This Research Was Sponsored by
THE NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

A Digital Controller Using Multirate Sampling for Gain Control

Report No. EDC 1-64-29

by
Richard W. Van Pelt

Harry W. Mergler
Professor of Engineering
Principal Investigator
Ns G 36-60

Digital
Systems
Laboratory

1965

ABSTRACT

The conventional continuous proportional plus integral control algorithm is converted to its sampled, incremental form, then approximated by a set of three differences which are computed digitally and applied to the process actuator at different times and rates. Computation of the three terms at different rates serves to reduce the amount of equipment needed to provide a wide range of integral gain while retaining the stabilizing effect provided by frequent sampling of the proportional term.

Tests demonstrate that the controller performs well for all types of industrial processes except flow loops with slow actuators. For this exception the controller can easily be modified to provide good response. The controller is designed to control a single loop either independently or in conjunction with a higher level control computer.

AUTHOR

9-10-65

Corrected Table of Contents
NsG-36/36-03-001
Report No. EDC-1-64-29
(Case Institute)

	Page
Abstract	ii
Acknowledgment	iii
List of Figures	vii
List of Tables	ix
List of Symbols	x
Logical Symbols Used	xiv
Chapter I	
Introduction	1
Chapter II	
Nature of the Process Control Problem	12
Process, Transducer and Actuator	12
Analog Controller and Typical Process	14
Chapter III	
Digital Control Algorithm	20
Preliminary System Assumptions	20
Derivation of the Three Mode Incremental Control Algorithm	23
Functions of the Two Mode Control Algorithm	26
Controller Gain Adjustment by Frequency Manipulation	28
Priority Allocation and Computation Frequency of Algorithm Terms	34
Intermediate Storage Requirements	38
Summary	42

Chapter IV

Detailed Design of the Controller 43
Sequence of Controller Operations 43
Conversion of Sequence Rules to Boolean Statements 52
Implementation of Boolean Statements by Logical Elements 60
Contoller Modification for Smoothed Integral Effect 64
System Photographs 67
Summary 67

Chapter V

Effect of Algorithm Approximations Upon Controller Performance 70
First Difference Approximation 71
Separation of Operate and Read Modes 77
Summary 83

Chapter VI

Theoretical Closed Loop Response and Stability 84

Chapter VII

Experimental Results 91
Observed Effects of Algorithm Approximation 91
Observed Effects of Multirate Sampling 96
Summary 103

Chapter VIII

Conclusions and Recommendations for Further Work 104
Conclusions 104
Recommendations for Further Work 105

Appendix A

Controller Tuning by Reaction Rate 109

Appendix B

Effect of Noise on Process Output 112

Appendix C

Use of Controller with a Higher Level Computer 117

List of References 121

LIST OF FIGURES

Figure		Page
1. 1	Multiloop Digital Computer Control System	7
1. 2	Single Loop Digital Controllers	8
2. 1	Elements of a Process Control Loop	13
2. 2	Schematic of Water Level Control System	15
2. 3	Block Diagram of the Continuous Analog..	17
2. 4	Response of the Three Control Modes to a Ramp Error Input	19
3. 1	Output m of Sampled Three Mode Digital Controller in Response to a Step Error Input	25
3. 2	Digital Controller in Sampled Control Loop	30
3. 3	Integrated Output of Multirate Sampled Digital Controller for $K_p = .05$	39
4. 1	Schematic of Controller and Encoder	45
4. 2	Controller Logic Diagram	54
4. 3	Controller Timing Diagram	61
4. 4	Logic Required for Smoothed Integral Effect	66
4. 5	Controller.....	68
4. 6	Controlled System	69

5. 1	Actuator Speed Requirements Established by (5-9)	75
5. 2	Process Reaction Curves for Varying Duration of Actuator Motion τ and Fixed Value of Process Time Constant T	82
6. 1	Sampled Data System with Sampling Interval τ	85
7. 1	Response of System to a Change of Set Point of 100 Quanta	92
7. 2	Response of Single Capacity System to a 100 Quanta Change of Set Point	95
7. 3	Response of Single Capacity System to 50 Quanta Change of Set Point without use of Read Mode	97
7. 4	Response of System to a 100 Quanta Step Change of Set Point	98
7. 5	Response of two Capacity System to 100 Quanta Change of Set Point	101
7. 6	Response of Two Capacity System to a 100 Quanta Change of Set Point	102
A . 1	Open Loop Response of Process to 5. 3 per- cent Step Change in Input	110
B. 1	Effect of Noise on Manipulated Variable ..	114

LIST OF TABLES

Table		Page
3 - 1	Priority of Computation	37
4 - 1	Operations Performed by Controller	47
4 - 2	Operations Performed by Controller	49
4 - 3	Operations Performed by Controller	51
6 - 1	Conditions for Marginal Stability	89

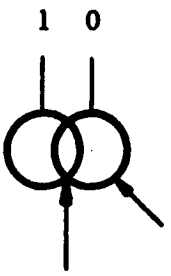
LIST OF SYMBOLS

a_i	Coefficient of characteristic equation
A	Binary number
A_i	The i^{th} bit of A
A_{kp}	Defined by equation (3-18)
A_{kr}	Defined by equation (3-19)
A_{ki}	Defined by equation (3-20)
b	A feedback variable
b (s)	Laplace transform of b
b_i	The i^{th} bit of b
b_k	The value of b at the k^{th} sampling instant
$B^{\mu n}$	An event B delayed in time by n microseconds
c	Controlled variable
d	Measure of noise amplitude
e	Natural logarithm base
e	Error signal
e_k	Value of e at the k^{th} sampling instant
f_a	Countdown frequency applied to register A
f_i	Frequency of integral mode sample order

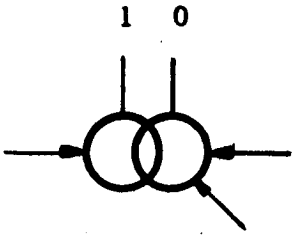
f_p	Frequency of proportional mode sample order
f_s	Frequency at which stepping motor is driven
FF	Flip flop
G_0	Open loop gain
h_2	Head of water in tank 2
$(J)_{\Delta r}$	Operation (J) to be performed when a change occurs in r
$(J)_{f_i}$	Operation (J) to be performed at a rate of f_i
$(J)_{f_p}$	Operation (J) to be performed at a rate of f_p
K	A Ziegler-Nichols reaction rate parameter
K_1	Open loop gain excluding controller
K_f	Gain of feedback transducer
K_p	Proportional gain of controller
K_v	Gain of valve
$k + \delta +$	An instant of time between $k + \delta$ and $k + 1$
L	A Ziegler-Nichols reaction rate parameter
m	Valve position
Δm_k	An increment of valve position computed immediately after taking the k^{th} sample
m_k	Valve position after application of increment Δm_k

M_i	The i^{th} control flip flop
N	An integer
N	A Ziegler-Nichols reaction rate parameter
r	Reference variable
r_i	The i^{th} bit of r
r_k	The value of r at the k^{th} sampling instant
R	Resistance of valve
R	A Ziegler-Nichols reaction rate parameter
s	Laplace operator
S	Sign of A
t	Time
Δt	Sampling interval
T	Time constant
T	A Ziegler-Nichols reaction rate parameter
T_d	Derivative time of controller
T_i	Integral time of controller
T_n	Period of oscillation of closed loop
T_{fs}	Time for full stroke of valve
u	Load variable

V	Amount of valve travel during a fixed period
w	A flow rate
z	z-transform operator
α_i	0 to 1 transition of binary variable "i"
β_i	1 to 0 transition of binary variable "i"
δ	A fraction of the sampling interval
τ	A delay
τ_k	A time interval computed at sample instant k

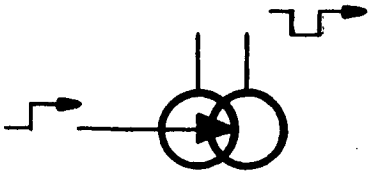


Flip Flop with
Trigger Input

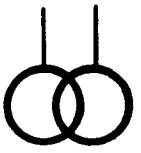


Flip Flop with
Set and Reset Inputs

Diagonal Arrows are Manual
Reset

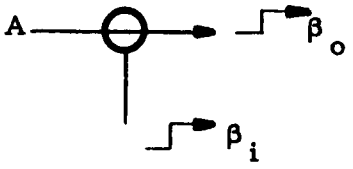


One Shot Multivibrator



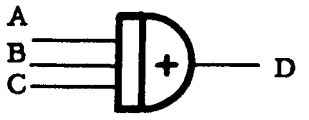
Free Running Multivibrator

LOGICAL SYMBOLS USED



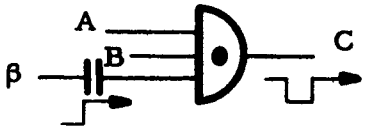
Steering Gate

$\beta_o = \overline{A}\,\beta_i$



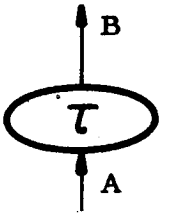
NOR Gate

$D = \overline{A + B + C}$



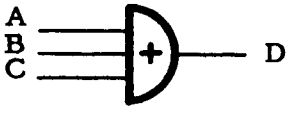
Gated Pulse Generator

$C = \overline{A}\,\overline{B}\,\beta$



Delay

B = A delayed by τ



OR Gate

D = A + B + C

LOGICAL SYMBOLS USED

CHAPTER I

INTRODUCTION

Continuous analog controllers have come into widespread use for the automatic regulation of industrial processes. Their function is to compare the desired value of a process output with the measured output and to adjust the process input in accordance with the error. In a three mode controller the process input calculated by the controller is a combination of the present error, its rate of change and its time integral. Such a controller, containing proportional, integral and derivative modes, is commonly called a PID controller. All such controllers have, until recently, been analog devices.

Since World War II, the advances made in digital computers have steadily broadened the areas in which digital techniques can be profitably employed. The insensitivity of digital data to noise, both in transmission and manipulation, resulted in the use of digital control systems for aircraft and missiles. Machine tools were subjected to digital control in order to produce complicated parts economically and accurately.[1] In process control, starting in the 1950's, digital computers were used for data collection and reduction at large chemical and petroleum installations

in an effort to improve plant identification and operation.[2] In some cases the computer ordered new controller set points, but the direct controller function of valve control was not taken over by the digital computer. The public utilities also started using digital computers, but at first only to compute set points or load assignments.[3]

A number of people began about 1958 to investigate the possibility of direct digital control of many individual loops using a shared digital computer.[2] There were two major objectives. The first objective was, for the same quality of control, to reduce capital investment below that required for the analog controllers that would be displaced. The second objective was to improve control and to provide basic equipment compatible with higher level computer control functions such as interaction compensation, ratio, cascade, and optimization. The author took part in a feasibility study of a "first level" direct digital control computer at the Systems Research Center of Case under the direction of the late Professor D. P. Eckman in early 1962.[4] The study, sponsored by the Corning Glass Works and the Conoflow Corporation, included simulation of various algorithms and quantization techniques and included system planning for a computer capable of PI control of 96 loops, plus some additional

features. The study failed to show that the computer would be less expensive than 96 analog controllers, although the additional capabilities of the computer tended to justify any difference. No attempt was made to estimate reliability. In June 1962 Williams of Monsanto reported on the successful use of an RW 300 computer for direct digital control of 10 control loops, using a PI algorithm.[5] In November 1962 Yetter and Sanders of duPont reported on their studies of direct digital control.[6] They concluded that the digital time-shared system would show capital savings over electronic analog controllers for installations with at least 75-100 loops. They devoted considerable attention to the difficult problem of coupling the central high speed computing facility to the individual slow speed valve actuators. At the same time a report was published that Imperial Chemical Industries, Ltd. in England had been operating an ammonia plant for over a year using a Ferranti digital computer providing PID control.[7] In this case pneumatic actuators were used, requiring an auxiliary control loop for each manipulated variable just to maintain valve position.

Also about 1962 some single loop digital controllers appeared on the scene. Minnesota Mining and Mfg. announced a digital controller with only the I mode, and the French Lignes Télégraphiques et Téléphoniques

announced a PID controller that was part digital and part analog, designed to drive a truly digital actuator, a valve containing seven flow-weighted on-off valves. Development of digital process controllers in Russia resulted in several reports in 1962.[8, 9, 10, 11] They included consideration of single loop and multiloop controllers, both operating in the PI mode, and included also discussion of the extent to which sampling improves or degrades performance for the various controller structures proposed. Analog to digital conversion at the input of these controllers was apparently done after an analog error had been generated, thus avoiding one digital subtraction but at the same time losing the opportunity to use digital set point data. The point was made, however, that in the absence of digital sensors this procedure causes no loss of accuracy. Mergler, Peatman and Walker of the Case Digital Systems Engineering Group have developed single loop digital controllers, both PI and PID, in which a digital set point is used.[12, 13] The controller designed by the author makes use of the accumulator flip flops designed for Mergler's controller, but is distinguished from all of them by the means of adjustment of integral effect and by the simplifications made in the control algorithm.

The present trend, at least outside of the Soviet Union, is toward the use of a single time-shared computing unit for the control of many loops. Several such computers have been sold in the last two years. The main reason usually given for following the shared computer approach is that the only real justification for shifting to digital control is to get an improvement in control over analog controllers. The feeling seems to be that if any savings in equipment results from the switch it will be very small, so we must look for profit from higher level computer control functions. Since such functions can only be performed on a central computer with access to all loops, why not use that same computer for basic "first level" direct digital control? This is a compelling question, and one that no proponent of single loop controllers has tried to answer. My answer, however, is that there are two good reasons why first level control should not be performed by a shared computer.

The first reason for using single loop controllers instead of a shared computer is to obtain better reliability. Loss of control of a single loop can generally be corrected soon enough to avoid plant upset, especially since manual control can be used in the interim. The spectre that haunts potential digital control buyers is the prospect of simultaneous failure of all loops. Using the shared computer

of Figure 1.1, this could occur if either the A/D converter or the computer failed. When using separate controllers, as in Figure 1.2, only the failure of the A/D converter or of the power supply could cause catastrophic failure of all loops. Loops that had their own digital encoders on their transducers would have no common failure mode at all, if a stand-by power supply were provided. Preventive maintenance of individual controllers could be performed by replacing a controller with a spare and submitting it to whatever off-line checks might be useful. With a shared computer this would be possible only by having two entire duplicate computers. This has, in fact, been proposed by some to ensure reliability. Other measures for increasing computer reliability can be taken, but each costs money. It would seem less expensive to use the single loop controllers, provide a few spares, and ensure reliability at the critical shared part of the system by providing a duplicate input multiplexer and A/D converter. Whether such a procedure would in fact be less expensive depends on the cost of the single loop controller. The design used in this thesis uses 100 transistors for the PI modes, and might require 60 more for extension to PID control.

The second reason for using single loop controllers is to simplify the job of valve actuation. Referring to Figure 1.1

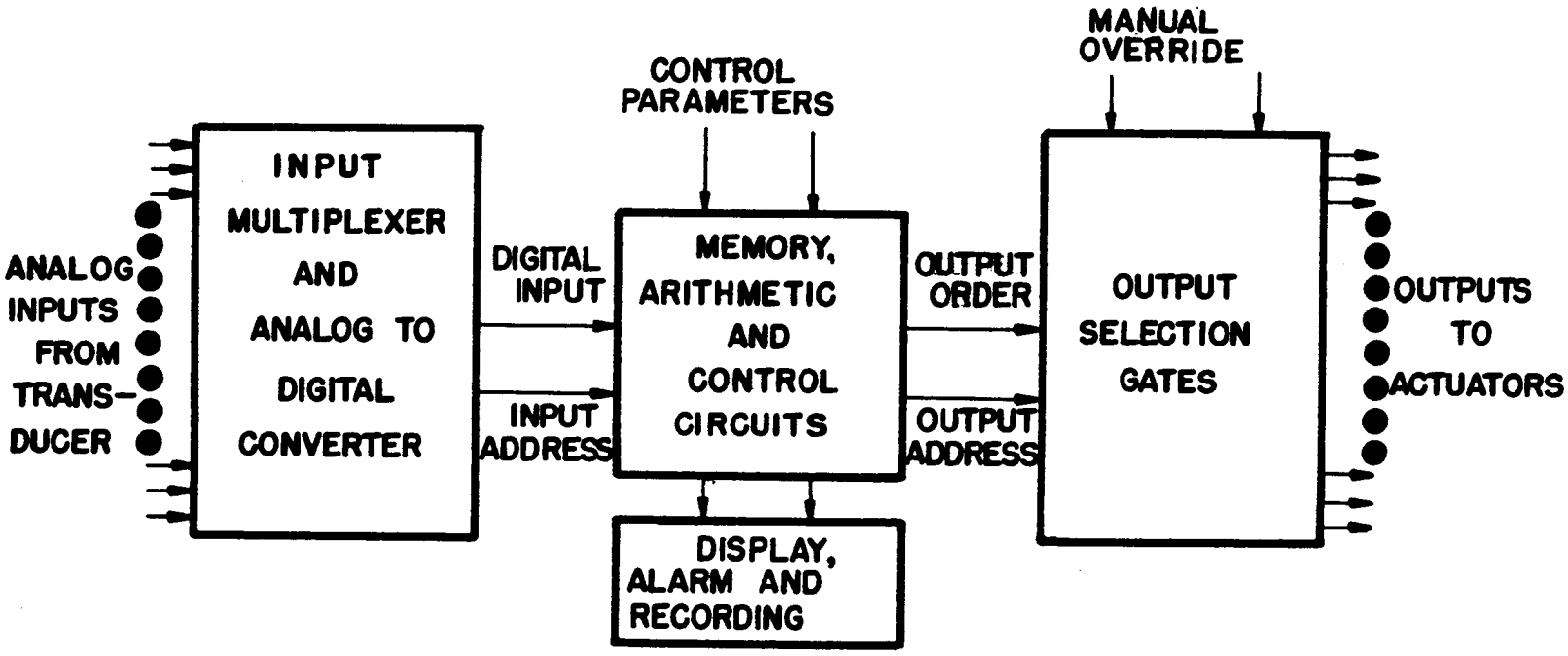


FIGURE 1-1 MULTILOOP DIGITAL COMPUTER CONTROL SYSTEM

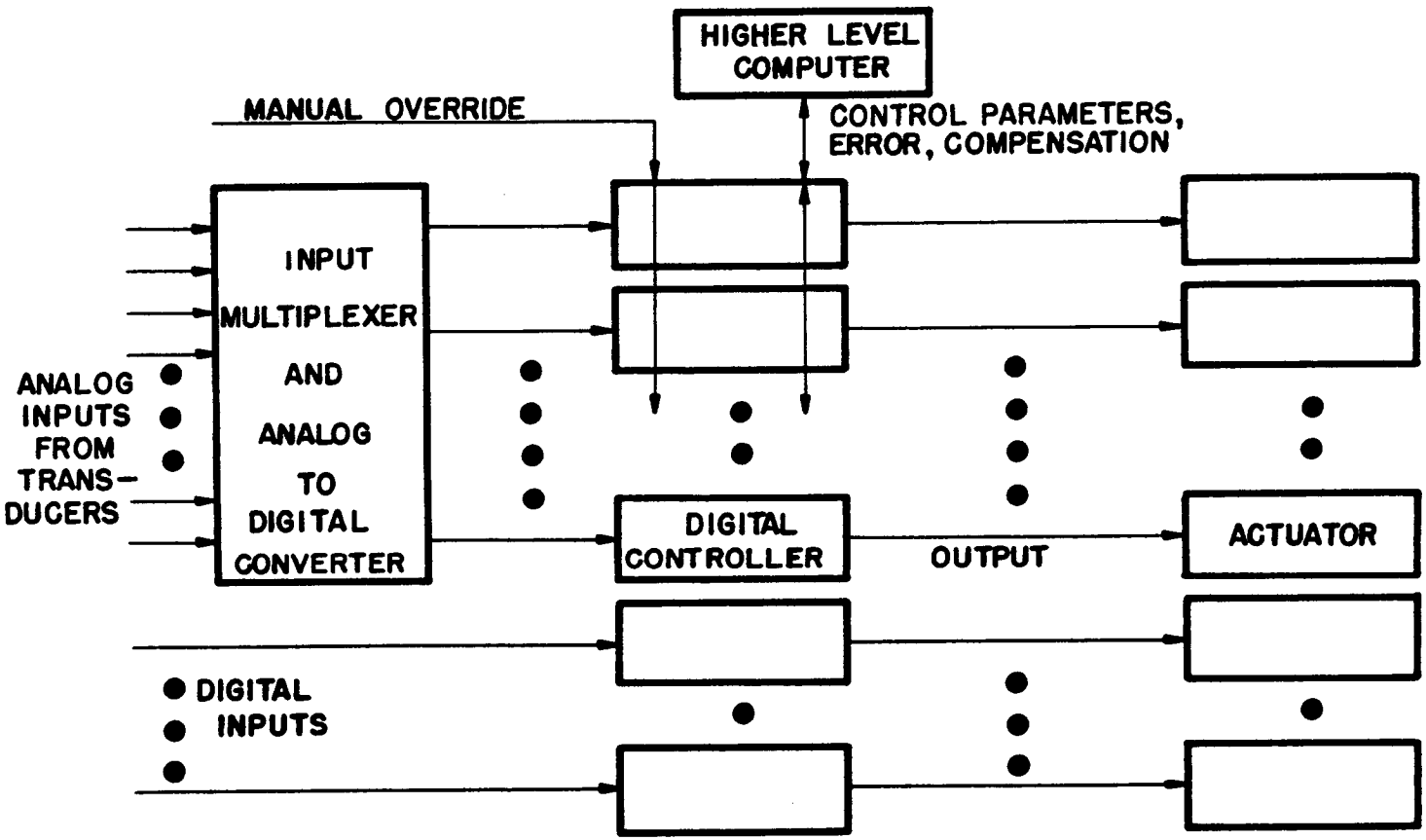


FIGURE 1-2 SINGLE LOOP DIGITAL CONTROLLERS

it is evident that multiplexing is necessary for both the input and output of the shared computer. The input transfer is relatively simple, since the digital feedback variable for each loop can be transferred in parallel in a matter of a few microseconds, and this needs to be done not more often than 10 times per second per loop. Transfer of the output orders to the actuators is a different sort of problem. If the output order is given only once, when the order is computed, then the actuator that receives the order must contain some sort of memory. The reason is that the actuators must move some specified distance in response to the order, and the time required for that motion is in many cases much longer than the time that the output multiplexer can afford to dwell on any one particular loop. The solutions to this problem can be broken into two classes; those in which the computer gives absolute actuator position orders and those in which it gives incremental actuator position orders.[6] One of the solutions proposed in the incremental category is to maintain one countdown register in the computer for each actuator. The position increment computed each sampling instant is put in this register and counted down to zero. The actuator is moved at a fixed rate in one direction or the other as long as the countdown is going on. This technique, explained more fully in Chapter III, can also

be used with single loop controllers. The advantage in doing so is that no output multiplexing is required, rather a given countdown register is simply wired to a given actuator. This is the technique chosen for the design proposed here, and is the one shown in Figure 1.2.

If the single loop controller approach does relieve the problems of reliability and actuator control, we still must show compatibility with higher level control computers. The plan is to require that a separate, shared computer perform all functions except the basic PI or PID control algorithm. It would have access to the shared A/D converter for its inputs, and would send its incremental output orders directly to the various single loop controllers, as shown in Figure 1.2. The controllers would apply these increments to their actuators, then return to their normal control duties until the next higher level control order comes along. Thus the higher level control computer is simplified by the fact that its output orders can be administered by the single loop controllers. Furthermore, its reliability requirements could be relaxed, since its failure would not effect first level control. If higher level control is not needed, the computer can be omitted and the user need pay only for what he uses. The detailed means of communications between the first level con-

trollers and the higher level computer will be covered in later chapters and in Appendix C. Chapters I and II are intended to provide background and motivation for the sections that follow on the actual design of the single loop digital controller.

CHAPTER II

NATURE OF THE PROCESS CONTROL PROBLEM

Process, Transducer and Actuator

The block diagram of Figure 2.1 identifies the elements of an industrial process control loop. The controller is provided in order to maintain the feedback variable b, an indication of the controlled variable c, equal to the reference variable or set point, r. There are two characteristics that typify industrial process control loops and distinguish them from servomechanism control loops. The first is that for industrial processes, especially continuous ones, the reference variable r is maintained at a constant value for long periods of time, so that the controller's principal job is to maintain the controlled variable at a constant value in the face of load changes or disturbances, u. This is in strong contrast to the duty cycle of a servomechanism, which is asked to produce a motion corresponding to an often varying reference signal r. The second characteristic of industrial processes is that the dynamic lag of the actuator is usually negligible in comparison to the lag inherent in the transducer or in the process being controlled. In flow loops, which may account for 40 percent of process loops,[6] the process time constant is usually negligible compared to that of the transducer, but where it is important one can generally select a valve actuator with quick

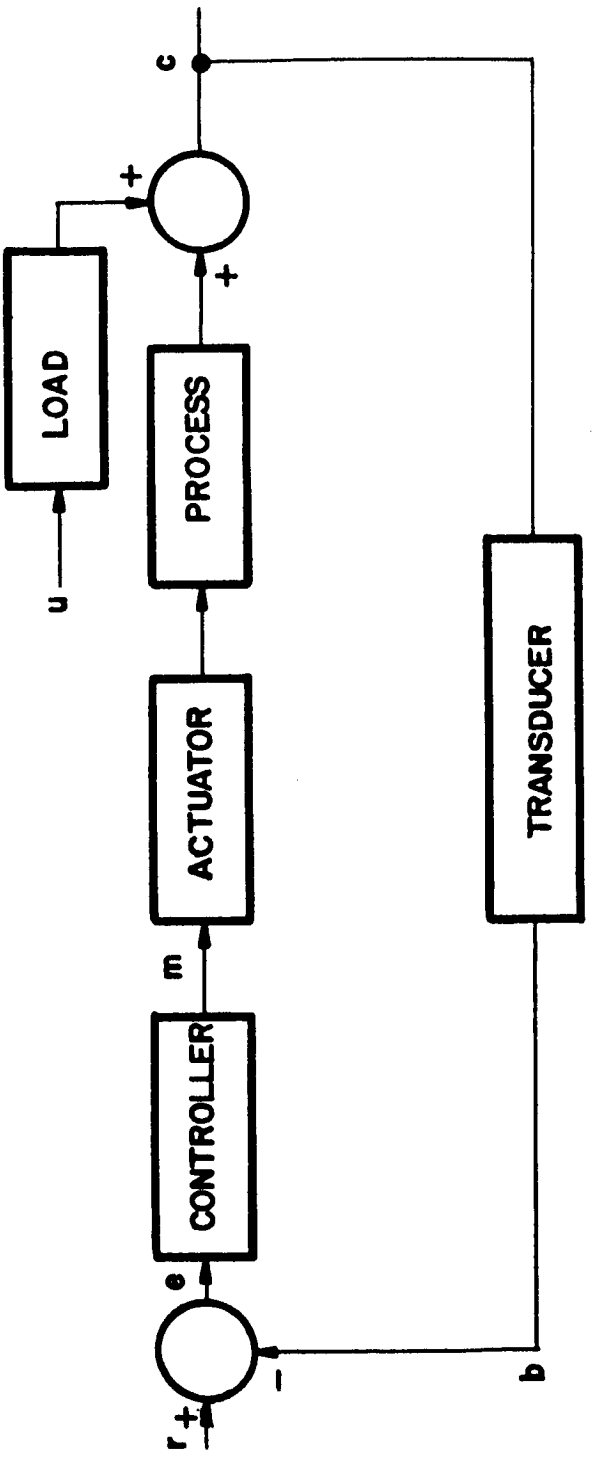


FIGURE 2-1 ELEMENTS OF A PROCESS CONTROL LOOP

enough response to justify neglecting its dynamics in the control loop. Chapter III demonstrates how these two traits can permit a considerable simplification to be realized in the design of a digital process controller and later chapters establish guidelines showing the range of control loop characteristics for which this simplified control algorithm is effective.

Analog Controller and Typical Process

Figure 2.2 shows a process consisting of two tanks in cascade. The level of the lower tank is to be controlled by manipulation of input flow to the upper tank in the face of any arbitrary disturbance. Although such a system is not a common industrial process, its dynamic response is similar to that of thermal processes.[14] This liquid level process is the one actually used in this research. It permitted great flexibility in testing, for two reasons. First, it was possible, when desired, to bypass tank 1, directing flow w_1 directly into tank 2, leaving only a single capacity system, with an attendant increase in response speed. Second, the valve actuator used was an electric stepping motor, which could be run at any speed up to 145 pulses per second, giving a time for full stroke of the valve of any value down to 32 seconds. It is shown in Chapter V that the ratio of time for full valve stroke to the dominant time constant of the system is an important parameter in

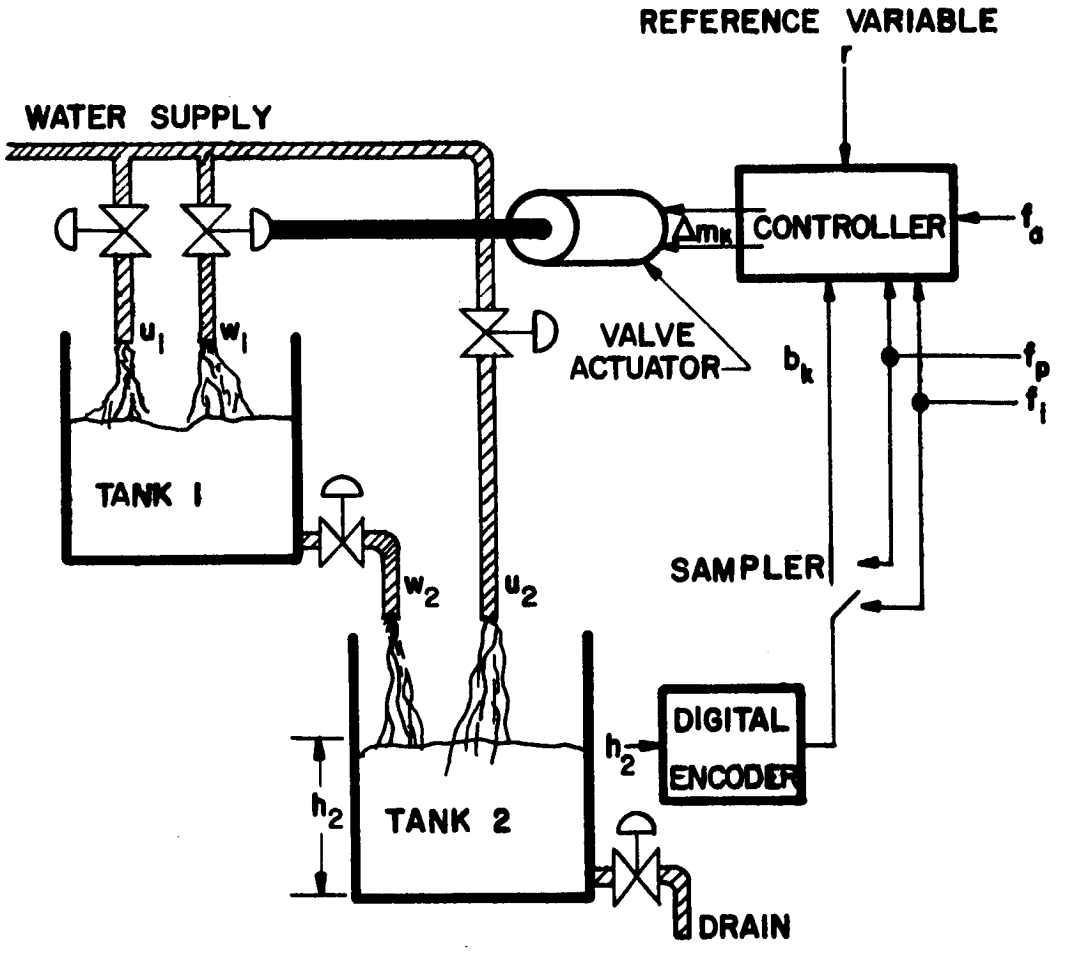


FIGURE 2-2 SCHEMATIC OF WATER LEVEL CONTROL SYSTEM

predicting the effectiveness of the control technique described in this paper.

When a process like that shown in Figure 2.2 is controlled by an analog controller, the loop can be described in block diagram form by Figure 2.3 , which gives the form of the transfer functions of the various elements of this analog control loop.

Analog controllers are most commonly available with either two or three "modes" of control. The LaPlace transformed output of a three mode analog controller is given by the equation

$$m(s) = K_p\left[\frac{1}{T_i s} + 1 + T_d s\right] e(s) \qquad (2\text{-}1)$$

The three modes are called respectively integral, proportional and derivative. Integral action provides a corrective ramp output from the controller for a steady error input and is included to reduce steady state error to zero. Proportional action provides a ramp output from the controller in a direction to stop the motion of an input error ramp. Derivative action provides an impulse output signal for a step input error signal, and, because of its stabilizing effect, is useful for permitting higher proportional and integral gains to be used.

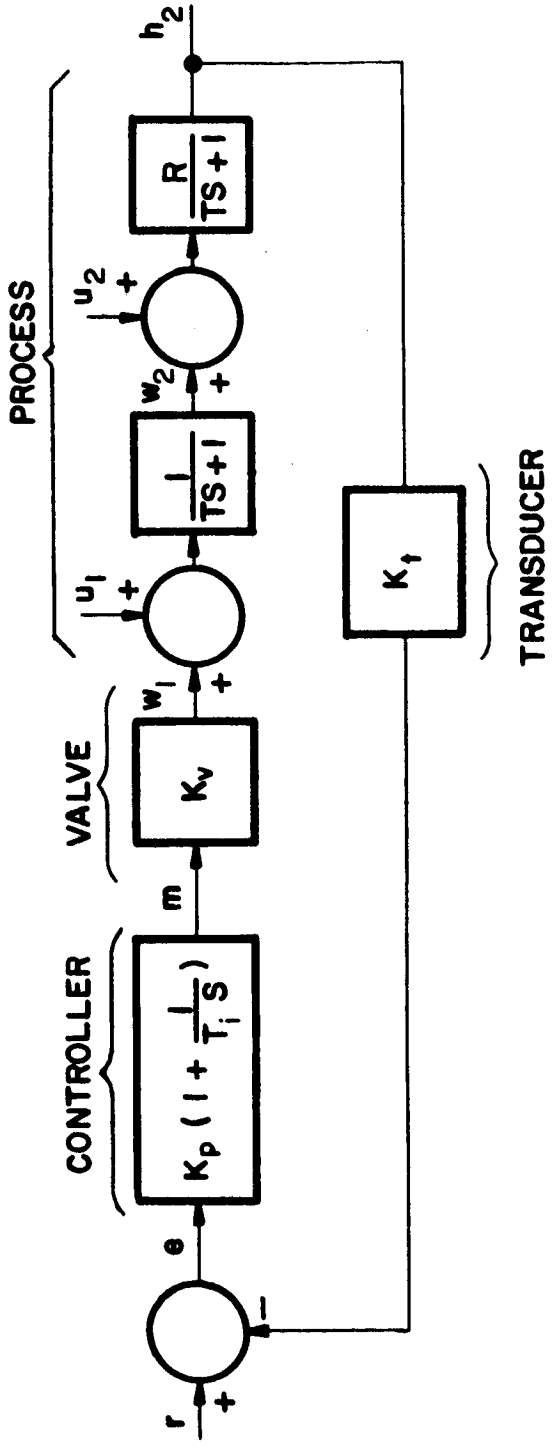


FIGURE 2-3 BLOCK DIAGRAM OF THE CONTINUOUS ANALOG

Figure 2.4 illustrates how each of the three control modes contributes to produce controller output m.[14] The upper plot shows an arbitrary ramp error signal starting from zero at time 1. At time zero m has an arbitrary initial value of $M_o = 1$. For such a test the controller is understood to be running open loop, that is without the feedback connection shown completed in Figure 2.1. The proportional action, the lower dotted line, is simply $K_p e$ added to M_o. In this case K_p is one. Therefore, the proportional part of m follows e exactly, in a direction that would tend to reduce error in a closed loop. Derivative action, shown in the figure with a value of 2, adds an increment of m equal to T_d multiplied by the slope of the error curve. This provides a lead action, yielding a given value of m a time T_d earlier than proportional action alone would have produced it. The integral mode adds a further increment of m. In the Figure $T_i = 10$, so that at any time the integral effect is equal to one tenth of the area under the error curve preceeding that time. The inverse of T_i is called "reset rate" and is defined as the number of times per minute that the proportional action is duplicated by the integral action.

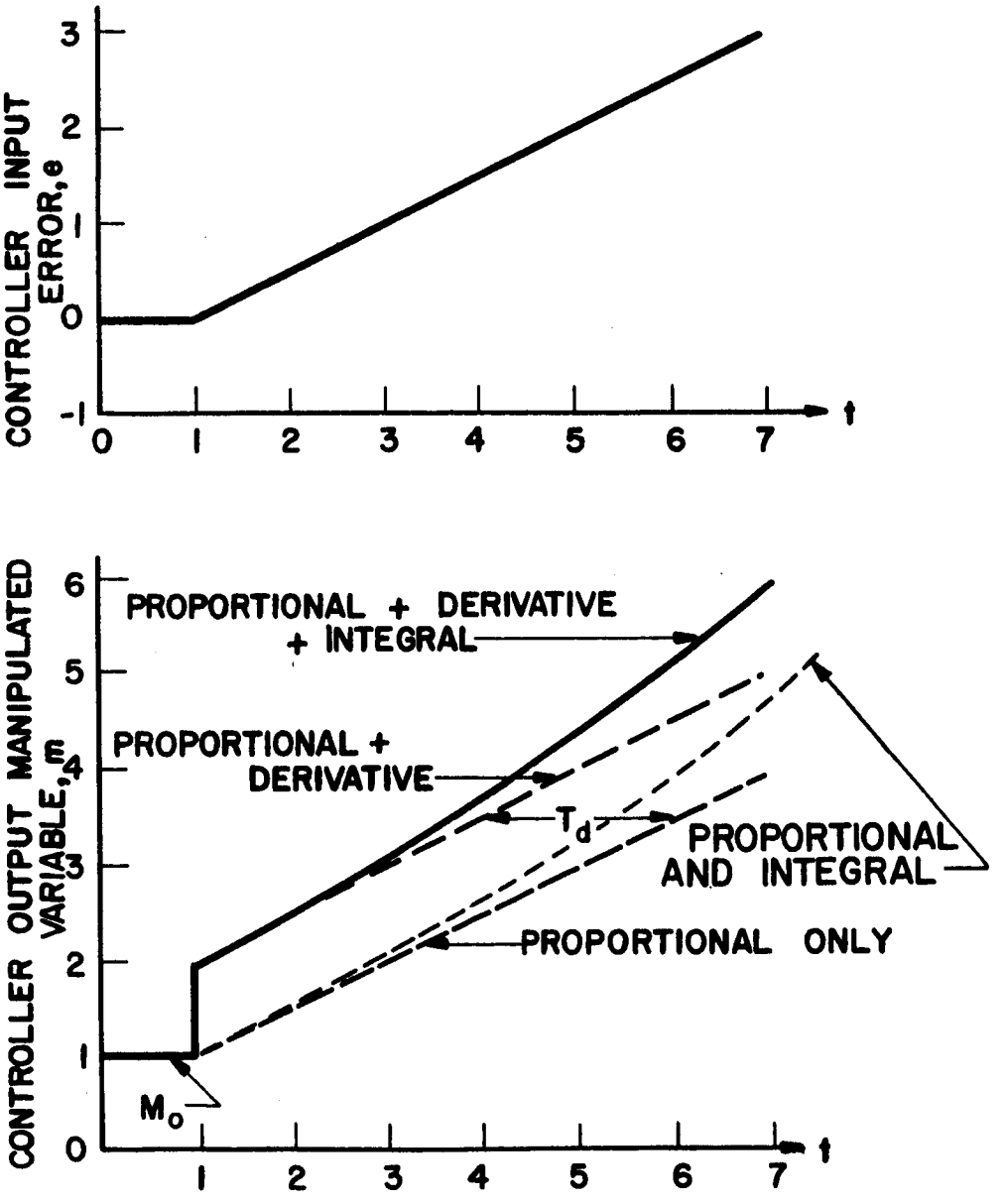


FIGURE 2-4 RESPONSE OF THE THREE CONTROL MODES TO A RAMP ERROR INPUT

CHAPTER III

DIGITAL CONTROL ALGORITHM

The aim of this research was to design a simple digital controller that would have roughly the same proportional and integral actions as the usual two mode, proportional plus integral analog controller, yet retain the inherent accuracy, resolution and flexibility of digital computation. In order to be usable on a wide variety of processes it was necessary to furnish the same wide range of gain and integral time that is provided on analog controllers. These requirements could not be considered, however, until it was decided what sort of digital input data the controller would have to work with, what sort of valve actuator the controller would drive, and how the controller's output would control the actuator. Once these decisions are made it is possible to develop a control algorithm for a digital controller using sampled data.

Preliminary System Assumptions

Controller Input Data:

The reference variable or set point, r, is provided in parallel binary form from a set of 10 on-off switches. The feedback variable, b, is available in 10 bit parallel form, either continuously from a digital encoder or upon command with negligible delay from an analog-to-digital converter.

Control parameters K_p and T_i are discussed later in this chapter.

Valve Actuator:

Analog controllers usually produce an output, m, which is a valve position order. The valve actuator is designed to open to a position proportional to m. Since the analog controller works on a continuous basis it is a simple matter for it to provide the continuous signal m for valve positioning. The output of a digital controller, however, must be converted to analog form before it can be used for actuator control. If the controller output is m, there are three choices open for actuator control:

1. Between controller sampling times the digital value m could be held in a memory register which would continuously be the input for a digital-to-analog converter whose output would then drive an analog actuator. This scheme is uneconomical because of the large expense for a DA converter for each manipulated variable.

2. After a new sample is taken and a new value of m is calculated, m could be converted to an analog signal which would then be stored by an analog memory device, a zero-order hold, until a new value of m is calculated. The stored analog value of m would drive the analog actuator. This technique has the advantage of requiring only a single DA converter for a number of control loops, but would still be complicated by the necessity for multiplexing the input and output of the converter between the various controller outputs and analog memory inputs. Furthermore, inexpensive analog memory units are subject to drift.

3. A combination of the first two methods could be used by converting m to an analog signal by a multiplexed DA converter and holding the converter on this value of m until the actuator had reached its new ordered position. Then the actuator could be mechanically or electrically locked in this position until the next sample time and the DA converter would be free for use on another controller output and actuator. If one chose an actuator designed so that the load could never drive the actuator, then no locking of the actuator would be necessary. This technique has the disadvantage of still requiring a DA converter, and it would not be possible to use it for many loops, because the converter must dwell on a given loop as long as the actuator of that loop is moving.

None of the above three schemes is well suited for actuator control, especially if separate digital controllers are used for each loop. It is considerably simpler, both from the point of view of actuator control and simplification of controller logic, to use an incremental scheme. In all three methods above the end effect of the DA conversion, actuation and holding was to move the actuator some increment Δm_k where

$$\Delta m_k = m_k - m_{k-1} \qquad (3\text{-}1)$$

and where m_k is defined as the value of m resulting from the computation following the k^{th} sample. If we use an actuator capable of maintaining its last position, as discussed in method three above, then it is not necessary to compute m_k after each sample, but only Δm_k. This

technique is what is meant by using an incremental scheme. The controller is designed to compute an increment of valve motion, Δm_k, and to apply this correction to a valve capable of maintaining its last position when the actuation signal is removed. Such an actuator can be called an incremental actuator, whether it be a stepping motor or perhaps a synchronous motor capable of moving a distance proportional to the time it is turned on, even if for only a small fraction of a second. The final preliminary design assumption is then, that the digital controller is to operate an incremental valve actuator.

Derivation of the Three Mode Incremental Control Algorithm

Let Δt = time interval between samples

For Δt small and quantum size small:

$$\frac{\Delta m_k}{\Delta t} \cong \frac{dm}{dt} \qquad (3\text{-}2)$$

From equation (2-1):

$$\frac{dm}{dt} = K_p \left[\frac{1}{T_i} e + \frac{de}{dt} + T_d \frac{d^2 e}{dt^2} \right] \qquad (3\text{-}3)$$

Converting the error terms of (3-3) to their sampled equivalents, and combining (3-2) and (3-3):

$$\frac{\Delta m_k}{\Delta t} = K_p \left[\frac{1}{T_i} e_k + \frac{e_k - e_{k-1}}{\Delta t} + T_d \frac{\dfrac{e_k - e_{k-1}}{\Delta t} - \dfrac{e_{k-1} - e_{k-2}}{\Delta t}}{\Delta t} \right] \qquad (3\text{-}4)$$

where the substitution of e_k for e comes from using the rectangular approximation for e.

Solving (3-4) for Δm_k:

$$\Delta m_k = K_p\left[\frac{\Delta t}{T_i} e_k + (e_k - e_{k-1}) + \frac{T_d}{\Delta t}\,(e_k - 2e_{k-1} + e_{k-2})\right] \qquad (3\text{-}5)$$

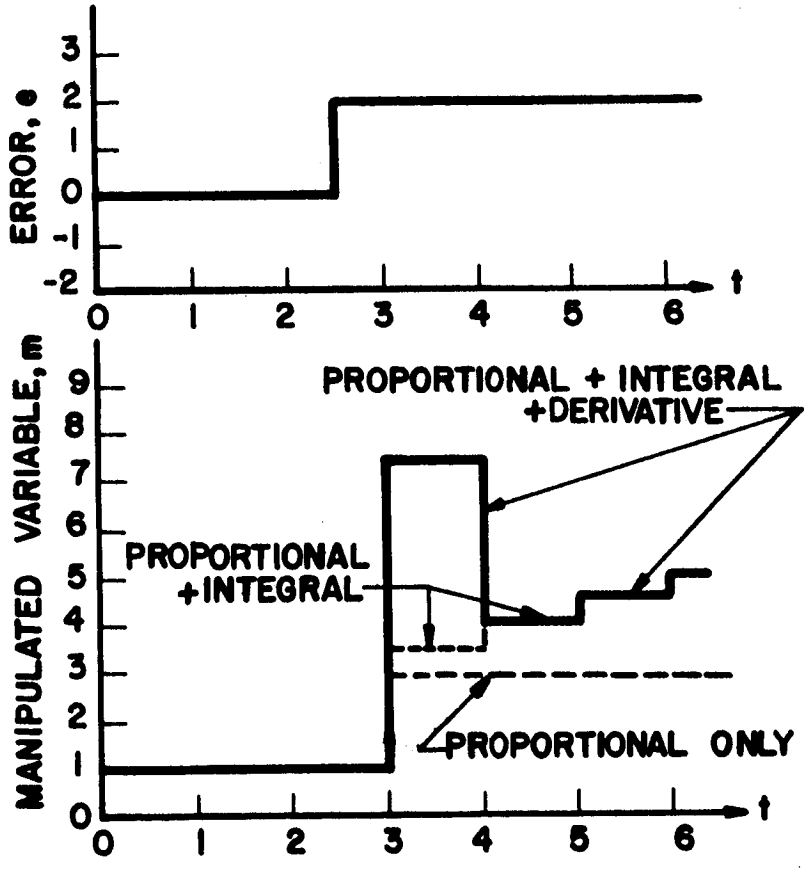
Figure 3.1 illustrates how this approximation to the continuous algorithm looks for finite Δt. At the first three sample instants the error e is zero and manipulated variable m remains at a constant value, arbitrarily chosen as one. A step change in e occurs between samples 2 and 3, but there can be no response at the output of the controller until sample 3 occurs. The controller output shown in the figure will be exactly the same whether the step change in e occurs at 2.01 time units or at 2.99. This is one reason for making the sampling interval as small as possible, to avoid adding unnecessary lag to the control loop. The response shown was computed using the following parameters:

$$\Delta t = 1 \qquad K_p = 1 \qquad T_i = 4 \qquad T_d = 2 \qquad (3\text{-}6)$$

The proportional effect is virtually the same as what would be produced by an analog controller, differing only by the sampling delay. The integral effect is a staircase instead of a ramp, and the derivative action is an impulse of the

$$\Delta m_k = K_p\left[\frac{\Delta t}{T_I}(e_k)+(e_k-e_{k-1})+\frac{T_d}{\Delta t}(e_k-2e_{k-1}+e_{k-2})\right]$$

$\Delta t = 1 \qquad K_p = 1 \qquad T_I = 4 \qquad T_d = 2$



SAMPLING INSTANTS AT INTEGRAL UNITS OF TIME

FIGURE 3-1 OUTPUT m OF SAMPLED THREE MODE DIGITAL CONTROLLER IN RESPONSE TO A STEP ERROR INPUT

proper area that has been broadened in duration and shortened in amplitude. This is not a great disadvantage, because it reduces the amount of non-linearity in response resulting from valve saturation, if Δt is properly chosen.

Functions of the Two Mode Control Algorithm

Because two mode, proportional plus integral, analog controllers are so widely used and so many users and vendors question the value of the derivative mode it was decided to limit this project to the design of a two mode controller. Therefore this paper will henceforth be concerned only with the two modes of control, allowing T_d of preceeding sections to go to zero. The only exception is in Chapter VIII, in which it is suggested that the technique developed later in this chapter could easily be extended to provide the third mode.

The two mode digital control algorithm is taken directly from (3-5), setting $T_d = 0$:

$$\Delta m_k = K_p\left[\frac{\Delta t}{T_i} e_k + (e_k - e_{k-1})\right] \qquad (3\text{-}7)$$

By our nomenclature from Figure 2.3 ,

$$e = r - b \qquad (3\text{-}8)$$

Hence, in sampled form:

$$e_k = r_k - b_k \qquad (3\text{-}9)$$

Substituting (3-9) into (3-7) :

$$\Delta m_k = K_p \left[\frac{\Delta t}{T_i} (r_k - b_k) + (r_k - b_k) - (r_{k-1} - b_{k-1}) \right] \qquad (3\text{-}10)$$

Rearranging:

$$\Delta m_k = K_p \left[(b_{k-1} - b_k) - (r_{k-1} - r_k) - \frac{\Delta t}{T_i} (b_k - r_k) \right] \qquad (3\text{-}11)$$

The first two terms in (3-11) make up the proportional part, and the third represents the integral. The second term, $(r_{k-1} - r_k)$ will be non-zero only when there is a change in reference variable, r. There is no need for computing that term at any other time. We can rewrite (3-11) with a subscript Δr on the second term to indicate that that computation should only be performed when there is a change in r.

$$\Delta m_k = K_p \left[(b_{k-1} - b_k) - (r_{k-1} - r_k)_{\Delta r} - \frac{\Delta t}{T_i} (b_k - r_k) \right] \qquad (3\text{-}12)$$

The term $(r_{k-1} - r_k)_{\Delta r}$ is now understood to mean the difference between the new value of r subtracted from the old value of r.

The first term of (3-12) represents change in the controlled variable between sampling instants k-1 and k. As

long as the reference variable r remains constant this first term is all that is required to give the incremental proportional effect.

To review the meaning of (3-12) recall that (3-7) was derived from the analog control equation (2-1). Based on the rectangular approximation for e and with very small Δt, (3-7) is the exact incremental equivalent of (2-1), minus the omitted derivative mode. Further, in going from (3-7) no approximations were made, since only differences equal to zero were left out. A digital controller that implements (3-12) will then provide precise proportional plus integral control, so called PI control, if sampling interval Δt is very small relative to the dominant time constant of the loop, and if quanta size is small enough to approximate analog resolution. The first condition, high sampling rate, can be fulfilled if desired, but it should be kept as low as possible while still retaining good control in order to reduce demands on the analog to digital converter to a minimum. The second condition, small quanta size, must be met in order to match the resolution capabilities of the analog transducer chosen.

Controller Gain Adjustment by Frequency Manipulation

An important part of the controller design problem is finding a good way of providing a wide choice of values for

the gain parameters K_p and $\Delta t/T_i$ in (3-12). In order for the controller to be applicable to a variety of processes these parameters should be variable over a range of at least 100 to 1. Analog controllers typically provide K_p from .5 to 50 and T_i from .1 minutes to 50 minutes, with other ranges available by field change.

Proportional Gain K_p

K_p, although called only proportional gain, also acts as a gain coefficient on the integral term. It is therefore possible in implementing the controller to perform the multiplication just as indicated in (3-12), that is to wait until the sum inside the brackets is formed and then apply K_p to that sum. The easiest way to accomplish this multiplication is at the interface between controller and actuator. The technique chosen for digital to analog conversion is to convert Δm_k to a period of time τ_k during which the actuator moves at full speed in one direction or the other depending upon the sign S of Δm_k. In the experimental loop used, shown in Figures 2.2 and 3.2, the actuator is a stepping motor, driven at a speed of f_s steps per second. At this speed the actuator requires T_{fs} seconds to move full stroke. The number actually calculated in the subtracter is $\Delta m_k/K_p$. This number, called A_k in Figure 3.2, is then converted to a time period by counting it down to zero at a frequency of f_a pulses per second. If full stroke of the valve is covered by m ranging from zero to one, then

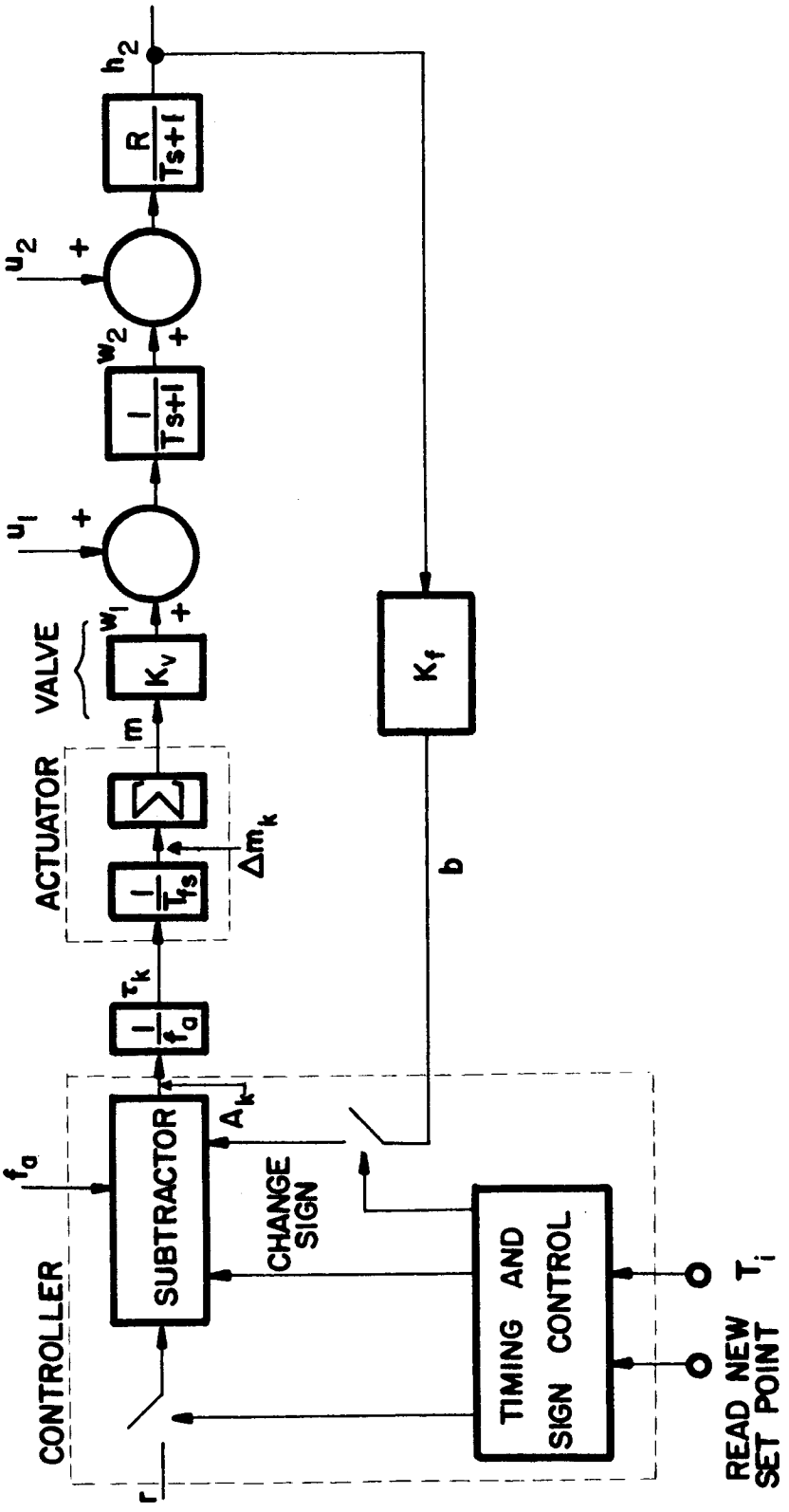


FIGURE 3-2 DIGITAL CONTROLLER IN SAMPLED CONTROL LOOP

Δm_k is finally represented in terms of loop parameters by

$$\Delta m_k = \frac{A_k}{f_a T_{fs}} = \frac{\Delta m_k}{K_p f_a T_{fs}} \qquad (3\text{-}13)$$

Then for this method of DA conversion:

$$K_p = \frac{1}{f_a T_{fs}} \qquad (3\text{-}14)$$

and there are now two ways to vary K_p for a given valve, process and encoder. Varying T_{fs} is the first way, but it would never be used because the tendency is to use the fastest valve actuator that is economically available. All that is required by the other method is to provide a variable frequency pulse generator for f_a. In the controller tested the pulse generator used is a free-running multivibrator with a frequency range of 5 to 100 cps. Higher frequencies and hence lower values of gain K_p are available simply by changing a pair of capacitors.

Integral Time T_i

Implementation of equation (3-12) requires forming the difference between the binary numbers b_k and r_k and multiplying that difference by a number equal to $\Delta t/T_i$, the sampling interval divided by a suitable value of the so-called integral time that was used on the analog controller. There are numerous ways to perform multiplication in a

digital device, but all of them require a considerable amount of hardware. Because the other two terms in (3-12) require no multiplication and because the multiplication by K_p of the expression in brackets can be performed in the DA conversion, a method was sought to avoid the necessity for multiplying the integral term by the coefficient $\Delta t/T_i$. The simplest way to do this is to let that coefficient be always equal to one. Thus far the only restriction on Δt has been that it should be small relative to the dominant system time constant. Letting $\Delta t/T_i$ go to one would then dictate what Δt would have to be for a T_i suitable for the system. The criterion would be:

$$\Delta t = T_i \qquad (3\text{-}15)$$

Using this sampling interval, however, the condition that Δt be small relative to the dominant time constant would no longer be satisfied, because the value of T_i used in process control is often equal to or greater than the dominant time constant of the system.* The only basis for this condition of small Δt is, however, the desire to keep the digital controller response as similar as possible to analog controller response. This paper will show that in fact satisfactory control is possible even when the simplification

---

* Selection of K_p and T_i for a given process is discussed in Appendix A and references

of (3-15) is made, provided that the proportional term of the control algorithm is sampled and computed at a rate which is different from and sufficiently higher than the integral rate. The reason it is necessary to sample the proportional part often while the integral part can be sampled only every T_i is concerned with system stability. The only function of the integral mode is to eliminate steady state error, whereas the proportional mode is a stabilizing action, acting to counteract change in the controlled variable. It is important that no extra lag be introduced in the determination of this rate of change, since in well tuned control loops the proportional mode generally causes the major share of manipulated variable motion. A sampling lag on the order of T_i introduced in the proportional effect would act like an effective dead time and would decrease phase margin, requiring a much reduced proportional gain and hence a reduced speed of response for a given degree of stability. If this large sampling lag is introduced only into the integral effect, a certain reduction is required in integral gain, but the proportional gain and effect can remain nearly unchanged. Chapter VII discusses the extent to which this reasoning was borne out in experimental tests, and Chapter VIII proposes a method for providing a smoother integral effect without abandoning the desired feature of gain adjustment by frequency manipulation. The effect upon stability of various sampling

schemes is discussed further in Chapter VI. Appendix B evaluates the effect of noise in feedback signal b on the different control modes as a function of the sampling rate used in the different parts of the control algorithm.

Priority Allocation and Computation Frequency of Algorithm Terms

Applying (3-15) to (3-12) :

$$\Delta m_k = K_p \left[(b_{k-1} - b_k)_{f_p} - (r_{k-1} - r_k)_{\Delta r} - (b_k - r_k)_{f_i} \right] \qquad (3\text{-}16)$$

where $f_i = \frac{1}{T_i}$

and $(b_k - r_k)_{f_i}$ means that this "integral" term is computed at a rate of f_i. The first term is computed at a higher rate f_p, while the second term is computed only when a change occurs in set point r. Since these terms are computed at different times it is no longer meaningful to keep them together in one equation, since the contributions to Δm do not all occur together at a given instant k. Therefore a new notation is introduced to describe the actual separate operations performed in the controller. In a previous section the binary output of the controller was defined as A_k, with $A_k = \Delta m_k / K_p$. Then

$$A_k = (b_{k-1} - b_k)_{f_p} - (r_{k-1} - r_k)_{\Delta r} - (b_k - r_k)_{f_i} \qquad (3\text{-}17)$$

Because the three components of A_k are generated at different times, a separate expression is defined for each term:

Proportional terms:

$$A_{kp} = (b_{k-1} - b_k)_{f_p} \qquad (3\text{-}18)$$

$$A_{kr} = -(r_{k-1} - r_k)_{\Delta r} \qquad (3\text{-}19)$$

Integral term:

$$A_{ki} = -(b_k - r_k)_{f_i} \qquad (3\text{-}20)$$

In normal operation set point r is a constant and A_{kr} does not need to be computed. A_{kp} and A_{ki} are then sufficient to generate the corrections required for two mode PI control. When a change in set point r is ordered, A_{kr} of (3-19) must of course be computed and applied as a correction to the valve. While this is being done, however, there is no reason why A_{kp} and A_{ki} must continue to be computed since the process is most likely very near steady state, making A_{kp} and A_{ki} very small compared to A_{kr}. After a change in r is initiated by application of the increment A_{kr} to the actuator the other two terms of the control algorithm can again be computed in their usual sequence. When the algorithm is broken up in this way the controller has two distinct modes of operation. In the "Operate" mode the controller computes A_{kp} and A_{ki}, representing both proportional and integral effects for con-

stant r. Whenever r is changed, but at no other time, the controller is shifted to the "Read" mode to enable it to compute and apply A_{kr}. The expression "Read" indicates that a new r is to be read in. The controller returns to the Operate mode as soon as this computation is performed. It is evident that the Read mode has the higher priority, since whenever we wish to make a change in set point it is important to get the process started in that direction. Once it is started by the A_{kr} term there is time enough to return to the Operate mode for proper transient response.

The motivation for dividing up the duties of the controller is to permit a small amount of logic to perform several functions. We have seen that the Read and Operate modes are mutually exclusive. It is therefore possible to design the controller so that the logic that performs (3-18) and (3-20) in the Operate mode also performs (3-19) in the Read mode. This is especially easy since all three equations are simple subtractions. In the Operate mode equations (3-18) and (3-20) must be performed, but at different rates. It has been shown that for stability the sampling rate f_i for the integral term (3-20) must be a small fraction of the rate f_p for the proportional term. It is therefore possible to perform the integral computation during the interval between successive proportional

computations without causing any significant variation in the effective value of f_i. Thus, the two terms used while the controller is in the Operate mode can be computed and applied to the actuator successively, making it possible to perform both subtraction operations in the same register without an intermediate storage register. The sequence of events chosen for implementation of the controller is this: During Operate mode proceed with computation of the proportional term (3-18) at frequency f_p. When an integral order occurs, calling for execution of (3-20), wait until the next proportional computation (3-18) is completed, then immediately perform the subtraction (3-20). When (3-20) is complete, return to the proportional computation (3-18).

The following table summarizes the priority of the three computations, and the conditions that must be satisfied before they are carried out:

Term	Conditions that must be met to enable computation of respective term			
	Computation Ordered	Operate Mode	Countdown not in progress	Countdown just completed
A_{kr}	x			
A_{kp}	x	x	x	
A_{ki}	x	x	x	x

Table 3-1

Figure 3.3 shows the valve position m as a function of time resulting from an arbitrary feedback signal b. It happens that the two integral corrections shown have signs opposite to the signs of the proportional corrections just preceeding them. The figure demonstrates how the addition of the proportional and integral effects is performed by the actuator itself, rather than by the controller. This same general principal was used in the controllers designed by Walker[12] and Peatman[13], but in both those cases the adjustment of T_i was accomplished by providing variable gain within the controller rather than variable integral sampling rate.

Intermediate Storage Requirements

Before starting the detailed logic design of the controller it is necessary to determine what intermediate storage capability is required.

In the Read mode the solution of (3-19) requires storage of the old value of r, r_{k-1}. It is sufficient to enter r_{k-1} into the subtraction register as soon as the controller is put into the Read mode. Then the switches representing r can be changed and the new value of r subtracted from the old. Thus no auxiliary storage is required, outside of the r switches and the subtraction register. The factor

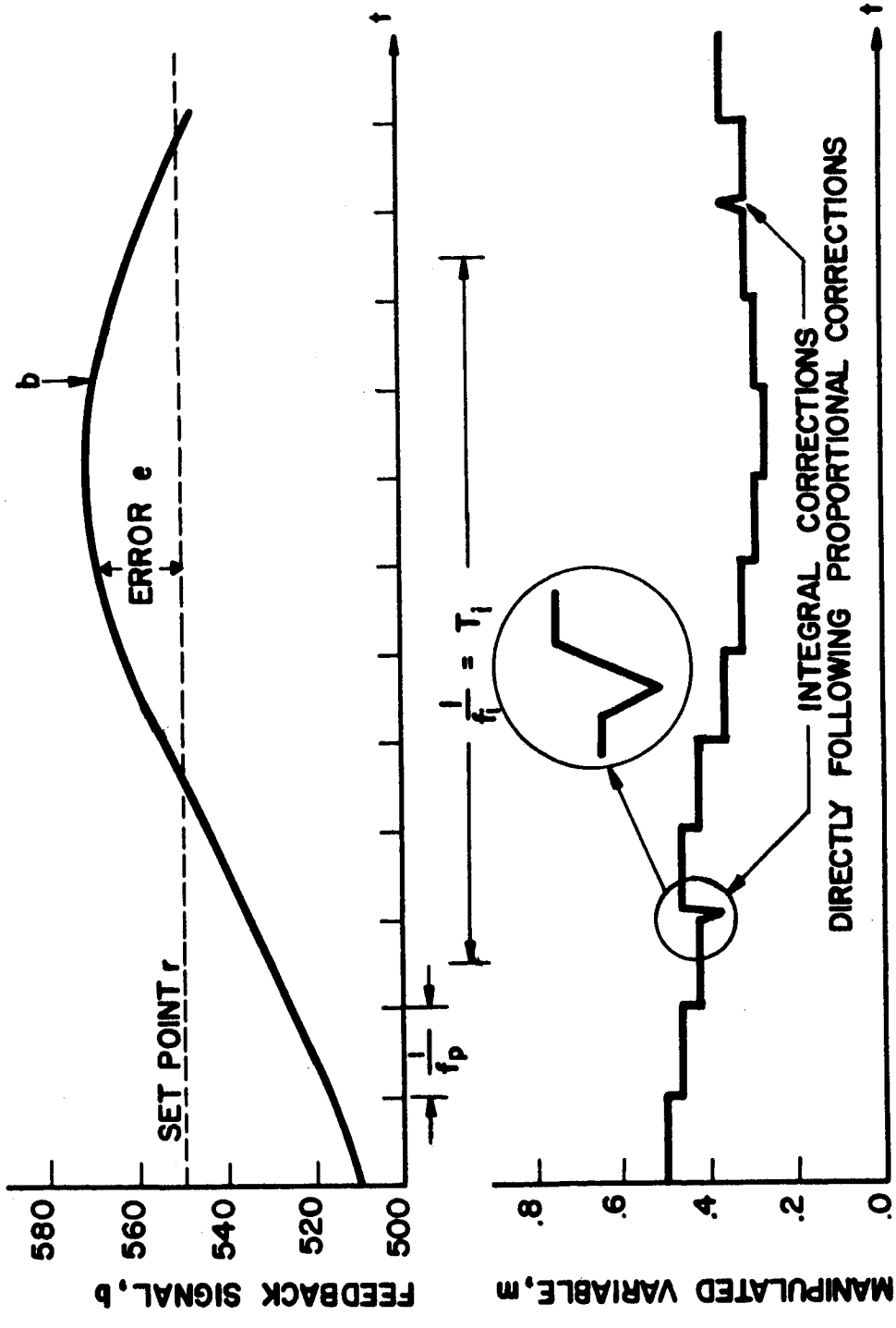


FIGURE 3-3 INTEGRATED OUTPUT OF MULTIRATE SAMPLED DIGITAL CONTROLLER FOR $K_p = .05$

that makes this savings possible is the restriction that no set point changes be made without first putting the controller into the Read mode.

In the Operate mode the solution of the integral term (3-20) requires the subtraction of present r from present b. No intermediate storage is required, but b must be entered into the subtraction register just before the subtraction is performed. This is done immediately after applying the proportional term, (3-18), to the actuator. (3-18) is the difference $b_{k-1} - b_k$, which obviously does require intermediate storage. According to the equation, the value of b_k that is used as the subtrahend in the calculation of A_{kp} would be retained and would then be used as the minuend in the computation of $A_{(k+1)p}$. To retain b_k would require an extra memory register. This extra register can be avoided, however, by making a further approximation. In Chapter II it was pointed out that in process control the actuator time constant is usually small compared to that of the process. This statement is further developed in Chapter V to show that when using a constant speed incremental actuator the absolute average speed the actuator requires to maintain good control is small compared to its full speed capability. This is another way of saying that the average time required for

actuation during a sampling interval is small compared to the length of that sampling interval. This is evident in Figure 3.3 , which is similar to test results obtained with the process of Figure 2.2 . If we now define δ as the fraction of the sampling interval required for computation and actuation we can, under certain conditions, make the approximation

$$b_k \cong b_{k+\delta} \qquad (3\text{-}21)$$

Using $b_{k-1+\delta}$ instead of b_{k-1} in (3-18) means that at a sampling instant k the computation (3-18) can be made without retaining the value of b_k. At time $k+\delta$, immediately after A_{kp} has been applied to the actuator, b is resampled and stored in the subtraction register where it is ready for the next computation. If an integral sample has been ordered (3-20) is performed as soon as $b_{k+\delta}$ is entered, since $b_{k+\delta}$ is then the present value of b and can serve as the minuend. At time $k+2\delta$, after A_{ki} has been applied to the actuator, b is again entered into the register where it serves as the minuend b_k for the next proportional term (3-18). δ is understood to depend upon the magnitude of the difference term A_k and the countdown rate f_a.

The conditions under which (3-21) is valid are discussed in Chapter V and include consideration of the amount of noise on signal b, and the average value of δ.

Summary

Data input and output requirements for a sampled digital controller were stated, including the decision to use 10 bit parallel binary input variables and an incremental output order in the form of a pulse width for the control of a constant speed actuator such as a synchronous motor or stepping motor. The continuous three mode control algorithm was converted to incremental, sampled form, then for the two most commonly used modes, proportional plus integral, the incremental equation was divided into three component parts well suited for serial computation in shared logic. The two salient characteristics of process control loops listed in Chapter II were the basis of a heuristic justification of the serial computation scheme, the method of gain adjustment by varying sampling rate and the approximation made in a first difference term to eliminate the need for an auxiliary storage register.

CHAPTER IV
DETAILED DESIGN OF THE CONTROLLER

The preceeding chapter has provided the sampled control algorithm roughly corresponding to continuous proportional plus integral control and has described how this algorithm can be implemented in a digital device without recourse to any electronic storage register beyond the basic subtraction register. In this chapter the steps necessary to perform the algorithm are tabulated and the digital logic built to implement those steps is described.

Sequence of Controller Operations

The controller must compute the three differences of equations (3-18), (3-19) and (3-20), each at a different time. (3-18) and (3-20) are computed while the controller is in its Operate mode, whereas (3-19) is computed only once each time the set point r is changed, when the controller is momentarily put into the Read mode. Grouping these three terms by controller modes :

Computed during Operate mode :

$$A_{kp} = (b_{k-1} + \delta - b_k)_{f_p} \qquad (4\text{-}1)$$

$$A_{ki} = -(b_k - r_k)_{f_i} \qquad (4\text{-}2)$$

Computed during Read mode :

$$A_{kr} = -(r_{k-1} - r_k)_{\Delta r} \qquad (4\text{-}3)$$

In accordance with the assumption made in Chapter III $b_{k-1+\delta}$ is used in (4-1) instead of b_{k-1} . The serial computation scheme proposed permits (4-2) to be performed immediately following any (4-1) computation.

It was shown in Chapter III that it would be possible to perform the necessary computations with a controller of the structure shown in figure 4.1 . The only electronic memory register of the controller is the accumulator A, which consists of a sign flip flop S and 10 data flip flops, all of which are trigger elements with sufficient input gating to permit a minuend to be entered into the register. Further gating is provided to permit a second number, the subtrahend, to enter the register and propagate the borrows necessary to form a difference.[15] Set point r is available as the level outputs of a set of 10 electromechanical switches. Feedback b is always available as level outputs from a binary encoder. The controller could also operate by requesting and waiting for a value of b from a shared A/D converter.

Using the structure shown in Figure 4.1 , Table 4-1 lists the series of steps necessary to perform equation (4-1).

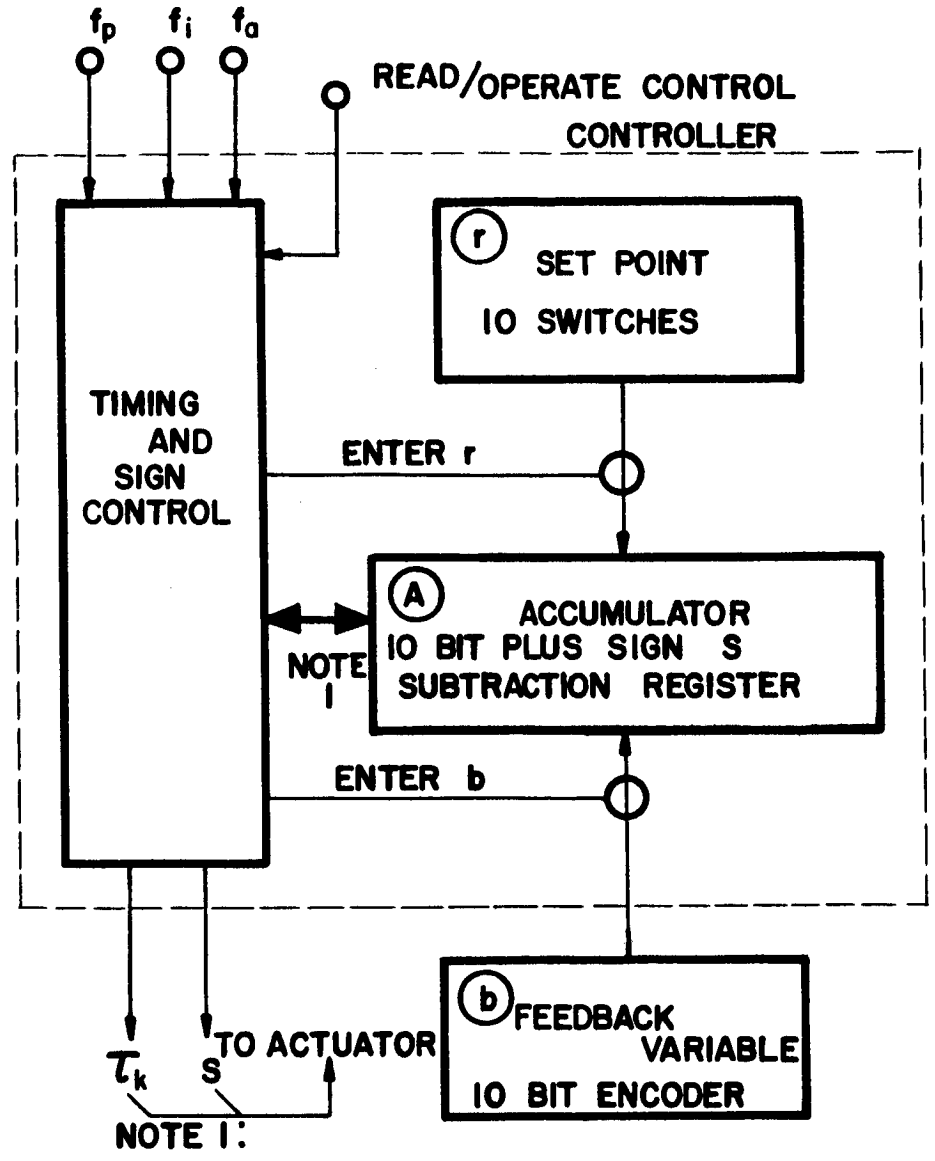


FIGURE 4-1 SCHEMATIC OF CONTROLLER AND ENCODER

Table 4-2 shows how the cycle of Table 4-1 must be varied to compute equation (4-2) between proportional samples.

Table 4-3 includes a change of set point to show a sequence in which all three algorithm equations are computed.

The first entry in Table 4-1 shows the contents of register A just before the k^{th} proportional sampling instant. When the sample order comes at time k the solution of (4-1) requires that b_k be subtracted from A. The subtraction takes place in 250 microseconds, after which the difference A_{kp} is counted down to zero at a frequency of f_a. As long as counting is in progress the actuator is driven at full speed in the direction corresponding to S, the sign of A. If S is positive the output is decreasing and the actuator is therefore driven to open the valve. When A becomes zero the countdown is stopped and the present value of b, $b_{k+\delta}$, is transferred to A. $b_{k+\delta}$ is then the minuend for the subtraction occurring at sample instant k + 1.

Time $f_p \cdot t$	Controller Mode	Order	Contents of Register A upon Completion of Order
k -	Operate	-	$b_{k-1+\delta}$
k	Operate	Prop. Order: Sub b from A Countdown A at f_a	A_{0}^{kp}
$k + \delta$		Transfer b to A	$b_{k+\delta}$
k + 1	Operate	Prop. Order: Sub b from A Countdown A at f_a	$A_{0}^{(k+1)p}$
$k+1+\delta$		Transfer b to A	$b_{k+1+\delta}$
k + 2	Operate	Prop. Order: Sub b from A Countdown A at f_a	$A_{0}^{(k+2)p}$
$k+2+\delta$		Transfer b to A	$b_{k+2+\delta}$

Operations Performed by Controller

Table 4-1

The same procedure is followed in Table 4-2 until after time $k + \delta$, when A_{kp} has been completely applied to the actuator. The pulse trains f_p and f_i for proportional and integral sampling orders need not be synchronized, making it possible for an integral order to arrive at the controller at any time in the proportional sample cycle. An integral order is shown occurring between proportional samples k and k + 1, at a time called "k + δ +" in the table. To minimize degradation of proportional performance no action is taken on the integral order until the next proportional sample is completed. At time $k + 1 + \delta$ the transfer of present b to A is completed and immediately thereafter the integral computation (4-2) is performed. The sign S of the difference is changed to provide the minus sign in front of the difference, then the countdown to the actuator proceeds exactly as for a proportional term. After countdown, b is transferred to A, ready for the next proportional order.

Time $f_p \cdot t$	Controller Mode	Order	Contents of Register A upon Completion of Order
k-	Operate	-	$b_{k-1+\delta}$
k	Operate	Prop. Order: Sub b from A	A_{kp}
		Countdown A at f_a	0
$k + \delta$		Transfer b to A	$b_{k+\delta}$
$k+\delta+$	Operate	Int. Order: Execute after next prop. order is complete	$b_{k+\delta}$
$k + 1$	Operate	Prop. Order: Sub b from A	$A_{(k+1)p}$
		Countdown A at f_a	0
$k+1+\delta$		Transfer b to A	$b_{k+1+\delta}$
		Sub r from A	$-A_{(k+1)i}$
		Change sign of A	$A_{(k+1)i}$
		Countdown A at f_a	0
$k+1+2\delta$		Transfer b to A	$b_{k+1+2\delta}$
$k + 2$	Operate	Prop. Order: Sub b from A	$A_{(k+2)p}$
		Countdown A at f_a	0
$k+2+\delta$		Transfer b to A	$b_{k+2+\delta}$

Operations Performed by Controller

Table 4-2

Table 4-3 follows the same pattern as Table 4-2 until after time $k+1+2\delta$. At some time between then and the next sampling instant, called "$k+1+2\delta+$" in the table, the manual mode control switch is turned to the Read position in preparation for a change of set point. A rotary Read switch can be used to permit register A to be manually reset by momentarily opening the ground line to the zero side of the flip flops just before the old value of r is transferred into A by means of the trigger inputs. With the old value of r recorded in A the set point can be changed at the operator's leisure. As long as the controller is in the Read mode no further computations can take place and the actuator stays where it is. As soon as the new value of r has been entered, shown in the table as time $k + 3 +$, the operator turns the switch from the Read position around to the Operate position. The transition causes the new r to be subtracted from A. The sign of A is changed and countdown to the actuator proceeds. After countdown, b is transferred to A and the controller is ready for further Operate mode computations.

Time $f_p \cdot t$	Controller Mode	Order		Contents of Register A upon Completion of Order
k-	Operate	-		$b_{k-1+\delta}$
k	Operate	Prop. Order:	Sub b from A Countdown A at f_a	$A_{0\,kp}$
k+δ			Transfer b to A	$b_{k+\delta}$
k+δ+	Operate	Int. Order:	Execute after next prop. order is complete	$b_{k+\delta}$
k+1	Operate	Prop. Order:	Sub b from A Countdown A at f_a	$A_{0\,(k+1)p}$
k+1+δ			Transfer b to A	$b_{k+1+\delta}$
			Sub r from A	$-A_{(k+1)i}$
			Change sign of A Countdown A at f_a	$A_{0\,(k+1)i}$
k+1+2δ			Transfer b to A	$b_{k+1+2\delta}$
k+1+2δ+	Read	Reset A Transfer old r to A		0 $r_{k+1+2\delta+}$
k+3+	Operate	New r ready:	Sub r from A	$-A_{k+3+}$
			Change sign of A Countdown A at f_a	$A_{0\,k+3+}$
			Transfer b to A	$b_{k+3+\delta+}$
k+4	Operate	Prop. Order:	Sub b from A Countdown A at f_a	$A_{0\,(k+4)p}$
k+4+δ			Transfer b to A	$b_{k+4+\delta}$

Operation Performed by Controller

Table 4-3

Conversion of Sequence Rules to Boolean Statements

The next step necessary in the design of the controller is to convert the preceeding tables and explanations into explicit logical statements that lend themselves to implementation in digital logic. Boolean algebra terminology will be used not only for the combinational logic involved but also to state under what conditions one event or transition follows another. The latter function, vital in the synthesis of asynchronous logic, is discussed by Mergler[16]. It permits one to write, for instance, an expression $T_C = A\,\alpha_B$, meaning "Trigger flip flop C when B makes an α transition if A is one at that time." α_B is defined as the 0 to 1 transition of binary variable B, and β_B as the 1 to 0 transition of B. The flip flops and delay devices used in the controller are all activated by β transitions.

In a sequential circuit it is often desirable to have one operation follow another after a finite delay. In one transfer operation, for instance, it is necessary to trigger all flip flops of register A to make them all zero, then trigger certain of them to enter a number. Depending upon how it is done, it may take anything from one to five microseconds for the first triggering operation to be completed. In such a case it is useful to be able to take the pulse that orders the first part of the operation, delay it about 10 micro-

seconds, and use the delayed pulse to order the second part of the operation. This technique is used extensively in the design of the controller. To provide a symbol to cover this case we follow the lead of Arnstein[17] and represent an event B delayed by n microseconds as B with superscript μn. Thus if A is to take place 20 microseconds after pulse B, we would write $A = B^{\mu 20}$.

We are now prepared to make generalized rules from the sequences of tables 4-1 through 4-3 and convert them into Boolean statements as a preliminary step to implementation of the controller in digital logic.

Transfer of a Number to a Register

For register A the decision was made to use a flip-flop with six separate gated trigger inputs and a manual reset capability. The controller logic diagram, Figure 4.2 , shows that not quite all of these inputs are used, although they would be used in a suggested modification to the controller. Figure 4.2 also shows that two of the trigger inputs are delayed a time τ . This time delay is about 4 μsec. Since there is no automatic set or reset input on the flip-flops chosen, transfer of a number to A is performed by taking advantage of the fact that every transfer order is immediately preceeded by a countdown order. As will be shown, the accumulator register A is all ones upon

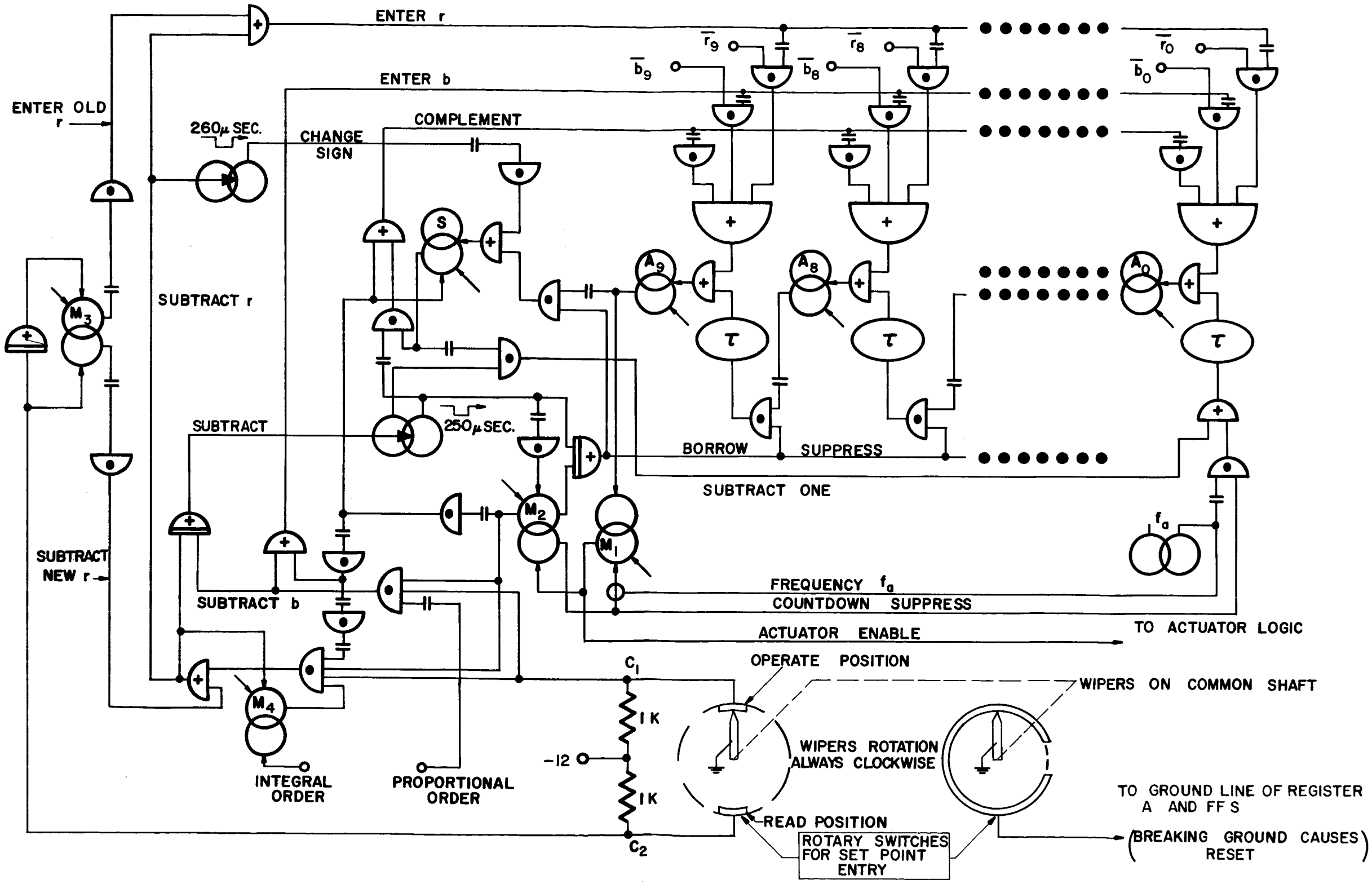


FIGURE 4-2 CONTROLLER LOGIC DIAGRAM

completion of a countdown order. To enter a number b, then, it would be possible to simply trigger those flip flops of A for which the corresponding bits of b are zero. The number b would then be in A. To keep the transfer operation compatible with the subtraction operation, however, it was decided to first complement A after countdown, making A all zeroes, then trigger the flip flops of A corresponding to the bit positions of the incoming number whose values are one.

Subtraction

The method of direct subtraction[15] is used, whereby each bit of the minuend in A is decreased by the value of the corresponding bit of the subtrahend, and if a bit in A thereby becomes less than zero, it must become one and the next higher order bit must be reduced by one; that is, a one must be borrowed from the next higher order. When all borrows have been propagated the difference appears in register A. If, however, the difference is negative, a borrow is generated by an α transition in A_9, making the sign bit S one, representing minus. The difference then appears in A in two's complement form and must be converted to one's complement form by subtracting one from A_0, the so-called "end-around-carry". A is then complemented to convert the one's complement form to natural binary form. Because borrows may propagate through all

ten stages of A, in the worst case, and further borrows may occur after the end-around-carry, it is necessary to delay the complement order until well after the subtraction begins. Another requirement is that no borrows can be permitted to propagate during the complement process, as is true also during the transfer process.

Countdown of a Number

The countdown operation produces a time period proportional to the difference generated in A. The number in A is repeatedly decreased by one until A makes the transition all zeroes to all ones. This transition is easier to recognize than the desired transition 00...01 to all zeroes. If the difference number to be counted down is n, however, then n + 1 pulses are required to produce register underflow. Therefore, to produce a period of time proportional to n, the period is begun by the first countdown pulse and ended by the underflow resulting from the (n + 1)th pulse. To keep the controller as versatile as possible the countdown clock is not required to be synchronized with any other operation in the controller. It was therefore necessary to use two control flip flops, M_1 and M_2, to provide the desired countdown characteristics. M_2 is set whenever a difference has been generated in A and is ready for countdown. M_1 is set when the first countdown pulse occurs after M_2 has been set. M_1 and

M_2 are both reset when A underflow (a_{A_9}) occurs. The actuator is then on as long as M_1 remains set.
Using the standard logical connectives:

Set M_2 = Subtraction Complete (4-4)

Set M_1 = (Countdown Pulse) · (M_2) (4-5)

Reset M_2 = a_{A_9} (4-6)

Reset M_1 = a_{A_9} (4-7)

Actuator Enable = M_1 (4-8)

Actuator Direction = S (4-9)

Operate Mode Cycle

Referring to Table 4-2 we can write:

Subtract b = ($\overline{M_2}$) · (Operate) · (Prop. Order)$^{\mu 10}$ (4-10)

The 10 microsecond delay is provided to allow the borrow gates time enough to be activated before they are needed.

The borrow enable line must be on for any subtraction, then go off to permit a possible complement. 250 microseconds were allowed in this design, which is more than adequate for worst case borrow propagation. The borrow line must also be enabled for countdown.
Therefore:

Borrow Enable = (Subtract + 250 μ s) + M_2 (4-11)

The "Subtract + 250 μs" indicates a pulse initiated by any subtract order and stretched to a width of 250 μs.

The "end-around-carry" requirement is met by:

$$\text{Subtract one} = (a_s) \cdot (\text{Subtract} + 250\ \mu s) \qquad (4\text{-}12)$$

Further, providing for complement before transfers:

$$\text{Complement} = (S) \cdot (\text{Subtract})^{\mu 260} + (\beta_{M_2})^{\mu 10} \qquad (4\text{-}13)$$

$$\text{Countdown Enable} = M_2 \qquad (4\text{-}14)$$

The reason for complementing A before transferring b into it is to make that entry operation the same as for subtracting b. That is, b is entered into A in the same way, whether it is minuend or subtrahend. The only difference is in whether the borrow gates are enabled. Therefore

$$\text{Enter b} = (\text{Subtract b}) + (\beta_{M_2})^{\mu 20} \qquad (4\text{-}15)$$

To permit incoming integral orders to be deferred until after the next proportional order a storage flip flop M_4 is provided.

$$\text{Reset } M_4 = \text{Integral Order} \qquad (4\text{-}16)$$

$$(\text{Subtract r})_i = (\overline{M_2}) \cdot (\text{Operate}) \cdot (M_4) \cdot (\beta_{M_2})^{\mu 40} \qquad (4\text{-}17)$$

$$\text{Set } M_4 = (\text{Subtract r})_i \qquad (4\text{-}18)$$

To allow for the minus sign in front of (4-2) and (4-3), and

yet not change sign until after any possible complement

$$\text{Change Sign} = (\text{Subtract r})^{\mu 260} \qquad (4\text{-}19)$$

Read Mode Cycle

In accordance with the Read entry in Table 4-3 the A register must first be reset upon entering the Read mode. Since mode selection is already a manual operation a rotary switch is used to momentarily break the ground connection to the emitters of the zero side of the A flip flops. Further rotation of the switch brings it to the Read position. This arrival transfers the old value of r to A. Flip flop M_3 serves only to square up the Read signal transitions enough for them to be able to drive other logic elements. In this one case it is useful to use a flip flop with a DC set and reset.

$$\text{Transfer } r = \beta_{M_3} \qquad (4\text{-}20)$$

After r has been changed the switch is turned further, back to the Operate position.

$$(\text{Subtract } r)_{\Delta r} = (\alpha_{M_3})^{\mu 10} \qquad (4\text{-}21)$$

The delay again allows the borrow gates to be enabled.

Combining (4-17) and (4-21)

$$\text{Subtract r} = (\alpha_{M_3})^{\mu 10} + (\overline{M_2}) \cdot (M_4) \cdot (\text{Operate}) \cdot (\beta_{M_2})^{\mu 40} \qquad (4\text{-}22)$$

This is the expression that was anticipated in (4-19).

For all cases in which r is applied to A, whether as minuend or subtrahend:

$$\text{Enter } r = (\text{Subtract } r) + (\text{Transfer } r) \qquad (4\text{-}23)$$

For control of the subtraction process we define an expression needed for initiation of the subtraction routine, whether for the one equation for which b is the subtrahend or for the two equations for which r is the subtrahend:

$$\text{Subtract} = (\text{Subtract } b) + (\text{Subtract } r) \qquad (4\text{-}24)$$

Implementation of Boolean Statements by Logical Elements

Figure 4.3 is a logic diagram of the controller that implements the preceeding algorithm. In tracing the steps in its operating cycle it is helpful to refer to Table 4-3 and to the controller timing diagram, Figure 4.3 . The timing diagram shows the states of various important parts of the circuit as a function of time for the cycle of events shown in Table 4-3 from time (k + 1) to time (k + 4). The lower of the two possible levels that a variable can take represents its "one" or "asserted" value.

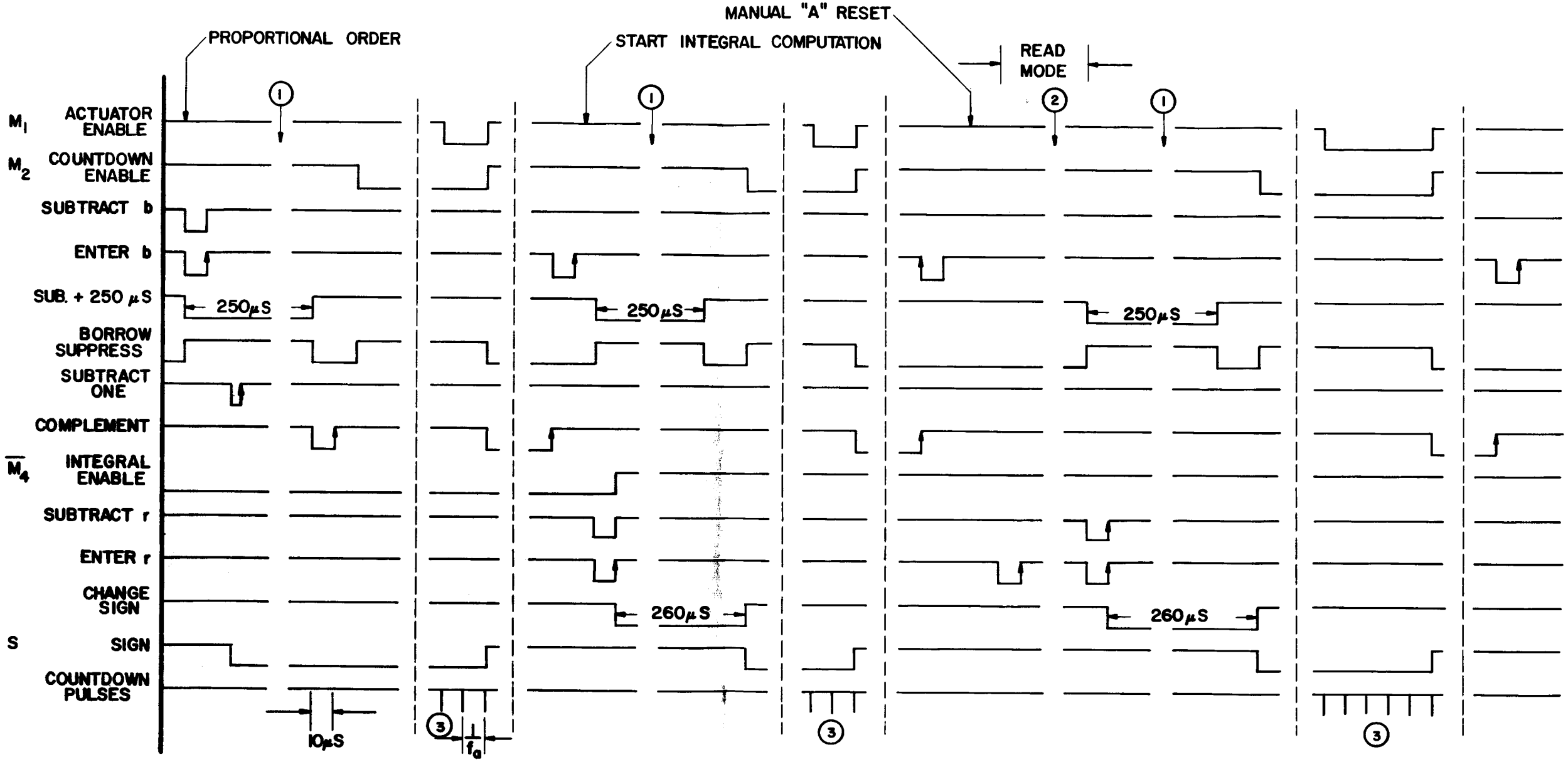


FIGURE 4-3 CONTROLLER TIMING DIAGRAM

Each flip flop in Figure 4.2 has a diagonal arrow indicating which side is made true by pushing the initialize button. This button opens a normally closed switch, opening the ground line to the emitter of one side of each flip flop. When this occurs the A register goes to all zeroes and M_1 through M_4 are set. This is the condition that prevails at the end of an A_{kp} countdown; one more countdown pulse drives A to all ones, M_1 and M_2 are reset and the present value of b is transferred into A. The controller is then ready for normal automatic operation.

According to (4-4) M_2 is to be set when the subtraction is complete. The complement operation of (4-13) is the last step in subtraction, so (4-4) is satisfied by setting M_2 10 μs after the complement order or 20 μs after the Subtract one-shot ends its 250 μs pulse. A pulse generator with a 20 μs delay is therefore chosen to set M_2. This delay is necessary to prevent borrows from being propagated by the complement operation. A steering gate is used to implement (4-5), since it simply calls for the ANDing of a pulse and a level. Reset of M_1 and M_2 occurs for underflow of A, as stated in (4-6) and (4-7).

A proportional computation for A_{kp} is made if the conditions of (4-10) are met; that is, if the controller is in the Operate mode and a countdown is not in progress. The equation is satisfied by a three input gated pulse generator

whose output pulse is 10 μs wide. The gates following this element are so arranged that the trailing edge of the pulse applies b to the A triggers, providing the desired delay. The expression for Borrow Enable is given by (4-11). Gated pulse generators and steering gates, however, require a zero or "false" level to enable an output pulse to be generated. Therefore, we provide

$$(\text{Borrow Suppress}) = \overline{(\text{Borrow Enable})} \qquad (4\text{-}25)$$

A two input NOR gate is sufficient to produce the combination of (4-25) and (4-11).

Equation (4-12) calls for ANDing a pulse and a level. The Subt ract One line is shown produced by a gated pulse generator. This function can also be provided by one of the steering gates included on the A_o card.

The Complement order of (4-13) is to occur for either of two conditions. Each of these conditions requires the presence of a certain instantaneous order or transition. A Complement order (a β transition) is required whenever either of two other transition occurs. The two input transitions never occur closer than 250 μs apart, so it is possible to implement the equation by converting the two input pulses into 10 μs wide pulses, using gated pulse generators, and ORing them. The effective Complement order, then, is the trailing edge, or β transition, of the

output pulse. This technique works because whenever an input pulse occurs from one of the gated pulse generators the other input is zero. The OR gates shown in Figure 4.2 were actually synthesized by connecting two NOR gates in series, thus preserving the polarity of the original pulses. The Subtract order, on the other hand, is produced by a NOR gate, which inverts incoming pulses. The β transition, which triggers flip flops and activates one-shots and gated pulse generators, is then the leading edge of the pulse at the NOR gate output when it is the trailing edge at the input. This effect can be observed in Figure 4.3, where the β transitions of Enter b pulses are the trailing edges, but the "Subtract + 250 μs " line is activated by the leading edges of "Subtract" pulses. The Enter r expression (4-23) uses an OR gate, providing a 10 μs delay between initiation of the pulse and actual entry. The Enter r pulse resulting from an integral order is already delayed 30 μs from the end of countdown. This delay allows time for completion of Complement and Enter b orders.

Controller Modification for Smoothed Integral Effect

In order to try out a different sampling scheme it was decided to provide the capability of taking integral samples more often than once per T_i, using a reduced gain for that

term. That is, instead of (3-20) we can write

$$A_{ki} = \frac{1}{4}(b_k - r_k)_{f_i} \qquad (4\text{-}26)$$

where $f_i = 4/T_i$. This has the effect of smoothing the integral effect. The method used to accomplish this is to use a countdown rate of $4f_a$ for the integral term countdown. This can be implemented with the logic of Figure 4.4. Flip flop M_5 is added to indicate how long $4f_a$ should be used for countdown. M_5 is set as soon as M_4 is set, to signal the beginning of an integral computation. It stays set until M_1 is reset, signifying the end of that countdown. The gated pulse generators permit $4f_a$ to pass through the NOR gate only during this integral countdown. At all other times f_a passes. The output of the NOR gate of Figure 4.4 then replaces the "f_a" line coming from the f_a FRMV in Figure 4.2. The f_a FRMV of Figure 4.2 is the one used in Figure 4.4 and called "Countdown FRMV". To add the smoothing capability, then, requires nine extra transistors; two for each flip flop and one for each gated pulse generator and NOR gate. Each additional stage of smoothing requires another flip flop and, of course, a doubling of the frequency of the countdown FRMV.

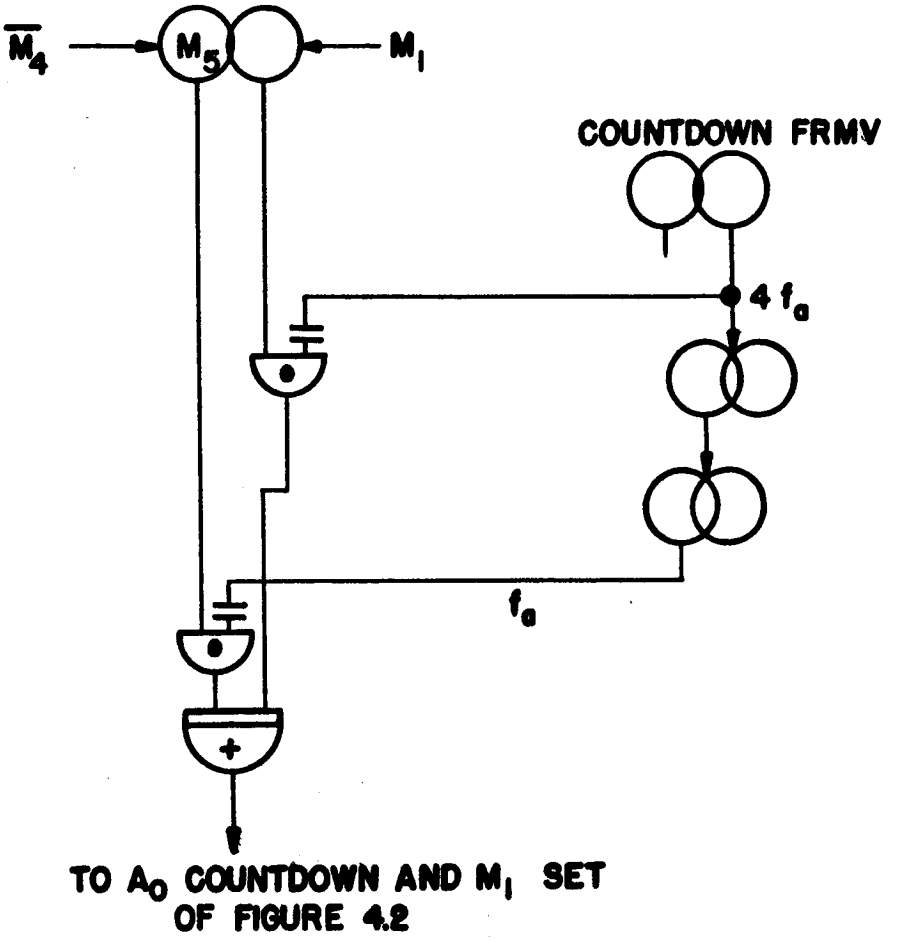


FIGURE 4-4 LOGIC REQUIRED FOR SMOOTHED INTEGRAL EFFECT

System Photographs

Figure 4.5 is a front view of the controller.
Figure 4.6 is a photograph of the controller with the system used in the experiments.

Summary

A single subtraction register with associated control logic was designed to perform the three computations called for in Chapter III. The resulting controller, shown in Figure 4.2, required five transistors for each bit of the subtraction register and fifty for timing and control logic, making a total of one hundred transistors per controller for ten bit resolution of set point and feedback variable. Provision was made for smoothed integral effect, using nine additional transistors. The design provides a greater range of control parameters than would be needed for control of any industrial process.

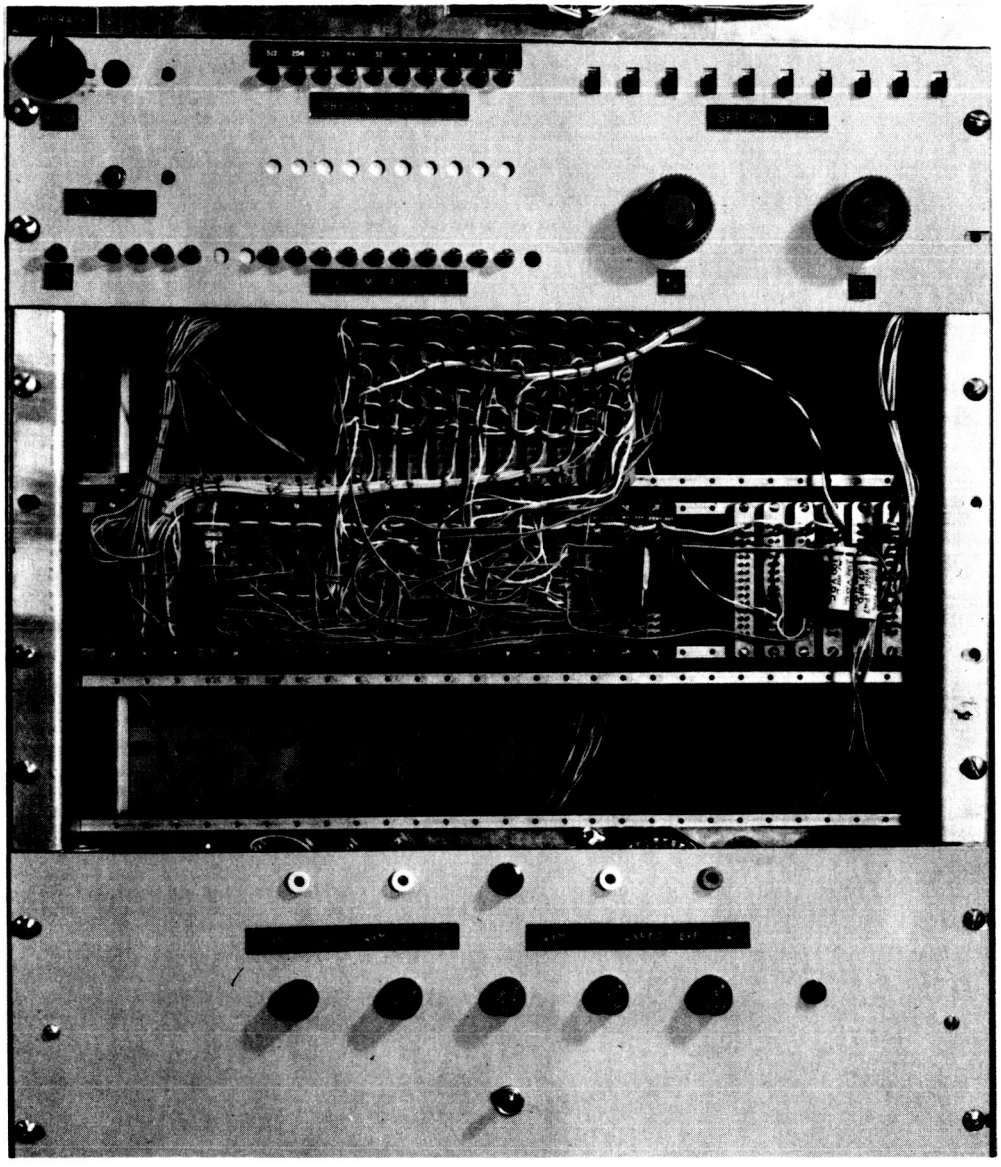


FIGURE 4-5 CONTROLLER

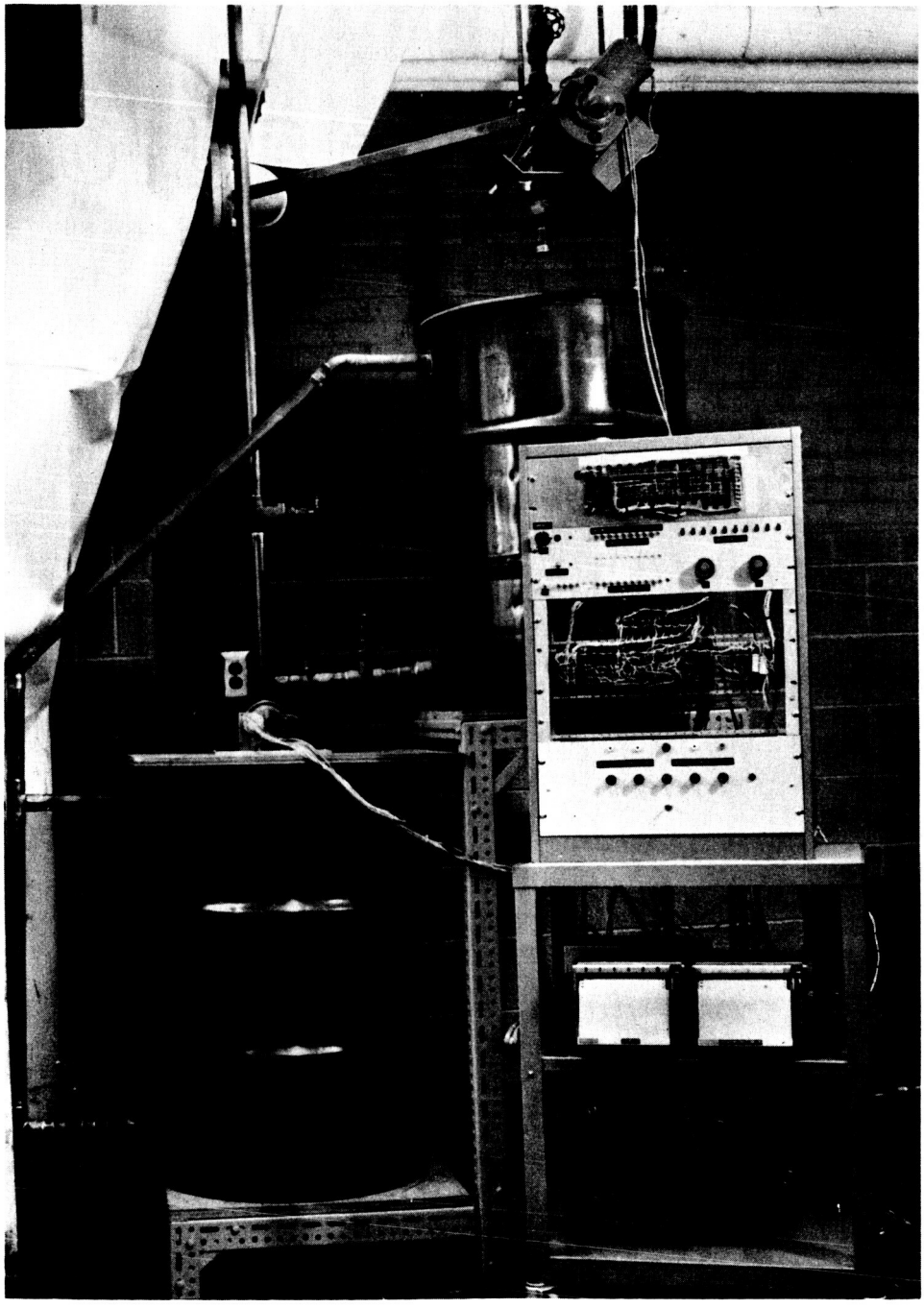


FIGURE 4-6 CONTROLLED SYSTEM

CHAPTER V

EFFECT OF ALGORITHM APPROXIMATIONS UPON CONTROLLER PERFORMANCE

The continuous PI control law was converted to the sampled, incremental form of (3-12). It was assumed that quantization was fine enough relative to transducer resolution to cause no significant effect on performance. (3-12) was converted to (4-1), (4-2) and (4-3) by making certain stipulations about sampling and computation rates for the different terms. The use of two different sampling rates for the two Operate mode terms (4-1) and (4-2) can cause process performance to vary considerably from normal PI control. This effect will be treated in Chapter VI by means of stability analysis. The proportional term (4-1) is an approximation to a first difference. The effect of this approximation is analyzed in this chapter. The other approximation to be treated in this chapter is the failure to compute the Operate mode terms (4-1) and (4-2) when the controller is in the Read mode. The effects of these approximations will be related to the dynamic characteristics of the actuator and process being controlled.

First Difference Approximation

The difference between last process output b_{k-1} and present output b_k was approximated in (4-1) by

$$A_{kp} = (b_{k-1} + \delta - b_k)_{f_p} \qquad (5\text{-}1)$$

where δ is the portion of the sampling interval required for computation of $A_{(k-1)p}$ and for its application to the actuator. The correct first difference would be

$$(A_{kp})_c = (b_{k-1} - b_k)_{f_p} \qquad (5\text{-}2)$$

Therefore, if b has a constant rate of change from time (k-1) to time (k),

$$A_{kp} = (1 - \delta) (A_{kp})_c \qquad (5\text{-}3)$$

Thus, the approximation has the effect of lowering proportional gain by a factor of (1-δ). Average effective proportional gain is further reduced by the fact that last output is not held in memory during the time that the integral term is being computed and counted out. Both sources of lost gain can be lumped together by noting that the proportional effect computation (5-2) does not include the change of b that occurs during computation and countdown periods. Computation time of 260 microseconds or

less is negligible in comparison to the minimum sampling interval of one tenth of a second.

We would like to evaluate the δ in (5-3) for a worst case to determine whether our approximation will have a serious effect upon stability. We can define a $\delta_{avg.}$ as the average value of δ that would result from a given pattern of b, choosing the pattern of b so as to require a large amount of valve motion.

The Ziegler-Nichols method of controller tuning set forth in Reference 18 preceeded the Cohen and Coon equations given in Reference 14 and worked out in Appendix A. While the former is not as precise as the latter, it is considerably simpler, and accurate enough to permit elimination of some of the variables in our evaluation of δ.

The maximum amount of valve travel occurs for the transient resulting from a large change in set point. With the controller properly tuned for moderate overshoot, process output has the form of Figure 7.1. According to Ziegler-Nichols, the period of oscillation of b in the closed loop is 5.7 L, where L is the apparent dead time in the loop, shown in Figure A.1. b can be roughly approximated, for the first part of the transient, by a straight line connecting

its starting point to the point of first crossing of new set point. If the period of oscillation is 5.7 L, then the time to first crossing is about 2 L. If we use a change in set point of two tenths of full range, we can express valve travel V between time zero and time 2 L as

$$V = K_p\left[\frac{.2\,L}{T_i} + .2\right] \qquad (5\text{-}4)$$

(5-4) was obtained from (2-1) by solving for m at time 2 L, letting $T_d = 0$, $m_o = 0$, and $K_f = 1$.

From Reference 18, for proper tuning,

$$T_i = \frac{L}{0.3} \qquad (5\text{-}5)$$

Substituting (5-5) into (5-4)

$$V = .26\,K_p \qquad (5\text{-}6)$$

To find $\delta_{avg.}$ we divide (5-6) by the maximum amount of motion the actuator is capable of making in the same period:

$$\delta_{avg.} = \frac{.26\,K_p}{2\,L/T_{fs}} = .13\,\frac{K_p T_{fs}}{L} \qquad (5\text{-}7)$$

Industrial processes are usually conservatively tuned, so that a moderate change in loop gain or load conditions will not cause instability. It should, for example, cause no

problem if controller gain is changed $\pm$20 percent. We can arbitrarily set this value as an upper limit for δ to find a zone of acceptability of our first difference approximation. Using (5-7), we can write

$$0.13\frac{K_p T_{fs}}{L} \leq 0.2 \qquad (5\text{-}8)$$

or

$$T_{fs} \leq 1.5\,\frac{L}{K_p} \qquad (5\text{-}9)$$

(5-9) sets an upper limit on time required for full stroke of the control valve given apparent dead time L and controller gain K_p. If (5-9) is satisfied, effective proportional gain will never be as much as 20 percent less than K_p for the transient case stated above. For small changes of set point and small disturbances effective gain will be proportionately closer to K_p. Based upon (5-9), Figure 5.1 shows lines of equal valve speed for given conditions of L and K_p. As shown in Appendix A, K_p is a function of L and the other loop response characteristics. If the inequality (5-9) cannot be satisfied for a certain loop, there is a danger of instability for large disturbances or large changes of set point.

As an example, the system of Figure 2.2 does satisfy (5-9). Appendix A gives $K_p = 3/10^3 q$. To convert this

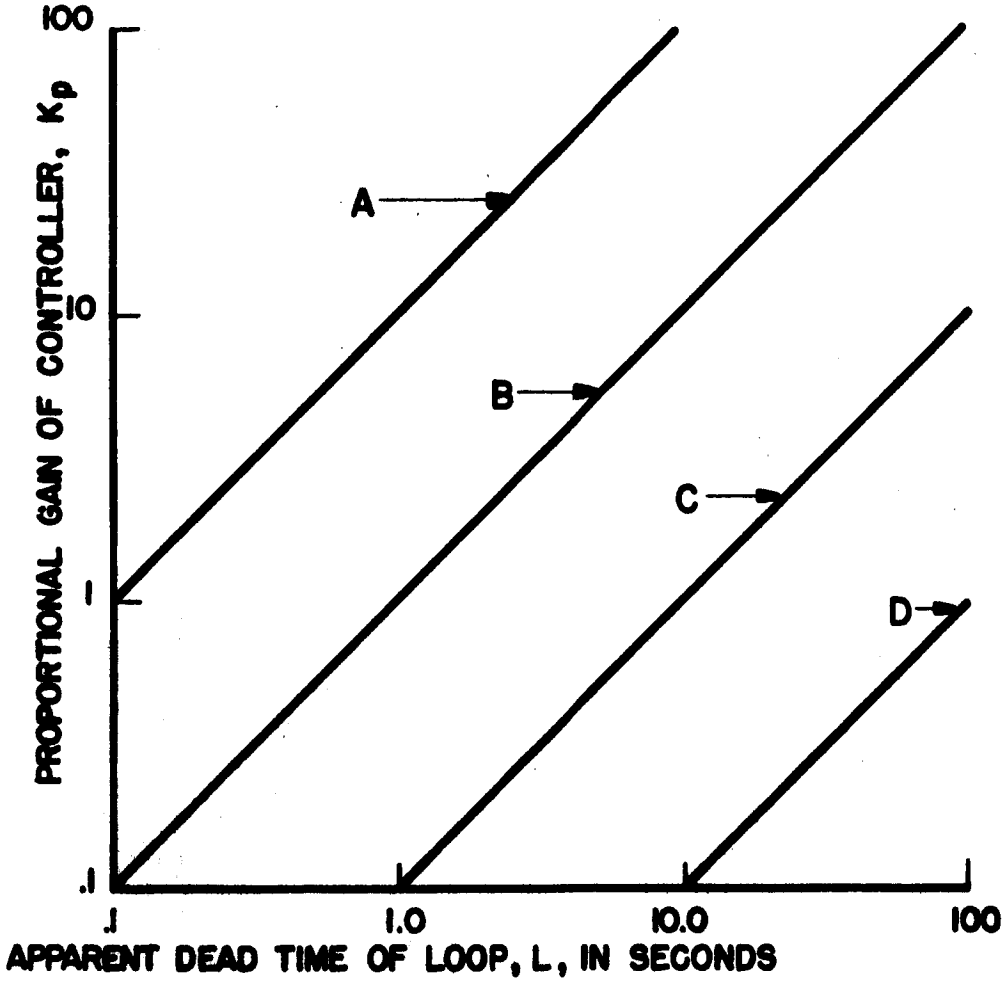


FIGURE 5-1 ACTUATOR SPEED REQUIREMENTS ESTABLISHED BY (5-9)

value to the normalized K_p used in deriving (5-9) we must multiply K_p by the number of quanta that represent full range of b. Therefore

$$K_p = \frac{3}{10^3 q} \times 2^{10} q = 3.1 \qquad (5\text{-}10)$$

Also from Appendix A

$$L = 1.4 \text{ minutes} = 84 \text{ seconds} \qquad (5\text{-}11)$$

Substituting (5-10) and (5-11) into (5-9) :

$$T_{fs} \leq 41 \text{ seconds} \qquad (5\text{-}12)$$

The valve used had a T_{fs} of 33 seconds.

It is possible to eliminate controller gain K_p from (5-9), if desired, by using the Cohen and Coon equation (A-1). The resulting equation is

$$T_{fs} \leq \frac{1.5\, LK_1}{.082 + .9\frac{T}{L}} \qquad (5\text{-}13)$$

where K_1 is process gain, or loop gain divided by K_p, and T is the characteristic Ziegler-Nichols time constant defined in Appendix A.

It is interesting to note that by our very definition of process control in Chapter II the first difference approximation is reasonably valid. If the process is so fast in response that the actuator must run at nearly full speed to control the process, then the control problem should really be classified as a servomechanism problem, and the PI controller described here should not be used. In this case, or if feedback data is noisy, an extra memory register can be added to the controller to restore performance to normal. This technique is mentioned further in Appendix B. If such a step is not taken, and process dynamics are negligible, the controller is effective only in the integral mode.

Separation of Operate and Read Modes

The separation of controller duties into the Operate and Read modes has the effect of opening the control loop while the new set point is being entered. This can have no adverse effect upon control action as long as the time the controller is in the Read mode is small compared to the dominant time constant of the process. The controller should not be left in the Read mode longer than necessary, lest the effect of uncorrected disturbances throw process output too far off.

A second complication results from separation of the two modes. It is closely related to the first difference approximation. When a new set point is entered and the mode selector switch is turned from Read to Operate the computation of A_{kr} takes place immediately and countdown of the resulting difference to the actuator begins. Even though the controller is in the Operate mode no further control action can take place until countdown of A_{kr} is complete. By (4-10) and (4-17) proportional integral samples are inhibited until M_2 is reset at the end of countdown. The fact that no integral samples can be taken during this period does no harm; integral action is intended only for correction of steady state errors, so that overshoot can be reduced if the first integral sample is taken well after the set point is changed. For this reason the integral order flip flop M_4 is set whenever a new set point is entered. The inability of the controller to detect change of process output and to compute A_{kp} during A_{kr} countdown could be detrimental to control. It is necessary to define the circumstances under which this degradation of control performance is small enough to permit this controller to be used.

Figure A.1 is an open loop response of a process to a step change in input. The response of any process can be characterized by the parameters shown on the figure. This process reaction curve is useful in analysis of the problem at hand because just before a change in set point is made the system can be assumed to be at steady state, so that process output immediately following a change in set point will resemble the shape of the beginning of the open loop process reaction curve. The resemblance would be exact if the actuator moved the valve in zero time. The fact that valve motion is a ramp instead of a step means that initial process response will be less rapid than that shown in the process reaction curve. To determine the exact amount of process output change that takes place while the actuator is moving would require a rather involved computation for each different process, much as in Appendix B. We can, however, obtain a good estimate of process output change by using the Ziegler-Nichols process model.

The time required for valve actuation is

$$\tau_{kr} = \frac{\Delta r}{f_a} \qquad (5\text{-}14)$$

$$\Delta m_{kr} = \frac{\Delta r}{f_a T_{fs}} \qquad (5\text{-}15)$$

Using the Ziegler-Nichols process model, change of process output during τ_{kr} is zero if

$$\frac{\Delta r}{f_a} \leq L \qquad (5\text{-}16)$$

For an upper value of Δr we can again take 20 percent of full range, so that using (3-14), (5-16) becomes

$$T_{fs} \leq 5\frac{L}{K_p} \qquad (5\text{-}17)$$

This minimum of time for full stroke of the valve happens to have the same form as (5-9), the minimum established to avoid problems arising from the first difference approximation. Comparison of (5-9) and (5-17) shows that (5-9) demands the higher performance, so that (5-17) does not constitute an additional restriction.

It is useful to consider further the meaning of the limitation imposed by (5-9). The fastest process control loops are fluid flow control loops. The process output, flow rate, follows valve position with no lag. Transducer lag as small as 0.2 seconds may be the only lag in the loop, although if a pneumatic actuator were used it would have a lag of about one second. If we now use an incremental actuator with a time for full stroke of 10 seconds, we introduce a nonlinearity into the analysis because valve actuation time, which varies with size of output order, is now comparable to the other lags in the loop.

This effect is very evident in the shape of a Ziegler-Nichols reaction curve for such a process. The shape of the curve depends very strongly on the amplitude of valve increment imposed. Figure 5.2 shows this variation. To arrive at a value of L for evaluation of (5-9) a small valve increment should be made. The Ziegler-Nichols model is then reasonably valid and a proper value of K_p can be calculated and substituted into (5-9). This will provide a minimum value of T_{fs} because if a larger amplitude reaction curve had been used, a smaller value of K_p would have been required to compensate for the longer apparent lag. If a smaller value of K_p is chosen for a wide range of closed loop stability it would be possible to use a proportionately larger value of minimum T_{fs}. In control of fast processes with no dead time, however, the entire problem may be academic, since adequate control is possible using only integral action.[14] A much simplified version of the controller proposed could provide integral action without encountering the limitations discussed in this chapter.

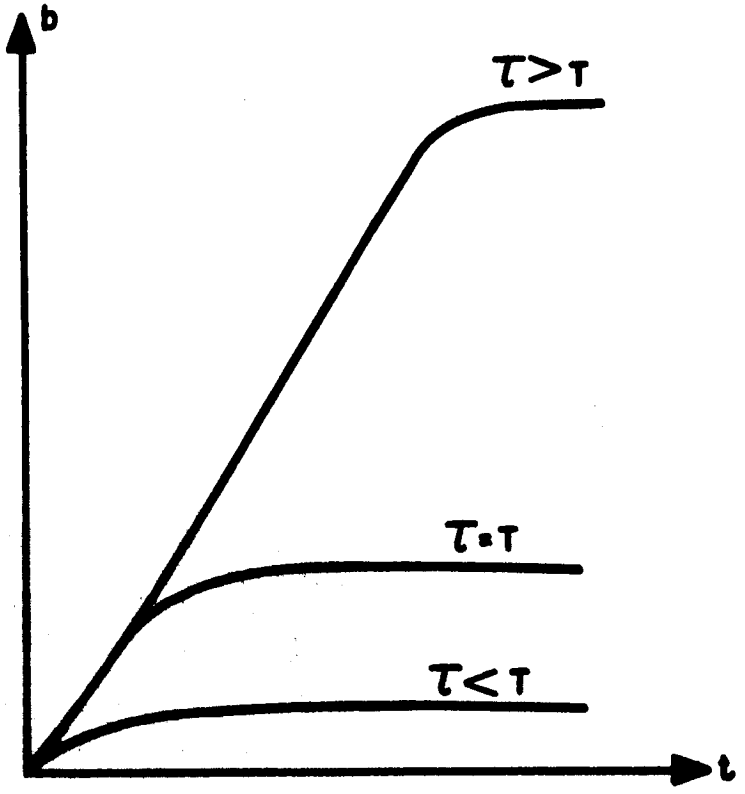


FIGURE 5-2 PROCESS REACTION CURVES FOR VARYING DURATION OF ACTUATOR MOTION τ AND FIXED VALUE OF PROCESS TIME CONSTANT T

Summary

It was shown that two modifications that were made in the controller algorithm in order to reduce memory requirements put an upper limit on time for full stroke of the valve. For process loops with very fast response, such as flow loops, it may not be practical to provide an actuator fast enough to meet this limit. In this case a simplified digital controller providing only integral control can be used, as is often done with analog controllers.[14]

CHAPTER VI

THEORETICAL CLOSED LOOP RESPONSE AND STABILITY

The two modes of control represented by (4-1) and (4-2) call for samples and computations at two different rates, f_p and f_i. The reason for doing this is to reduce the amount of equipment needed to provide a wide range of integral time T_i while retaining the stabilizing effect provided by frequent proportional sampling. While multirate sampling simplifies implementation of the controller, it also complicates the mathematical description of the system. No attempt will be made to derive an exact mathematical model of the control loop using the two sampling rates. Rather, the well known z transform method for a single sampling rate will be used to put outer bounds on stability limits for the more complicated system, using the process of Figure 2.2 as an example.

Using the nomenclature of Figure 6.1 and the two mode control algorithm (3-7) :

$$G_1(z) = \frac{K_p \left[z(1+\tau/T_i) - 1 \right]}{z - 1} \qquad (6\text{-}1)$$

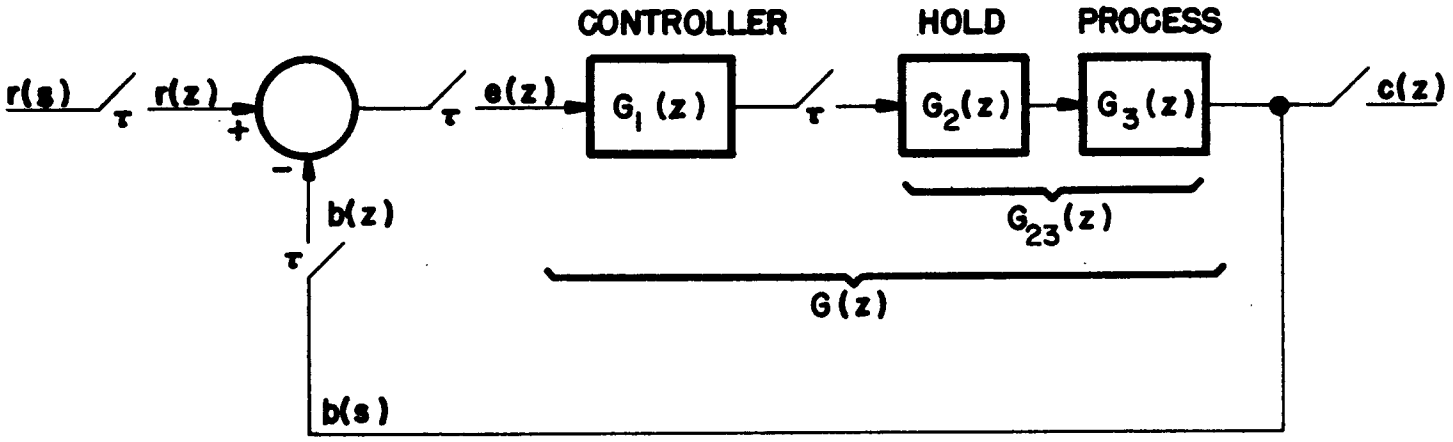


FIGURE 6-1 SAMPLED DATA SYSTEM WITH SAMPLING INTERVAL τ

The product $K_v R K_f$ of Figure 3.2 was found to be

$2.1 \times 10^3 \frac{\text{quanta}}{\text{full stroke}}$.

Hence :

$$G_2(s)\, G_3(s) = \frac{1-e^{-s\tau}}{s} \times \frac{2.1 \times 10^3}{T^2 (s + \frac{1}{T})^2} \qquad (6\text{-}2)$$

For the process used, T = 5 minutes. Therefore

$$G_2(s)G_3(s) = \frac{2.1 \times 10^3 (1-e^{-s\tau})}{25s\,(s+.2)^2} \qquad (6\text{-}3)$$

From Reference (19) :

$$G_{23}(z) = 2.1 \times 10^3 \left[1-z^{-1}\right]\left[\frac{z}{z-1} - \frac{z}{z-e^{-2\tau}} - \frac{.2\tau e^{-.2\tau} z}{(z-e^{-2\tau})^2}\right] \qquad (6\text{-}4)$$

$$G(z) = G_1(z)\, G_{23}(z) =$$

$$\frac{2.1 \times 10^3 K_p \left[(1+\frac{\tau}{T_i})z-1\right]\left[(z-e^{-.2\tau})^2 - (z-e^{-.2\tau})(z-1) - .2\tau e^{-.2\tau}(z-1)\right]}{(z-1)(z-e^{-.2\tau})^2} \qquad (6\text{-}5)$$

The characteristic equation $F(z) = 1 + G(z) = 0$ is :

$$
\begin{aligned}
F(z) = \quad & z^3 \{1\} \\
& + z^2 \{-1-2e^{-.2\tau}+2100K_p[(1+\tfrac{\tau}{T_i})(-e^{-.2\tau}+1-.2\tau e^{-.2\tau})]\} \\
& + z \{e^{-.4\tau}+2e^{-.2\tau}+2100K_p[(1+\tfrac{\tau}{T_i})(e^{-.4\tau}-e^{-.2\tau} + \\
& \quad .2\tau e^{-.2\tau}) + e^{-.2\tau}(1+.2\tau) - 1]\} \\
& + \{-e^{-.4\tau}-2100K_p[e^{-.4\tau}-e^{-.2\tau}+.2\tau e^{-.2\tau}]\} = 0
\end{aligned}
\qquad (6\text{-}6)
$$

(6-6) is the characteristic equation for the sampled system with both modes sampled at the same rate $1/\tau$, with proportional gain K_p and integral time T_i. We shall now develop stability criteria for the system and evaluate them for several sets of the three parameters involved.

The controlled system is stable only if all roots of the characteristic equation lie within the unit circle of the complex plane. We can follow the method given by Oldenbourg and Sartorius[20] to transform the z-plane into the w-plane in such a way that we map the interior of the unit circle of the z-plane into the left half of the w-plane. This will permit us to use the Hurwitz stability criteria. The simplest suitable transformation is

$$w = \frac{z + 1}{z - 1} \qquad (6\text{-}7)$$

Substituting (6-7) into a cubic characteristic equation of form

$$a_3 z^3 + a_2 z^2 + a_1 z + a_0 = 0 \qquad (6\text{-}8)$$

and forming the Hurwitz stability criteria, we find that the stability conditions are expressed by the following relations between the coefficients of the original cubic:

$$\left.\begin{aligned} &3(a_3 - a_0) + a_2 - a_1 > 0 \\ &3(a_3 + a_0) - a_2 - a_1 > 0 \\ &a_3 - a_2 + a_1 - a_0 > 0 \\ &a_3^2 - a_0^2 + a_2 a_0 - a_3 a_1 > 0 \end{aligned}\right\} \qquad (6\text{-}9)$$

Various values of K_p, T_i and τ are substituted into (6-6) and (6-9) to establish stability boundaries. The results are shown in Table 6.1, together with results obtained for the case of continuous control. Open loop gain G_0 is also shown in the table, where

$$G_0 = K_p K_v R K_f = 2100 K_p \frac{q}{fs} .$$

τ, minutes	K_p	T_i, minutes	G_0
0	.0155	2.39	32.5
0	.0021	2.00	4.5
0	.00055	1.30	1.15
0.31	unstable	5.0	-
0.5	.0137	8.0	28.8
1.0	.0066	8.0	13.9

Table 6.1 Conditions for Marginal Stability

When $\tau = 0.31$ and $T_i = 5$ minutes, no positive value of K_p satisfies (6-9). Using continuous control with $T_i = 5$ minutes stable performance is possible even with very high values of K_p. This demonstrates the effect a slow sampling rate can have on stability, even though the ratio of dominant time constant of the loop to sampling interval is over 10 : 1. Using a T_i of 8 minutes the analysis shows that quite high values of K_p are possible even with sampling intervals as high as one minute.

The model used for the stability analysis above does not correspond to the design of Chapter IV. The proportional sampling interval used in control of the two capacity system was from one to ten seconds. The integral sampling interval

was much longer, usually about 7 minutes. The correlation between the multirate sampling system used and the single rate sampling system analyzed theoretically will be discussed in Chapter VII. The theoretical results for the single rate system can apply directly only when the controller is modified as explained in Chapter IV to provide a smoothed integral effect, and when the smoothing is sufficient to make f_p the same as f_i.

CHAPTER VII

EXPERIMENTAL RESULTS

The controller described in Chapter IV was built and used to control the process of Figure 2.2. For tests that required fast process reaction rate the first tank was bypassed. Presentation of experimental results is divided into two parts; the first concerns the observed effects of the two approximations discussed in Chapter V, and the second is on the effect of multirate sampling.

Observed Effects of Algorithm Approximations

The first difference approximation and the separation of Operate and Read modes had no observable adverse effect upon control of the two capacity process with a 33 second actuator. Figure 7.1 is typical of change of set point response curves taken with the two capacity system. The lower curve represents valve position m, and the upper curve is the corresponding process output, b.

The 10 percent change of set point order took place at time zero, with the system in steady state. With the controller

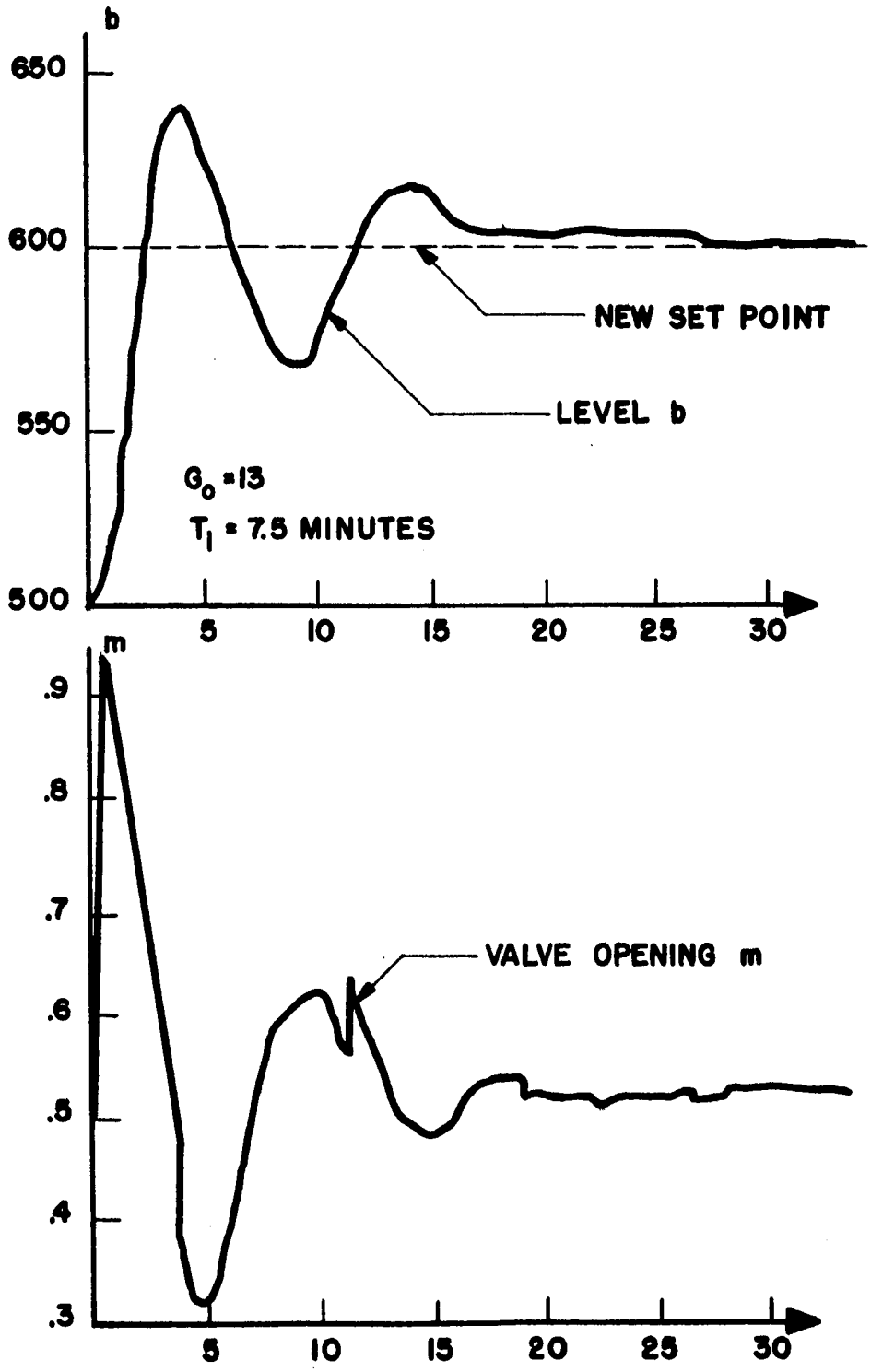


FIGURE 7-1 RESPONSE OF SYSTEM TO A CHANGE OF SET POINT OF 100 QUANTA

gain set as shown on the figure, the initial motion of the actuator required 15 seconds. In that time virtually no change in process output could be observed. Thus, for such a system, the separation of Read and Operate modes is fully justified. The time intervals required for subsequent proportional corrections to m were so short that no change in the binary value of b could be observed during countdowns. During integral countdowns b did not change more than one quantum. This low loss of feedback information had been anticipated because the parameters of the control loop used satisfied the requirement for actuator speed imposed by (5-9).

The test process was modified by bypassing the first tank and slowing the actuator speed down to give a T_{fs} of 47 seconds. The resulting process reaction rate curve was much like that of Figure 5.2 for $\tau < T$. The allowable gain and reset rate $(1/T_i)$ for such a process were very high, and apparent dead time for small Δm was nearly zero. As a result, (5-9) was not satisfied. Test behavior was as expected, in that system stability was much reduced for large values of error and rate of change of error. When the initial large amplitude oscillations did finally become smaller, the effective damping increased, showing very stable and satisfactory performance for small disturbances near the set point. This performance can be completely

explained as effects of the two approximations discussed in Chapter V. As shown in Figure 7.2, the initial change of set point order saturated the valve. Because of the high gain used, the countdown for that order did not end until well after saturation occurred, and no stabilizing proportional order could be executed during that interval. It is also evident that the first difference approximation was scarcely approximate during the first part of the transient, since such a large amount of time was devoted to moving the actuator, during which all change in b went unnoticed by the controller.

Figure 7.2 could also represent a flow control loop, where m is flow rate proportional to valve position and b is the measured value of m from a slow transducer. It is evident that this controller should not be used on a flow loop because the use of the actuator as a summer produces unnecessary fluctuations in flow rate. This effect is, however, much reduced when integral effect is smoothed using an expression like (4-26), as was done in Figure 7.2, and is completely eliminated if the controller is modified to provide only integral control, as recommended in Chapter V.

The instability evident in Figure 7.2 can be much reduced, without eliminating the proportional effect, if changes in

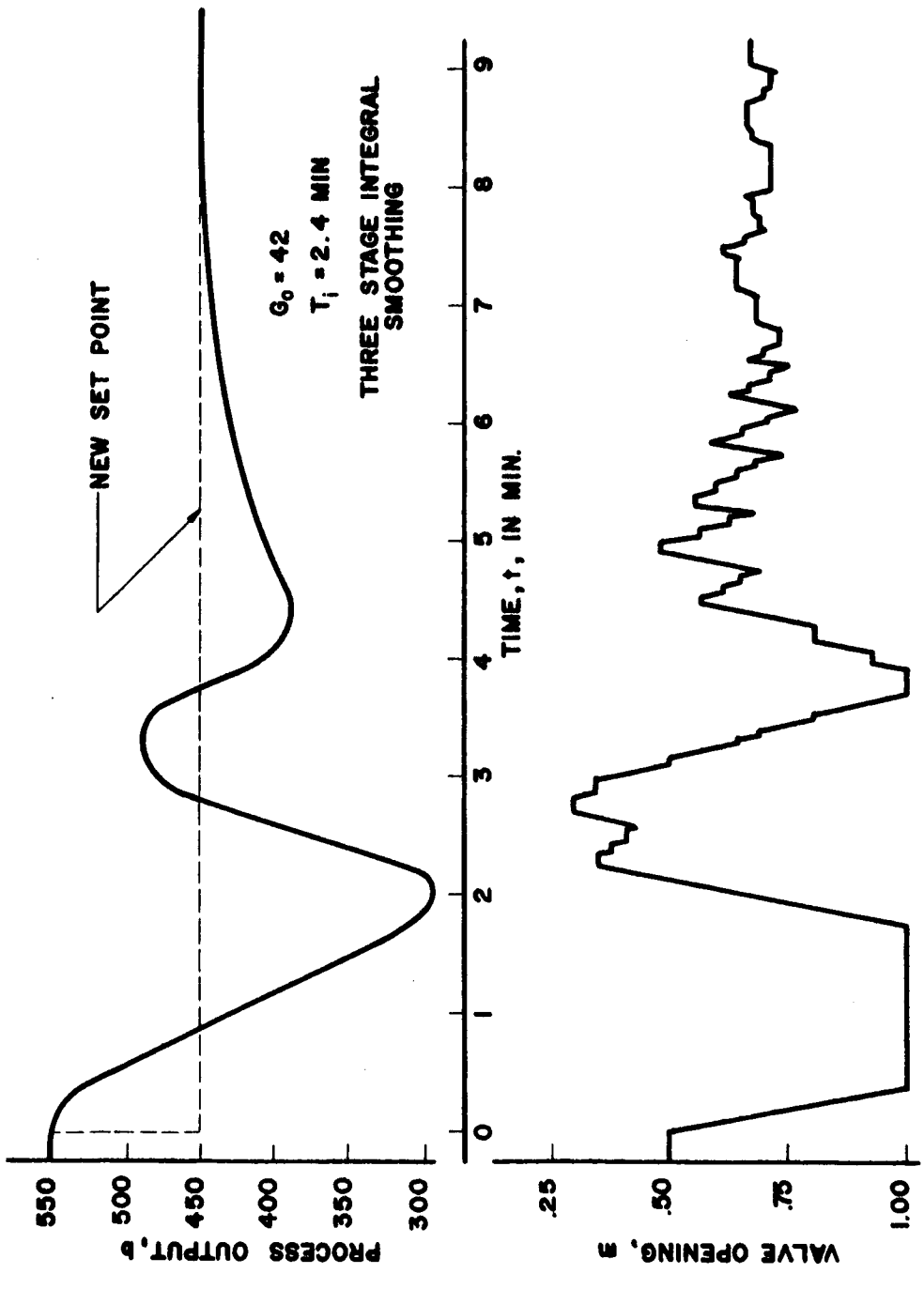


FIGURE 7-2 RESPONSE OF SINGLE CAPACITY SYSTEM TO A 100 QUANTA CHANGE OF SET POINT

set point are performed without using the Read mode; that is, without performing the proportional computation (4-3) for A_{kr}. Figure 7.3 demonstrates this technique. The new set point is attained in much less time than in Figure 7.2, and with no overshoot.

Observed Effects of Multirate Sampling

Figure 7.1 is an example of a successful run using the multirate sampling scheme originally proposed. Proportional gain used is about the same as that recommended in Appendix A, but it was found necessary to increase integral time T_i to about double the Ziegler-Nichols value in order to ensure good stability. The reason this reduction in integral effect was necessary becomes evident from inspection of Figure 7.4, for which integral time was only 30 percent greater than the Ziegler-Nichols value. The controller settings used here were such as to create a period of oscillation T_n approximately double the integral time. It was therefore possible for the integral samples to be taken on successive maximum and minimum points of the oscillation, providing sufficient over-correction to the process to sustain the oscillation. This problem was avoided in Figure 7.1 by increasing T_i. The instability problem can also be avoided by reducing T_i, if K_p is sufficiently reduced at the same time. The main point is to avoid having $T_i = 1/2\ T_n$.

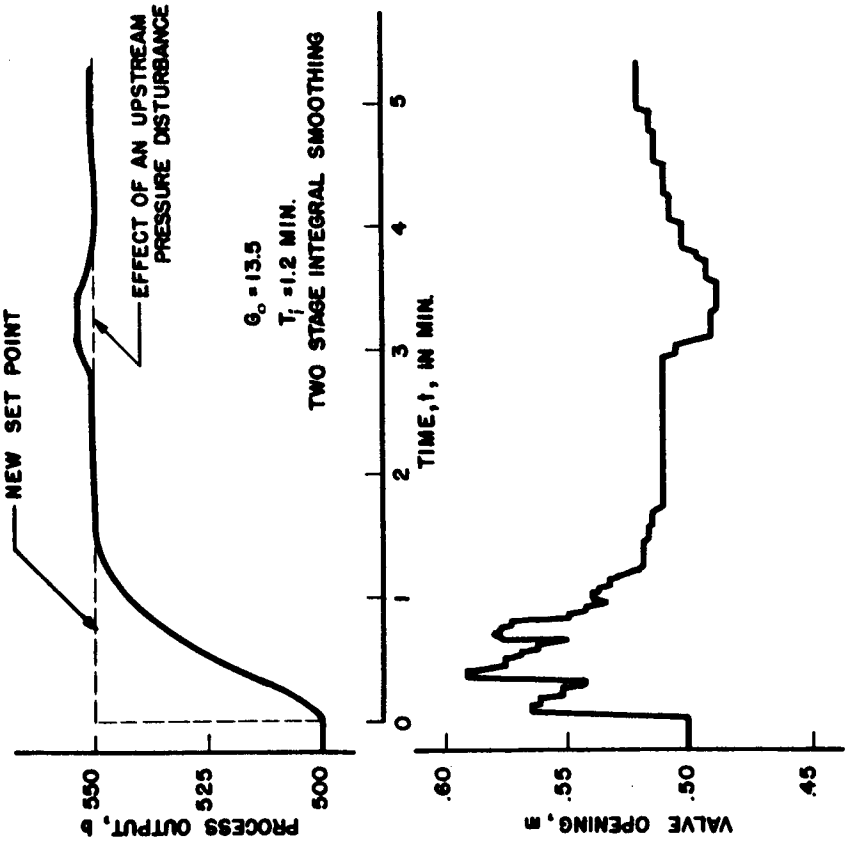


FIGURE 7-3 RESPONSE OF SINGLE CAPACITY SYSTEM TO 50 QUANTA CHANGE OF SET POINT WITHOUT USE OF READ MODE

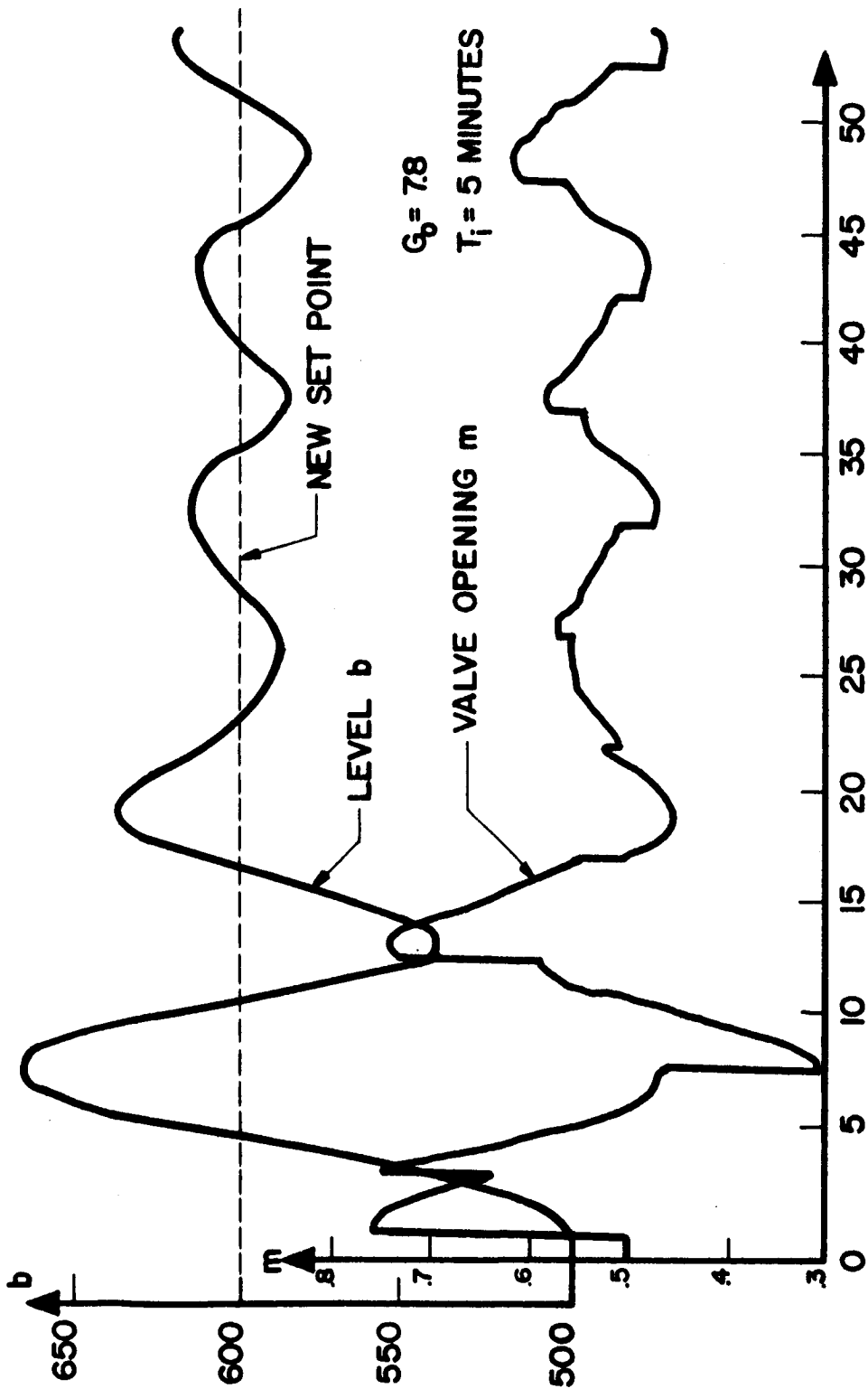


FIGURE 7-4 RESPONSE OF SYSTEM TO A 100 QUANTA STEP CHANGE OF SET POINT

The rules for controller tuning again prove useful here. Reference (18) states that a control loop with a properly tuned PI controller has a natural period of oscillation of about 5.7L, where L is the apparent dead time of the loop. Substituting this fact into (A-2) gives:

$$T_i = \frac{T_n (30+3R)}{5.7(9+20R)} \qquad (7\text{-}1)$$

where R is not valve resistance, as before, but the ratio L/T of Appendix A. For R large, $T_i << T_n$ and there is no danger of the instability found with $T_i = 1/2\ T_n$. For the two capacity process R = .144 and $T_i = 1/2\ T_n$ and we have poor stability. For R = 0, T_i is still close enough to $1/2\ T_n$ to be of concern.

While the solution of increasing T_i above that dictated by tuning equations was successful, it also caused weaker control action than that available with analog controllers. A better solution was to use the integral smoothing technique described at the end of Chapter IV. According to (7-1) the largest possible ratio of T_i to T_n would be 0.58. To make this ratio always less than 1/2 would require only one stage of smoothing, that is a gain reduction on each integral sample of 1/2. But it is best to use three stages of smoothing to ensure that integral samples are not taken every $1/2\ T_n$ even in loops where K_p and T_i are both

set above optimum. Figure 7.5 shows response of the two capacity system to a change in set point using three stages of integral smoothing. Performance is considerably better than Figure 7.4, though still lightly damped. It was noted that when four stages of smoothing were used the integral correction order τ_{ki} for a one quantum error called for less than one step of the actuator, and hence produced no correction at all.

As long as integral sampling intervals did not occur just twice per natural period, the stability of the control loop seemed uneffected by slow integral sampling. This is demonstrated by Figure 7.6 for which T_i was 1.3 minutes, unsmoothed, and K_p was .00049. Its response is just barely stable. Table 6.1 shows that the stability boundary for the continuous system lies at T_i = 1.3 minutes and K_p = .00055. Thus the multirate controller, with proportional sampling interval of 5 seconds and integral sampling interval of 1.3 minutes, yielded about the same performance as what was predicted for a continuous PI controller with the same gains.

Load change tests were also made, but are not presented here because they demonstrated nothing more than was shown by set point change runs.

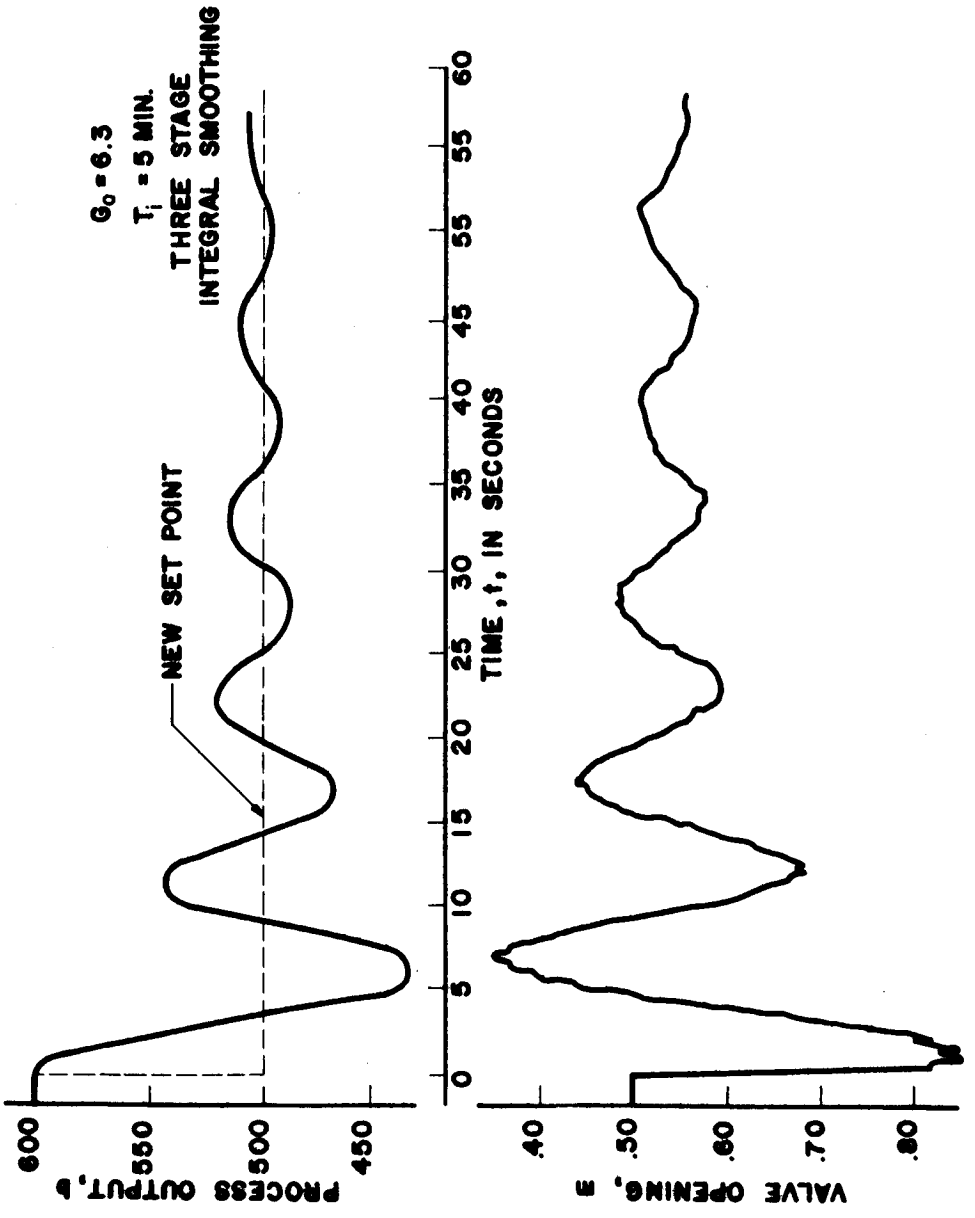


FIGURE 7.5 RESPONSE OF TWO CAPACITY SYSTEM TO 100 QUANTA CHANGE OF SET POINT

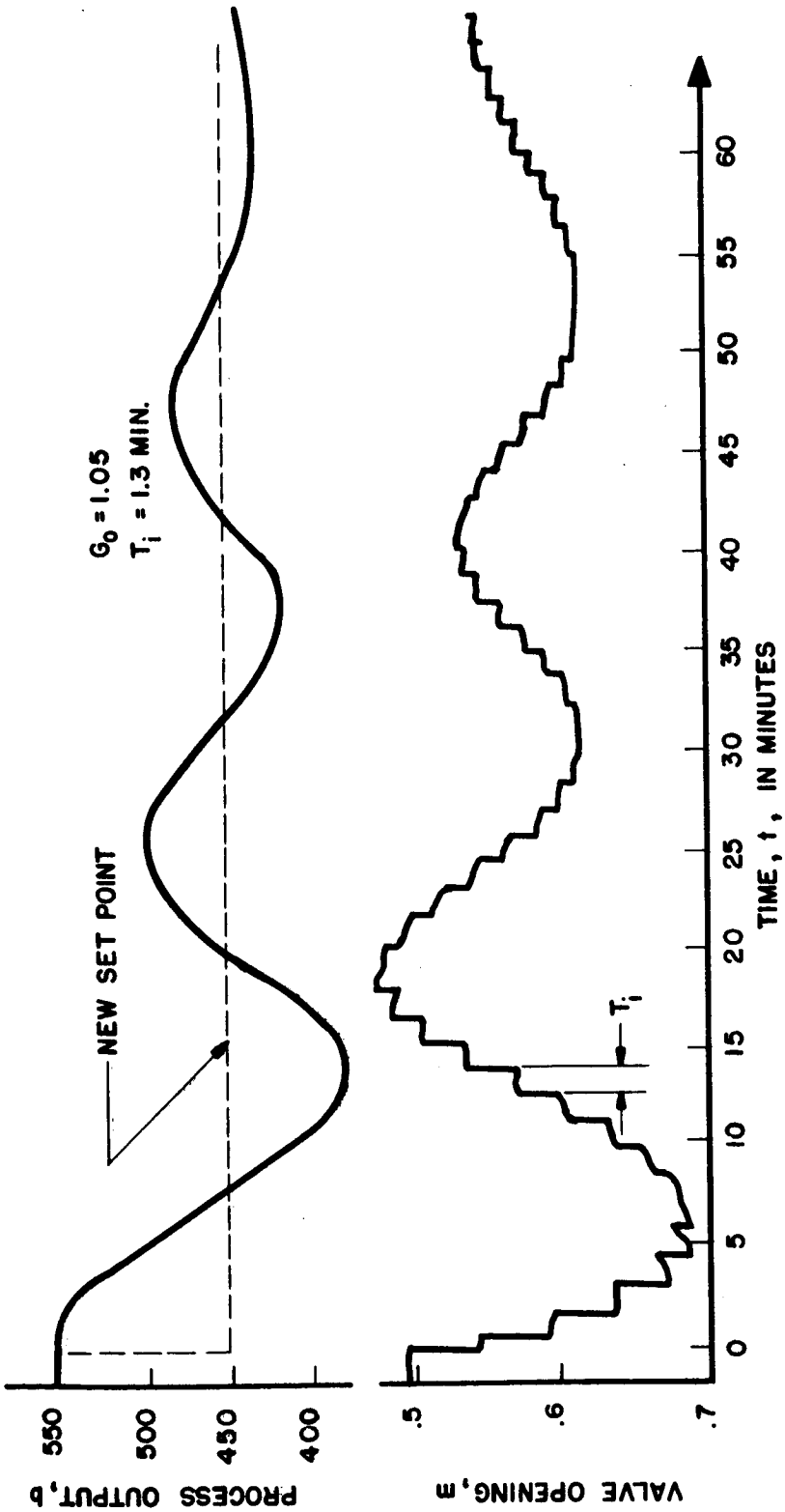


FIGURE 7-6 RESPONSE OF TWO CAPACITY SYSTEM TO A 100 QUANTA CHANGE OF SET POINT

Summary

The simplifying approximations made in the original design of the controller led to instability in certain cases. To restore a guarantee of stable performance it was necessary to smooth the integral effect by sampling that term eight times per T_i instead of just once. To avoid instability for large errors for certain fast responding processes it was found necessary to either add another memory register to the controller, omit use of the Read mode, or leave off the proportional effect for those cases. The latter step is a common procedure with analog flow controllers, and has little adverse effect on control performance.

CHAPTER VIII

CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER WORK

Conclusions

The features of the controller described in this paper that constitute new contributions to the field are the decomposition of the control algorithm into parts that can be computed and applied serially, and the variation of the sampling rate for the integral term, independent of the proportional term sampling rate, to effect adjustment of integral time. The effectiveness of this approach has been demonstrated, subject to certain limitations that are a function of the reaction rate of the process being controlled and of the speed of the actuator used. The rules for application of the controller can be stated as follows:

The original PI controller of Figure 4.2 can only be used with gains set somewhat below the Ziegler-Nichols optimum settings. For best performance the integral smoothing circuitry of Figure 4.4 should be added, requiring no more than 11 extra transistors for three stages of smoothing.

With this addition and with a suitable incremental actuator the controller can be tuned to provide virtually the same performance as an ideal continuous PI controller. To maintain stable performance for large set point changes, however, it is necessary to check that actuator speed is fast compared to process reaction rate, specifically that $T_{fs} \leq 1.5\ L/K_p$, where T_{fs} = time for full stroke of the valve, L is apparent loop dead time, and K_p is controller gain normalized for unity feedback. If the actuator provided is not capable of this speed, one of two steps must be taken: either the proportional mode must be left off the controller entirely or else the part of the proportional mode used to initiate the set point change must be omitted, permitting the change to be made by the more gradual integral action. If noise on the feedback signal is a serious problem it is necessary to add an extra 10 bit memory register to the controller.

Recommendations for Further Work

It should prove possible to add the derivative mode to the controller, using the same frequency technique for gain adjustment as was used for the integral term. Consider the PID incremental algorithm (3-5) :

$$\Delta m_k = K_p\left[\frac{\Delta t}{T_i}\, e_k + (e_k - e_{k-1}) + \frac{T_d}{\Delta t}\,(e_k - 2e_{k-1} + e_{k-2})\right] \quad (8\text{-}1)$$

Using (3-9) for e_k:

$$\Delta m_k = K_p\left[\frac{\Delta t}{T_i}\ (r_k - b_k) + (r_k - r_{k-1}) + (b_{k-1} - b_k) + \frac{T_d}{\Delta t}(r_k - 2r_{k-1} + r_{k-2}) - \frac{T_d}{\Delta t}(b_k - 2b_{k-1} + b_{k-2})\right] \quad (8\text{-}2)$$

If we can do without the derivative effect for initiation of changes of set point, with the knowledge that such events seldom occur, we can drop the first of the two derivative terms of (8-2). Using (3-14) and the decision to use three stages of integral smoothing:

$$\Delta m_k = \frac{1}{f_a T_{fs}}\left[\frac{1}{8}(r_k - b_k)_{f_i} + (r_k - r_{k-1})_{\Delta r} + (b_{k-1} - b_k)_{f_p} - N(b_k - 2b_{k-1} + b_{k-2})_{f_d}\right] \quad (8\text{-}3)$$

where $f_i = \frac{8}{T_i}$ and $f_d = \frac{N}{T_d}$. The subscript Δr again identifies the term that is computed only when r is changed. N is an integer. f_p is the rate at which the proportional term $(b_{k-1} - b_k)$ is computed. For the PI controller the setting of f_p has no great importance, since it has no effect on gain. It need only be kept above a certain minimum rate in order to avoid excessive sampling phase lag. A value of $f_p = 50/T_n$ was found to be adequate, where T_n is the period of oscillation of the closed control loop. Values down

to $20/T_n$ might serve as well, but were not tried. As long as this flexibility in f_p is available it would seem possible to use this sampling rate also for the derivative rate f_d to permit certain economies in the logical implementation of (8-3). N could be selected to make f_d always well over $20/T_n$, then f_d could be adjusted for fine adjustment of T_d. From Reference (21), the optimum ratio of T_i to T_d should lie between 6 and 1/2. Then f_d must lie between 6 and 1/2 times N/T_i. From Reference (18), $T_i = .4T_n$. Therefore, f_d must lie between 6 and 1/2 times $2.5M/T_n$. Then to ensure an adequate proportional sampling rate

$$f_p = f_d \geq \frac{20}{T_n} \qquad (8\text{-}4)$$

Hence

$$\frac{2.5N}{2T_n} \geq \frac{20}{T_n} \quad \text{and } N \geq 16 \qquad (8\text{-}5)$$

For N = 16, f_d will range from $20/T_n$ to $240/T_n$. (8-3) can be rewritten as:

$$\Delta m_k = \frac{1}{f_a T_{fs}} \left[\frac{1}{8} (r_k - b_k)_{f_i} + (r_k - r_{k-1})_{\Delta r} + (b_{k-1} - b_k)_{f_p} \right.$$

$$\left. -16 (b_{k-2} - b_{k-1} - b_{k-1} + b_k)_{f_d} \right] \qquad (8\text{-}6)$$

It appears that two extra memory registers beyond the one in the PI controller would be required to implement (8-6). A total of 60 transistors in addition to the 109 used in the modified PI controller should be sufficient to provide the PID algorithm of (8-6).

Oldenbourg and Sartorius[20] stated that stability of loops containing true transportation lags, or dead times, could be improved by sampling and computing corrections at a maximum rate of once per dead time. Having now built a controller whose different modes are sampled at different rates, it is interesting to speculate on whether it would be possible to achieve improved control over loops with dead time by observing the Oldenbourg and Sartorius rule only for the integral effect, while sampling the proportional and derivative terms more often. It might be that only the derivative mode could be sampled more often than once per dead time to provide the best response. In any case, further research might reveal that the multirate sampling scheme has definite advantages in the control of such special kinds of processes.

APPENDIX A

CONTROLLER TUNING BY REACTION RATE

Reference (14) presents the Ziegler-Nichols method for tuning controllers. It is based upon the characterisation of any process by a dead time L and a first order lag T. A step change in valve position is made and the resulting open loop response is plotted. This was done on the two capacity system of Figure 2.2, resulting in the response shown in Figure A.1. The controller countdown rate was 14.3 q/sec. The time for valve full stroke was 33 sec. Therefore by (3-14)

$$K_p = \frac{\text{sec.}}{14.3\ q \times 33\ \text{sec.}} = \frac{2.12}{10^3 q} = .212\ \frac{\text{percent full stroke}}{\text{quantum}}$$

The step input to the controller was 25 q. The valve motion was therefore

$$M = \frac{2.12}{10^3 q} \times 25\ q = .053$$

Five percent of full stroke was considered small enough to preserve linearity.

From Figure A.1 :

L = 1.4 min.

T = 9.7 min.

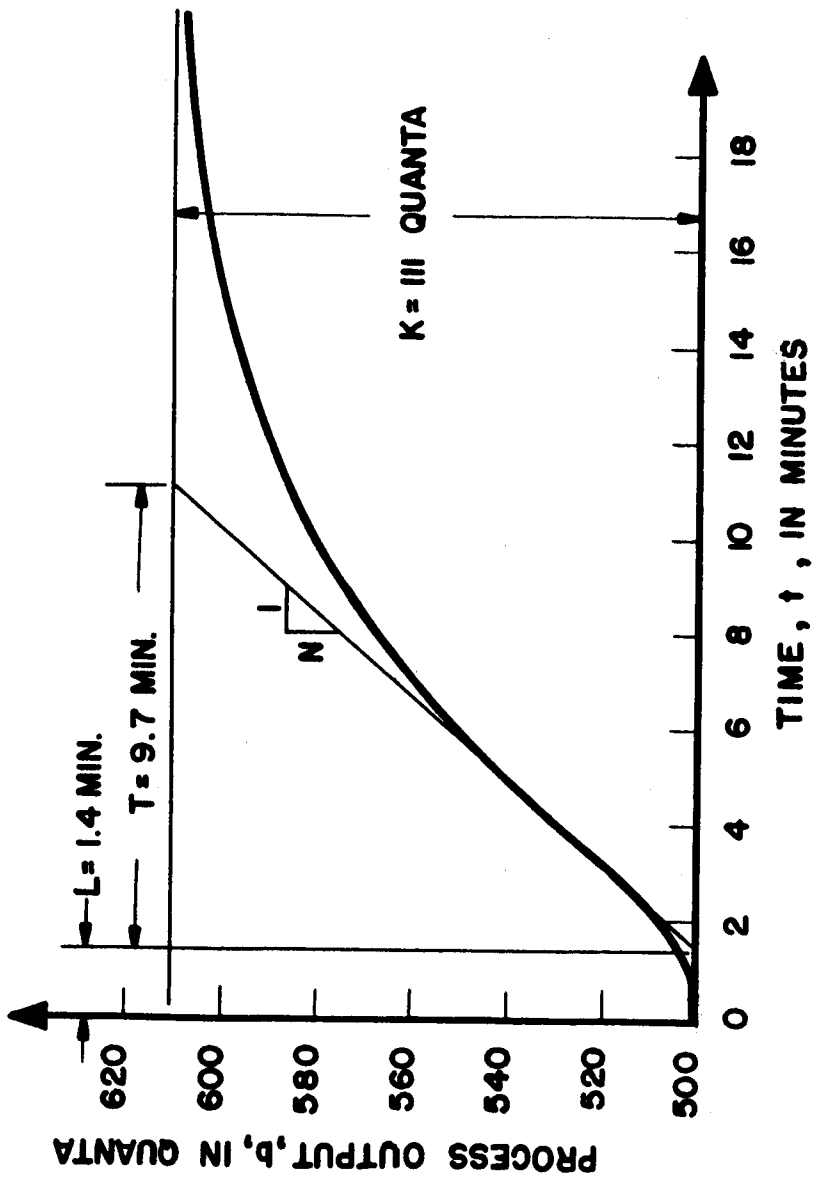


FIGURE A-I OPEN LOOP RESPONSE OF PROCESS TO 5.3% STEP CHANGE IN INPUT

$$K = 111\ q$$

$$N = \frac{K}{T} = \frac{111\ q}{9.7\ \text{min.}} = 11.46 \frac{q}{\text{min.}}$$

$$R = \frac{L}{T} = \frac{1.4\ \text{min.}}{9.7\ \text{min.}} = .144$$

For one-quarter ratio of amplitudes and a compromise between minimum area and period, by this method :

$$K_p = \frac{M}{NL}\left(\frac{9}{10} + \frac{R}{12}\right) = \frac{.053\ \text{min.}}{11.46q \times 1.4\ \text{min}}\left(\frac{9}{10} + \frac{.144}{12}\right)$$

$$= .003\ \frac{(\text{full stroke})}{(\text{quantum})} \qquad (A\text{-}1)$$

$$T_i = L\left(\frac{30+3R}{9+20R}\right) = 1.4\ \text{min}\ \left(\frac{30.43}{11.9}\right) = 3.57\ \text{min.} \qquad (A\text{-}2)$$

(A-1) and (A-2) should then give "best" response. To try these values we adjust countdown rate to comply with (A-1). By (3-14) :

$$f_a = \frac{1}{K_p T_{fs}} = \frac{10^3 q}{3.0 \times 47\ \text{sec.}} = 7.1\ q/\text{sec.} \qquad (A\text{-}3)$$

Figure 7.5 is a response curve using the K_p of (A-3) and a T_i of 5 minutes. The 3.57 minutes of (A-2) was found to be too oscillatory.

Three stage integral smoothing was used for this run.

APPENDIX B

EFFECT OF NOISE ON PROCESS OUTPUT

The float that measured water level in the process of Figure 2.2 was geared to 10 bit binary encoder with U-brush scan to provide a quantum size of .02 inch. Wave motion, gear backlash and static friction were at a low enough level that noise in the binary feedback number b was not a problem. A steady increase in water level would produce a monotonic increase in b. Hysteresis of the float-encoder combination amounted to somewhat less than one quantum. Such an accurate and noise-free measurement can not, however, always be provided in an industrial environment, especially if the transducer produces an analog voltage that must be sent to a shared analog-to-digital converter. We should, then, investigate what effect a noisy signal at the input to the controller will have on the output of the process.

The integral term of (3-5) is $(r_k - b_k)$. r is a fixed binary number not subject to noise. If b_k is d quanta higher than its true value, the controller will make a false correction which, if the loop were then opened, would lead to a dG_0 steady state error. In closed loop operation this sort of

error can only be corrected after its effect is seen at the process output. This could be a seriously disturbing factor with the controller proposed in Chapter II, but the problem can be greatly reduced by using the integral smoothing technique described later in the paper. Using three stage smoothing integral samples are taken eight times per T_i instead of once, and the gain of the term is one-eighth of its former value. From the Ziegler-Nichols tuning equations we can predict that these integral corrections will occur about five times per L, where L is the apparent loop dead time of Figure A.1. Therefore no single noisy integral sample will have time to effect process output significantly before the next integral sample is taken. The randomness of the noise will then determine the extent to which one noisy sample will correct the effect of previous samples.

Both components of the proportional term $(b_{k-1}-b_k)$ are subject to error arising from noise. Sample number two in Figure B.1 is assumed to be incorrect. m takes the excursion A_{2p} shown at time 2, but returns to its original level at time 3 because the incorrect value b_2 was retained in storage and used in the calculation of A_{3p}. The maximum resulting transient effect of this pulse can be calculated graphically by the method of Taft [16], using the reaction rate curve of Figure A.1. The maximum excursion of c,

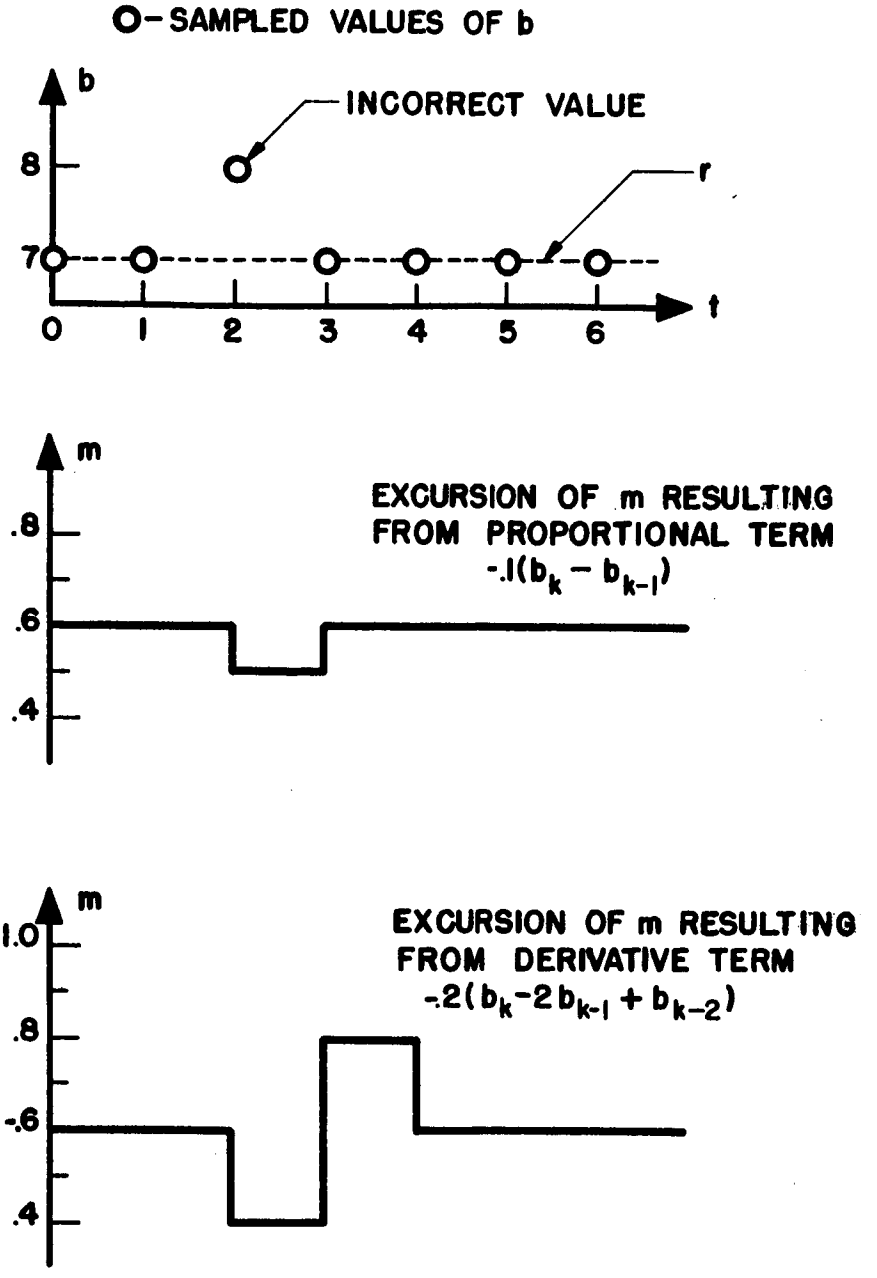


FIGURE B-1 EFFECT OF NOISE ON MANIPULATED VARIABLE

if uncorrected by feedback, would be $\frac{dG_0}{Tf_p G_0}$, where d is the amount of error on the sampled signal, G_0 is open loop gain, T is Ziegler-Nichols time constant and f_p is proportional sampling rate. The true effect would be less because of feedback. The greatest difficulty arises from the fact that in the controller described in this paper the value of b_k used in computation of A_{kp} is not retained for use in computation of $A_{(k+1)p}$. Rather, b is resampled at time $k + \delta$, where δ is some small fraction of a sampling interval, and $b_{k+\delta}$ is retained in place of b_k. If the noise level d changes in time δ, which should be expected, m of Figure B.1 will not return to its proper value at sample 3, and we are once again faced with a permanent error of the type arising from the integral term. If noise does occur in b, this error will be worse the more often the proportional term is sampled. The way to avoid the problem is either to provide temporary storage for b_k in the A/D converter, or add another memory register to the controller.

The effect of a noisy signal at time 2 upon the derivative effect is also shown in Figure B.1. Not only does m return eventually to its proper value, but a negative compensatory square pulse even brings the time integral of m to the value it would have had without the error. The fact that the gain of the derivative term is usually high might mean that this doublet could still cause some deviation in m. Taft's

graphical convolution method will again put an outer bound on error in m, but it cannot be expressed in terms of Ziegler-Nichols parameters as was done for the proportional error. Once again the problem is considerably complicated if the value of b_k used in the calculation for A_{kd} is not retained in its exact form for the calculation of $A_{(k+1)d}$ and $A_{(k+2)d}$. If noise is present in b it is virtually essential that they be retained. If they are retained, some of the fears about the folly of taking a second difference should be dispelled.

APPENDIX C

USE OF CONTROLLER

WITH A HIGHER LEVEL COMPUTER

Any control function beyond the basic PI algorithm must be handled by a separate higher level computer. The plan outlined in Chapter I calls for the use of a single computer to perform the higher level control functions for a large number of single loop digital controllers. From a reliability standpoint it is desirable to be able either to set control parameters into the controller manually, or to put the controller under the automatic supervision of the higher level computer. It is the purpose of this section to determine just what communications between controllers and computer are necessary to achieve this goal.

Set Point :

For independent use each controller has 10 switches for manual setting of the binary set point, r. The controller should also have the capability of accepting a set point from the central computer. This could be provided by 10 levels from a register in the computer. One additional switch is required, either at the computer or the controller, to select which of the two possible set points should be effective.

Controller Gain K_p and Integral Time T_i :

Adjustment of K_p is performed by varying the countdown frequency f_a. A free running multivibrator within the controller provides the necessary pulse train, whose frequency can be continuously varied over a wide range by manual adjustment of a pair of potentiometers. The central computer could also have the capability of controlling f_a by gating one of a variety of clock outputs to the controller. Only a single line is required for this job, in addition to the line for the selector switch mentioned above. The same provisions would be made for integral time, T_i, since it is also adjusted by the frequency of a pulse train.

Proportional Sampling Rate f_p :

This pulse train can also be generated either in the controller or the central computer, but there is less reason to vary it, since for the PI controller its value has no effect on gain. When several controllers receive their digital input data b from a shared A/D converter it would be best to generate f_p from a common source for all the controllers. All loops with similar dynamics could be sampled at the same rate, the sampling instant for one controller being offset from that of the next by the use of a ring counter which would also control the multiplexer of the A/D converter.

Information Required by Central Computer :
When the controllers are using their own set points the central computer needs to receive either the values of error from each controller, produced eight times per T_i, or else the value of r and of b, available each time an A/D conversion is made for each loop. It could be expected that if the central computer is in operation at all, it would be providing set points to the controllers. In that case it would need only values of b for each loop. They could be provided either directly from the A/D converter or else from the A register of each controller, which contains the last value of b at any time a computation or countdown is not in process.

Actuation Orders from Computer to Controllers :
The higher level control functions assigned to the central computer must be programmed in such a way as to produce incremental output orders for the various actuators. When one of these orders is generated within the computer it can be immediately transferred to the A register of the proper controller, and immediately be counted down to the actuator. To do so, however, might mean destroying the value of b in that register just before a proportional sample was about to be made. To avoid undue degradation of the proportional effect, the best course would be to hold the output orders in computer memory until a signal from the controller

notifies the computer that it has just finished a countdown and is ready to accept any other order. Such a signal could be generated by

$$\text{Transfer Order} = (\beta_{M_2})(M_4)(\text{Order Ready}) \qquad (C\text{-}1)$$

In such a case the Enter b order of Figure 4-2 would have to be inhibited, and to start countdown the flip flop M_2 would have to be set as soon as transfer was accomplished. At the end of countdown b would be resampled and normal PI operation would continue until the next order was ready in the computer.

LIST OF REFERENCES

1. Mergler, H.W. "A Digital-Analog Machine Tool Control System", Proceedings of the Western Joint Computer Conference, American Institute of Electrical Engineers, 1954.

2. Schall, W.C., "Direct Digital Control - The First Two Years", Proceedings of the Fifth National Chemical and Petroleum Instrumentation Symposium of May 4-5, 1964, Plenum Press, New York, 1964, pp. 1 - 6.

3. Kirchmayer, L.K., "Lowering Generating Costs with Computing Controllers", Control Engineering, December, 1958, p. 85.

4. Eckman, D.P., Bublitz, A., and Holben, E., "A Satellite Computer for Process Control", Instrument Society of America Journal, November, 1962, pp. 57 - 64.

5. Guisti, A.L., Otto, R.E., and Williams, T.J., "Direct Digital Computer Control", Control Engineering, June, 1962, pp. 104 - 108.

6. Yetter, E.W. and Sanders, C.W., "A Time-Shared Digital Process-Control System", Instrument Society of America Journal, November, 1962, pp. 53 - 56.

7. Instrument Society of America Journal, November, 1962, p. 13.

8. Krug, E.K., "The Dynamic Properties and Setting of Systems With Discrete Controllers", Automation and Remote Control, Vol. 23, No. 4, 1962, pp. 436 - 448.

9. Diligenski, S.N., "Certain Block Diagrams and Dynamic Characteristics for Digital Controllers", Automation and Remote Control, Vol. 23, No. 11, 1962, pp. 1364 - 1375.

10. Aleksandridi, R.M., "Certain Problems in Selecting the Structure of a Multichannel Digital Controller", Automation and Remote Control, Vol. 24, No. 2, 1963, pp. 191 - 200.

11. Aleksandridi, Diligensky and Krug, "Digital Controllers", Paper No. 519 of 1963 Conference of I.F.A.C. in Basel, Switzerland.

12. Walker, J.F., "A Study of Logical Synthesis Techniques in Digital Compensation", PhD Thesis, Case Institute of Technology, 1965.

13. Peatman, J.B., "A Digital Controller Employing Truncated Logarithmic Quantization", PhD Thesis, Case Institute of Technology, 1965.

14. Eckman, D.P., Automatic Process Control, John Wiley & Sons, New York, 1958.

15. Richards, R.K., Arithmetic Operations in Digital Computers, D. VanNostrand, New York, 1958.

16. Mergler, H.W., "Notes on Digital Control Systems Engineering", Case Institute of Technology.

17. Arnstein, W., "Analysis and Synthesis of Transition-Coupled Asynchronous Counters", PhD Thesis, Case Institute of Technology, 1964.

18. Ziegler, J.G., and Nichols, N.B., "Optimum Settings for Automatic Controllers", Transactions ASME, November, 1942.

19. Tou, J.T., Digital and Sampled-Data Control Systems, McGraw-Hill, New York, 1959.

20. Oldenbourg, R.C. and Sartorius, H., The Dynamics of Automatic Control, ASME, 1948.

21. Volgin, V.V., "Determination of the Optimum Adjustment of PID Regulators", Automation and Remote Control, Vol. 23, No. 5, 1962.