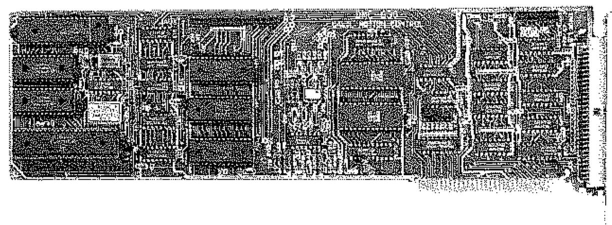
GALIL

DMC/600

Advanced Motion Controller

Features

- IBM* PC/XT/AT compatible
- Controls motion of up to three servo motors
- Independent or coordinated motion
- Circular and linear interpolation of 2 axes
- Continuous contouring for up to 3 axes
- FIFO buffer for fast communication
- User-definable application programs
- Conditional statements for controlling program execution real-time
- Programmable time and position trippoints
- Variables for entering and changing system parameters
- Arithmetic functions for manipulating parameters
- Uncommitted I/O
- Digital filter with gain, damping and integrator for optimum performance
- Error handling, end of travel, emergency stop, status reporting
- 500,000 counts/sec maximum speed
- Computer-Aided Servo Design Kit available



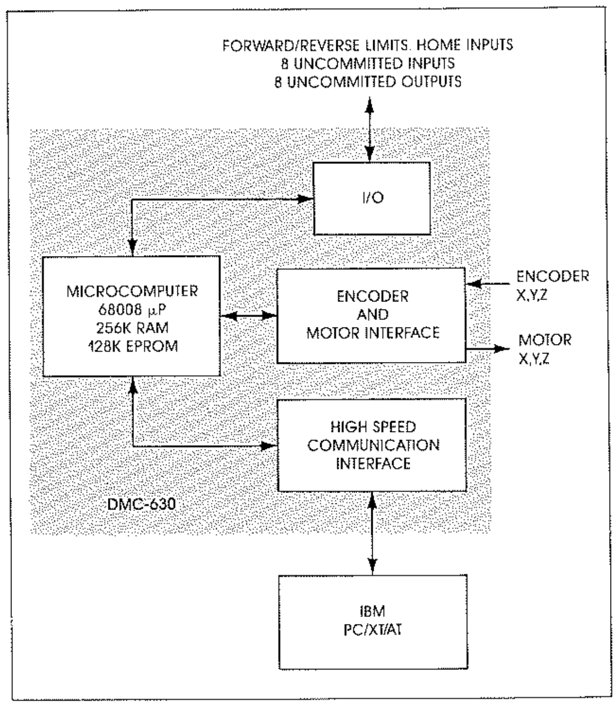


Figure 1. Motion Control System

Applications

- NC machines
- X-Y stages
- Web Processing

General Description

The DMC-600 series is a fully programmable servo motion controller contained on an IBM PC compatible card. It controls the motion of up to three DC or brushless motors with incremental encoder feedback. The DMC-610 controls one motor, the DMC-620 controls two motors, and the DMC-630 controls three servo motors.

Modes of motion include independent or vector positioning, contouring, jogging and homing. The motion profiles for each motor may be specified separately or as a sequence of coordinated vectors. The coordinated mode provides linear and circular interpolation of two axes with continuous motion at the programmed vector velocity and acceleration. Up to 256 straight line or arc segments may be specified in one continuous motion sequence. The contouring mode generates

*IBM PC is a registered trademark of International Business Machines Corporation.

motion for all the axes along any user-defined position path.

The controller contains an extensive instruction set for executing complex motion programs. Instructions are sent to the DMC-600 via the PC Bus. A FIFO buffer allows fast pipelining of instructions. Programs can also be downloaded from the PC into the DMC-600 memory.

Execution of motion commands in a program can be controlled using conditional statements which make decisions based on the logical state of I/O lines and motion parameters. Variables allow parameters to be input or modified during program execution. Arithmetic operations of variables are permitted.

The DMC-600 contains a digital filter with an integral gain term for eliminating position error at stop. The filter coefficients can be changed on-the-fly for optimum dynamic performance. Feedforward parameters are provided for reducing error during acceleration.

Several error handling features are available including automatic shut-off for excessive position error, limit switch inputs, emergency stop inputs and programmable torque limits. Complete status reporting and position monitoring functions are included. The DMC-630 also contains 8 uncommitted input lines and 8 output lines.

System Elements

The elements of the DMC-630 servo system are shown in Figure 1. The elements include the IBM PC, DMC-630 motion controller, a motor, incremental encoder and power amplifier for each axis of motion, and external switches such as end-of-travel and homing inputs. Connection of these elements is simplified with the ICB-960 interconnect board.

IBM PC—Sends high level commands to the DMC-630. Use of the PC computer may be minimized by storing motion programs in the DMC-630 memory.

DMC-630—Performs all the time-intensive functions of motor control. These functions include generating motion profiles and position trajectories, decoding the encoder feedback and comparing it with the command position, stabilizing the servo system, outputting a motor command signal for driving the power amplifier, and providing error and status reporting.

Motor—The DMC-630 controls up to three DC or brushless motors. Any size motor may be used as long as the power amplifier provides sufficient voltage and current to drive the motor.

Power Amplifier—This element amplifies the DMC-630 command signal to the appropriate current necessary for driving the motor and load. Each axis requires its own amplifier. The amplifier should be configured as a current source when no velocity feedback is used, and as a velocity amp when tachometer feedback is included. The analog output from the DMC-630 to the amplifier varies between −10 and +10 volts. The DMC-630 also provides a pulse-width-modulated (PWM) output for switching power transistors directly. The PWM output is available in two formats: Inverter and Sign Magnitude. In the Inverter mode, the PWM signal is 0% duty cycle for full negative voltage, 50% for 0 voltage and 99.6% for full positive voltage. In the Sign Magnitude Mode (Jumper SM), the PWM signal is 0% for 0 voltage, 99.2% for full voltage and the sign of the motor command is available at the sign output.

Encoder—The encoder translates motor motion into an electrical signal which is decoded by the DMC-630 as the motor position. Each axis requires feedback from an incremental encoder with two channels in quadrature, CH A and CH B. The encoder may be TTL or analog with magnitude up to 12 Volts. For noise immunity, differential encoder inputs, CH A− and CH B−, may also be input. The DMC-630 also accepts an encoder index signal which is useful for referencing the encoders during the Home instruction.

The DMC-630 performs quadrature decoding of the encoder signals, resulting in a resolution of quadrature counts (4 x encoder cycles).

External Inputs—Limit switches and emergency stop inputs may be connected to the DMC-630 to prevent system damage. There are also uncommitted inputs and outputs the user can define.

Stabilizing the Servo System

The DMC-630 implements a digital filter for compensating the closed-loop system. This filter provides system gain, damping and integration for optimizing system dynamic response and eliminating position error. The filter coefficients (Gain, Zero, Pole and Integral Gain) can be adjusted on-the-fly.

To simplify the tuning process, Galil's Digital Motion Monitor, DMM-900, is recommended. The DMM-900 converts the digital encoder position into an analog signal for display on an oscilloscope. The designer writes a program to issue repeated step commands to the motor and then observes the actual motor response. The filter parameters are adjusted until the desired response is observed.

The DMC-630 also provides acceleration feedforward (FA) for reducing the following error during acceleration.

Computer Interface

The DMC-630 is programmable, receiving commands over the PC Bus. The controller is configured as a standard IBM PC Bus card that is mapped into I/O space. Communication between the DMC-630 and PC is in the form of ASCII characters, where data is sent and received via 256 character READ and WRITE FIFO buffers on the DMC-630. A handshake is required for sending and receiving data.

Communication Registers—Description

Register Name	Description	Address	READ/WRITE
READ	for receiving data	N	Read only
WRITE	for transmitting data	N	Write only
STATUS	for handshaking	N+1	Read only
CLEAR BUFFER	for clearing FIFO buffer	N+1	Write only
SYNCH START	for synchronizing multiple controllers	N+2	Write only

Address Selection

The READ and WRITE registers occupy the same address, N, in the I/O space. The STATUS and CLEAR BUFFER registers occupy the next address, N + 1.

The selection of the address, N, is done by inserting the appropriate jumpers labeled A2 through A8. The address can be every fourth number between 512 and 1016, where A2 represents the 2^2 bit and A8 represents the 2^8 bit. A jumper corresponds to a binary 1. The default address is 1000 decimal.

Programming the DMC-630

Instruction Set

The DMC-630 contains an extensive instruction set for programming a variety of motion sequences. Each instruction is represented by a two-character operation code followed by the applicable motion parameters.

DMC-630 instructions are upper case ASCII, except for some special commands. A semicolon or carriage return terminates the motion commands.

For example, the instruction

PR 4000;

is the Position Relative command. PR is the command and 4000 represents the required position value. The ; terminates the command.

Where applicable, instruction parameters may be specified for the X,Y or Z axes independently or simultaneously. Some instructions, such as the tell position command (TP) request that data be returned to the host.

For example,

```
SP   20000, 40000, 70000;
AC   100000,, 200000;
PR   ,200;
BGY;
TP   XY;
```

specifies the speed of the X axis as 20000 counts/sec, Y axis as 40000 counts/sec and Z axis as 70000 counts/ sec. The acceleration of the X-axis is specified as 100000 counts/sec^2 and the Z-axis as 200000. The Y-axis position only is set at 200 counts. The command BGY causes the Y axis to begin motion. TP XY tells the position of the X and Y axis.

A complete listing of the DMC-630 instructions is given in Table 1.

Motion Programs

Instructions can be combined to form motion programs. Instructions can be sent from the PC as they are executed or they may be downloaded into the DMC-630 memory. The DMC-630 memory stores up to 200 lines of 32 characters per line. Multiple commands may be stored on a line. Separate programs or subroutines are distinguished by labels.

Programs are entered and edited using the DMC-630 editor. The edit mode is entered using the ED command.

The execution of statements in memory is controlled by conditional commands which make logical decisions based on controller status, I/O lines, and motor operation.

The conditional commands cause the program to branch on a condition or to hold further execution until an event occurs. For example, the wait (WT) command causes the program to hold execution until the specified time has elapsed. The After Motion (AM) command waits until the current motion is complete. The Jump on Condition (JP) instruction causes a jump to another program line if a logical condition is satisfied. The JP instruction has the format:

JP Destination, logical condition

The destination is a program line number or label. The condition may be the status of an input or output line or variable. The use of logical operators =, <, > is permitted.

Example:

JP #A, I1 = 1	Jump to label A if input 1 is high
JS #B, V1 + V2<4	Jump to subroutine B if V1 + V2 is less than 4
JP 2, PEX = 0	Jump to line number 2 if position error of X equals zero
JP 5, PY = PZ	Jump to line number 5 if position of Y is equal to position of Z axis

Example:

000 #A; CB1	Program starts at label A
001 PR 1000	Define Position
002 SP 10000	Define Slew Speed
003 AC 100000	Define Acceleration
004 WT 500	Wait here 500 msec
005 BGX	Begin Profile
006 AMX	Wait here until after motion complete
007 SB1	Set output bit 1
008 JP #A, I1 = 1	Repeat program If input 1 is high
009 EN	End Program

Variables

The DMC-630 provides 64 variables specified by the command V0 through V63. Variables can be written into motion programs and later be assigned a specific value. Variables can also be specified as the actual motor position (PX, PY, PZ), position error (PEX, PEY, PEZ), as the state of input lines (I1 through I8), or as the state of output lines. Once defined, variables can be manipulated with the arithmetic operations of add, subtract, multiply and divide. Arithmetic operations are performed from left to right. Variables can be assigned to motion parameters such as position. They can also be tested with logical operations using the Jump on Condition (JP) instruction.

Variables are real numbers between the range of ±8388607.000. The fractional part has a resolution of 1/256th.

Example:

000	#A	Begins program A
001	PA V1	Assign position X as V1
002	BGX	Begin motion
003	#B; AMX	After motion complete
004	WT 5	Wait 5 msec
005	JP #C, PX = V1	Jump to #C if no error
006	PR V1 − PX; BGX	Perform a correction
007	JP #B	Jump to B
008	#C; EN	End program

In this program, the X target position is left as a variable to be assigned. After the motion is executed and is complete, line 5 checks the X axis position error, PEX. If it is not zero, a correction is made.

Modes of Motion

The DMC-630 controller can operate in independent or vector positioning, contouring, jogging or homing modes. Motion profiles can be specified for each of the X, Y and Z axes separately, or the X,Y velocity profiles can be coordinated for motion along a vector. Both linear and circular interpolation algorithms are provided for vector motion along straight line and arc segments. The various modes of motion are described below.

Independent Positioning

The acceleration rate (AC), slew speed (SP) and end position (PA) or (PR) for each axis are specified. On Begin (BG), the DMC-630 generates a trapezoidal velocity profile and position trajectory. The Begin command can be issued for all axes either simultaneously or independently.

Example:

PA	1000, 2000, 3000	Specify X, Y, Z position
SP	40000, 100000, 200	Specify X, Y, Z speed
AC	100000, 100000, 1000000	Specify X, Y, Z accel
BG	X	Begin X only
WT	500	Wait 500 msec
BG	YZ	Begin Y, Z motion

The speed can be changed at any time during motion. The acceleration cannot be changed during motion. A Stop (ST) can be issued at any time to decelerate motion to a stop. The Increment Position (IP) instruction allows the position target to be extended while the motor is in motion.

Jog Mode

In this mode, an end position is not specified. The acceleration (AC) and slew speed (JG) are given. On Begin (BG), the motor begins accelerating to the slew speed and runs at that speed until a new speed is entered or a stop (ST) command is issued. The speed, direction, and acceleration may be changed during motion. The direction of motion is specified by the sign of the (JG) command.

Example:

JG	40000	Specify speed
AC	100000	Specify acceleration
BG	X	Begin motion
WT	1000	Wait 1000 msec
AC	50000	Change acceleration
JG	−40000	Change direction and speed
WT	200	Wait 200 msec
ST	X	Stop motion

Coordinated Motion Sequences

Operation in this mode is specified in terms of the trajectory coordinates and the vector velocity and acceleration. Up to 256 different straight line and arc segments may be specified in a sequence. The command BG S (Begin Sequence) causes the DMC-630 to begin generating the trajectory of the continuous path. The total distance traveled along the path must not exceed 8388607 counts. Linear segments are specified by the X-Y coordinates of their final points.

For example:

VP 10000, 20000

defines a linear segment ending at the given X,Y coordinates. Coordinates must always be specified with respect to the start of the move.

Circular arcs are expressed in terms of the radius, initial angle and travel angle. The units for the angle are in degrees, but fractional degrees are permitted. For example,

CR 1000, 135.125, 90.0

defines a circular arc with a radius of 1000 counts, starting at 135.125° and moving in a positive direction of 90°. The definition of the angles is as indicated by Figure 2.

The vector velocity and acceleration are defined independently with vector acceleration (VA) and vector speed (VS) commands. The vector speed may be changed during motion.

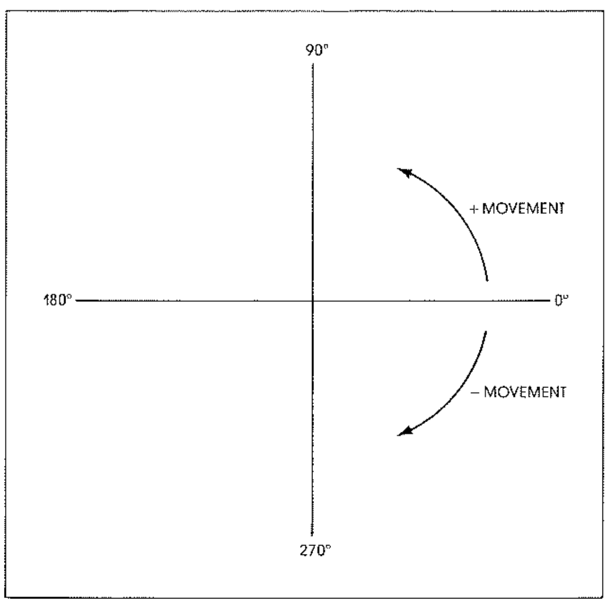


Figure 2. Definition of Angle Movement

Example: Generate the motion sequence to scribe a rectangle of dimensions 10000 x 6000 counts with radiused corners of 500 counts. Define the starting point as shown in Figure 3 and divide the rectangle sequence into 9 segments.

VS	5000; VA 40000	Define vector speed and acceleration
VP	−4500, 0	Linear, Segment #1
CR	500, 270, −90	Circular, Segment #2
VP	−5000, 5500	Linear, Segment #3
CR	500, 180, −90	Circular, Segment #4
VP	4500, 6000	Linear, Segment #5
CR	500, 90, −90	Circular, Segment #6
VP	5000, 500	Linear, Segment #7
CR	500, 0, −90	Circular, Segment #8
VP	0, 0	Linear, Segment #9
BGS		Begin Sequence

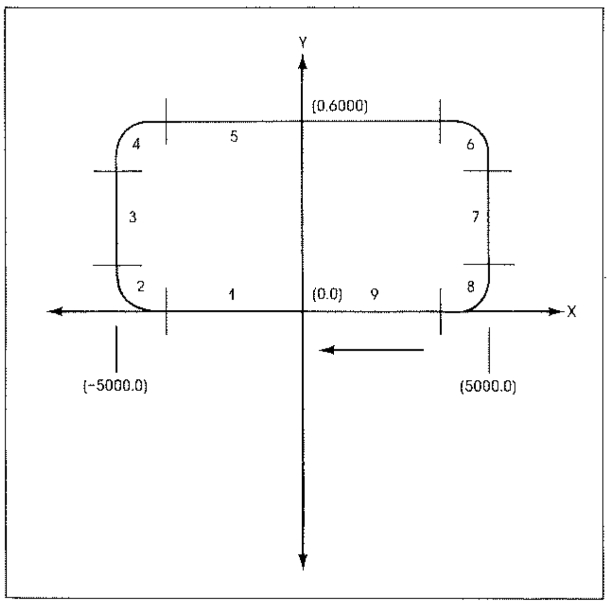


Figure 3. Scribing a Rectangle with Radiused Corners

Contouring

The contouring mode enables the generation of position trajectories of any shape with all the axes. The user describes the required motion trajectories by a sequence of increments of the form:

DX, DY, DZ, DT

Each motion increment is characterized by the relative distance for the active axes and the associated time increment. The controller then performs linear interpolation between the specified points for smooth motion.

The contouring mode is activated with the instruction

CM XYZ

This specifies the contouring axes. Any combination of one, two or three axes is permitted. Non-contouring axes may perform other motions.

The position data is transmitted in the form of data records. The contouring mode is terminated with an end record as shown in Fig. 4.

The form of each data record is shown in Fig. 5. The data record starts with a header consisting of the byte 80H. It is followed by a time increment byte, which defines the time interval DT. The time increment byte defines a number, n, which sets the time interval as:

$$DT = 2^n \text{ milliseconds}$$

As n is allowed to vary between 1 and 8, the range of DT is between 2 and 256 ms.

Following are the position increments for the three axes. Each increment is described by two bytes, where the upper byte is limited to ± 125. This results in position increments in the range

$$-32{,}000 < DX < 32{,}000$$

To end the contouring mode, the user transmits an end record consisting of the byte 80H twice, as shown in Figure 6.

The contouring mode includes a learn mode, where position data is reported to the host in a format which makes it ready for play back. To activate the learn mode, the user commands the instruction

LN n

This selects the time interval and causes the controller to send to the host computer data records in the form shown in Fig. 5. The value of n may be varied while the learn mode takes place. This mode is terminated with the instruction

LN or LN 0

which causes the controller to transmit an end record.

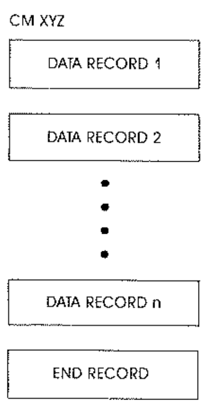


Figure 4. Structure of Information Flow

80H	Header
80H + n	DT
MX	DX
NX	
MY	DY
NY	
MZ	DZ
NZ	

Figure 5. Data Record

80H
80H

Figure 6. End Record

Homing

Using the Home Instruction (HM), the DMC-630 can home each motor to an external home reference signal and an encoder index. Upon the HM and BG command, the motor begins moving at the specified speed until the Home input line changes state. The direction of motion is specified by the initial level of the Home input (Low is forward, high is reverse). At the transition of the Home input, the motor is commanded to stop. Next, the motor slews very slowly back to the Home transition again. At this point, the motor slews forward until the encoder index is detected. The zero position is defined here.

For custom homing applications, a user defined homing routine can be created as part of a program.

Error Handling

The DMC-630 provides several error handling features to prevent system damage. Forward and reverse limit switch inputs prevent motion in the respective direction. An abort input brings all motors to an immediate stop. The DMC-630 has an error output line that goes low when the position error limit specified by the ER instruction is exceeded. This signal can be connected to the system computer or an emergency shut-off line to prevent system damage. There is also an automatic off-on-error shut-off instruction (OE) and automatic error handling subroutines.

The user can interrogate the position error of any axis with the Tell Error (TE) instruction.

Uncommitted I/O

The DMC-600 provides eight uncommitted input lines and eight uncommitted outputs. These may be connected to external signals such as relays, triggers or system timing signals.

The output lines are toggled by the Set Bit (SB) and Clear Bit (CB) instructions. For example, the instruction SB 2 sets output line 2. The OP instruction defines the state of all output lines. The state of the input lines may be checked with the conditional statement, JP, or the After Input command, AI.

For example:

JP #A, I2 = 0	Jump to label A if input 2 is zero
JP #B, I3 = 1	Jump to label B if input 3 is high
AI 1	Wait until input 1 is high

The statement, I0, checks the status of the least significant 4 input lines. For example, I0 = 7 means I4 is zero, I3 is one, I2 is one, I1 is one. The state of the input lines can also be interrogated with the Tell Input (TI) instruction.

The DMC-630 also provides an interrupt for specified inputs. The II command specifies input interrupts. Upon the occurrence of that input, the DMC-630 program sequencer will jump to the subroutine defined by label, #ˆ.

Inputs

Encoder CHA, CHB, Index: Quadrature encoder. Analog or TTL. 500,000 quadrature counts per second maximum speed.

Encoder CHA – , CHB – , Index – : Differential inputs. Optional. Jumpers required.

Forward Limit Switch*: Low input inhibits motion in forward direction.

Reverse Limit Switch*: Low input inhibits motion in reverse direction.

Home: Transition causes motor to stop during Homing sequence.

Abort*: Low input stops motion instantly without controlled deceleration.

Input 1–Input 8: Uncommitted.

Increment: Each rising edge increments the position command counter by one quadrature count. TTL levels.

Decrement: Each rising edge decrements the position command counter by one quadrature count. TTL levels.

Reset*: Low input resets the state of the controller to its power-on condition. TTL levels.

Outputs

Analog Motor Command: ± 10 volt range signal for driving power amplifier.

PWM, sign: Pulse-width-modulated motor command. 20KHz. Optional.

Error*: Goes low when position error on any axis exceeds specified limit. TTL level.

Output 0–Output 7: Uncommitted. Set and cleared with SB and CB instruction. TTL level.

Motion Complete*: Goes low when the DMC-630 processor has completed generating motion profile. Motor may not actually be at position yet.

*Active low