A Microcomputer-Based Self-Tuning IP Controller for DC Machines

RAMACHANDRAN BALASUBRAMANIAN AND KONG HUNG WONG

Abstract—**The problem of designing an adaptive controller for dc drives operating under varying load conditions is addressed. A robust self-tuning controller that can track a constant reference and reject constant load disturbances is developed. The controller is shown to have an adaptive integral–proportional (IP) structure. The proposed controller is implemented in a microcomputer and a dc motor is controlled under varying load and operating conditions. Experimental results show that the controller possesses excellent adaptation capability as well as transient recovery under load changes.**

I. Introduction

THE TWO important requirements for high-performance controlled electrical drives are their steady-state and dynamic tracking ability to set point changes and their ability to recover from load disturbances. Conventional proportional–integral (PI) controllers for such drives are designed on the basis of local linearization about an operating point. These controllers are very effective if the load changes are small and the operating conditions do not force the system too far from the linearizing equilibrium point.

In certain applications, such as precision servos for robots or machine tools and high power drives used for power shovels, mine hoists, etc., the drive operates under a wide range of changing load characteristics and the parameters of the system vary substantially [1]. Thus to ensure a specified dynamic response independent of variations of the parameters, some form of adaptive control is called for. One of the earliest suggestions to this end was to use a model-following adaptive controller [2]. Although there have been some recent extensions of this technique [3] it is not widely accepted, perhaps for two reasons. In this scheme the parameters of the system are not estimated and hence are not available for control or other purposes; secondly there are some difficulties in implementing this scheme using limited word-length microcomputers.

The adaptive control strategy that has received widespread acceptance in process control and other industrial control applications is the self-tuning controller [4]. This scheme proceeds in two stages: in the first, the parameters of the system are estimated in real time from the input to the output of the system, and in the second a control signal is generated based on these parameter estimates and some control design algorithm. These two stages are carried out in a digital computer and the parameters estimated correspond, of course, to those based on a discrete model of the continuous system. This scheme is very attractive since the estimates of system parameters are also available in real time. Recently it has been suggested that these estimates may be used in on-line detection and localization of faults in electrical drives [5].

In [1] a self-tuning controller has been proposed for dc drives; here parameter estimation is based on the recursive least squares method [4] and the control algorithm on the pole placement technique [6]. The pole placement scheme requires solution of a Diophantine equation at each sampling interval; there may be some limitations in applying this scheme for more complicated drives. Also in [1] the motor current is not explicitly included in the adaptive scheme; current peaking is avoided via a fixed nonadaptive, current loop.

In this paper a new self-tuning controller is proposed wherein the control algorithm is based on minimizing the sum of variances of signals occurring in the system. Although the basic methodology is similar to that in [4], an integrator driven by the tracking error is included in the control loop, ensuring robust tracking and excellent transient recovery under load changes, even under imperfect parameter estimation. The resulting integral–proportional self-tuner is believed to be new and novel and is particularly relevant for drive applications. Also, by including the motor current explicitly in the performance index, current peaking is avoided.

In Section II the basic theory of the robust self-tuner is developed when the parameters of the system are known. Section III considers the self-tuning version of the controller for systems with unknown parameters. In Section IV the discrete model of a dc drive is outlined and, using the results of Section II, a self-tuning controller is developed for the drive. To assess the viability of the method, simulation results pertaining to a drive control using the proposed scheme are presented in Section V. In section VI implementation of the proposed self-tuner on a 16-bit microcomputer is discussed. Section VII describes the results of tests performed on a dc machine controlled using the microcomputer based self-tuner.

Paper IPCSD 86-2, approved for publication in this TRANSACTIONS by the Industrial Drives Committee of the IEEE Industry Applications Society. Manuscript released for publication June 3, 1985.

R. Balasubramanian was with the Electrical Engineering Department, University of New Brunswick, Fredericton, NB, Canada E3B 5A3. He is deceased.

K. H. Wong was with the Electrical Engineering Department, University of New Brunswick, Fredericton, NB, Canada. He is presently with the Projects Section, Wisma SES Corporation, 3673 Kuchung, Sarawak, Malaysia.

IEEE Log Number 8611178.

II. A Robust Self-Tuning Controller

A. Development of the Controller

Consider first a system with known parameters. Let the system be described by

$$A(z^{-1})y_1(k) = z^{-d}1B_1(z^{-1})u(k) + \xi_1(k) \tag{1}$$

where

$$A(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_{n_A} z^{-n_A}$$

$$B_1(z^{-1}) = b_{10} + b_{11} z^{-1} + \cdots\ b_{1n_{B_1}} z^{-n_{B_1}}$$

and where

- z^{-1} backward shift operator,
- d_1 system time delay,
- $y_1(k)$ system output at k-th sampling instant,
- $u(k)$ control input at $t = kT$,
- T sampling interval,
- $\{\xi_1(k)\}$ sequence of uncorrelated random variables arising from measurement noise.

The object is to synthesize a control $u(k)$ such that $y_1(k)$ follows a reference signal $w(k)$ with zero steady-state error. We shall design the controller for the case when $w(k)$ is a constant. In this case the internal model of $w(k)$ is an integrator, and for robust control the control loop should necessarily include this internal model driven by the tracking error [7]. Accordingly define

$$e(k) = w(k - d_1) - y(k) \tag{2}$$

and

$$(1 - z^{-1})v(k) = e(k). \tag{3}$$

In practice there are also certain auxiliary outputs $y_i(k)$, $i = 2, \cdots m$ of the system which can be measured and used for control purposes. These satisfy

$$A(z^{-1})y_i(k) = z^{-d_i} B_i(z^{-1})u(k) + \xi_i(k). \tag{4}$$

The control $u(k)$ should be such that

$$\lim_{k \to \infty} E[e(k)] = 0$$

where $E(\cdot)$ stands for the expected value. Also the magnitude of $y_i(k + d_i)$ and $u(k)$ should be restricted. With this in mind the target control is to minimize

$$\begin{aligned} J = {} & \lambda_1 E[P_1(z^{-1})\{y_1(k + d_1) - w(k)\}]^2 \\ & + \sum_2^m \lambda_i E[P_i(z^{-1})y_i(k + d_i)]^2 + \lambda_v E[S(z^{-1})v(k + d_1]^2 \\ & + \lambda_u E[Q(z^{-1})u(k)]^2 \end{aligned} \tag{5}$$

where

$$P_i(z^{-1}) = 1 + p_{i1} z^{-1} + \cdots + p_{in_i} z^{-n_i}$$

$$S(z^{-1}) = 1 + s_1 z^{-1} + \cdots + s_{n_s} z^{-n_s}$$

$$Q(z^{-1}) = 1 + q_1 z^{-1} + \cdots + q_{n_Q} z^{-n_Q}$$

and λ's are non-negative weighting parameters. In the sequel we shall drop z^{-1} whenever this will not cause any confusion.

Consider $P_i y_i(k + d_i)$ first. Evidently

$$P_i y_i(k + d_i) = \frac{B_i P_i}{A} u(k) + \frac{P_i}{A} \xi_i(k + d_i).$$

Let

$$\frac{P_i}{A} = F_i + z^{-d_i} \frac{G_i}{A} \tag{6}$$

where

$$F_i = 1 + f_{i1} z^{-1} + \cdots + f_{i(d_i - 1)} z^{-d_i + 1}.$$

Substituting from (6) and simplifying, one readily gets, say,

$$\begin{aligned} P_i y_i(k + d_i) &= B_i F_i u(k) + G_i y(k) + F_i \xi_i(k + d_i) \\ &= \phi_i(k + d_i/k) + F_i \xi_i(k + d_i). \end{aligned} \tag{7}$$

In (7) one can recognize $\phi_i(k + d_i/k)$ to be the d_i step-ahead predictor of $P_i y_i(k + d_i)$.

Consider $Sv(k + d_1)$ next. We have

$$Sv(k + d_1) = \frac{S}{(1 - z^{-1})} [w(k) - y(k + d)].$$

Let

$$\frac{S}{P_1(1 - z^{-1})} = F' + \frac{z^{-d_1} G'}{P_1(1 - z^{-1})} \tag{8}$$

where

$$F' = 1 + f_1' z^{-1} + \cdots + f'_{(d_1 - 1)} z^{-d_1 + 1}.$$

Then substituting from (8) we have

$$Sv(k + d_1) = G' v(k) + F' P_1 w(k) - F' P_1 y_1(k + d_1).$$

We now define

$$F' F_1 = F_a + z^{-d_1} F_b \tag{9}$$

where

$$F_a = 1 + f_{a1} z^{-1} + \cdots + f_{a(d_1 - 1)} z^{-d_1 + 1}.$$

Substituting from (9) and (7) and simplifying one gets, say,

$$\begin{aligned} Sv(k + d_1) &= G' v(k) + F' P_1 w(k) - (F' G_1 + A F_b) y_1(k) \\ &\quad - B_1 F_a u(k) + F_a \xi_1(k + d_1) \\ &= \psi(k + d_1/k) + F_a \xi_1(k + d_1). \end{aligned} \tag{10}$$

Again $\psi(k + d_1/k)$ is the d_1 step-ahead predictor of $Sv(k + d_1)$.

The steps leading to (7) and (10) look involved. However, the central idea is to be able to write $P_i y_i(k + d_i)$ and $Sv(k + d_1)$ in terms of quantities known at the k-th sampling instant. In particular, in (7) and (10) the noise terms are uncorrelated with the prediction values. This enables one to write J of (5) in the form

$$\begin{aligned} J = {} & \lambda_1 [\phi_1(k + d_1/k) - w(k)]^2 + \sum_2^m \lambda_i \phi_i^2(k + d_i/k) \\ & + \lambda_v \psi^2(k + d_1/k) + \lambda_u [Qu(k)]^2 + \sum_1^m E\{F_i \xi_i(k + d_i)\}^2 \\ & + E\{F_a \xi_1(k + d)\}^2. \end{aligned} \tag{11}$$

The noise terms are thus independent of $u(k)$ and J is a minimum with respect to $u(k)$ when $\partial J/\partial u(k)$ is set to zero. This yields after some simplification

$$u(k) = -[\Sigma H_i Y_i(k) + Mv(k) + Lw(k)]/J \tag{12}$$

where

$$\left.\begin{aligned} H_1 &= \lambda_1 b_{10} G_1 + \lambda_v b_{10}[F' G_1 + AF_b] \\ H_i &= \lambda_i b_{i0} G_i;\ i > 1 \\ M &= -\lambda_v b_{10} G' \\ L &= -\lambda_1 b_{10} - \lambda_v b_{10} F' P_1 \end{aligned}\right\} \tag{13}$$

and

$$J = \sum_{1}^{m} \lambda_i b_{i0} B_i F_i + \lambda_v b_{10} B_1 F_a + \lambda_u Q \ \}.$$

The control $u(k)$ given by (12) may appear to be complicated. However, it simply involves feedback of measured quantities $y_i(k)$, $y_i(k-1)$ and is easily implemented. The parameters of H_i, M, $\cdots$ are readily found once the system parameters are known or are estimated.

B. Properties of the Proposed Controller

1) Transfer Function Relating Set Point and Output: Since the system is linear, superposition holds; hence we shall take $\xi_i = 0$, $i = 1, \cdots m$ and find the pulse transfer function relating $w(k)$ to $y_1(k)$. Substituting from (12) in (1) and simplifying one gets

$$y_1(k) = \frac{L(1-z^{-1}) - Mz^{-1}}{(JA + \Sigma z^{-d_i} B_i H_i)(1-z^{-1}) - M} w(k). \tag{14}$$

2) Closed-Loop Characteristic Equation: From (14) the closed-loop characteristic equation is given by

$$D = (JA + \Sigma z^{-d_i} B_i H_i)(1-z^{-1}) - M.$$

Substituting from (13) and simplifying one readily gets

$$D = \left[\sum_{1}^{m} \lambda_i B_i P_i b_{i0} + \lambda_u QA\right](1-z^{-1}) + \lambda_v B_1 b_{10} S = 0. \tag{15}$$

Thus the weighting constants λ's provide some flexibility in designing the control system. Evidently for the system to be stable, $S = 0$ or $B = 0$ should not have a root at unity.

3) Tracking Error: The tracking error was defined in (2) as

$$e(k) = w(k - d_1) - y(k).$$

Substituting from (14) and simplifying one has

$$e(k) = \frac{-L(1-z^{-1})}{D} w(k). \tag{16}$$

Since the reference signal is a constant w, the z-transform of $e(k)$ is

$$E(z^{-1}) = -\frac{L}{D} w \tag{17}$$

and

$$\lim_{k\to\infty} e(k) = \lim_{z\to 1} (1-z^{-1}) E(z^{-1}) = 0 \tag{18}$$

provided that the closed-loop system is stable. Thus the steady-state tracking error is zero.

4) Disturbance Rejection Property: We have assumed until now that the disturbance or load on the system is zero. When a constant disturbance $r(k) = r$ is acting on the system, (1) and (4) are appropriately modified as

$$Ay_i(k) = z^{-d_i} B_i[u(k) + Rr(k)] \tag{19}$$

where

$$R(z^{-1}) = r_0 + r_1 z^{-1} + \cdots + r_{n_R} z^{-n_R}.$$

If the control $u(k)$ is chosen as per (12), the pulse transfer function relating $r(k)$ to $e(k)$ is readily evaluated to be

$$e(k) = \frac{z^{-d_1} B_1 RJ(1-z^{-1})}{D} r(k). \tag{20}$$

Evidently, as before, for a constant disturbance r

$$\lim_{k\to\infty} e(k) = 0 \tag{21}$$

provided that the closed-loop system is stable. Thus the proposed controller asymptotically rejects constant disturbances.

5) Robustness of the Controller: As noted earlier the parameters of the system are not known in advance and have to be estimated in real time using input–output data. In practice this is done using a microcomputer and A/D converters of limited word lengths. The control $u(k)$ of (12) is implemented using the estimated values of parameters. Let

$$u_1(k) = -[\Sigma \hat{H}_i y_i(k) + \hat{M}v(k) + \hat{L}w(k)]/\hat{J} \tag{22}$$

where $\hat{H}_i$, $\cdots$ are found from the estimated parameters. Then one can easily find the pulse transfer functions relating $w(k)$ to $e(k)$ and $r(k)$ to $e(k)$ as

$$e(k) = -\frac{\hat{L}(1-z^{-1})}{\hat{D}} w(k) \tag{23}$$

and

$$e(k) = \frac{z^{-d} B_1 R\hat{J}(1-z^{-1})}{\hat{D}} r(k) \tag{24}$$

where

$$\hat{D} = (\hat{J}A + \Sigma z^{-d_i} B_i \hat{H}_i)(1-z^{-1}) - \hat{M}. \tag{25}$$

Evidently the factor $(1 - z^{-1})$ continues to appear on the numerators of these transfer functions and hence

$$\lim_{k\to\infty} e(k) = 0$$

provided that the closed-loop system continues to be stable.

Thus the ability of the controlled system to track the reference and reject the disturbance is no longer subordinate to the accuracy and speed of estimating the parameters. Tracking error continues to be asymptotically zero, even when the parameter estimates are not perfect and have a small bias. It is emphasized that a conventional minimum variance self-tuner does not have this property [4], [6].

III. The Adaptive Controller

When the parameters of the system are not known or are varying slowly they are estimated in real time using the input–output data. The specific parameters estimated depend on the application. For example, it is often unnecessary to estimate the parameters of the system, that is, those associated with A, $B_1, \cdots, B_m$. It is quite adequate to estimate the so-called prediction or regression parameters associated with H_i, M, L, and J in (12). However, in applications involving controlled electrical drives, it has been reported that knowledge of system parameters may be useful in fault detection or isolation [5]. Accordingly in this work we shall estimate the parameters of the system and then use these parameters to generate the control of (12) by using the identities in (6).

It is also worth noting that a practical system is always subject to unknown load disturbances. It is therefore necessary to estimate the system parameters in the presence of such loads. Thus, although knowledge of the disturbance is not required to generate the control $u(k)$ of (12), one has to estimate it just to be able to detect and estimate parameter changes. Hence we write (19) in the form

$$y_i(k) = -a_1 y_i(k-1) - a_2 y_i(k-2) \cdots + b_{i0} u(k-d_i) + \cdots + z^{-d_i} B_i R r(k) + \xi_i(k).$$

Since the disturbance $r(k)$ is assumed to be a constant, $z^{-d_i} B_i R r(k)$ is also a constant and we shall denote it by δ_i. Then the estimation equation becomes

$$y_i(k) = -a_1 y_i(k-1) - a_2 y_i(k-2) \cdots + b_{i0} u(k-d_i) + \cdots + \delta_i + \xi_i(k). \quad (26)$$

With

$$\underline{X}^T(k) = [-y_i(k-1) - y_i(k-2) \cdots u(k-d_i) \cdots 1]$$

and

$$\underline{\theta}^T = [a_1 a_2 \cdots b_{10} \cdots b_{20} \cdots b_{i0} \cdots b_{m0} \cdots \delta_1 \cdots \delta_m]$$

(26) can be written as

$$\rho(k) = y_i(k) = \underline{X}^T(k) \underline{\theta} + \xi_i(k). \quad (27)$$

The unknown parameters θ are estimated using the recursive least squares method or a variation thereof. The estimate of θ after k samples are processed is denoted by $\hat{\theta}(k)$. The recursive least squares algorithm now yields

$$\underline{\hat{\theta}}(k) = \underline{\hat{\theta}}(k-1) + \underline{K}(k)[\rho(k) - \underline{x}^T(k-d)\underline{\hat{\theta}}(k-1)]\}$$

$$\underline{K}(k) = \underline{P}(k)\underline{x}(k-d)/[\beta + \underline{x}^T(k-d)\underline{P}(k)\underline{x}(k-d)]\}$$

$$\underline{P}(k+1) = [\underline{I} - \underline{K}(k)\underline{x}^T(k-d)\underline{P}(k)]/\beta\} \quad (28)$$

where β is the so-called exponential forgetting factor.

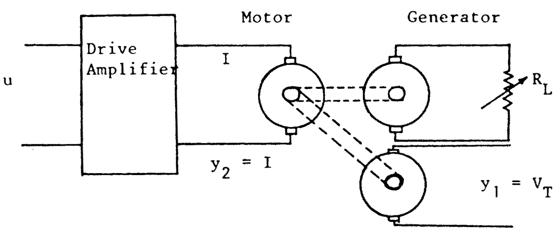


Fig. 1. Schematic of experimental drive system.

The self-tuner is usually implemented in a small or microcomputer of limited word length. Then the least squares algorithm may suffer from convergence problems due to round-off error [4]. To avoid these difficulties one often uses the square root algorithm. In this case, estimation proceeds according to the following scheme

$$\underline{\hat{\theta}}(k) = \underline{\hat{\theta}}(k-1) + \underline{K}(k)[\rho(k) - \underline{x}^T(k-d)\underline{\hat{\theta}}(k-1)\}$$

$$\underline{K}(k) = \underline{g}^{\rho}/\sigma_{\rho}^2\}$$

$$\underline{P}(k+1) = \underline{T}(k+1)\underline{T}^T(k+1)\} \quad (29)$$

where

$$\{\underline{T}(k+1)\}_{ij} = \sigma_{j-1}[\{T(k)\}_{ij} + f_i t_i^{j-1}/\sigma_{j-1}^2]/\sigma_j\sqrt{\beta}$$

$$\sigma_0 = \sqrt{\beta}$$

$$\sigma_j = (\sigma_{j-1}^2 + f_j^2)^{1/2}$$

$$f_j = \sum_{i=1}^{j} \{\underline{T}(k)\}_{ij} \{\underline{X}(k-d)\}_i$$

and

$$t_i^l = \begin{cases} \sum_{m=i}^{\rho} \{\underline{T}(k)\}_{im} f_m \\ 0 \text{ for } i > l. \end{cases}$$

Thus after the sample $y_i(k)$ is made available, one can set up the vector $\underline{X}^T(k)$ based on quantities known at that instant and find $\underline{\hat{\theta}}(k+1)$.

IV. Application to DC Machines

The self-tuning method developed in Sections II and III will now be utilized to design a controller for a dc machine. The experimental setup, to be discussed in more detail in Section VII, consists of a dc motor coupled to a dc generator as shown in Fig. 1. When the load on the system is changed by changing R_L, as we shall show below, the entire dynamics of the system changes. Hence this setup is a natural choice for testing adaptive control algorithms. The parameter estimation algorithms should be capable of tracking the load-dependent system parameters. An experimental setup similar to that in Fig. 1 has been used in other microcomputer-based adaptive speed control studies [8].

A. Discrete Model of the Machine

If $ku(t)$ is the voltage applied across the terminals of the armature, the dynamic equations describing the system of Fig. 1 are readily written as

$$\left.\begin{aligned} ku(t) &= R_m i_m(t) + e_b(t) \\ T_m(t) &= k_m i_m(t) \\ J\dot{\omega}(t) + B\omega(t) &= T_m(t) - T_L(t) \\ e_b(t) &= k_m \omega(t) \\ e_g(t) &= k_g \omega(t) \\ i_g(t) &= e_g(t)/(R_g + R_L) \\ T_L(t) &= k_g i_g(t) \\ y_1(t) &= k_T \omega(t) \end{aligned}\right\} \tag{30}$$

where

$i_m(t)$ ($i_g(t)$)	motor (generator) current,
$R_m(R_g)$	armature resistance of motor (generator),
$e_b(t)$	motor back EMF,
$e_g(t)$	generator voltage,
$T_m(t)$ ($T_L(t)$)	motor (load) torque,
J	total inertia,
B	total friction constant,
k_T	tachometer constant.

We have assumed that the armature inductances of the motor and generator are negligible. It is easy to modify the results when the inductances are not negligible as we shall demonstrate in Section VII.

It is now straightforward to simplify the identities in (30) to get the transfer function $G_1(s)$ relating the output $y_1(s)$ to $u(s)$ as, say,

$$G_1(s) = y_1(s)/u(s) = \frac{kk_m k_T}{R_m Js + R_m B + k_m^2 + \alpha k_g^2} = \frac{kk_m k_T}{D} \tag{31}$$

where $\alpha = R_m/(R_g + R_L)$. It is now clearly seen that the time constant and dc gain of the system depend on R_L.

The auxiliary output available in this particular drive application is the motor armature current. The control should be such that current peaks are avoided. If k_i is the constant associated with the current transducer, the auxiliary output $y_2(t)$ is

$$y_2(t) = k_i i_m(t). \tag{32}$$

Again it is straightforward to get the transfer function $G_2(s)$ relating $y_2(s)$ and $u(s)$ as

$$G_2(s) = y_2(s)/u(s) = \frac{[Js + B + \{k_g^2/(R_g + R_L)\}]}{D} k_i k. \tag{33}$$

The self-tuning control generated by the microcomputer will be applied to the motor after suitable power amplification via a D/A converter. Thus the zero-order hold equivalents of $G_1(s)$ and $G_2(s)$ are of interest. These are easily found to be of the form [11]

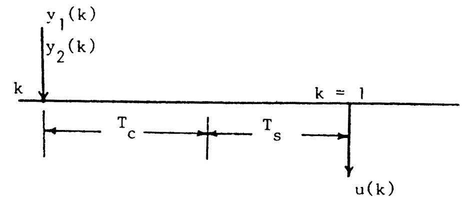


Fig. 2. Timing diagram. T_s added to T_c to delay $u(k)$ by one sampling interval.

$$G_1'(z^{-1}) = \frac{z^{-1} b_{10}}{1 + a_1 z^{-1}} \tag{34}$$

and

$$G_2'(z^{-1}) = \left[\frac{1 + (a_1 - \mu b_{10}) z^{-1}}{1 + a_1 z^{-1}}\right] \eta \tag{35}$$

where $\eta = kk_i/R_m$ and $\mu = k_m/kk_T$.

In practical implementation of the control proposed in Sections II and III additional care is required. The control $u(k)$ of (12) is generated subsequent to sampling the outputs $y_1(k)$ and $y_2(k)$. In practice it is not possible to generate $u(k)$ instantaneously. Parameter estimation and control design algorithms consume an appreciable fraction of the sampling interval as shown in Fig. 2. It is not impossible to account for the computation time τ_c in this figure and modify $u(k)$ accordingly. However, the most direct way of accounting for τ_c is to introduce a fictitious additional delay τ_s and send $u(k)$ to the D/A channel exactly at the $(k + 1)$-th sampling instant. This, of course, has the effect of introducing an additional delay of one sampling interval. Thus the discrete model of the machine is in effect changed from (34) and (35) to

$$G_1'(z^{-1}) = \frac{z^{-2} b_{10}}{1 + a_1 z^{-1}} \tag{36}$$

and

$$G_2'(z^{-1}) = \left[\frac{z^{-1}(1 + (a_1 - \mu b_{10}) z^{-1})}{1 + a_1 z^{-1}}\right] \eta. \tag{37}$$

Hence (26) in the specific case of dc drives becomes

$$(1 + a_1 z^{-1}) y_1(k) = b_{10} u(k-2) + \delta_1 + \xi_1(k) \tag{38}$$

and

$$(1 + a_1 z^{-1}) y_2(k) = \eta u(k-1) + (a_1 - \mu b_{10}) \cdot u(k-2) + \delta_2 + \xi_2(k). \tag{39}$$

B. Development of the Control Algorithm

Although in Section II the self-tuner was developed in its generality for practical use in drive applications, it is appropriate to take $P_i = Q = S = 1$. In this case with

$$A(z^{-1}) = 1 + a_1 z^{-1}; \; d_1 = 2, \; d_2 = 1$$

one readily gets from (16) and (8)

$$F_1=1-a_1z^{-1};\ F_2=1$$
$$G_1=a_1^2;\ G_2=-a_1$$
$$F'=1+z^{-1};\ F_a=1+(1-a_1)z^{-1}$$
$$G'=1;\ F_b=-a_1. \tag{40}$$

Hence the control algorithm of (12) can be written as

$$u(k)=-[h_{10}y_1(k)+h_{20}y_2(k)+m_0v(k)+1_0w(k)+1_1w(k-1)+j_1u(k-1)]/j_0$$

where

$$h_{10}=[\lambda_1a_1^2-\lambda_va_1(1-a_1)]b_{10}$$
$$h_{20}=-\lambda_2a_1\eta$$
$$m_0=-\lambda_vb_{10}$$
$$l_0=-(\lambda_1+\lambda_v)b_{10}$$
$$l_1=-\lambda_vb_{10}$$
$$j_1=\lambda_2\eta^2(a_1-\eta b_{10})-\lambda_1a_1b_{10}^2+\lambda_vb_{10}^2(1-a_1)$$

and

$$j_0=(\lambda_1+\lambda_v)b_{10}^2+\lambda_2\eta^2+\lambda_u. \tag{41}$$

C. Parameter Estimation and Self-Tuning Control

When the parameters of the drive are not known or are slowly varying they are estimated in real time using input–output data. Parameter estimation algorithms are by now well established; however, with particular reference to controlled electrical drives the following points are worth noting.

- If estimation is performed using a microcomputer of 16 bits or less, many ordinary recursive least squares schemes suffer from divergence problems [1] and considerable care is required in implementing the adaptive controller.
- Estimation using a modified steepest descent scheme is possible. This is easy to implement in a microcomputer but the rate of convergence is so slow that it is of very limited use in adaptive control applications [9].
- The square root estimator outlined in Section III does require additional computations, and for many practical systems with a large number of parameters, programming in a lower-level language may be necessary. However, the rate of convergence of the estimates and their reliability often justify these programming complexities, particularly in self-tuning applications. Accordingly we shall use the square root algorithm for estimation in this paper.

From (38) and (39) we note that six parameters have to be estimated. Hence the parameter vector is

$$\underline{\theta}^T=(a_1b_{10}b_{20}b_{21}\delta_1\delta_2). \tag{42}$$

The two observation equations can be written as

$$y_1(k)=[-y_1(k-1)-u(k-2)\ 0\ 0\ 1\ 0]\underline{\theta}+\xi_1(k) \tag{43}$$

and

$$y_2(k)=[-y_2(k-1)0-u(k-1)-u(k-2)\ 0\ 1]\underline{\theta}+\xi_2(k). \tag{44}$$

Thus, after $y_1(\cdot)$ and $y_2(\cdot)$ are sampled at the k-th sampling constant, the square root algorithm of (29) can be used twice to update the parameters. This in effect updates the A-parameters twice and all other parameters once per sampling. The above algorithm has been simulated and works reasonably well.

However, inspection of (38) and (39) indicates that whenever R_m is known and does not change it is sufficient to use (38) and estimate only a_1, b_{10}, and δ_1. The parameters b_{20} and b_{21} are found from a_1 and b_{10}; δ_2 need not be estimated since it is not needed in generating $u(k)$. Even when R_m does change, one can recognize that

$$\rho_2(k)=k_i[ku(k-1)-y_1(k)k_m/k_T]=y_2(k)R_m+\text{noise}.$$

Hence parameter estimation can be carried out using the observation equation

$$\rho_1(k)=y_1(k)=[-y_1(k-1)-u(k-2)\ 1]\begin{bmatrix}a_1\\ b_{10}\\ \delta_1\end{bmatrix}+\xi_1(k) \tag{45}$$

and the surrogate observation equation

$$\rho_2(k)=y_2(k)R_m+\text{noise}. \tag{46}$$

The parameters

$$\underline{\theta}_1^T=[a_1b_{10}\delta_1]$$

can be estimated from (45) using the algorithm of (29). Also, R_m can be separately estimated from (46). The estimation thus becomes decoupled; there are also some advantages in explicitly estimating R_m in fault diagnosis [5].

Comparing the two methods for the aforementioned parameter estimation proposal, it is clear that the first involves operations on a 6×6 matrix whereas the second involves a 3×3 matrix and a scalar. The first method also estimates more parameters than necessary. It is known that such overparametrization usually results in slow convergence. For these reasons in this work we shall adopt the second scheme in conjunction with the square root algorithm.

Thus the procedure for generating the self-tuning control can be summarized as follows.

Step 1 (Initialization): Choose a sampling interval consistent with the anticipated speed of response of the drive. Choose from past experience the initial value $\hat{\underline{\theta}}_1(0)$ of the parameters. It is usually convenient to take $\hat{\underline{P}}(0)$ as a diagonal matrix; for example one may take $\hat{\underline{P}}(0)=\alpha\underline{I}$ where α is large.

Step 2: At the k-th sampling instant, sample $y_1(k)$ and $y_2(k)$. Using (45) and (46) find $\hat{\underline{\theta}}_1(k)$, $\hat{\underline{T}}(k+1)$, and $\hat{R}_m$.

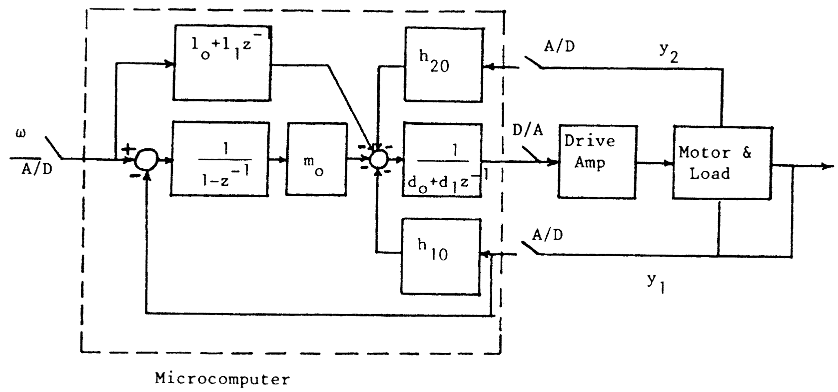


Fig. 3. Block diagram of proposed self-tuning controller for dc drives.

Step 3: Using the estimated parameters find $\hat{h}_{10}$, $\hat{h}_{20}$, $\hat{m}_0$, $\hat{l}_0$, $\hat{l}_1'$, $\hat{j}_0$, and $\hat{j}_1$. Find the control $u(k)$ as per (41).

Step 4: At the $(k + 1)$-th sampling instant, output $u(k)$ to the D/A channel. Proceed to Step 2.

Although we have not proved the convergence of the self-tuner it can be shown that under certain general conditions the scheme converges [4].

D. Characteristics of the Controlled System

- A block diagram of the proposed controller is shown in Fig. 3. The closed-loop system includes an integrator driven by the tracking error as well as a "proportional" control derived from the output. The current-feedback loop limits current peaks. The feed-forward path guarantees fast tracking as the set point changes. Thus the system possesses fast transient response and excellent transient recovery under disturbance. It is interesting to note that except for the addition of the feed-forward path, the controller obtained here is very similar to the recently proposed IP controller [10]. However, the IP controller is a fixed controller without any self-tuning capabilities.
- Substituting the control of (40) in (38) and simplifying one can readily get the characteristic equation of the system as

$$\{\lambda_1 b_{10}^2 - \lambda_2 b_{20}(b_{20}+b_{21}z^{-1}) + \lambda_u(1+a_1z^{-1})\}(1-z^{-1}) + \lambda_v b_{10}^2 = 0. \quad (47)$$

  Thus a wide variety of controllers can be designed by suitable choice of λ's.
- As discussed in Section IIB, the controlled system rejects constant disturbances and tracks constant set points with zero steady-state error. Also these properties are robust with respect to the accuracy and rate of parameter estimation.

V. Digital Simulation of the Proposed Controller

With a view to assess the performance of the proposed controller, digital simulations were undertaken. The nominal model of the system is given by

$$(1-0.7575z^{-1})y_1(k) = 0.1021u(k-2) + \xi_1(k) \quad (48)$$

and

$$(1-0.7575z^{-1})y_2(k) = 0.2u(k-1) - 0.1719u(k-2) + \xi_2(k). \quad (49)$$

The numerical values of the parameters chosen correspond to those of the experimental system described in Section VII. A large number of simulations were carried out and due to space limitations only a selected few are given below.

Case (i): The parameters of the system are assumed known. For various combinations of λ_1, λ_2, λ_u, and λ_v the transient performance of $y_1(k)$ and $y_2(k)$ to a unit step reference is shown in Fig. 4. As can be seen from Fig. 4(b), the effect of increasing λ_2 is to limit current peaks. Also, by choosing the λ's suitably, effective shaping of the transient performance is possible.

Case (ii): The parameters of the system change and the model of the system now becomes

$$(1-0.8291z^{-1})y_1(k) = 0.064u(k-2) + \xi_1(k) \quad (50)$$

and

$$(1-0.8291z^{-1})y_2(k) = 0.2u(k-1) - 0.179u(k-2) + \xi_2(k). \quad (51)$$

As we shall see in Section VII this model is obtained when the inertia of the rotor is increased. If one uses a fixed controller of the form of (12) based on the original set of parameters of Case (i), the resulting performance is as shown in Fig. 5. If, however, the control is such that it tunes itself to the new set of parameters, the resulting response is superior as shown in Fig. 5. Thus the advantage of an adaptive controller over a fixed controller is evident.

Case (iii): The model of the system now is the same as in Case (i) with a constant disturbance added. Thus

$$(1-0.7575z^{-1})y_2(k) = 0.1021u(k-2) + 0.3 + \xi_1(k). \quad (52)$$

All the parameters of the system are assumed unknown and are estimated using the square root estimator of Section IVC. The initial values $a_1(0)$ and $\delta_1(0)$ are taken as 1 and $b_{10}(0)$ as 0.075. The initial variance was assumed to be $\underline{P}(0) = \alpha \underline{I}$, $\alpha = 900$, and $\beta = 0.98$. The reference was taken as a square wave to

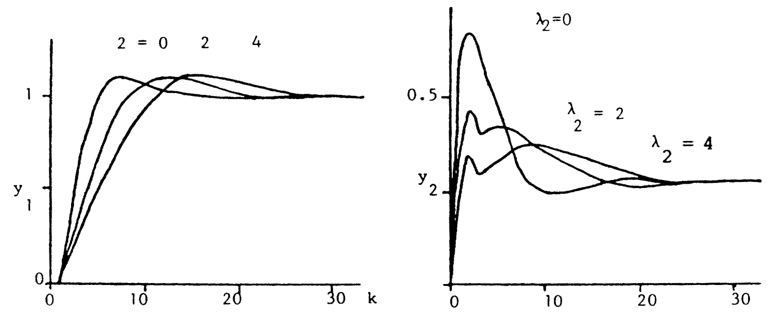


Fig. 4. Performance of proposed controller to unit step refernece. ($\lambda_1 = 1$, $\lambda_v = 1$, $\lambda_u = 0.05$.)

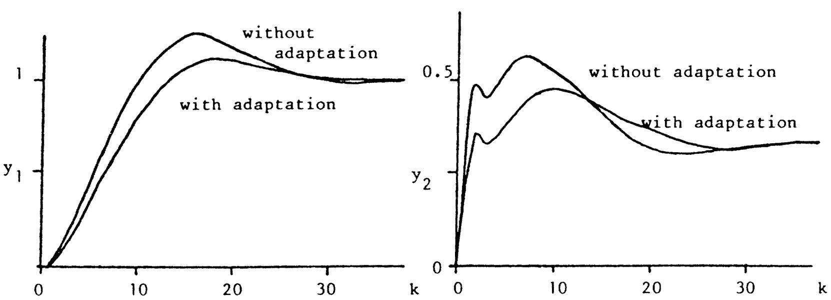


Fig. 5. Performance of controller with and without adaptation under parameter change. ($\lambda_1 = \lambda_v = 1$, $\lambda_u = 0.05$, $\lambda_2 = 2$.)

simulate varying set points. The results corresponding to one set of λ's are given in Fig. 6, where σ is the standard deviation of $\xi_1(k)$. Thus, even under a large amount of measurement noise, the self-tuner operates very satisfactorily.

To study the effect of armature inductance the observation equation (43) is taken as

$$(1+a_1z^{-1}+a_2z^{-2})y_1(k)=z^{-2}(b_{10}+b_{11}z^{-1})u(k)+\delta_1+\xi_1(k). \quad (53)$$

Evidently the additional parameters a_2 and b_{11} arise out of the armature inductance. The parameter estimation routine now included estimation of a_2 and b_{11} along with a_1, b_{10}, and δ_1. As can be seen from Fig. 6, the estimation algorithm correctly identified a_2 and b_{11} as zero. Parameter estimation is effectively completed in about 15 iterations. Other simulations included the effect of changing R_m and J, the effect of applying a sudden load R_L, and the effect of choosing different λ's. In all cases the performance of the self-tuner was satisfactory. The effect of these changes on system performance will be studied in Section VII on the actual test system.

VI. Microcomputer Implementation

Encouraged by the preliminary simulations described in Section V, it was decided to implement the robust self-tuner on a microcomputer system. A schematic of the test setup is shown in Fig. 7. The signal conditioning units are appropriate gainers/attenuators so that maximum advantage can be taken of the resolution of A/D converters.

The microcomputer used in this study is based on an INTEL 8086 CPU cascaded with an 8087 coprocessor, allowing fast and accurate execution of floating-point arithmetic. There is also a timer board with two 8253 timers and a 12-bit A/D–D/A converter board with eight A/D channels and six D/A channels with ±10-V full-scale range.

The self-tuner routines were written in assembler code. The routines are general and not restricted to the particular model derived in Section IVA. Thus it is quite easy to adapt the routines so that more than three parameters can be estimated, perhaps because of a large series inductance in the motor armature circuit. Details of the hardware and the routines are available in [11].

VII. Results

As shown in Fig. 7 the test system consisted of a dc motor coupled to an almost identical generator and a tachogenerator. The gains on the signal conditioning units and the power amplifier were set to take maximum advantage of the dynamic range of A/D and D/A converters. Of course appropriate scaling was done in the microcomputer in generating $u(k)$ so as to account for these gains. The transfer function from $U(s)$ to $Y_1(s)$ under no-load was

$$\frac{Y_1(s)}{U(s)}=\frac{0.421}{1+0.072s}$$

as was measured in an open-loop step response test. A sampling interval of 20 ms was chosen. It should be noted that estimation of the six parameters, namely a_1, a_2, b_{10}, b_{11}, δ_1, and R_m and generation of $u(k)$ as per (41) took only about 10 ms; hence the sampling could have been reduced to 10 ms. However, a T of 20 ms was considered reasonable.

For this sampling interval and for the values of k_i and k_T known for the experimental system, the discrete model under no-load is

$$(1-0.7575z^{-1})y_1(k)=0.1021u(k-2)+\xi_1(k) \quad (54)$$

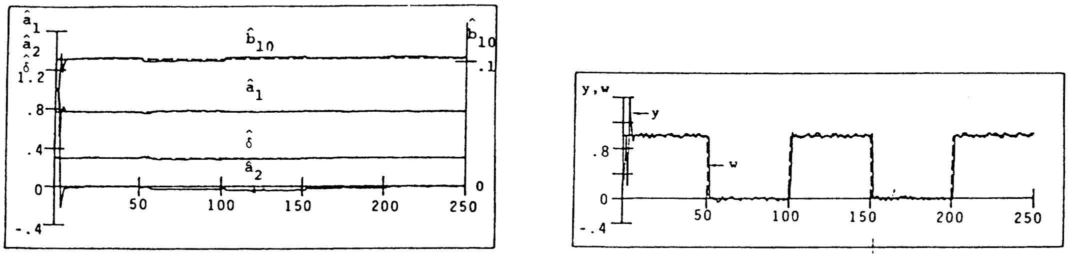


Fig. 6. Performance of the estimator and controller. ($\lambda_1 = \lambda_v = 1$, $\lambda_u = 0.05$, $\lambda_2 = 2$, $\lambda = 0.1$.)

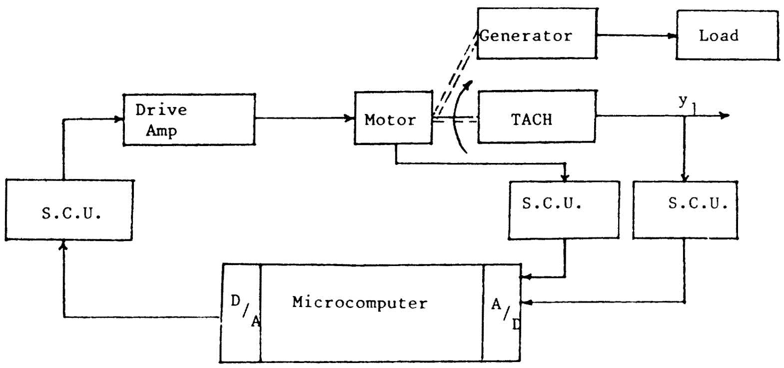


Fig. 7. Microcomputer-based self-tuning drive system. (S.C.U. = signal conditioning unit.)

and

$$(1 - 0.7575z^{-1})y_2(k)0.2u(k-1) - 0.1719u(k-2) + \xi_2(k). \tag{55}$$

A large number of tests were carried out and we describe a few of these below.

Case (i): The reference w is a constant plus a square wave signal. The latter is included to simulate varying set points. The machine was started from rest with the initial values of parameters in the estimation algorithm as $\hat{a}_1(0) = 0$, $\hat{b}_{10}(0) = 1$. Also, $\underline{P}(0) = 900\underline{I}$ and $\beta = 0.98$. As is clear from Fig. 8, the parameters $\hat{a}_1$ and $\hat{b}_{10}$ are correctly estimated and the output $y_1(k)$ follows the reference with zero steady-state error. Of course the excessive overshoots in $y_1(k)$ in the first two cycles are due to the choice of $\hat{a}_1(0)$ and $\hat{b}_{10}(0)$. These were deliberately chosen as above to study the performance of the self-tuner under a large error in $\hat{a}_1(0)$, $\hat{b}_{10}(0)$. It was ascertained that when $\hat{a}_1(0)$ and $\hat{b}_{10}(0)$ were chosen nearer to their true values of 0.7575 and 0.1021, these overshoots decreased. The parameter estimates shown are recorded from other D/A channels.

To study the performance of the estimation algorithm and the self-tuning controller when armature inductance effects are to be accommodated in them, the system model was assumed as

$$(1 - a_1z^{-1} + a_2z^{-2})y(k) = z^{-1}(b_{10} + b_{11}z^{-1})u(k) + \xi_1(k)$$

and the parameters a_2 and b_{11} were also estimated and used in generating the control. As shown in Fig. 8 the estimation algorithms correctly identified $\hat{a}_2$ to be zero.

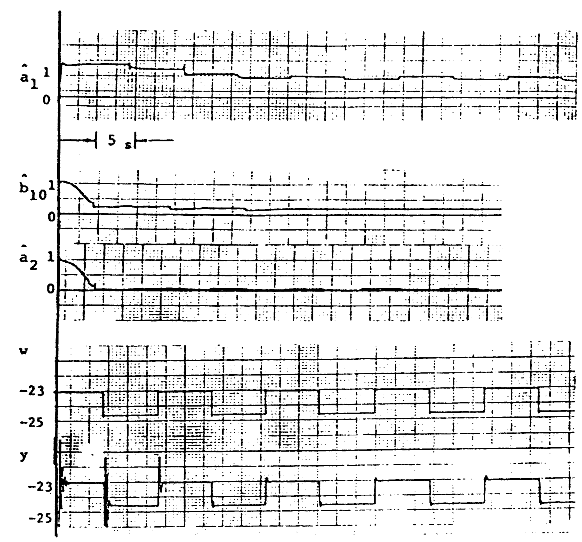


Fig. 8. Performance of controlled system to step changes in reference.

Case (ii): To study the normal starting performance of the self-tuner, the set point was changed slowly to achieve full speed by varying w from a small value to that corresponding to rated speed. As shown in Fig. 9 the resulting changes in the system were adequate to correctly estimate the parameters. This shows that parameter estimation is possible under normal operating conditions. Note that the system follows the set point with zero steady-state error.

Case (iii): The system was started under no-load with the set point w at a constant value. The values of $\hat{a}_1(0)$, $\hat{b}_{10}(0)$, etc., were as in Case (i). Also $\underline{P}(0) = 900\underline{I}$ and $\beta = 0.98$. After about 30 s, when parameter estimation is completed, a

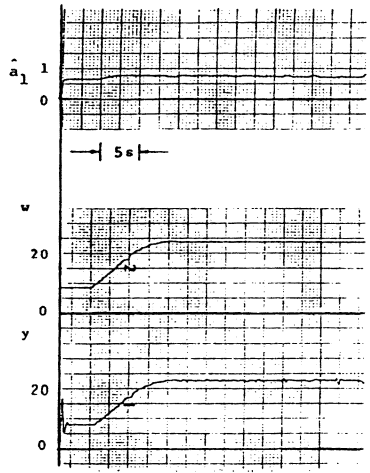


Fig. 9. Performance of controlled system under normal starting conditions.

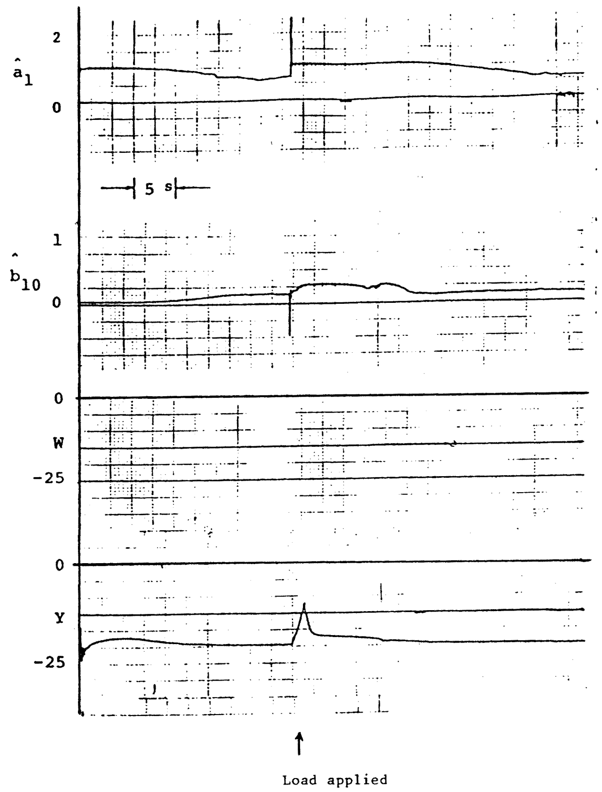


Fig. 10. Performance of controlled system under load changes.

step load is included by switching in a load resistance R_L. The system model changes from (54) and (55) to

$$(1-0.727z^{-1})y_1(k)0.1u(k-2)+\xi_1(k) \qquad (56)$$

and

$$(1-0.727z^{-1})y_2(k)=0.2u(k-1)-0.164u(k-2)+\xi_2(k).$$

As shown in Fig. 10, the self-tuner estimates the new parameters reasonably fast and maintains a zero state error under load.

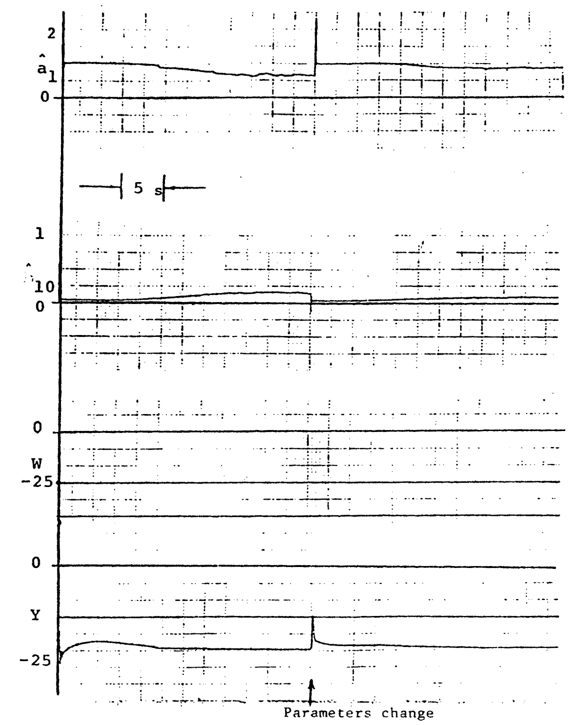


Fig. 11. Performance of controlled system under parameter changes.

Case (iv): The system was started again under no-load, and when parameter estimation is completed, an additional series resistance R was switched in the motor armature circuit. This has the effect of essentially increasing the rotor inertia, and the model after the resistance is switched in becomes

$$(1-0.8291z^{-1})y_1(k)=0.064u(k-2)+\xi_1(k)$$

and

$$(1-0.8291z^{-1})y_2(k)=0.2u(k-1)-0.179u(k-2)+\xi_2(k).$$

As shown in Fig. 11, the self-tuner tracks the changed parameters and the output follows the reference with zero steady-state error.

VIII. Conclusion

A robust self-tuning controller for the control of dc machines is developed. The controller consists of an estimator that estimates the parameters of the machine in real time and a controller that generates the control input to the machine. The controller is capable of tracking constant reference inputs and rejecting constant load disturbances with zero steady-state error even under imperfect parameter estimation. It is shown that the proposed controller has the structure of an IP controller; in addition it also possesses adaptation capability. It is also demonstrated that by choosing the parameters in the control algorithm suitably, current peaks can be restricted.

The proposed controller is first simulated digitally and its properties are ascertained. It is then implemented in a microcomputer system to control a dc motor. The results of various experiments conducted on the system indicate that the proposed scheme is viable.

References

[1] A. Brickwedde, "Microprocessor-based adaptive control for electrical drives," in *Proc. IFAC Conf. on Control in Power Electron. and Electric. Drives*, Lausanne, Switzerland, 1983, pp. 119–124.

[2] B. Courtiol and I. D. Landau, "High speed adaptation system for controlled electrical drives," *Automatica*, vol. 11, pp. 119–127, 1975.

[3] A. Balestrino, G. DeMaria, and L. Sciaviceo, "Adaptive control design in servosystems," in *Proc. IFAC Conf. on Control in Power Electron. and Electric. Drives*, Laussanne, Switzerland, 1983, pp. 125–131.

[4] D. W. Clarke and P. J. Gawthop, "Self-tuning control," in *Proc. IEE*, vol. 126, 1979, pp. 633–640.

[5] G. Geiger, "On-line fault detection and localization in electrical d.c. drives based on process parameter estimation and statistical decision methods," in *Proc. IFAC Conf. on Control in Power Electron. and Electric. Drives*, Lausanne, Switzerland, 1983, pp. 475–483.

[6] K. J. Astrom and B. Wittenmark, "Self-tuning controllers based on pole-zero placement," in *Proc. IEE*, 1980, 127D, pp. 120–130.

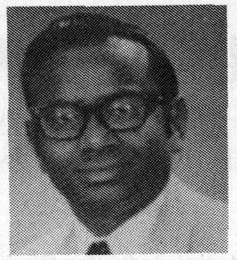
[7] E. J. Davison, "The robust control of a servomechanism problem for linear time-invariant multivariable system," *IEEE Trans. Auto. Control*, pp. 25–34, 1976, AC-21.

[8] F. Depping and M. Voits, "Automatic selection of control algorithms for an electrical drive with microcomputer-based speed control," in *Proc. IFAC Conf. on Control in Power Electron. and Electric. Drives*, Lausanne, Switzerland, 1983, pp. 507–514.

[9] U. Waschatz, "Adaptive control of electrical drives employing microprocessors," *ETG-Fachbericht*, vol. 11, pp. 135–140.

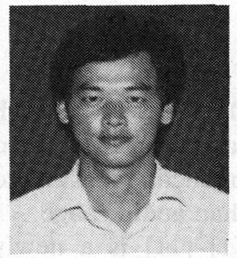
[10] F. Harashima and S. Kondo, "A design method for digital speed control system of motor drives," presented at the Power Electronics Specialists Conf., Boston, June 1982.

[11] K. H. Wong, "A self-tuning controller for multi-input multi-output systems," M.Sc.E. thesis, University of New Brunswick, Fredericton, Canada, 1984.



Ramachandran Balasubramanian was born in Periakulam, India on June 5, 1942 and died on June 23, 1985, a tragic victim of the crash of Air India 182. He received the B.E. degree from the Indian Institute of Technology (IIT), Bangalore, in 1965, the M.E. degree from the University of Madras in 1965, and the Ph.D. from the University of New Brunswick in 1968, all in electrical engineering.

Since 1968 he had been with the University of New Brunswick, where he was a Professor. He was a Visiting Professor at the Shaanxi Mechanical Engineering Institute in China in 1983, at the Indian Institute of Science, Bangalore, 1982–1983, and at ITT, Madras, 1974–1975. At the time of his death he was a very active Researcher in the areas of power systems, control, and signal processing, having published numerous works in a wide variety of conferences and journals.



Kong Hung Wong was born in Sibu, Sarawak State, Malaysia, on February 7, 1958. He received the B.Sc.E. and M.Sc.E. degrees in electrical engineering from the University of New Brunswick, Fredericton, NB, Canada, in 1982 and 1984, respectively.

After graduation he returned to Malaysia and joined the Sarawak Electricity Supply Corporation (SESCo) in 1984, where he is currently employed as an Electrical Engineer in the Projects Section, SESCo Headquarters, Kuching, Malaysia. His main interests are in the areas of adaptive controllers for multi-input/multi-output systems and HVAC-HVDC system control. If circumstances permit, the author plans to return to active research and pursue the Ph.D. degree in these areas by 1988.

Correction to "Failure Analysis of Components Due to 480-V Ground Faults"

In the above paper,[1] on pages 620–621 the vertical axes of Figs. 8–11 should have been labelled: "1", "10", "100", and "1000" current amperes.

Manuscript received October 9, 1986.

[1] D. J. Love and N. Hashemi, *IEEE Trans. Ind. Appl.*, vol. IA-22, no. 4, pp. 617–622, July/Aug. 1986.