsilon = \omega_1 - \omega_2$$

and after truncating Maclaurin series expansions for the exponential functions again, one obtains

$$|N^*(j\delta)| = \frac{K'_{NO}}{\sqrt{[1 + \varepsilon/\delta]^2 + [(1 - r')/\delta T]^2}} \tag{34}$$

It follows that the centre-frequency is now at ω_1 $(\delta = 0)$ and this shift is evaluated directly from equations (27) and (33) as:-

$$\Delta\omega_o = \frac{1}{T}\left[\cos^{-1}(a'/2) - \cos^{-1}(a/2)\right] \tag{35}$$

Furthermore, the bandwidth (B') of the realisation is derived from equation (34) as:

$$(B')^2 = 2\varepsilon B' + \varepsilon^2 + \left(\frac{1 - r'}{T}\right)^2 \tag{36}$$

In terms of derivatives at the point $(0, r)$ in $(\varepsilon; r')$- space, it follows that:-

$$|\Delta B| \leqslant \left|\frac{\partial B'}{\partial \varepsilon}\right| |\Delta\varepsilon| + \left|\frac{\partial B'}{\partial r'}\right| |\Delta r'| \tag{37}$$

From equations (32), (33) and (36), terms in the above expression are obtained as:

$$|\Delta\varepsilon| \leqslant 2|\Delta\omega_o| \; ; \; |\Delta r'| \leqslant q/4r$$
$$|\partial B/\partial\varepsilon| = 1 \quad ; \; |\partial B/\partial r'| = 1/T \tag{38}$$

so that

$$|\Delta B| \leqslant 2|\Delta\omega_o| + \frac{q}{4T(1 - BT)} \tag{39}$$

Consider for example the realisation of a notch filter with centre-frequency 12 rad/s, bandwidth ± 1 rad/s and a sampling period of 0.1s. Equation 27 specifies:-

$$a = 0.724716$$

With an 8 bit machine format, the above coefficient is best approximated by:-

$$a' = 0.718750 \qquad (0.1011100)$$

from which equations (35) and (39) give:-

$$\Delta\omega_o = 0.032 \text{ rad/s} \quad ; \quad |\Delta B| \leqslant 0.087 \text{ rad/s}$$

Therefore an 8-bit machine is about sufficient for realising the above filter specification.

5. CONCLUSIONS

Computationally simple methods of quantifying the effects of arithmetic and coefficient rounding in digital compensatons have been described. More comprehensive and accurate procedures are to be found in references (4, 5, 14). Generally speaking, the designer should be most wary of arithmetic roundoff in the realisation of phase-lead compensators, and coefficient rounding with high-Q anti-resonance filters.

5. REFERENCES

1. BERTRAM J E: "The effect of quantization in sampled-feedback systems". Trans AIEE, (1958), 77, pt 1, p 177.

2. SLAUGHTER J B: "Quantization errors in digital control systems", Trans IEEE PTGAC (1964), p70.

3. TSYPKIN Ya Z. "An estimate of the influence of amplitude quantization on processes in digital automatic control systems". Automatika i Telemekhanika, (1960), 3.

4. KNOWLES J B and EDWARDS R. "The effect of a finite word-length computer in a sample data feedback system". Proc IEE (1965), 112, p1197.

5. KNOWLES J B and EDWARDS R. "Finite word-length effects in multi-rate direct digital control systems". Proc IEE, (1965) 112, 2376.

6. KANEKO T and LIU B. "Round off error of floating-point digital filters". 6th Annual Allerton Conference, October (1968).

7. KNOWLES J B and EDWARDS R: "Computational error effects in a direct digital control system". Automatica, (1966), 4, p7.

8. JURY E I: "Sampled data control systems". (Wiley, 1958).

9. RAGAZZINI J R and FRANKLIN G: "Sampled data control systems". (McGraw-Hill, 1958).

10. KNOWLES J B: "Some DDC System design procedures", Chapter 3, Computer Control of Industrial Processes, Editors S Bennett and D A Linkens, IEE Control Engineering Science, Volume 21 (1982).

11. KNOWLES J B: "Some DDC system design procedures". Chapter 12, Design of modern control systems - Editors D J Bell et al, IEE Control Engineering Series, Volume 20 (1982).

12. AGARWAL R C and BURRUS C S: "New Recursive Digital Filter structures have a very low sensitivity to Round-Off Error Noise", IEEE Trans. on Circuits and Systems, 22, No 12, 921, (1975).

13. WEINSTEIN C J: "Quantization Effects in Digital Filters". MIT Lincoln Laboratory, Report 468, (1969).

14. KNOWLES J B and OLCAYTO E: "Coefficient Accuracy and Digital Filter Response". IEEE Trans. Circuit Theory 15, 31 (1968).

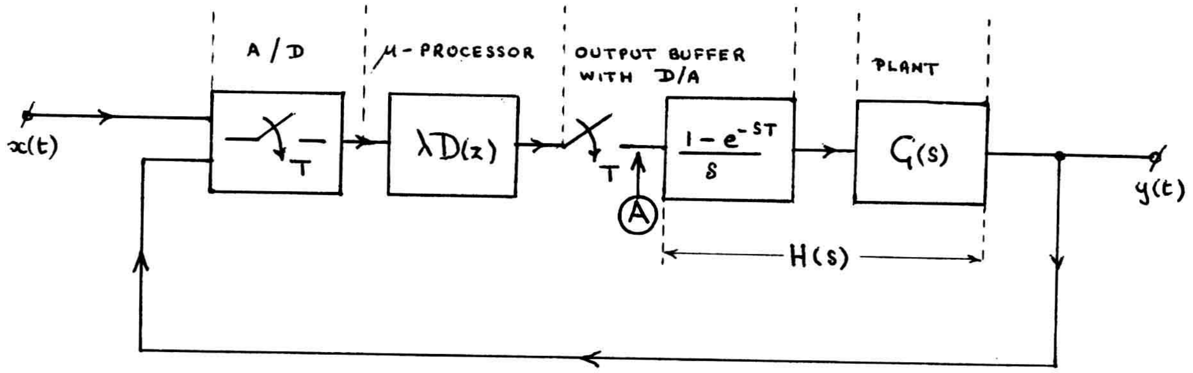

FIG. 1: A SINGLE INPUT DDC SYSTEM

DIRECT DIGITAL CONTROL ALGORITHMS FOR A CLASS OF D.C. DRIVES

E.S.TEZ *

SUMMARY

The direct digital control algorithms employed for speed regulation in a microprocessor-controlled, reversible and regenerative d.c. drive system are described. The closed-loop control of motor speed is achieved by means of only a single, speed feedback loop without using any A/D or D/A converters, and the need for a separate current loop is obviated by the real time control algorithms employed. The results obtained from a digital computer simulation of the overall drive system are provided as the basis for assessing the system performance under various operational conditions.

1. Introduction

The paper considers a reversible, regenerative d.c. drive system as an electromechanical plant where the process to be controlled by a micro-computer is the motor speed variations under various operational conditions.

Conventional control systems for variable-speed drives generally contain an outer speed feedback loop and an inner current feedback loop (ref. 1). Although the speed information can be fed back to a microprocessor-based controller in a digital form, feeding back the motor current data necessitates the use of an A/D converter in the current loop. Recently there has been an increasing interest in the direct digital control of variable speed drive systems. However, considerations appear to have been directed towards low power servo applications or simpler types of d.c. drives. The technical specific-ations of industrial drives, such as maximum current limit, controlled acceleration/ deceleration, fast speed reversal etc., have not yet been considered for implementation by soft-ware in a real time control situation.

The microprocessor-controlled d.c. drive system considered in the present paper is based on the armature control of a separately-excited d.c. motor fed by a 4-quadrant chopper. The hardware configuration of the system is shown as a block diagram in Fig. 1, and its practical realization has been described in ref. 2. At each sampling interval (fixed by Timer 2 of Fig. 1), the motor speed is sensed by a digital speed monitor, and the microcomputer determines the voltage to be applied to the motor according to the control algorithms employed. The micro-computer then controls the chopper via the duration of four gate signals, TS1 to TS4, in order to produce the necessary output voltage to the armature of the motor. As obvious in Fig. 1, the system contains only a single speed feedback loop, and the need for a separate current loop which would necessitate the use of A/D converters is obviated by the direct digital control technique employed. As described later, the function of the current feedback loop in limiting the armature current during large transient changes of motor speed is implemented totally by software in the microcomputer.

The present paper discusses the direct digital control algorithms used in the real-time control of the drive system. Based on a linear mathematical model presented, a digital computer simulation is established for the overall speed control system including the control algorithms. The results from the digital simulation are provided as the basis for assessing the system performance under various operational conditions. As an alternative to the popular PID algorithm, a nonlinear control algorithm is proposed, which essentially does not require tuning for different loads. This algorithm is shown to provide smooth transient changes with no overshoot in the controlled speed variation.

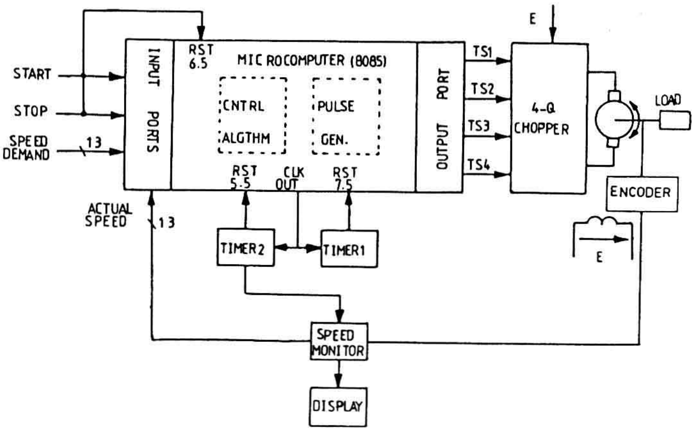

Fig.1: Microprocessor-controlled D.C. Drive

* Dept. of Electronic and Electrical Eng., University of Technology, Loughborough.

2. Mathematical Model

The dynamic behaviour of the digital control system of Fig. 1 can be described by the block diagram shown in Fig. 2, where transfer functions are used for representing various components of the system. The functions contained within the broken lines are performed by the microcomputer, and these include sampling the actual and required speeds and calculating the duration of the chopper gate signals so as to produce the necessary armature voltage. The block with the transfer function G(s) represents the control algorithm by which the gate signal duration D_1 is calculated from the speed error. The value of D_1 calculated at the nth sampling instant, D_{1n}, remains constant throughout the sampling interval, and this is represented by the zero-order-hold block in Fig. 2.

Since the chopper operational frequency is chosen high in comparison to the electrical time constant of the motor armature, the chopper is assumed to produce a continuously varying output voltage proportional to the duration of gate signals. Together with the software process of gate signal generation in the microcomputer, the chopper can thus be represented by the transfer function

$$\frac{V_{an}}{D_{1n}} = kE \qquad (1)$$

where k is a constant fixed by microcomputer hardware and E is the d.c. supply voltage.

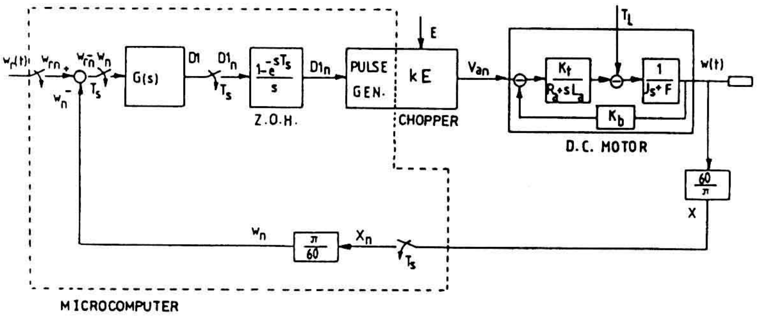

Fig.2: Transfer function block diagram

In the actual system hardware, the speed monitor measures the motor speed in r.p.m. The speed monitor in conjunction with the shaft encoder is therefore represented in Fig. 2 by a transfer function which provides conversion of units from rad/s to r.p.m.

A linear mathematical model is chosen for the separately-excited d.c. motor as characterised by the equations

$$\frac{dI_a}{dt} = \frac{V_a - R_a I_a - K_b \omega}{L_a} \qquad (2)$$

$$\frac{d\omega}{dt} = \frac{K_t I_a - F\omega - T_L}{J} \qquad (3)$$

where (see Nomenclature for symbols) I_a and ω are considered as state variables. The corresponding transfer function of the motor is shown in Fig. 2 with a block diagram representation taking V_a and ω as the input and output variables, respectively.

3. Control Algorithms

Since regulation of both the armature current and motor speed is intended with only a single feedback loop, there exists a need for two different control algorithms. These are such that, when the speed error is greater than a certain value, the maximum acceleration and torque are both determined by a current limiting algorithm (Algorithm 1). When the speed error reduces to below the value fixed, a different control algorithm (Algorithm 2), such as PID control, comes into effect to provide an optimal system response.

The decision as to which algorithm should be brought into action is made at each sampling instant according to the process in Fig. 3. The comparison of the speed error ε_n with a fixed error limit $\Delta\omega$ results in various speed zones, shown in Fig. 4, in each of which a different algorithm becomes operational. The error limit $\Delta\omega$ fixes the width of the zone for Algorithm 2 and a suitable value may be found for it by means of tuning the controller. However, a specific relationship is considered below between $\Delta\omega$ and other system parameters.

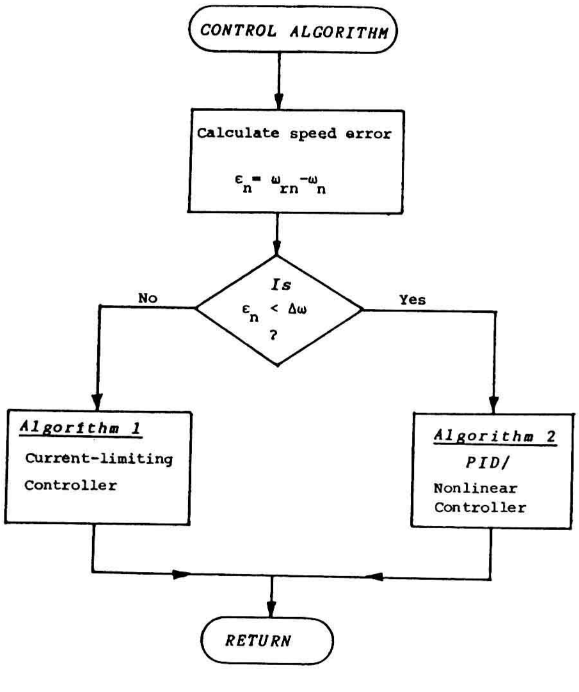
CONTROL ALGORITHM
Calculate speed error
$\varepsilon_n = \omega_{rn} - \omega_n$
Is $\varepsilon_n < \Delta\omega$?
No
Yes
Algorithm 1
Current-limiting Controller
Algorithm 2
PID/
Nonlinear Controller
RETURN

Fig.3: Process of algorithm selection

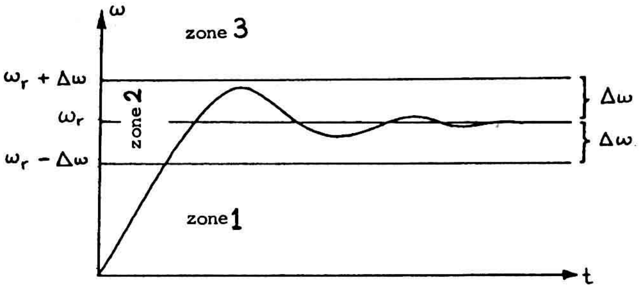

Fig.4: Typical step response

3.1 Algorithm 1 : Current-Limiting

The limiting of the motor current is achieved by restricting the difference between the applied armature voltage and the generated back e.m.f. to a maximum value ΔV_{max} fixed by software. Thus, the condition

$$|V_a - K_b\ \omega| \leqslant \Delta V_{max} \tag{4}$$

leads to the basis of current-limiting control algorithm as

$$V_{an} = K_b \omega_n \mp \Delta V_{max} \tag{5}$$

where

$$\Delta V_{max} = R_a . I_{amax} \tag{6}$$

and I_{amax} is a parameter with a value chosen anywhere below the maximum physical current that the drive system components can withstand. The appropriate sign in equation (5) depends on whether the motor speed falls into zone 1 or zone 3 of Fig. 4, and ΔV_{max} can therefore be given the same sign as the speed error ε_n. Accordingly, the current-limiting control algorithm can be formulated as

$$V_{an} = K_b \omega_n + \text{sign}\ (\varepsilon_n) . \Delta V_{max} \tag{7}$$

which is also valid for negative speeds, i.e. reverse running of the motor.

Equation (7) shows that V_{an} is determined from the current sample of motor speed and depends on neither the previous values of V_a or ω nor the sampling period. The sampling period can therefore be made as short as that needed to obtain a speed sample and to perform the mathematical operations involved.

The parameters K_b, R_a and I_{amax} are stored in the microcomputer's memory as constants. However, the value of I_{amax} can be changed in order to produce different rates of acceleration/ deceleration under software control.

3.2 Algorithm 2

A number of digital control algorithms would be suitable for use as Algorithm 2. The paper considers two different algorithms: firstly, the proportional-integral-derivative (PID) control, and, secondly, a nonlinear control algorithm which is based on an extension of the current limiting algorithm.

3.2.1 PID Control Algorithm

The most popular control algorithms used in industry are variants of PID control. The application of PID control to the system of Fig. 2, when the transfer function G(s) becomes

$$G(s) = K_p \times K_d s + \frac{K_I}{s} \tag{8}$$

results in the expression to be calculated by Algorithm 2 to have the form

$$V_{an} = V_{an-1} + C_1(\omega_{rn}-\omega_n) - C_2(\omega_{rn-1}-\omega_{n-1})$$
$$+ C_3(\omega_{rn-2}-\omega_{n-2}) \tag{9}$$

where C_1 to C_3 are constants related to the control gains by

$$C_1 = K_p + K_I + K_d$$
$$C_2 = K_p + 2K_d \tag{10}$$
$$C_3 = K_d$$

To obtain an optimum overall performance, suitable values are determined for the control gain coefficients by 'tuning' the control system. A tuning process is also needed to ensure that the armature current does actually not exceed I_{amax} at the instant when a switch-over takes place from the current-limiting algorithm to the PID algorithm.

3.2.2 Nonlinear Control Algorithm

This algorithm is developed from the condition that the armature current should not exceed I_{amax} even at the instant when the controller switches to Algorithm 2. Assume that, at the (n-1)th sampling instant, V_{an-1} is determined according to the current-limiting algorithm as

$$V_{an-1} = K_b \omega_{n-1} + \Delta V_{max} \tag{11}$$

and the controller switches to Algorithm 2 at the nth sampling instant when the armature voltage is incremented by ΔV, i.e.

$$V_{an} = V_{an-1} + \Delta V \tag{12}$$

Application of the condition in equation (4) to the above situation gives

$$V_{an} - K_b \omega_n \leqslant \Delta V_{max} \tag{13}$$

which, upon substitution of V_{an} from equation

(12) and then V_{an-1} from equation (11), becomes

$$K_b\omega_{n-1} + \Delta V_{max} + \Delta V - K_b\omega_n \leq \Delta V_{max} \tag{14}$$

leading to the condition

$$\Delta V \leq K_b(\omega_n - \omega_{n-1}) \tag{15}$$

This condition can be converted into an equation by multiplying the term on the right hand side by a factor always smaller than unity. Since $|\omega_r - \omega| < \Delta\omega$ within zone 2 of Fig. 4, $(\omega_r - \omega)/\Delta\omega$ has a magnitude smaller than unity as long as Algorithm 2 is operational. With this chosen as the fractional factor, the control algorithm resulting from equations (12) and (15) is obtained as

$$V_{an} = V_{an-1} + K_b(\omega_n - \omega_{n-1})\frac{(\omega_{rn} - \omega_n)}{\Delta\omega} \tag{16}$$

The above algorithm is in essence a modified integral control process in which the voltage change at a sampling instant is a function of both the speed error and the difference between the present and previous samples of the actual speed. Because of this relationship, the algorithm is termed a nonlinear control algorithm.

The only parameter in equation (16) without a specific value is $\Delta\omega$. Although this can be determined from a tuning process, it seems sensible to relate $\Delta\omega$ to ΔV_{max} through the dimensionally-correct relationship

$$\Delta\omega = \frac{\Delta V_{max}}{K_b} = \frac{R_a\ I_{amax}}{K_b} \tag{17}$$

which results in zone 2 of Fig. 4 to have a variable width depending on I_{amax}, the parameter controlling the rate of acceleration/deceleration.

4. Computer Simulation

On the basis of the mathematical model and control algorithms above, a digital computer simulation of the overall d.c. drive has been established with the values in the Appendix assigned to various system parameters.

The simulation program performs the same functions as the microcomputer does in the practical system. Besides, it simulates the motor/load combination by solving equations (2) and (3) continuously, although the armature voltage is determined by the control algorithms at each sampling instant in a discrete manner. The sampling period is taken as 16 ms, which is nearly equal to that in the practical hardware. The fact that the maximum output voltage available from the chopper cannot exceed the supply voltage is also included in the simulation. If, in a sampling interval, V_a is calculated by the control algorithms to be greater than E, the simulation limits the voltage applied to the armature to the supply level. The mechanical load exerts varying magnitudes of torque on the motor shaft and is assumed to have constant inertia and friction coefficients combined with those of the motor. The load torque is always given the same polarity as that of the actual motor speed.

The system response against step changes in the speed demand is illustrated in the following section for the cases of PID and nonlinear control algorithms used in conjunction with the current-limiting algorithm.

5. Results

5.1 Case with PID algorithm

A typical response when the speed demand is suddenly changed from zero to 1500 r.p.m. at no load is shown in Fig. 5, which exemplifies the tuning process for ensuring that I_a does not exceed I_{amax} even after the switch-over from Algorithm 1 to Algorithm 2. Initially, the current-limiting algorithm is in operation and I_a remains below the chosen maximum value while the motor accelerates at a constant rate. When the motor speed approaches the required level, the PID algorithm takes over, producing a

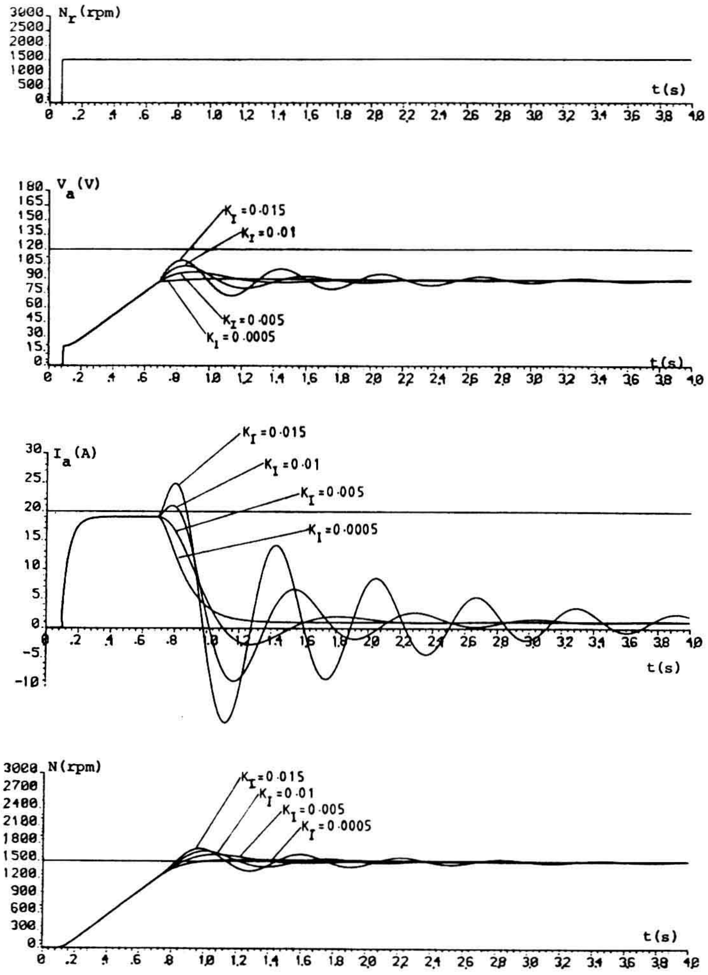

Fig.5: Step responses with PID algorithm

$K_p=10^{-4}, K_d=1, I_{amax}=20$ A, $T_L=0$ Nm response which depends on the control gain coefficients. As the figure shows, the appropriately chosen coefficient values result in no alternations in the armature current while the speed quickly settles to a steady state at the required level.

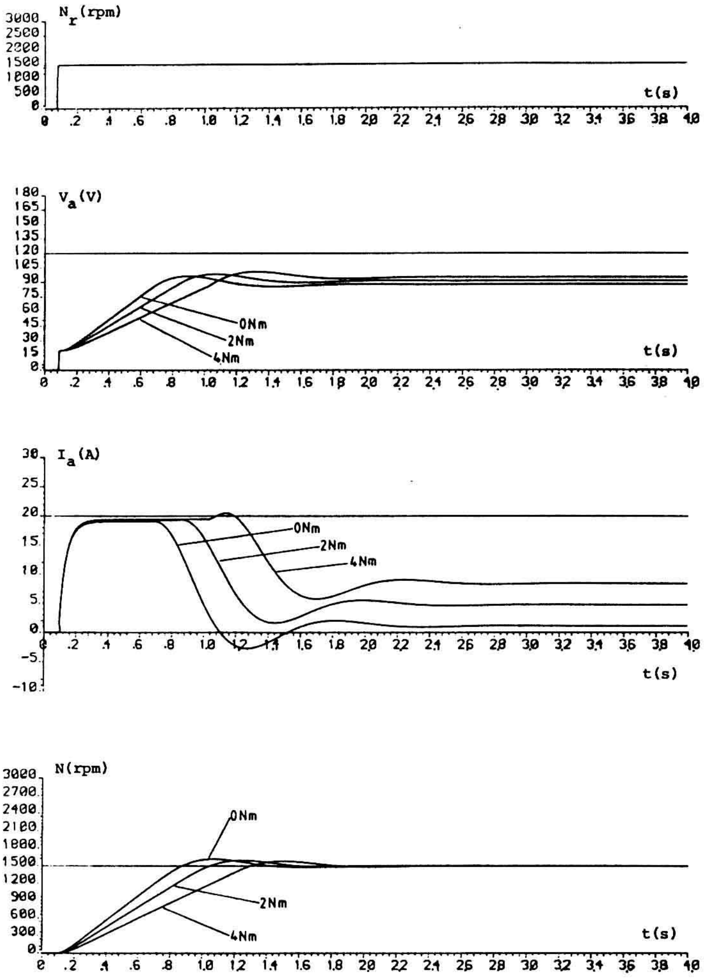

Fig.6: Effects of varying T_L on response with PID algorithm. $K_p=10^{-4}$, $K_I=0.005$, $K_d=1$, $I_{amax}=$ 20 A.

The effect of increased load torques can be seen in Fig. 6 as the changes in the acceleration rate produced by the current-limiting algorithm, together with an increase in the final steady-state motor current.

As evident in the figures, the armature current assumes negative values at certain intervals, although a positive speed (forward running) is required of the motor. This indicates that, during these intervals, the drive system is operating in the regenerative braking mode.

5.2 Case with nonlinear algorithm

The response resulting from the use of the non-linear control algorithm as Algorithm 2 is shown in Figs. 7 and 8. In order to illustrate that the drive system is both reversible and regenerative, the variation of speed demand has been given the profile shown in the figures.

The nonlinear control algorithm does not contain any coefficients which need to be determined by a tuning process. The only parameter to be fixed externally is I_{amax}, and Fig. 7 shows the system performance under different loading conditions with I_{amax} chosen as 40 A. As also occurs in the previous case, the current-limiting algorithm remains in operation for longer periods when the load torque is increased. This is because the maximum torque developed by the motor is fixed by I_{amax} and the torque available for acceleration decreases with increasing load torque.

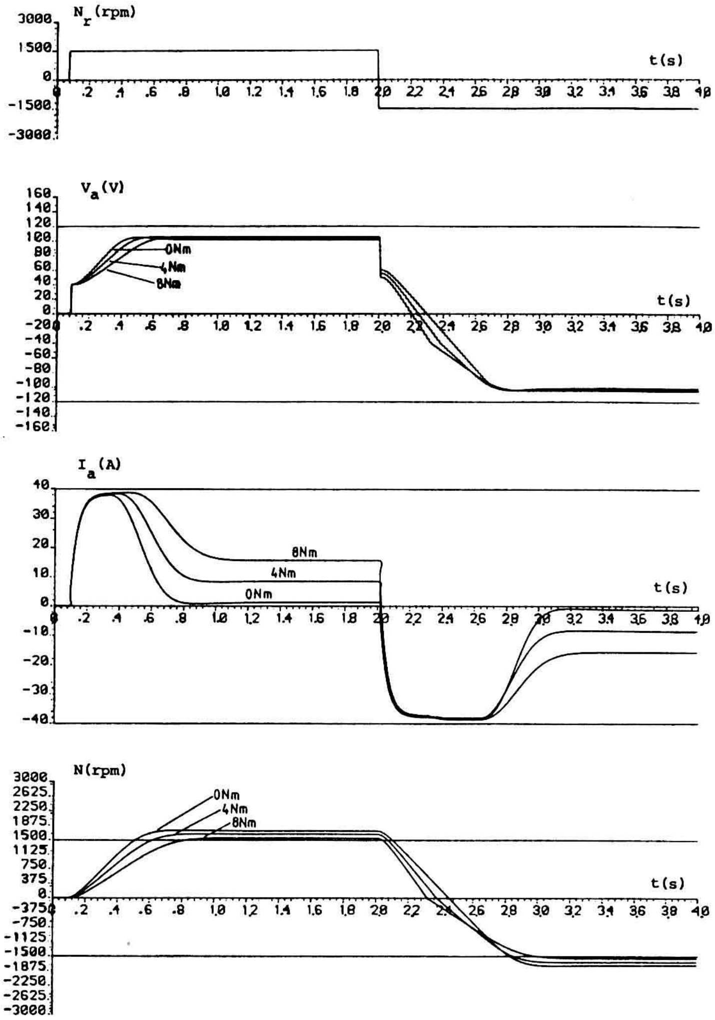

Fig.7: Responses with nonlinear algorithm under varying T_L . I_{amax}=40 A.

The switch-over from the current-limiting algorithm to the nonlinear algorithm is seen to be followed by very smooth transient changes, with the motor speed and current both exhibiting no oscillations while attaining their steady states in a gradual manner. The same is also evident in Fig. 8, which illustrate the effect of different values given to I_{amax} under a constant load torque. It is obvious that I_{amax} can be used as a parameter to produce various