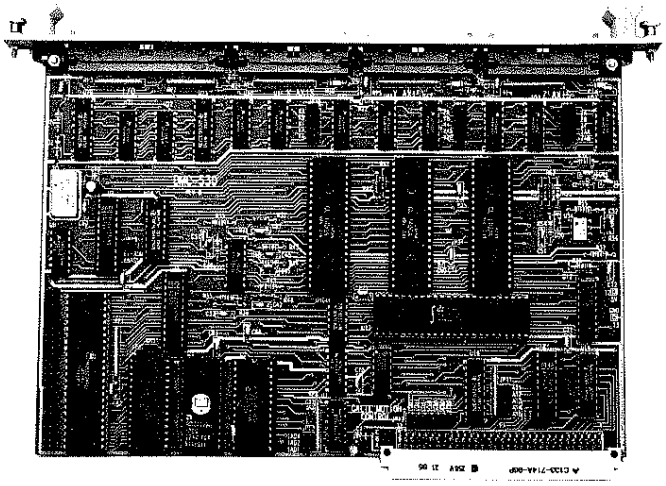
GALIL

DMC/500

VME Advanced Motion Controller

Features

- VME Bus Interface
- Controls motion of 1, 2 or 3 servo motors per card
- Independent or coordinated motion
- Circular and linear interpolation of 2 axes
- Up to 256 segments in a continuous sequence
- Dual-Port communication
- Binary or ASCII mode
- Real-Time Position Access
- Fully Supported VME Interrupts
- User-definable application programs
- 500 lines of on-board program memory
- Conditional statements for controlling program execution
- Programmable time and position trippoints
- 64 variables for entering and changing system parameters
- Arithmetic functions for manipulating parameters
- 8 user-definable inputs and outputs
- Digital filter with gain, damping and integrator
- Error handling, end of travel, Abort, position latch
- 2,000,000 counts/sec maximum speed
- EEPROM for storing controller settings
- Watch-Dog Timer



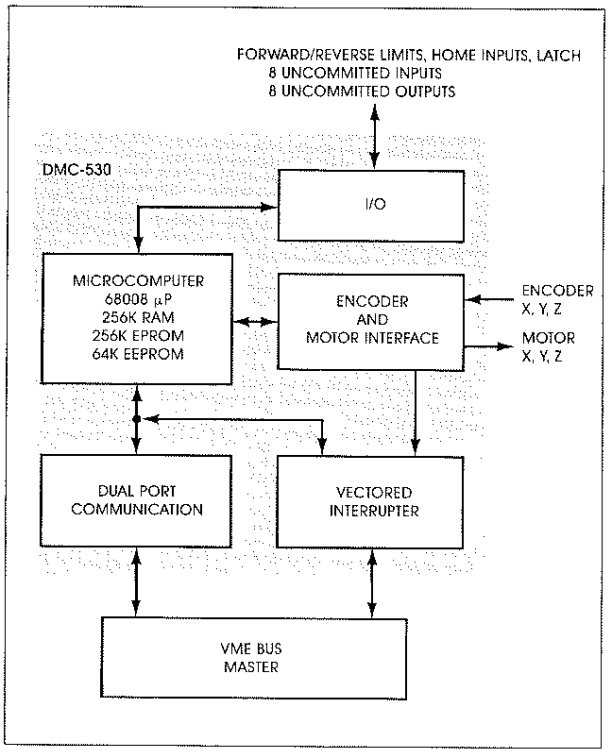


Figure 1. Advanced VME Motion Control System

General Description

The DMC-500 Series is a fully programmable servo motion controller contained on a VME Bus card. It controls the motion of up to three DC or brushless motors with incremental encoder feedback. The DMC-510 controls one motor, the DMC-520 controls two motors, and the DMC-530 controls three servo motors.

Modes of motion include independent positioning and slewing of one, two or three axes, coordinated positioning of two axes, three axis contouring and homing. The motion profiles for each axis 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

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-500 via the Dual-Port RAM. Programs can also be downloaded into the DMC-500 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-500 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. A watch-dog timer monitors proper microprocessor operation. The DMC-530 also contains 8 uncommitted input lines and 8 output lines.

System Elements

The elements of the DMC-530 servo system are shown in Fig. 1. The elements include the VME Host, DMC-530 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-930 interconnect board for each axis.

VME Host—Sends high level commands to the DMC-530. Use of the host computer may be minimized by storing motion programs in the DMC-530 memory. Any VME CPU board will be able to communicate with the DMC-530.

DMC-530—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-530 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-530 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-530 to the amplifier varies between −10 and +10 volts. The DMC-530 also provides a pulse-width-modulated (PWM) output for switching power transistors directly.

Encoder—The encoder translates motor motion into an electrical signal which is decoded by the DMC-530 as the motor position. Each axis requires feedback from an incremental encoder with two channels in quadrature, CH A and CH B, which gives 4x the number of encoder cycles. 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-530 also accepts an encoder index signal which is useful for referencing the encoders during the Home instruction.

External Inputs—Limit switches and emergency stop inputs may be connected to the DMC-530 to prevent system damage. There are also uncommitted inputs and outputs the user can define.

Communications Interface

The DMC-500 is configured for the VME short I/O space. All data transfers are 1 byte wide and go through the 1K dual-port RAM.

The dual-port RAM is divided into buffers for sending commands, application programs, contour commands and receiving status information. Handshake registers control communication timing.

In the dual-port RAM, information such as variables, axis positions, torque and the axis status are always available.

Programming the DMC-530

Instruction Set

The DMC-530 contains an extensive instruction set for programming a variety of motion sequences. Instructions may be sent in ASCII or binary format. An ASCII command is represented by two characters followed by its parameters. Binary commands are fixed format with a command number followed by numeric fields for each axis.

For example, in ASCII format:

PR 4000,9000, −1000;

PR is the Position Relative command, 4000,9000, −1000 is the X, Y, Z, positions, respectively. The ; terminates the command.

In binary, the equivalent command is:

C9 07 00 00 0F A0 00 00 23 28 FF FF FC 18

where C9 is the Position Relative command code, 07 represents the format and 00000FA0 00002328 FFFFFC18 represents the X, Y and Z positions.

A complete listing of the DMC-530 instructions is given in Table 1.

Motion Programs

Instructions can be combined to form motion programs. Instructions can be sent from the VME host as they are executed or they may be downloaded into the DMC-530 memory by writing to the Pro-

gram Buffer. The DMC-530 memory stores up to 500 lines of 32 characters per line. 190 lines may be stored in non-volatile memory. Multiple commands may be stored on a line. Separate programs or subroutines are distinguished by labels.

Programs are entered and edited using the DMC-530 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.

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-530 provides 64 variables specified by the command V0 through V63 in a program, or variables are specified in the variable buffer. 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.

The variable format is 3 bytes of integer and one byte of fraction (± 8388607 with fraction of 1/256).

Modes of Motion

The DMC-530 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-530 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-530 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 degrees and moving in a positive direction of 90 degrees.

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.

Example: Generate the motion sequence to scribe a rectangle of dimensions 10000x6000 counts with radiused corners of 500 counts. Define the starting point as shown in Figure 2 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
```

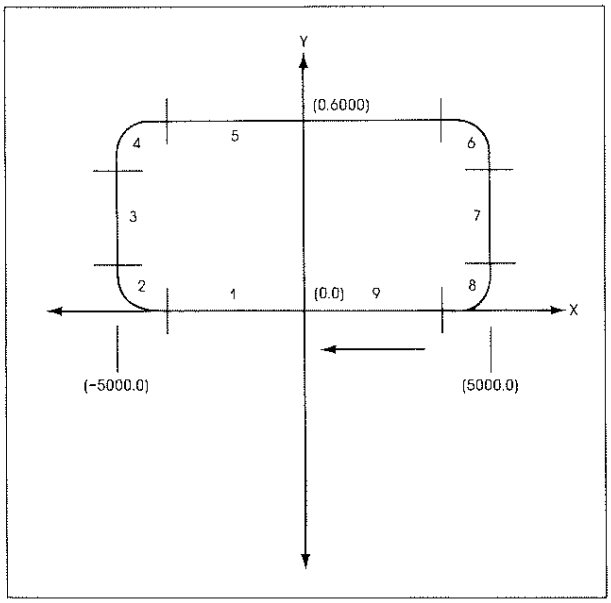


Figure 2. Scribing a Rectangle with Radiused Corners

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 in the form:

DX, DY, DZ, DT

Each motion increment is characterized by the relative distance for the active axes and the associated time increment in the range of 2-256 msec. The controller then performs linear interpolation between the specified points for smooth motion.

Homing

Using the Home Instruction (HM), the DMC-530 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-530 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-530 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 instruction (OE) and automatic error handling subroutines. A watch-dog timer output will toggle if the microprocessor ceases to function properly or loses power. Any of the error functions can be configured to generate a VME interrupt.

Uncommitted I/O

The DMC-500 provides eight TTL uncommitted inputs 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 DMC-530 also provides an interrupt for specified inputs. The II command specifies input interrupts. Upon the occurrence of that input, the DMC-530 program sequencer will jump to the subroutine defined by label #^.