MOTOROLA
SEMICONDUCTOR
APPLICATION NOTE

AN1120

Basic Servo Loop Motor Control Using the MC68HC05B6 MCU

By Jim Gray

This application note describes a basic circuit and software implementing proportional derivative (PD) closed-loop speed control for a brush motor using four integrated circuits (ICs), two opto discretes, and less than 200 bytes of code.

Feedback control systems using digital algorithms implemented on microcontroller units (MCUs) are becoming increasingly commonplace. The use of an MCU in this type of control application is justified when system flexibility is needed, such as varying drive motors or storing wear parameters in electrically erasable programmable read-only memory (EEPROM). Typically, the system would be modeled mathematically in the discrete time domain due to the use of sampled rather than continuous data. The linear difference equations describing the transfer function of the system are solved using z-transforms, allowing, in the case of proportional-integral-derivative (PID) control, the determination of constants for proper system performance and stability. However, this level of analysis is not necessary to illustrate how straightforward the implementation is using the MC68HC05B6 and the MPM3004 TMOS™ H-bridge. The generalized flow of a PD loop is shown in Figure 1. The transfer function of $G_C(s)$ consists of the PD control, and $G_P(s)$ represents the power amplifier, motor, and load. Here s is a complex variable having both real and imaginary parts. The proportional term K_P can be accomplished with shifting operations, at least to the resolution of powers of 2. The derivative term, K_Ds, of $f(t)$ is approximately

$$\left.\frac{df(t)}{dt}\right|_{t = kT} \cong \frac{1}{T}[f\,(kT) - f\,(k-1)T]$$

where $f(kT)$ is the current value of the controlled parameter, and $f(k-1)T$ is the value of the same parameter at the previous sampling time. In this example, k_Ds is realized as the rate of change of the difference between the measured and the desired period of motor-shaft rotation.

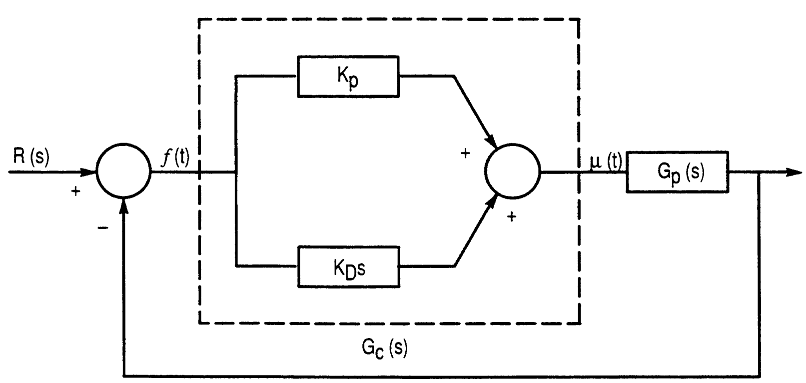

Figure 1. PD Loop Flow

TMOS is a trademark of Motorola Inc.

The MC68HC05B6 is an M68HC05 MCU Family member with two channels of programmable pulse-length modulation on-chip. When used with an H-bridge device such as the MPM3004, these channels can control bidirectional currents of up to 10-A continuous (25-A peak) at 60 V (see Figure 2). Two I/O pins and both pulse-length modulation (PLM) channels are used to control the MPM3004. Proper gate drive and level conversion is provided by the MC34151 dual inverting gate drivers. Input to the control loop consists of the MLED71 infrared emitter and MRD750 photo Schmitt trigger detector coupled through a slotted disc on the motor shaft. The TCAP2 pin and associated input capture registers are used to convert the optical index marks into a time measurement. Great care must be taken to ensure an adequate current source for the MPM3004 and to isolate the supply for the MC34151s. Separate circuit runs and 0.1-µF bypass capacitors on the MC34151 ICs were used in this case.

The justification for adding a derivative term to a proportional controller can be easily understood by examining the reasons for the overshoot and ringing typical of an underdamped proportional-only controller. When proportional control applies additional power to correct an underspeed condition, it does so continuously until the error term is zero, resulting in a power setting that ensures an overspeed condition. The converse occurs when reducing motor speed. The rate of change of the error signal as excessive power is being applied to correct underspeed will be a relatively large negative value (the error term is being rapidly reduced). Thus, the derivative of the error term is of the correct sign to compensate the proportional gain term. One effect of this compensation is to retard the loop's response time, but the proportional gain can be increased to offset this.

The listing (see Figure 3) shows the assembly source code for speed measurement and the PD control of PLMA, which drives the power H-bridge in one direction. The opposite direction of rotation is obtained by complementing bits 0 and 1 of port A and driving the opposite lower leg of the H-bridge with PLMB. Eight-bit arithmetic was used exclusively in this example for space and clarity. Although this approach is functional, 16-bit routines for multiply and divide, given in Reference 2, are better for finer control. Routines to set initial values, control direction of rotation, and check for motor stall are also necessary, although they are not shown in this application note.

Figure 4 shows the response of the system to various changes in load. The data was captured in an emulator trace buffer (Motorola CDS8 Jewelbox) and plotted using a data base program. Beginning from a no-load condition at 4 s, loading (an uncalibrated friction brake) was ramped to cause approximately a 50-percent duty cycle. Starting at 10 s, the load was then increased again until the system was at the limit of compliance — i.e., at full power and still maintaining the desired speed. Next, at 14 s, approximately half the load was rapidly (0.1 s) removed. The gain of the proportional term was 2, and the derivative constant was 1. In systems where a low-pass filter would be beneficial or the steady state error is potentially large, an integral term could be added for full PID control.

REFERENCES

1. Kuo, Benjamin C., *Automatic Control Systems*, New Jersey: Prentice-Hall, 1987.
2. M6805UM/AD2, *M6805 HMOS/M146805 CMOS Family User's Manual*, New Jersey: Prentice-Hall, 1983.
3. MC68HC05B6/D, *MC68HC05B6 Data Sheet*, Motorola, 1988.
4. M68HC05AG/AD, *M68HC05 Applications Guide*, Motorola, 1989.

Figure 2. Block Diagram of Servo Loop Motor Control

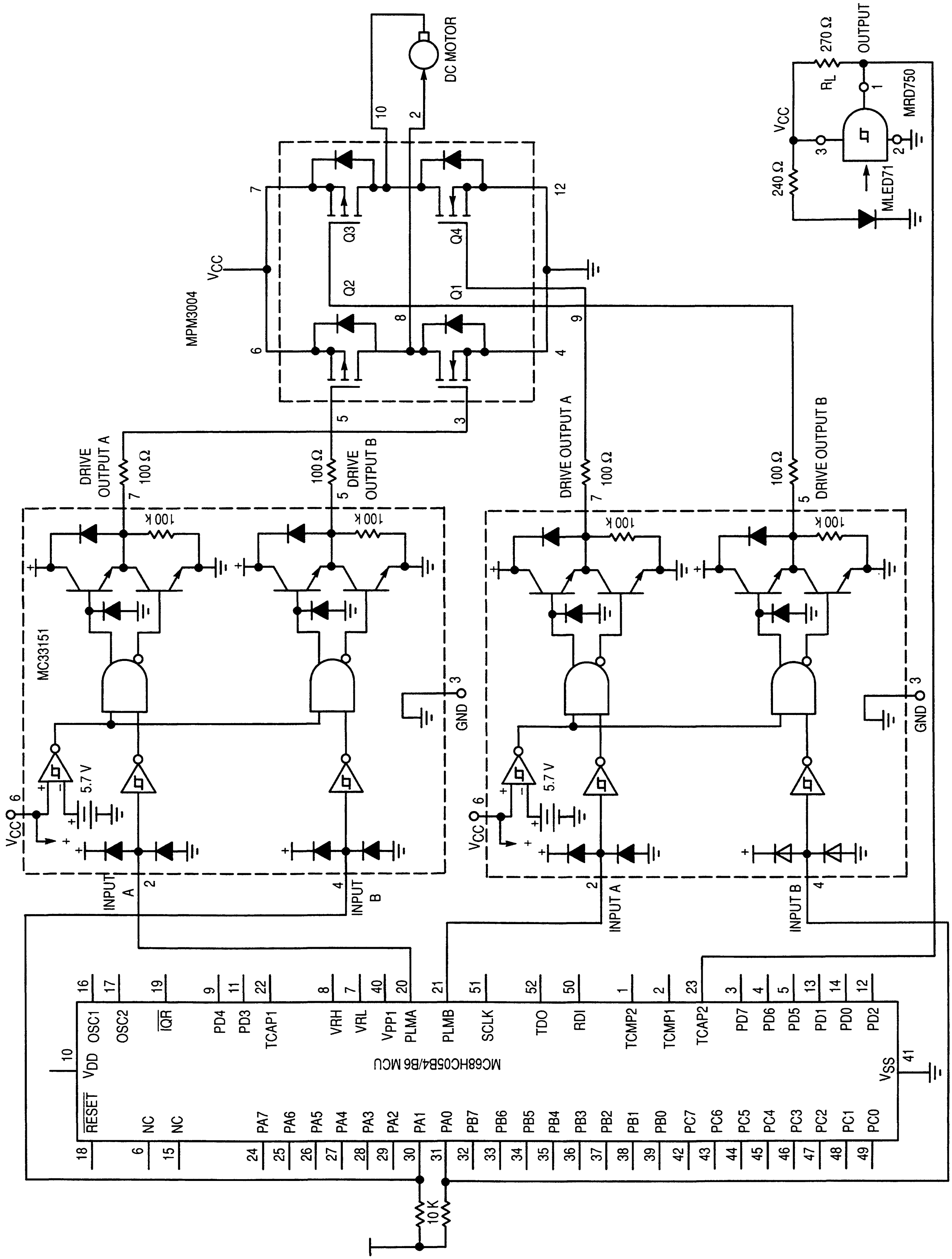

Figure 2. Block Diagram of Servo Loop Motor Control

```
                           ***************************************************************************
                           *                  MC68HC05B6 SERVO LOOP MOTOR CONTROL EXAMPLE            *
                           * This program performs a closed loop servo speed control using PLMA for  *
                           * output. Speed is measured optically with a slotted disk. The optically  *
                           * detected index  mark,controls TCAP2 which  allows  calculation  of the  *
                           * period  of revolution for the  loop  input.                             *
                           ***************************************************************************

0000                                org    .$0
                                    cycles off
0000
0000                       PADR     RMB     1
0001                       PBDR     RMB     1
0002                       PCDR     RMB     1
0003                       PDIDR    RMB     1
0004                       PADDR    RMB     1
0005                       PBDDR    RMB     1
0006                       PCDDR    RMB     1

000A                                ORG      $0A

000A                       PLMA     RMB     1
000B                       PLMB     RMB     1
000C                       MISC     RMB     1

0012                                ORG     $12

0012                       TCR      RMB     1
0013                       TSR      RMB     1
0014                       CAHR1    RMB     1
0015                       CALR1    RMB     1
0016                       COHR1    RMB     1
0017                       COLR1    RMB     1
0018                       CNTHR    RMB     1
0019                       CNTLR    RMB     1
001A                       ACNTHR   RMB     1
001B                       ACNTLR   RMB     1
001C                       CAHR2    RMB     1
001D                       CALR2    RMB     1

0050                                ORG     $50

0050                       BCNTH    RMB     1
0051                       BCNTL    RMB     1
0052                       ECNTH    RMB     1
0053                       ECNTL    RMB     1
0054                       PERIOD   RMB     1
0055                       PLMTMP   RMB     1                MUST BE INITIALIZED WITH STARTING VALUE
0056                       DESPRD   RMB     1                MUST BE INITIALIZED WITH DESIRED PERIOD COUNT
0057                       DELTAN   RMB     1
0058                       DELTAO   RMB     1
0059                       DELTADC  RMB     1
005A
0F00                                ORG     $F00

0F00 A604                  BEGIN    LDA     #$4              SELECT SLOW PLM REPETION RATE
0F02 B70C                           STA     MISC             SPEED
0F04 B655                           LDA     PLMTMP           LOAD PLM VALUE
0F06 B70A                           STA     PLMA
0F08 B613                  KEYS     LDA     TSR              CLEAR FLAG AND ANY PENDING INT.
0F0A B61C                           LDA     CAHR2
0F0C B61D                           LDA     CALR2
0F0E 1E12                           BSET    7,TCR            SET INPUT CAPTURE INTERRUPT ENABLE
0F10 9A                             CLI                      CLEAR I BIT ALLOWING TIMER INTERRUPTS
0F11 20FE                  WAIT     BRA     WAIT             WAIT FOR OPTO INDEX TCIC INTERRUPT
0F13 B613                  RPM      LDA     TSR              CLR TSR BIT 4 TO ENSURE
0F15 B61C                           LDA     CAHR2            SYNCHRONIZATION TO INDEX
0F17 B61D                           LDA     CALR2
```

Figure 3. MC68HC05B6 Servo Loop Motor Control Example

```
0F19 081302    TFLAG1   BRSET  4,TSR,INDEX1   TEST FLAG FOR INDEX1
0F1C 20FB               BRA    TFLAG1
0F1E B61C      INDEX1   LDA    CAHR2          STORE COUNT
0F20 B750               STA    BCNTH
0F22 B61D               LDA    CALR2
0F24 B751               STA    BCNTL
0F26 4F                 CLRA                  DELAY TO AVOID RETRIGGER ON SAME INDEX
0F27 4A        DEC1     DECA
0F28 26FD               BNE    DEC1
0F2A B613               LDA    TSR            CLEAR FLAG AND WAIT
0F2C B61C               LDA    CAHR2          FOR INDEX2
0F2E B61D               LDA    CALR2
0F30 081302    TFLAG2   BRSET  4,TSR,INDEX2
0F33 20FB               BRA    TFLAG2
0F35 B61C      INDEX2   LDA    CAHR2          STORE SECOND COUNT
0F37 B752               STA    ECNTH
0F39 B61D               LDA    CALR2
0F3B B753               STA    ECNTL
0F3D B652               LDA    ECNTH          CALCULATE PERIOD
0F3F B050               SUB    BCNTH          THEN
0F41 B754               STA    PERIOD         STORE.
0F43 B657               LDA    DELTAN         GET PREVIOUS ERROR AND
0F45 B758               STA    DELTAO         STORE IT.
0F47 B656               LDA    DESPRD         LOAD DESIRED PERIOD, SUBTRACT ACTUAL
0F49 B054               SUB    PERIOD         TO FORM DELTAN.
0F4B 2529               BLO    INCSPD         GO TO INCREMEMTING PLM
0F4D 48                 LSLA                  MULTIPLY ERROR BY 2.
0F4E B757               STA    DELTAN         OR FALL THRU TO DECREMENTING HERE.
0F50 B658               LDA    DELTAO         FORM RATE OF CHANGE
0F52 B057               SUB    DELTAN         OF ERROR
0F54 B759               STA    DELTADC        AND STORE.
0F56 B657               LDA    DELTAN         GET CURRENT ERROR
0F58 B059               SUB    DELTADC        AND APPLY DE/DT CORRECTION
0F5A B759               STA    DELTADC        THEN STORE.
0F5C B655               LDA    PLMTMP         GET CURRENT PLM
0F5E B057               SUB    DELTAN         AND APPLY CORRECTION.
0F60 2208               BHI    ADJDN          BRANCH TO DECREMENT IF RESULT POSITIVE
0F62 A610      PLMMIN   LDA    #$10           OTHERWISE IN LOW SATURATION SO
0F64 B70A               STA    PLMA           KEEP PLM AT MINIMUM.
0F66 B755               STA    PLMTMP
0F68 2023               BRA    DONE
0F6A A110      ADJDN    CMP    #$10           SEE IF PLM AT MINIMUM
0F6C 2202               BHI    DECSPD
0F6E 20F2               BRA    PLMMIN
0F70 B70A      DECSPD   STA    PLMA           DECREMENT PLMA
0F72 B755               STA    PLMTMP         UPDATE PLMA TEMPORARY LOCATION
0F74 2017               BRA    DONE
0F76 48        INCSPD   LSLA                  MULTIPLY ERROR BY 2
0F77 B757               STA    DELTAN         INCREMENT WITH SATURATION
0F79 B658               LDA    DELTAO         FORM RATE OF CHANGE
0F7B B057               SUB    DELTAN         OF ERROR.
0F7D BB57               ADD    DELTAN         NOW ADD IT TO CURRENT DELTA
0F7F B759               STA    DELTADC        TO FORM RATE OF CHANGE COMPENSATED ERROR.
0F81 B655               LDA    PLMTMP         GET CURRENT PLM
0F83 B059               SUB    DELTADC        AND APPLY CORRECTION.
0F85 2502               BLO    ADJUP
0F87 2004               BRA    DONE           IN SATURATION OR CORRECTION EQUALS 0
0F89 B70A      ADJUP    STA    PLMA
0F8B B755               STA    PLMTMP
0F8D 80        DONE     RTI                   RETURN TO WAIT


1FF0                    ORG    $1FF0          set vectors
1FF0 0F00               FDB    BEGIN          R
1FF2 0F00               FDB    BEGIN          SCI
1FF4 0F00               FDB    BEGIN          TOV
1FF6 0F00               FDB    BEGIN          TOC
1FF8 0F13               FDB    RPM            TIC
1FFA 0F00               FDB    BEGIN          IRQ
1FFC 0F00               FDB    BEGIN          SWI
```

```
139 1FFE 0F00                    FDB     BEGIN              RES
140 2000                         END

Symbol Table:

Symbol Name     Value  Def.#    Line Number  Cross Reference

ACNTHR          001A   *00036
ACNTLR          001B   *00037
ADJDN           0F6A   *00110   00105
ADJUP           0F89   *00126   00124
BCNTH           0050   *00043   00072  00088
BCNTL           0051   *00044   00074
BEGIN           0F00   *00056   00132  00133 00134  00135 00137  00138 00139
CAHR1           0014   *00030
CAHR2           001C   *00038   00061  00067 00071  00079 00083
CALR1           0015   *00031
CALR2           001D   *00039   00062  00068 00073  00080 00085
CNTHR           0018   *00034
CNTLR           0019   *00035
COHR1           0016   *00032
COLR1           0017   *00033
DEC1            0F27   *00076   00077
DECSPD          0F70   *00113   00111
DELTADC         0059   *00052   00099  00101 00102  00121 00123
DELTAN          0057   *00050   00090  00096 00098  00100 00104  00117 00119  00120
DELTAO          0058   *00051   00091  00097 00118
DESPRD          0056   *00049   00092
DONE            0F8D   *00128   00109  00115 00125
ECNTH           0052   *00045   00084  00087
ECNTL           0053   *00046   00086
INCSPD          0F76   *00116   00094
INDEX1          0F1E   *00071   00069
INDEX2          0F35   *00083   00081
KEYS            0F08   *00060
MISC            000C   *00024   00057
PADDR           0004   *00016
PADR            0000   *00012
PBDDR           0005   *00017
PBDR            0001   *00013
PCDDR           0006   *00018
PCDR            0002   *00014
PDIDR           0003   *00015
PERIOD          0054   *00047   00089  00093
PLMA            000A   *00022   00059  00107 00113  00126
PLMB            000B   *00023
PLMMIN          0F62   *00106   00112
PLMTMP          0055   *00048   00058  00103 00108  00114 00122  00127
RPM             0F13   *00066   00136
TCR             0012   *00028   00063
TFLAG1          0F19   *00069   00070
TFLAG2          0F30   *00081   00082
TSR             0013   *00029   00060  00066 00069  00078 00081
WAIT            0F11   *00065   00065

                Errors: None
                Labels: 47
  Last Program Address: $1FFF
  Last Storage Address: $FFFF
         Program Bytes: $009E 158
         Storage Bytes: $0020 32
```

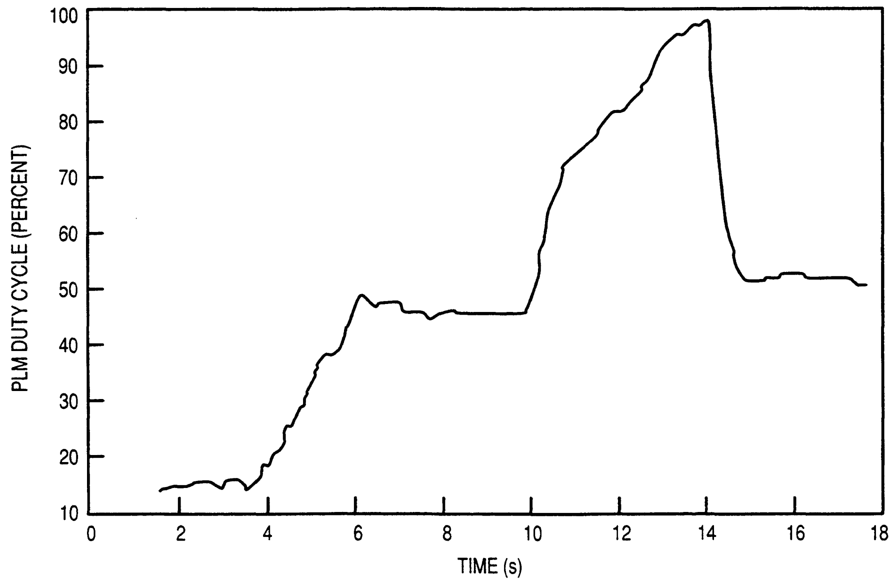

Figure 4. Step Response of PLM Motor Control

Literature Distribution Centers:
USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Center; 88 Tanners Drive, Blakelands, Milton Keynes, MK14 5BP, England.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141 Japan.
ASIA-PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Center, No. 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.

MOTOROLA

A30024 PRINTED IN USA 5/91 IMPERIAL LITHO 79535 18,000 MCU YCACAA

AN1120/D