Distributed Arithmetic for Microcomputer Implementation of Control Algorithms

S. I. Ahson and Uday Shankar Triar

ABSTRACT: The paper describes a method for implementing direct-digital-control (DDC) algorithms on readily available microcomputers using distributed arithmetic. With the use of distributed arithmetic, fast execution of the control program can be achieved on 8-bit microcomputers without loss of precision. A position control system for a DC servomotor is described to illustrate the distributed arithmetic technique. The control system is developed around Rockwell’s AIM-65 single-board microcomputer. A single-chip version of the position control system based on Intel’s 8748 single-chip microcomputer is also described. The AIM-65 interpretive PASCAL proves to be a powerful feature that allows certain off-line computations to be easily implemented.

Introduction

In a microcomputer-based control system, direct-digital-control (DDC) algorithms are implemented in software [1]. Use of assembly language on conventional 8-bit microcomputers increases the speed of execution of the control program but at the expense of reduced accuracy due to the limited number of bits representing each coefficient and the round-off error. In order to achieve high processing speed without loss of precision, the use of distributed arithmetic [2] is proposed for the implementation of control algorithms on 8-bit microcomputers. In distributed arithmetic, multiplication—the most dominant arithmetic operation of DDC algorithms—is replaced by accesses to look-up tables containing calculated values and shift/add operations.

This paper describes a position control system for a DC servomotor that has been developed in the laboratory around two microcomputers: (a) the AIM-65 single-board microcomputer of Rockwell International and (b) the Intel 8748 single-chip microcomputer. The software for various DDC algorithms (e.g., PID, lead/lag) for the AIM-65 and the 8748 have been written in their assembly languages, and the distributed arithmetic approach has been used. In the case of AIM-65, the availability of PASCAL has been exploited for entering and modifying controller parameters for creating the look-up tables and for linking of assembly-level control program with PASCAL.

Experimental results pertaining to the proportional-integral-derivative (PID) algorithm is presented. The control objective is to change the angular position of the rotor of the servomotor in a desired manner. The hardware interfaces described in this paper and the control software developed can be used for other industrial plants.

Distributed Arithmetic for Control Algorithms

The use of a microprocessor for the realization of a direct-digital-control algorithm is limited to slow systems with large time constants due to the relatively long time taken for the multiplications. In order to improve the processing speed, a method based on distributed arithmetic [1] is proposed. This method is explained below through the PID algorithm.

The three terms of a PID controller [3] are written as

\[ Y_p(k) = K_p x(k) \]  \[ Y_d(k) = (K_d/T)x(k) - sI(k - 1) \]  \[ Y_i(k) = \int_0^k x(j) \, dt \]  \[ Y_j(k) = y(k) = Y_p(k) + Y_d(k) + Y_i(k) \]

Now consider the representation of an 8-bit number \( z(k) \) in 2’s complement form:

\[ z(k) = \sum_{j=1}^{7} z_j(k) 2^{-j} - z_0(k) \]

where \( z_j(k) \) represents the \( j \)th bit from the m.s.b. Substituting for \( x(k) \), \( x(k - 1) \), and \( y(k - 1) \) in Eq. (4) and using the 2’s complement representation of Eq. (3), we obtain

\[ y(k) = (K_p + K_T + K_d) x(k) \]

or

\[ y(k) = \sum_{j=1}^{7} \left( \frac{K_p + K_T + K_d}{2^j} \right) 2^j \]

Hardware Interface Design

The AIM-65 [4] is a single-board microcomputer consisting of an 8-bit 6.502 CPU, an 8K ROM monitor, language ROMs...
Table 1
Look-up Table for PID Algorithm

<table>
<thead>
<tr>
<th>$x_i(k)$</th>
<th>$x_j(k - 1)$</th>
<th>$y_j(k - 1)$</th>
<th>$F[x_i(k), x_j(k - 1), y_j(k - 1)]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-Ko/T</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$1 - K_p T$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$K_p + TK_i + K_o T$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$1 + K_p + TK_i + K_o T$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$K_o + TK_i$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$1 + K_o + TK_i$</td>
</tr>
</tbody>
</table>

(BASIC, PASCAL, FORTH, PL/65), 4K assembler and peripheral equipment in the form of a 20-character alphanumeric display, a 20-character thermal printer, and a full ASCII keyboard. On- and off-board expansion capabilities and user dedicated R6522 Versatile Interface Adapter (VIA) enhance the usefulness of the system. A 16-bit address bus allows the CPU to directly address 65,536 memory locations. An 8-bit bi-directional data bus carries data from 6502 to/from memory and interface devices. The control bus carries various timing and control signals. The R6502 CPU on the AIM-65 board operates at 1 MHz and produces a clock output for external use.

The R6522 VIA contains two 8-bit YO ports (PA and PB), four control/status lines (CA1, CA2, CB1, CB2), two 16-bit counter/timers, an 8-bit shift register and interrupt logic. A hardware interface between the AIM-65 and the process/plant to be controlled is shown in Fig. 1. It uses eight multiplexed inputs (CHO-CH7) for acquiring analog signals, using PA lines PAO-PA2 for channel addressing. PA3 is used for sample/hold mode control for the DatelIntersil sample/hold circuit, which has a 200 US acquisition time. Port A bits PA4 and PA5 are used to start the type ADC0800 analog-to-digital converter and to test for end of conversion.

The address decoder allows the microprocessor to read the digital output of the ADC as memory location 9FFF. The output latch of the ADC0800 is enabled with the address 9FFF. The 500 KHz clock for the

Fig. 1. Hardware details for AIM-65/process interface.

May 1985
ADC is obtained from the microprocessor clock via a divide-by-two circuit.

Conversion of the digital output on port B of the VIA is done into an analog voltage, using the digital-to-analog converter. The error signal is generated using a six-input analog adder, formed with type LM324 OP amps, as shown in Fig. 1.

The hardware interface described above is for an 8-bit single-board microcomputer with sufficient memory and VO lines. Considering possibilities for large distributed control systems where dedicated controllers may be used for local monitoring and control of a small section of a plant, an interface was designed for the Intel 8748 single-chip microcomputer [5]. The 8748 has a less powerful arithmetic unit than the 6502, a shorter 12-bit program counter, and uses the 64-byte on-chip RAM to provide for working registers and stack, as well as data storage area. It contains 1K of EPROM as program memory and 27 YO lines. Fig. 2 shows how the 8748 can be interfaced to a process/plant through the ADC0809 and the DAC0800. Hardware facilities for single stepping an 8748 program and for resetting the microcomputer are also shown.

Control Software and Test Results

The difference equation corresponding to the PID algorithm was implemented in software by distributed arithmetic, using the assembly languages of the 6502 microprocessor and the 8748 single-chip microcomputer. In the case of AIM-65, the software written in PASCAL consisted of instructions for an interactive session with the user to select the control algorithm and its parameters from the keyboard, followed by the creation of the look-up table.

AIM-65 PASCAL (20K ROM) is a subset of the standard PASCAL language [6] and incorporates extensions that permit linkage to assembly level routines and allow direct control of memory-mapped I/O of the microcomputer. These features are very useful for control applications.

The PASCAL program calls the assembly language routine, named CONTROL, located at memory address 0550H, which implements data acquisition and real-time control. The CONTROL routine is called by an instruction SUBR (CONTROL), where CONTROL is declared as a variable by the following declaration

VAR CONTROL = $0550: CHAR

90550 stands for the memory location assigned to the variable CONTROL. The CONTROL routine is an infinite-loop, so a return to the PASCAL program never occurs.

Software for the single-chip version of the PID controller was developed in the assembly language of the 8748, using the Intel-ec Series 2 Development System. The PROM programmer for the 8748, which is available on the development system, was used for programming the EPROM.

The hardware and software developed for the control systems based on the AIM-65 and the 8748 were tested for position control of an armature-controlled DC servomotor.

Fig. 2. 8748 based control system.
Before performing any test, the transfer function of the motor was determined. The transfer function is given by

\[ T(s) = \frac{37.5}{s(0 + 4.4)} \]

Theoretical analysis is based on the assumption that the motor is linear and that frictional characteristics are viscous.

The control objective is to maintain or change the angular position of the rotor in a desired manner. To achieve this, a pair of potentiometers are used to generate the error signal, which is fed to the microcomputer that computes the control signal according to the PID algorithm and outputs the same through a D/A converter to the preamplifier of the servomotor. A schematic representation of the laboratory setup (8748 version) is shown in Fig. 3.

Experiments were performed to correlate actual system performance with results predicted from theory. The angular position response for a 90 degree step in the set point was obtained for different controller settings. The plot of output pot position for \( K_p = 1.0 \), \( Kr = 0.5 \), and \( Ko = 0.0 \) is shown in Fig. 4.
Changes in controller coefficients produced predicted changes in the output response.

Conclusions

The experience gained on the AIM-65 and the Intel 8748 microcomputers can be used with other 8-bit microcomputers for control applications. A wide variety of processes can be controlled using the developed systems, which are not limited to servomotor control. With the availability of high-level languages, like PASCAL on single-board microcomputers, software development can be greatly facilitated. Efficient development of software for single-chip microcomputer-based control systems requires the use of a development systems.

References


IEEE Launches Individual Learning Packages

The IEEE announces publication of three Individual Learning Packages (ILPs) in Digital Signal Processing, Spread Spectrum Signals and Systems, and Advanced Microprocessors. Individual Learning Packages are intended for professional engineers and researchers who require current state-of-the-art knowledge in emerging high technology apart from a formal classroom mode.

Developed under the guidance of the IEEE Educational Activities Board, Individual Learning Packages contain multimedia curriculum designed to be learning-effective. Each ILP includes: (1) an IEEE Press Book and/or Course Text, (2) a Study Guide to Course Text, (3) an Audiodisc, and (4) a Final Examination evaluated by an IEEE that measures learning mastery. Continuing Education Achievement Units and an IEEE Certificate of Achievement and/or Award Label verify educational achievement.

Digital Signal Processing ILP (HS0047-1) — developed by Dr. Andrew Sekey, Department of Electrical and Computer Engineering, University of California, Santa Barbara — is an advanced and extensive three-trimester postgraduate program (400 hours) designed for the electronics or computer engineer who is familiar with linear and sampled-data systems, and who requires intensive state-of-the-art knowledge to remain highly competitive in the digital signal processing field. Priced at $179.00 for IEEE members and $225.00 for nonmembers, the Digital Signal Processing ILP also includes two optional additions: (1) ILS-IEEE Simulation Software System (HS0048-9), a user-friendly "electronic study partner" for the user who requires hands-on experience. This software package includes user's manual, workbook, and a set of 5 inch floppy diskettes that can be run on an IBM PC. This ILS-IEEE interactive software is available only with the Digital Signal Processing ILP. Only one-tenth the price of its nearest commercially available counterpart, this optional addition is $150.00 for both IEEE members and nonmembers (2) mainframe tape for the IEEE Press Book Programs for Digital Signal Processing contains 29 advanced and tested Fortran programs for the serious researcher or designer. The tape is standard 1/2-inch width on a 600-foot reel and is available in either 800 (PT05004) or 1600 (PT05017) bits-per-inch (bpi) in EBCDIC. This optional addition is priced at $62.95 for both IEEE members and nonmembers.

Spread Spectrum Signals and Systems ILP (HS0046-3) — developed by Mr. Robert C. Dixon, Senior Member of IEEE and engineering faculty member at the George Washington University and UCLA — introduces the concepts of spread spectrum systems through eight independent learning sessions (80 hours) from basic spread spectrum theories and techniques to design and analysis for complex spread spectrum system design. Spread Spectrum Signals and Systems ILP is priced at $179.00 for IEEE members and $225.00 for nonmembers.

Advanced Microprocessors ILP (HS0045-5) — developed by Dr. G.B. Soss Gray under the direction of Longman Crown, Inc. — introduces in nine independent learning sessions (80 hours) the theories and design processes in single-chip technology. Major emphasis is placed on 16- and 32-bit microcomputer systems (LIPUS). Advanced Microprocessors ILP will be available in the fall of 1985.

The three Individual Learning Packages in Digital Signal Processing, Spread Spectrum Signals and Systems, and Advanced Microprocessors can be ordered from the IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854, USA. For each ILP ordered, a $8.50 shipping and handling charge is added to the total ILP purchase.

Reader Service #77