

T. J. Cameron
M. H. Dost

Paper Servo Design for a High Speed Printer Using Simulation

Abstract: The design of feedback systems for control of paper motion through the IBM Printing 3800 Subsystem is described. The main tool for analysis and optimization is DSL, a digital simulation language for systems of continuous nature. The several stages in the evolution of the design are discussed.

Introduction

Maintaining accurate position control, while paper moves at 80 cm/s, and providing rapid start/stop motion without tearing the fanfolded paper was a significant design problem for the paper transport system of the IBM 3800 printer, the mechanical features of which are described in [1]. The objectives were 1) to synchronize at the transfer station predetermined paper positions with the photoconductor positions, so that transfer of the toned image can occur without blurring; 2) to regulate paper speed through the fuser and assure stopping of paper between the hot rolls only at perforations, where no print is located, and 3) to control the amount of paper slack between transfer station and fuser.

Specifications for this printer were so different from those in previous IBM line printers that new concepts had to be developed to meet the requirements accurately, reliably, and with reasonable cost. Initial attempts to use conventional mechanical schemes were soon abandoned in favor of servomechanisms.

In addition to the technical challenges in the design of these systems, logistics and schedule problems were difficult for this new printer. All other electromechanical systems, as well as the electrophotographic process and the optical systems, were equally new and uncertain, requiring much experimentation and testing with laboratory models as well as complete prototype machines. It soon became clear that dependence on the prototype machine had to be minimized in favor of analytic design methods, and computer simulation proved to be the answer.

These complex electromechanical systems were partly continuous, partly discrete in nature. Their transient behavior as well as steady state stability had to be well understood, and computational flexibility had to be guaranteed to allow exploration of many design alternatives. Fortunately, a digital simulation language was available that could solve large sets of coupled nonlinear differential equations with arbitrarily varying coefficients. This computer language, together with a graph generating program, enabled us to model all our problems as accurately as desired in time, frequency or other domains, and to display the results graphically for easy interpretation.

After a brief characterization of the simulation language, this paper shows how the design of the subject control systems evolved through several cycles of concept formulation, mathematical modeling, simulation and validation, iteration about the critical parameters, physical buildup, and testing.

On simulation languages

The DSL language referred to here is a digital language for the simulation of continuous systems (a CSSL [2]), rather than discrete systems. It is particularly suited for the study of physical phenomena as expressed by ordinary differential equations or modeled through transfer functions. Nonlinearities, coefficient variations in time, and changes in the model structure are readily accommodated, and there is almost no limit to complexity or the number of inputs to a system.

Copyright 1978 by International Business Machines Corporation. Copying is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract may be used without further permission in computer-based and other information-service systems. Permission to *republish* other excerpts should be obtained from the Editor.

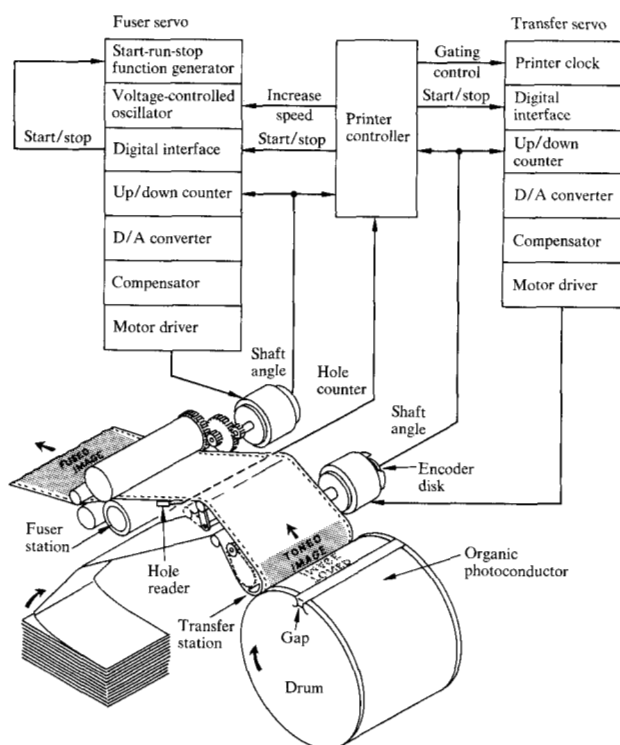


Figure 1 The paper path and some of the main components of the paper control systems.

A high-level language like DSL is easy to learn and use. Like CSMP [3, 4] which evolved from an early version of DSL, this language is largely nonprocedural, i.e., the sequence of program statements is not critical. This fact and the Englishlike syntax minimize communication problems between the person who poses the problem and the computer. Simple statements, such as

```
SPEED = INTGRL (V0, ACCEL)
ACCEL = FORCE/MASS
IF (SPEED. GT. TOP) CALL ENDRUN
```

allow compact programs to be written in a short time and to be understood by others, at least in conjunction with block diagrams that show the functional interrelation of system variables.

While digital simulation is in many ways similar to analog simulation, which is used widely in the process, power and aerospace industries, there are unique differences. The independent variable may be called TIME, but it is inherently discrete, unlike the real time in nature, requiring some caution in numerically critical problems. The problems of noise, drift, and resolution associated with analog electronics and programming problems, such as scaling and sign inversion in analog equipment, do not exist in digital simulation and the computer can be time shared with many other types of applications.

One disadvantage of early digital simulation languages, namely the slow turnaround time of batch mode computation, has been overcome with the advent of interactive computing systems and visual display consoles. Another handicap of those simulation languages, which possess only printed or print-plotted output capabilities, has been eliminated in DSL: a graph control language called GRAFAEL [5] was incorporated, enabling the user to obtain high resolution plots of flexible format on a variety of plotting devices with little programming effort.

Paper motion control systems

Of many automatically controlled operations in the printer, the paper motion variables were among the more demanding. Rapid accelerations occur during normal operation and absolute position control is necessary for precise print registration. After image transfer, the fuser servo has to maintain smooth paper motion to avoid separation of toner particles from the paper. This is accomplished by closely matching the start and stop motion of the two servos. No relative velocity error can be allowed in steady state.

Figure 1 shows the paper path and the primary components of the paper motion control systems. The motion is directed by a supervisory controller or control computer, which coordinates the actions of these and other control functions in the printer. The transfer and fuser servos respond to commands to start, stop, and increase speed, and they supply information on paper position by means of shaft angle encoders and a paper hole counter.

Both servos have independent closed loop control. After a start signal from the controller, a series of input pulses increments an up/down counter and a series of feedback pulses decrements the counter. The difference count is converted to a drive voltage, causing the motor to rotate. The printer controller thus maintains control of both closed loop systems by communicating through pulse trains.

• Transfer servo design

The initial design of the transfer servo centered around the concept of using a high-torque, low-inertia dc motor with permanent magnets, which had been used in another printer. Acceleration and deceleration of constant magnitude were selected for starting and stopping the paper. This resulted in a trapezoidal velocity input profile and meant that paper displacement would increase during starting and decrease during stopping according to a quadratic (parabolic) function of time.

A digital shaft angle encoder was envisioned from the start. A linear transducer, however, was assumed at first, and a continuous system with lead-lag cascade compensation was designed by conventional methods, using DSL to calculate and plot frequency response curves.

Specifically, phase and gain margins were optimized by studying families of curves associated with the movement of the poles and zeros of the compensator. This approach worked well, but a digital encoder resembles a linear system only at full printing speed, where the encoder pulse rate is very high. Analytically, such a system could be treated as neither a linear nor a sampled data system during start and stop modes, because the encoder pulse frequencies would vary widely, making the system unstable at low speeds and, therefore, useless without additional measures. The linear design was a good way to get started. However, transient simulation was needed to check the operation of the system as a function of time, including nonlinearities and discrete events; it was readily accomplished with DSL. As a result, a good understanding of the system was obtained, particularly with respect to the effect of discretization on stability.

The parabolic input command of position vs time (in digital form, Fig. 2) was accomplished by incrementing an up/down counter through a prescribed succession of pulses such that the time between pulses decreases steadily until the proper pulse frequency for printing is reached. While the paper accelerates from standstill to full speed for a distance x_a , the drum moves a distance of $2x_a$. Thereafter, they both move nominally at the same speed for a short time (until transients have died out) before printing is actually started, i.e., before toner is transferred from the photoconductor to paper. The actual shaft angle displacement was similarly parabolic in time, which resulted in a pulse sequence from the digital encoder much like the input pulse sequence; it, in turn, was used to decrement the counter. The difference count, via a digital-to-analog converter (DAC), served as the error signal for this system.

To approximate the proven continuous system, a concept of analog fill between the discrete steps of the input and feedback signals was pursued and found to work well. To obtain a continuous signal roughly proportional to paper position, motor drive voltage and armature current were judiciously mixed (to yield a "velocity estimate," explained subsequently), integrated, and reset in synchronism with the encoder pulses.

Similarly, an analog command velocity signal was integrated and reset whenever an input pulse was emitted by the microprogram, according to a stored table of integers which represent the number of periods of a clock between emission of pulses. Together with these fill signals, the DAC output became a much smoother function that did not make the system unstable at low speeds. Figure 3 is a block diagram of this system as designed and used during the initial stages of machine development. Figure 4 is a frequency response diagram for the linearized system whose open loop transfer function consisted of $G(s)$ for the amplifier/motor combination,

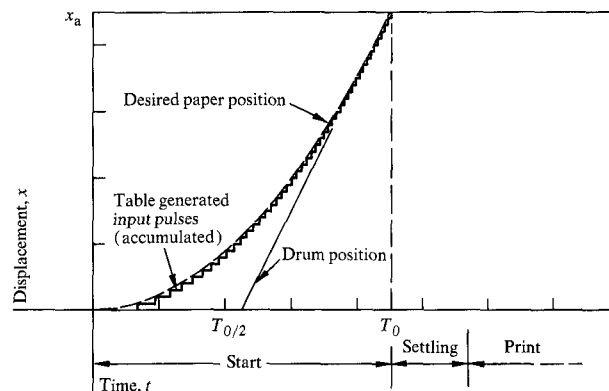


Figure 2 The parabolic form of digital input command of position vs time.

$$G(s) = \frac{14\,000}{s \left(\frac{s}{151} + 1 \right) \left(\frac{s}{3860} + 1 \right)}, \quad (1)$$

and $G_e(s)$ for the compensating filters,

$$G_e(s) = \frac{\left(\frac{s}{145} + 1 \right) \left(\frac{s}{300} + 1 \right)}{\left(\frac{s}{20} + 1 \right) \left(\frac{s}{3000} + 1 \right) \left(\frac{s}{8000} + 1 \right)}. \quad (2)$$

This linearization was deemed adequate, because the position feedback and feedforward paths for fill merely provided the signals which restored the continuous nature of actual and desired position that was initially postulated.

The motor velocity ω can readily be seen to be related to applied voltage V , armature current i , resistance R , inductance L , and back-EMF constant K_B :

$$K_B \omega(t) = V(t) - Ri(t) - L \frac{di(t)}{dt}. \quad (3)$$

In Laplace operator notation, and with the substitution $\tau_e = L/R$, Eq. (3) becomes

$$\omega(s) = [V(s) - RI(s)(1 + \tau_e s)]/K_B, \quad (4)$$

which is more easily implemented if divided by $(1 + \tau_e s)$. Because the electrical time constant τ_e is very small, the following approximation holds true:

$$\omega(s) \approx \frac{\omega(s)}{1 + \tau_e s} = \left[\frac{V(s)}{1 + \tau_e s} - RI(s) \right] / K_B. \quad (5)$$

This is the basis for the velocity estimation, mentioned previously.

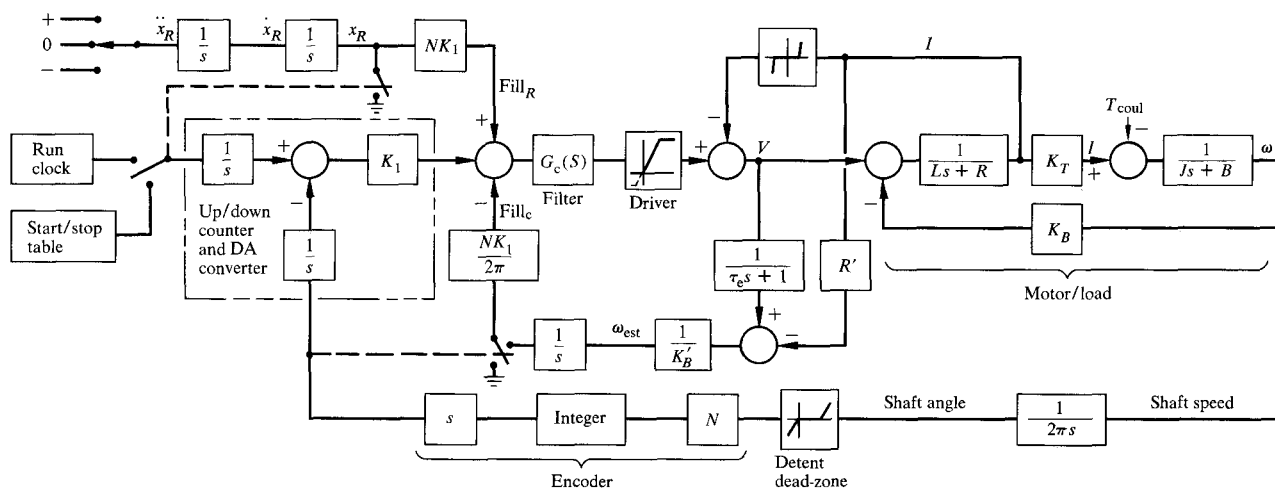


Figure 3 Block diagram of the initial design of the transfer servo.

Note that none of the system nonlinearities shown in Fig. 3 came into play during high speed printing because the sampling rate was high and saturation did not occur. Rather, they were effective only during the start and stop modes of operation, leaving the system linear during the print mode, with the exception of Coulomb friction.

The system was found to function as intended in the simulation and also in actuality. It served in early prototypes of the printer for a long time, and its success resulted in the decision to design servo systems with linear motors, both for the fuser and the drum drive. The initial fuser drive system, which used a harmonic transmission to the drum drive and on/off control through a particle clutch and an electrical brake, showed severe wear problems. The drum drive system, which used a precision gear drive and a synchronous motor, was not accurate enough and was too expensive. At that time, it was recognized that the same type of motor could be used to drive the transfer servo and the fuser rolls, except that the latter would be driven through a gear train.

When a moving coil motor was custom designed to serve both purposes, certain compromises had to be made. This resulted in an occasional instability, which was not expected from the above mathematical model. Experimental frequency response diagrams were carefully obtained, but neither gain nor phase curves vs frequency agreed with predicted ones as indicated in Figs. 4 and 5(a). Other tests showed that removal of the fill signals eliminated the oscillations. However, with fill removed, acoustical noise problems from the gears arose, along with power system problems due to current surges during start/stop. Because the discrepancies between observed and postulated frequency response curves indicated a deficiency in our mathematical model, we decided

to pursue model improvements. First, we derived the transfer function of a resetting integrator to better represent the circuitry in the velocity feedback path. It was found to be

$$\frac{\text{Fill}}{X_{\text{est}}} \approx \frac{1}{s} (1 - SH), \quad (6)$$

where SH is the conventional transfer function of a zero-order sample and hold circuit

$$SH = \frac{1 - e^{-sT}}{sT} \quad (7)$$

and where T is the encoder pulse period.

This did not prove to have the desired effect, and we modeled the mechanical load more accurately. When that effort also failed to explain the observed instabilities, it was suspected that the system might be highly sensitive to mismatch between parameters of components in the velocity estimator and those of the motor (K_B , R , L). This, finally, was found to be the answer to our question; Fig. 5(b) shows the drastic effect of resistance mismatch which, together with mismatch in other parameters, would explain the observed phenomena in the frequency domain, Fig. 5(a). Such sensitivity studies would have been virtually impossible to do experimentally.

This led to abandonment of the fill concept in favor of providing additional inputs to the system, which were designed to anticipate known loading in this case of known forcing: 1) a voltage proportional to acceleration (to overcome inertia effects), and 2) a voltage proportional to velocity (to overcome motor back-emf). These signals, when added to the compensated discrete error signal (without fill), together with the design of a complex pole filter, resulted in transients of desirable characteristics

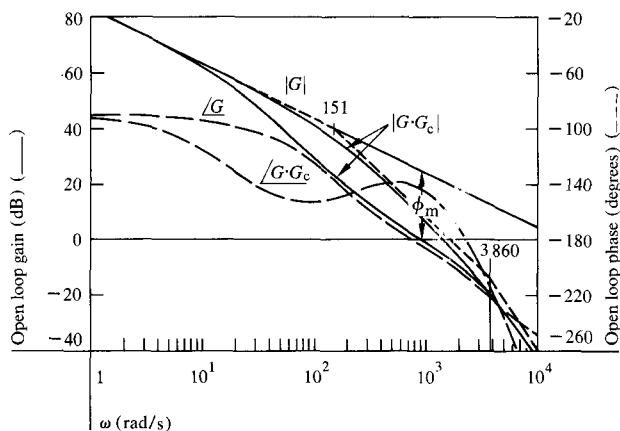


Figure 4 Frequency response of transfer servo. Open loop, uncompensated (G) and compensated (GG_c).

and inherent stability in the printing mode. Since the position command of the servo is given as piecewise parabolic, velocity and acceleration are known to be piecewise linear and constant functions, as sketched in Fig. 6. It is, therefore, easy to generate the extra system inputs to the transfer servo as well physically as in the simulation program. Figure 6 shows a simplified simulation block diagram of the transfer servo as it finally evolved.

A comparison of the position error signals obtained for the original design (with fill) and the final design (with additional forcing functions) showed the latter to be clearly superior. This proves the validity of the concept that it is advantageous to add open loop control to a system if its input or forcing function is predetermined. Figure 7 illustrates the superior capability of the system with extra inputs to track the desired position profile while achieving the necessary position lock in a shorter time.

When the first design of the transfer servo resulted in a usable system, it not only provided the foundation for tests of paper-motion-dependent subsystems, but also stimulated the design of other servomechanisms in the 3800 printer. An on-off control system for the fuser had been plagued with wear problems and was changed to a continuous control system with dc motor drive, as described subsequently. A precision gear drive system for the photoconductor drum was redesigned to a direct drive dc motor scheme that improved speed regulation greatly while reducing cost.

• Fuser servo design and slack loop control

The design of the fuser servo, as finally implemented, was guided by the idea of using as many of the transfer servo elements as possible, e.g. motor, encoder, and driver. Besides the obvious cost reduction advantage of common parts, the benefits of replacement part interchangeability

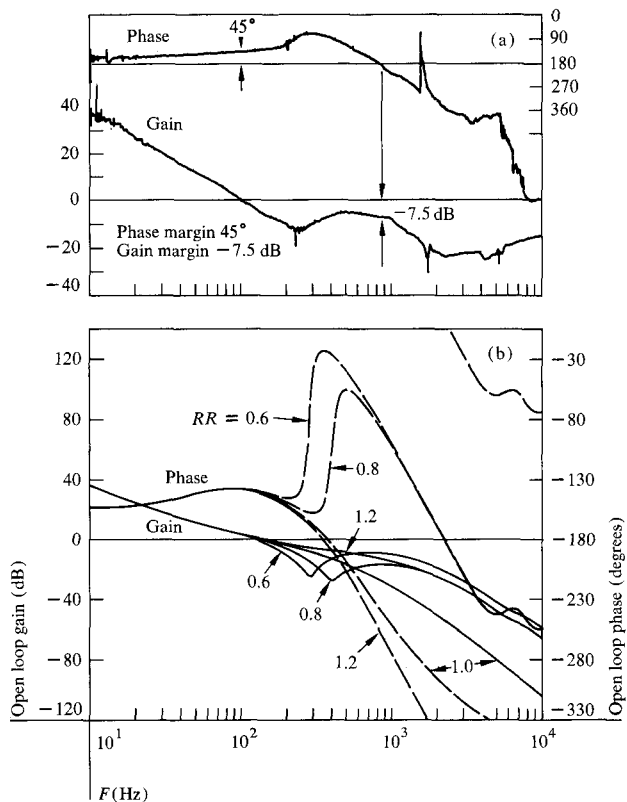


Figure 5 Open loop frequency response curves of transfer servo. (a) Experimental curves. (b) Computations for varying ratios of $RR = R'/R$.

as well as simplifications in the trouble-shooting procedure also resulted.

Paper motion through the fuser is caused by the pinch action of two rubber coated rolls of 10.2-cm diameter. The hot roll, which contacts the print side, is motor driven through a gear train, and the backup roll is normally loaded against the hot roll. Pressure and heat fuse the toner particles onto the paper, fixing the image permanently. While slippage between paper and fuser rolls is negligible, dimensional changes in the rolls and the paper cause paper displacement variations, which need to be corrected in order to assure that the paper stops between rolls only in the vicinity of the perforations; this avoids offset (sticking) of toner to the hot roll.

Differences between transfer and fuser servo and timing problems required a slack loop of paper between the two stations. Its length variations had to be minimized to prevent toner from coming off the paper because of weak electrostatic attachment prior to fusing. Control of the paper loop is, therefore, another requirement associated with the fuser control.

Figure 8 shows how the basic fuser and transfer servo systems are interconnected to assure proper operation of the paperline as a whole. The start and stop commands

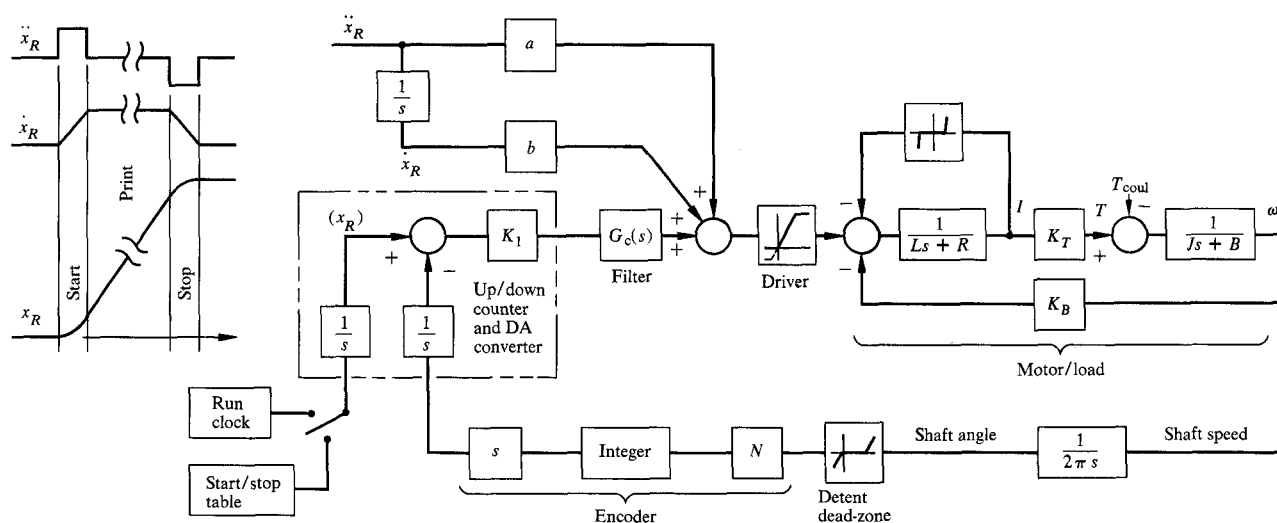


Figure 6 Simplified simulation block diagram, showing the final design of the transfer servo.

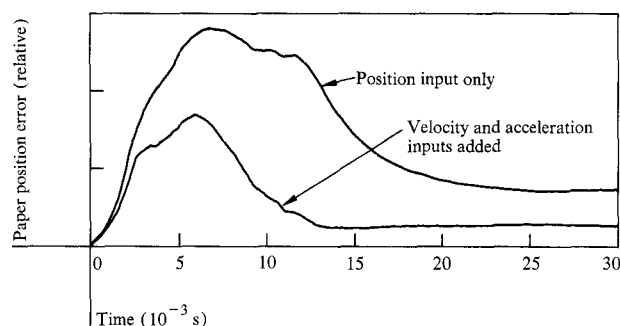


Figure 7 Position error during start of transfer servo, showing how the system tracks the desired position profile (simulation results).

cause the generation of a trapezoidal signal (accelerate/print/decelerate), which varies the frequency of a voltage controlled oscillator (VCO) such that an impulse train is generated to serve as primary input to the fuser servo, like the table-generated impulse train to the transfer servo. The resulting paper displacement is determined by a microprogram in the controller by combining information from the fuser shaft angle encoder and a pin feed hole counter ahead of the fuser. Absolute paper positions at the fuser and the transfer stations are then compared and used to supply additional impulses to the fuser servo, when it lags behind the transfer servo. The VCO is designed so that the fuser servo operates nominally slower than the transfer servo. Note the 4-ms delay in starting the acceleration table for the transfer servo; this reduces start transients in the slack loop.

The simulation program developed for the transfer servo served as the basis of a program for the fuser servo, since some similarities existed. A number of modifications were readily made to model the system properly: the gear drive in the mechanical load, the VCO input scheme, the slack loop (based on a simplified transfer servo representation), the hole count mechanism and a filter in the outer loop. This allowed us to study the interactions of the two servos and to optimize parameters throughout the evolution of the system design. As for the transfer servo, the fuser servo simulation programs and frequency response calculations provided essential tools to develop the system, minimizing the experimentation with hardware. Because of the high paper speeds, experimentation on the interaction between the servos to control paper slack would have been difficult.

Summary

Several systems for automatic control of paper motion in the high speed IBM 3800 printer have been designed to control acceleration during starting and stopping and to obtain position lock at printing speed despite widely varying mechanical loads. A mixture of analog and digital hardware was used to accomplish the task with high reliability at reasonable cost.

Accessibility of prototype machines for experimentation was very limited, yet many novel approaches had to be evaluated. Computer programs, written in a high level digital simulation language, provided the means to analyze many alternatives in time and frequency domain studies. Their flexibility, computing power and precision, combined with good graphic output capabilities, made the difficult task possible, limited only by the accuracy of the mathematical models.

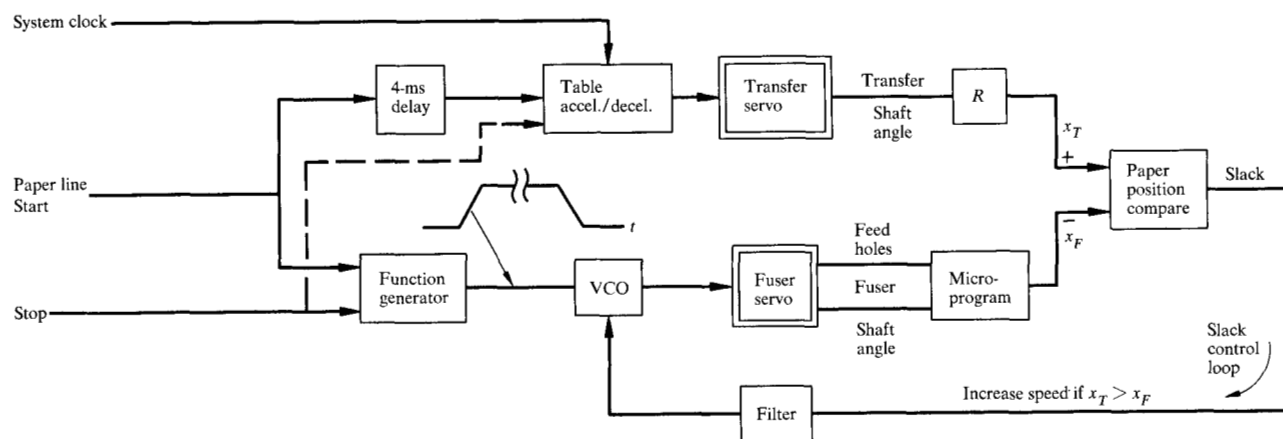


Figure 8 Fuser servo and associated slack control loop.

Acknowledgments

The contributions of many colleagues, who helped to develop the paper control systems over several years, are gratefully acknowledged. R. J. Froess, R. P. Crawford, R. L. Diehl, and L. D. Tipton were the primary contributing engineers under K. J. Elsey's leadership, and G. J. Thaler and K. L. Deckert were instrumental in systems design and analysis aspects.

References

1. R. G. Svendsen, "Paper Path of an On-Line Computer-Output Printer," *IBM J. Res. Develop.* **22**, 13 (1978, this issue).
2. The SCi Simulation Software Committee, "The SCi Continuous System Simulation Language (CSSL)," *Simulation* **8**, 281 (1967).
3. *Continuous System Modeling Program III (CSMP III) and Graphic Feature* (Program No. 5734-XW9), GH19-7000-1 and SH19-7001-2, 1972, IBM World Trade Corporation, 821 United Nations Plaza, New York, NY 10017.
4. F. H. Speckhart and W. L. Green, *A Guide to Using CSMP—The Continuous System Modeling Program*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1976.
5. M. H. Dost, W. M. Syn, and N. N. Turner, Jr., "High-Level Control Language for Scientific and Engineering Graph Generation," *Winter Simulation Conference*, Sacramento, CA, Dec. 18-19, 1975.

Received June 3, 1977

The authors are located at the IBM General Products Division laboratory, 5600 Cottle Road, San Jose, California 95193.