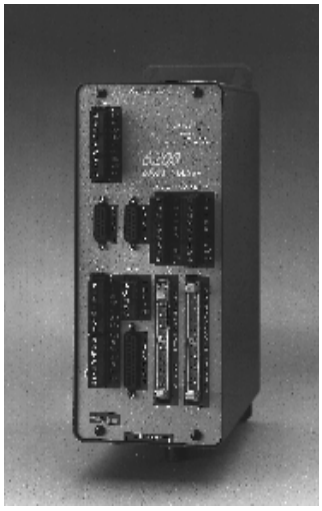
6000 Series . . .

A powerful lineup of controllers and software

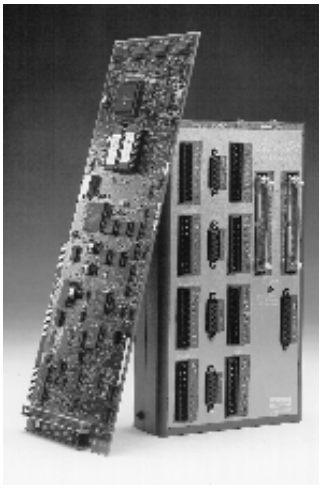
There are four 6000 Series controllers (indexers) that provide a ±10V analog voltage output. All units share the following features:

- Powerful 6000 programming language
- Up to 28 programmable inputs and 26 programmable outputs
- Fast inputs for position capture registration, etc.
- Analog inputs for joystick control and variable input
- End-of-travel and home limit inputs for each axis
- Encoder feedback
- Software provided
  - Motion Architect
  - DOS®-based 6000 Series disk
  - Dynamic Link Library (DLL) for use with Microsoft® Windows™ and Microsoft Windows NT application development kits
- Optional software
  - CAD-to-Motion (CompuCAM)
  - Motion VIs for LabVIEW® (Motion Toolbox)
  - Dynamic Data Exchange (DDE) servers
  - Graphical icon-based software (Motion Builder)
  - OCX Custom Control Interface (Motion OCX Toolkit)



6200

- One or two axes of step & direction control with encoder feedback
- Operates standalone or interfaces with PCs, PLCs, and thumbwheels
- Two RS-232C communication ports
- 150,000 bytes of non-volatile memory for program path and storage
- Complete specifications begin on C109



AT6200/AT6400

- PC/AT®-based indexer
- One to four axes of step & direction control with encoder feedback
- Separate adaptor board to simplify connection and optically isolate I/O
- 2 axes of circular interpolation and 4 axes of linear interpolation
- 1,500,000 bytes of RAM for program path and storage
- Complete specifications begin on C113

Software and hardware options completes the system . . .

For more information on the support software available for the 6000 Series indexers, see the following:

Dynamic Data Exchange (DDE) Server

The DDE6000 is a software driver that enables communications between Microsoft Windows applications that support the DDE protocol and our 6000 Series products.

The DDE6000 supports NetDDE, allowing any computer on any Windows for Workgroups™ or NT Server Network, which supports NetDDE to control any 6000 Series product connected to a networked computer.

Compumotor's DDE6000 not only supports all 6000 Series products, but also supports multiple 6000 Series products simultaneously.

DDE is a standard Microsoft protocol that automatically updates data to and from Windows applications. The DDE servers provide the 6000 Series with fast status information to a Windows application that wants it. Information that can be provided by the DDE server and the corresponding equivalent 6000 command is listed in the table:

Status Report (Status Command)

Motor Positions (TPM)	Time Frame Counter (TFC)
Encoder Positions (TPE)	Inputs (TIN)
Motor Velocities (TVEL)	Outputs (TOUT)
Axis Status (TAS)	CW Limits (TLIM)
System Status (TSS)	CCW Limits (TLIM)
Interrupt Status (TINT)	Home Limits (TLIM)
User Status (TUS)	Other Inputs (TINO)
Timer (TTIM)	Analog Inputs (TANV)
Variables (VAR)	Hi-Res Analog (TANI)
Following Status (FS)	Master Cycle Position (PMAS)
Phase Shift (PSHF)	Master Cycle Number (NMCY)
Master Velocity (VMAS)	Extra Encoder Position (TPE)
6000 Response	

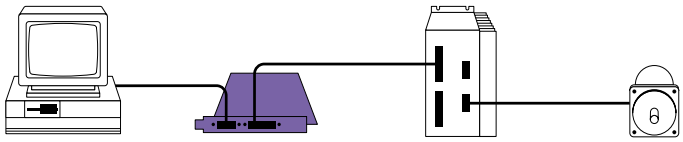
These DDE6000 servers allow our products to communicate easily with a wide variety of applications, including several factory automation or man-machine interface software packages. Listed here are some of the most popular products that have DDE support and the companies who produce them. The list to the left is a small subset of applications that support DDE.

Company	Product(s)
East Point Software Corporation	IPC-Xpert
Iconics	WinWorx
ICCOM, Inc.	WinView Wintrend LinLinx WinLogic 5
LabTech	Real-Time Windows Real-Time Vision
Intellution	FIX DMACS™
National Instruments	LabVIEW®
TA Engineering	AIMAX®-WIN
Canary Labs, Inc.	Trend Link
Wonderware Software Development Corp	InTouch
United States Data Corporation	FactoryLink IV

FIX DMACS™ is a registered trademart of Intellution.

AIMAX®-WIN is a registered trademark of TA Engineering.

All others are trademarks of their respective companies.



Motion Architect does the work for you . . . configure, diagnose, debug

Motion Architect has been designed for use with all 6000 Series products—for both servo and stepper technologies. The versatility of Microsoft Windows and the 6000 Series language allow you to solve applications ranging from the very simple to the complex.

Motion Architect comes standard with each of the 6000 Series products and is a tool that makes using these controllers even simpler—consid-erably shortening the project development

time.

Using Motion Architect, you can open multiple windows at once. For example, both the Program Editor and Terminal Emulator windows can be opened to run the program, get information, and then make changes to the program.

On-line help is available throughout Motion Architect, including interactive access to the contents of the Compumotor 6000 Series Software Reference Guide.

Standardizing your motion control software requirements around Motion Architect protects your software investment.

- The code is transferable to any 6000 Series indexer or servo controller
- Reduce your learning curve and development
- Reduce maintenance and support training
- This is the platform for future products and enhancements

Motion Architect requires:

- Microsoft Windows, release 3.1 or later or Windows NT. Motion Architect automatically detects the operating system on your computer and installs the proper drivers.
- At least 2 MB of RAM
- At least 3 MB of hard disk space

This does not include the memory requirements for additional add-on modules.

The heart of Motion Architect is the shell

The shell provides an integrated environment to access four main modules. These modules consist of:

- **System Configurator and Code Generator (Setup):** Automatically generate controller code of basic system setup parameters (I/O definitions, encoder operations, drive setup, etc.), based on answers you give to dialog boxes.
- **Program Editor (Editor):** Create blocks or lines of 6000 Series controller code, or copy portions of code from a previous file. You can save Editor files for later use in a high-level program (e.g., BASIC, C, etc.), or in the Terminal or Panel modules.
- **Terminal Emulator (Terminal):** Providing communication directly with the 6000 Series product, the terminal emulator allows you to type in and execute controller code and transfer code files to and from the controller. Owners of 6000 Series bus-based controllers can transfer (download) the soft operating system.
- **Test Panel and Program Tester (Panel):** Create your own test panel to exercise your programs and check the activity of I/O, motion, system status indicators, timers and counters, and the communications interface.

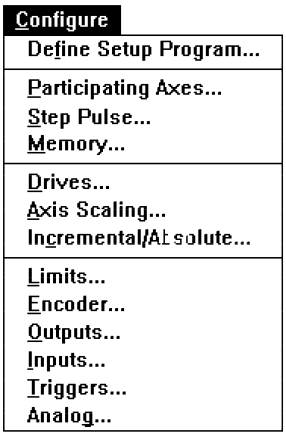
In addition to the functions of the preceding modules, Motion Architect gives you these on-line resources (see **Help** pull-down menu):

- **Context-Sensitive Help:** Access via the Help pull-down menu, the Help buttons in the dialog boxes, or by pressing the F1 key at any time. This resource provides comprehensive help information about the modules.
- **On-Line User Guides:**
  - *6000 Series Following User Guide*
  - *6000 Series Software Reference Guide*

The modules listed below are sold separately as add-on utilities to Motion Architect. After they are installed, you can access them from the utilities menu. To purchase one of these options, contact your local Automation Technology Center.

- **CompuCAM™:** CAD-to-Motion (CAM) software allows you to translate DXF, HP-GL, and G-code files into 6000 Series language motion programs.

Motion Architect . . . system configurator and code generater

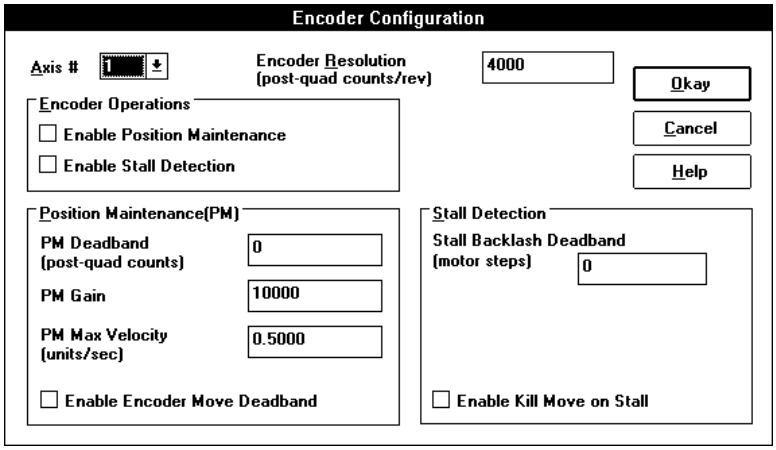


Setup

The Setup module takes much of the drudgery out of motion control programming. This module automatically generates configuration code in response to your answers to a series of dialogs concerning I/O definitions, encoder operations, homing operations, end-of-travel limits, drive setup, joystick setup, execution modes, etc.

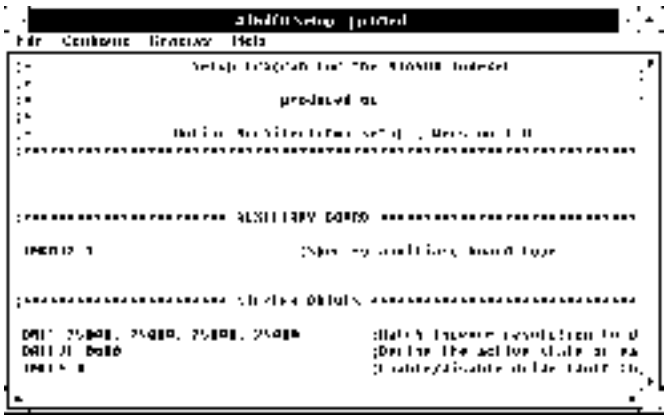
Establishing system set-up parameters (Configure)

The setup module includes the **Configure** and **Generate** menu items. Under the **Configure** pull-down menu are the configuration parameters to consider for your application. When you select one of these parameters, a dialog box appears. Shown here is an example of the encoder configuration dialog box. Note that as you complete the configuration of the parameters, a check mark (✓) appears next to the menu item.



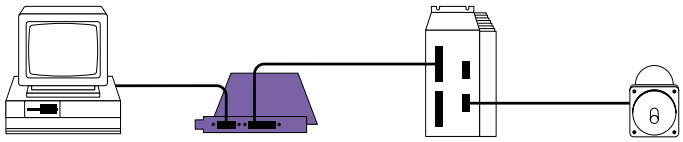
Generating controller code for system set-up parameters (Generate)

After you finish answering the dialog boxes under the **Configure** menu, click the **Generate** menu to view the resultant controller code. A program (.prg) file is automatically saved to store the newly generated code. You can later open this .prg file in the **Editor** and use it as a building block for a set-up program.



Controller set-up code is automatically generated, based on your answers to the confuration dialog boxes.

C Step Motor Systems

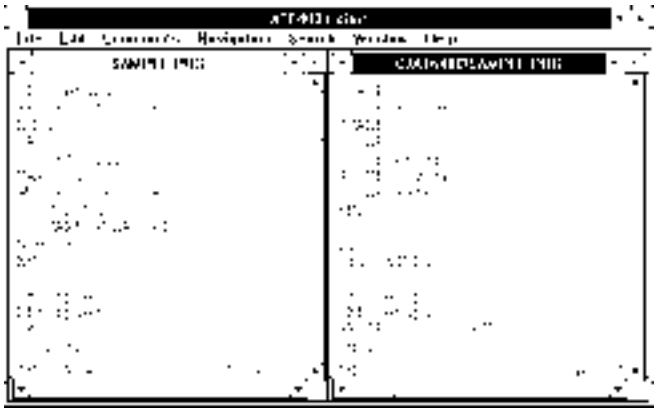


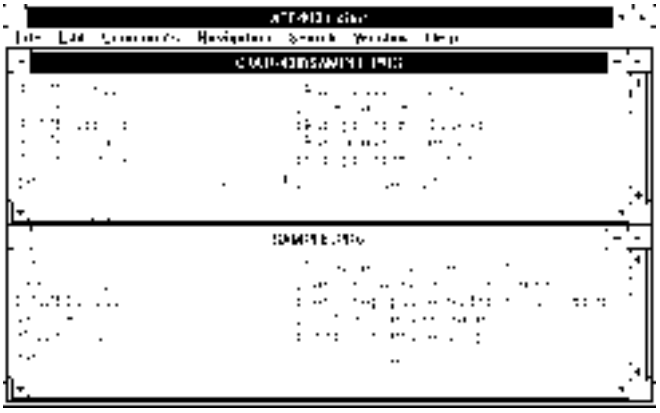
Program Editor

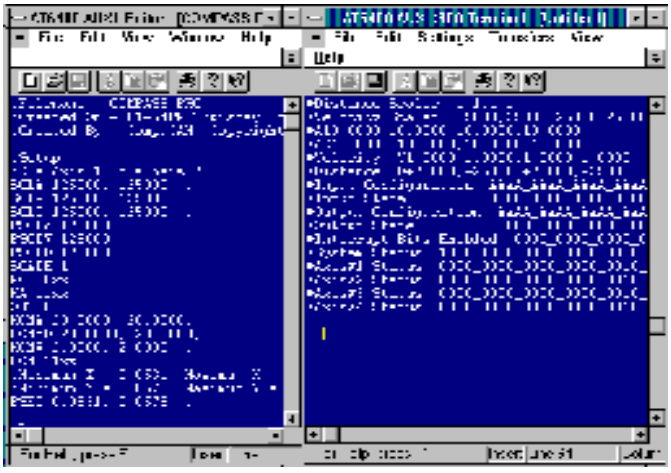
The **Editor** module is a tool you can use to create and edit controller programs using the 6000 Series programming language. You can then save these programs as separate files for later use in high-level programs (e.g., BASIC, C, etc.) or in the Terminal and Panel modules.

After you have created or edited a controller code file in the Editor, you can save it as a separate file. This file can then be used in the following ways:

- Before running the program in an application, you can test-run the file from the Panel module.
- From within the Editor module, you can open any program file and print its contents.
- From within the Editor module, you can copy the contents (part or all) of the file and then paste it into the Terminal module for immediate execution.
- Within the Terminal module, download the file that was created in the Editor module to the 6000 Series product for immediate execution.
- Use the file's controller code in a high-level program (e.g., BASIC, C, etc.).







Terminal Module

Communicating directly with the 6000 Series product, the **Terminal** module allows you to type in and execute controller code and transfer code files to and from the controller. From the Terminal module, AT6200, AT6400, AT6250 and AT6450 users can transfer (download) the soft operating system.

Parker Automation

Parker Hannifin Corporation
Compumotor Division
Rohnert Park, California

Test Panel

The **Panel** module allows you to test your controller programs with your own test panel. You can customize the panel with multiple windows (monitoring controller output) and programmable buttons (for user input).

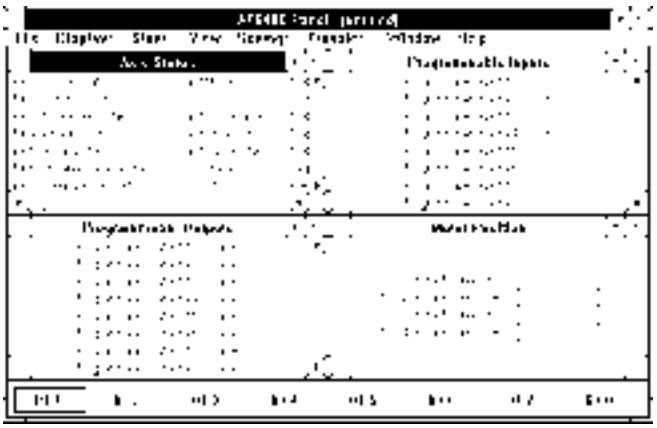
One or more windows can be opened to view the following controller information:

- I/O (programmable I/O, analog I/O, limits)
- Motion (motor and encoder position, velocity)
- Status (axis, system, interrupt, user status, and following)
- Miscellaneous (timer and counter values)
- Communications interface

Four tiers of 8 user-definable buttons are available which allow you to send controller code fragments (up to 256 characters each) to a 6000 Series product.

You can download controller programs to the 6000 Series product via the file-transfer **(Transfer)** menu, and execute them using programmable buttons or external I/O.

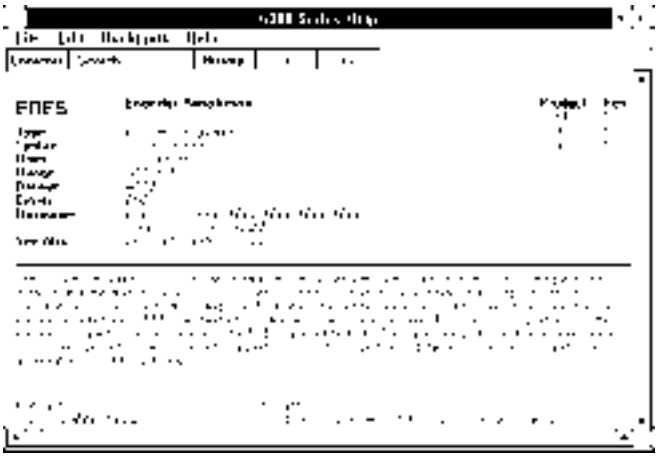
Additionally, you can create as many test panel configurations as you wish and save them as separate files.



On-line Help

From the menu bar in the Shell or from any module, you can access the on-line help system by clicking on Help, or by pressing the F1 key when a menu item is selected. The Help pull-down menu items are briefly described here.

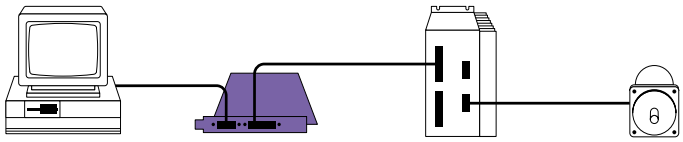
- **Help for help.** The same help resource for all modules, describes how to use the on-line help system's features.
- **Keys help.** Defines the shortcut (hot) keys you can use within a particular module to help you select certain commands or menu items faster.
- **Help Index.** Provides information on how to use the particular module from which you accessed the on-line help system. The information is categorized by the items that appear in the module's menu bar.
- **6000 Commands.** Displays the Command Dialog Box, from which you can look up commands, edit them, and insert them into an active program editor session or terminal emulator session.
- **6000 Software Reference.** Brings you directly to the Contents menu of the on-line version of the *6000 Series Software Reference Guide.* This is a valuable resource for detailed command descriptions and programming guidelines.



- **6000 Following Reference.** Brings you directly to the Contents menu of the on-line version of the *6000 Series Following User Guide.*
- **About.** Displays revision and copyright information.

Parker Automation

Parker Hannifin Corporation
Compumotor Division
Rohnert Park, California



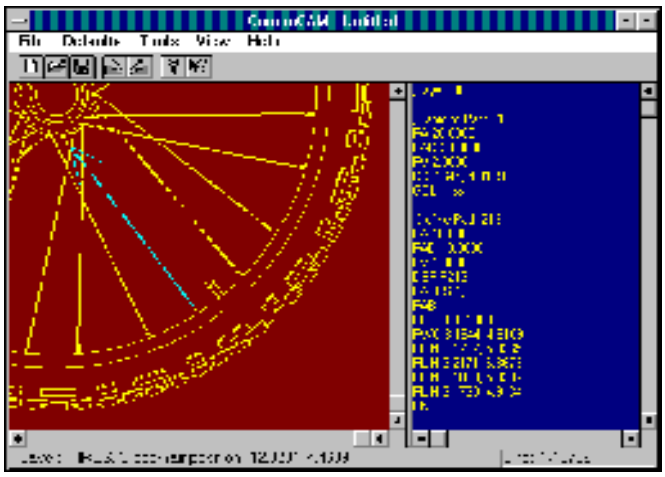
CompuCAM

CompuCAM is a Microsoft Windows-based programming package that imports geometry from CAD programs, plotter files, or NC programs and generates 6000 code compatible with Compumotor's 6000 Series motion controllers.

From CompuCAM, run your CAD software package. Once a drawing is created, save it as either a DXF file, HP-GL plot file, or G-code NC program. This geometry file is then imported into CompuCAM where the 6000 code is generated. After generating the program, you may use Motion Architect functions such as editing or downloading the code for execution.

Complex moves can be programmed without any 6000 Series coding experience. Simply invoke CompuCAM and import DXF, G-code, or HP-GL files. All set-up can be accomplished within Motion Architect and CompuCAM's menu-driven format.

CompuCAM runs on an IBM PC/AT, or compatible, with 2M RAM, 1M hard disk space, and a VGA color monitor. Microsoft Windows 3.1 (or greater) and Compumotor's Motion Architect 2.2 (or greater) are required. A math coprocessor is highly recommended.



CompuCAM Features

- CompuCAM's three versions import and translate shapes defined in DXF, HP-GL, or G-code
- Compatible with most off-the-shelf CAD packages that output DXF, HP-GL, or G-code files (e.g., AutoSketch)
- Supports:
  - Point-to-point positioning
  - 2-axis circular interpolation (when available)
  - Multi-axis linear interpolation
- CompuCAM's DXF version translates circles and polygons
- Allows imported line and arc segments to be ordered for sequential execution
- Operations such as turning on and off an output can be specified at the beginning and end point of each motion path
- Ability to step through the program to verify motion
- Customized M and H codes can be used to increase CompuCAM's functionality with G-code translation
- Utilizing the menu bar, the user may:
  - Ascribe motion and I/O to the imported geometry
  - Save work
  - Retrieve previous sessions
  - Zoom/pan geometry
  - Access Help system
  - Sequentially order the execution of motion paths
- Compatible with Compumotor's 6000 Series controllers
- Two-way splitter window provides simultaneous views of the imported geometry and the generated 6000 code
- The Motion Architect editor can edit 6000 code generated by CompuCAM
- The Motion Architect terminal emulator can download the generated code to a 6000 Series controller
- CompuCAM offers two alternative code generation schemes: Define, Run, Delete or Define All, Run All

Whether you are outputting geometry from a CAD program as a DXF file, or programming in G-code, CompuCAM has a version for you. Specific information regarding each CompuCAM version follows.

DXF Import Filter

The DXF import filter will handle the following geometric entities:

- Lines
- Arcs
- Circles
- Polygons
- Polylines representing complex geometry—i.e., ellipses, splines, bezier curves, etc.

The DXF filter will only import curve geometry. This filter does not handle points and text.

HP-GL Import Filter

The HP-GL import filter will handle the following subset of HP-GL:

AA	Arc Absolute
AR	Arc Relative
CI	Circle
PA	Plot Absolute
PD	Pen Down
PR	Plot Relative
PU	Pen Up
IP	Input P1 and P2
SC	Scale

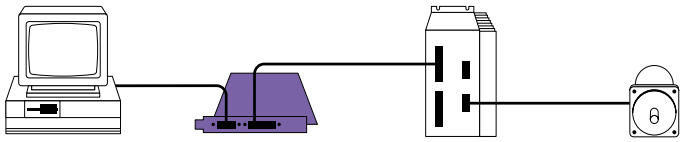
The HP-GL import filter does not handle instructions from the polygon group, the line and fill attributes group, the character group, or any of the advanced extensions.

G-Code Import Filter

The G-Code import filter will handle the following subset of EIA RS-274-D G-Codes:

F Codes	Feedrate
G Codes	
G00	Rapid traverse positioning mode (Default)
G01	Linear interpolation mode
G02	CW contouring mode
G03	CCW contouring mode
G04	Time delay
G28	Move to home switch input
G61	Turn off continuous path motion
G64	Turn on continuous path motion (Default)
G79	Execute a subroutine prior to each subsequent move
G80	Cancel G79 mode (Default)
G90	Absolute positioning mode
G91	Incremental positioning mode
G92	Set current position
H Codes	Add offset to specified axis
I Codes	1st axis incremental distance to center point
J Codes	2nd axis incremental distance to center point
L Codes	Number of repeats for a subroutine call
M Codes	
M01	End of program
M06	Same as M01
M98	Execute a gosub
M99	Return from a subroutine
Mxx	M codes (other than those listed) are customizable for I/O control
N Codes	Optical line number
O Codes	Begin definition of a subroutine
P Codes	Time delay or subroutine definition number
R Codes	Radius of an arc
X Codes	Commands a move on the X-axis
Y Codes	Commands a move on the Y-axis
Z Codes	Commands a move on the Z-axis
“ ” Codes	All characters between “ ” are sent directly to 6000 product
( ) Codes	All characters between ( ) are ignored

The G-Code filter will import geometry from NC programs that do interpolated motion as well as XY motion with an unrelated Z motion. The XYZ interpolated motion includes linear XYZ motion and arcs in the XY plane. It does not include arcs in the XZ or YZ planes.



Motion Toolbox enhances LabVIEW for 6000 Series programming

Create stunning user interface using controls and indicators built into LabVIEW.

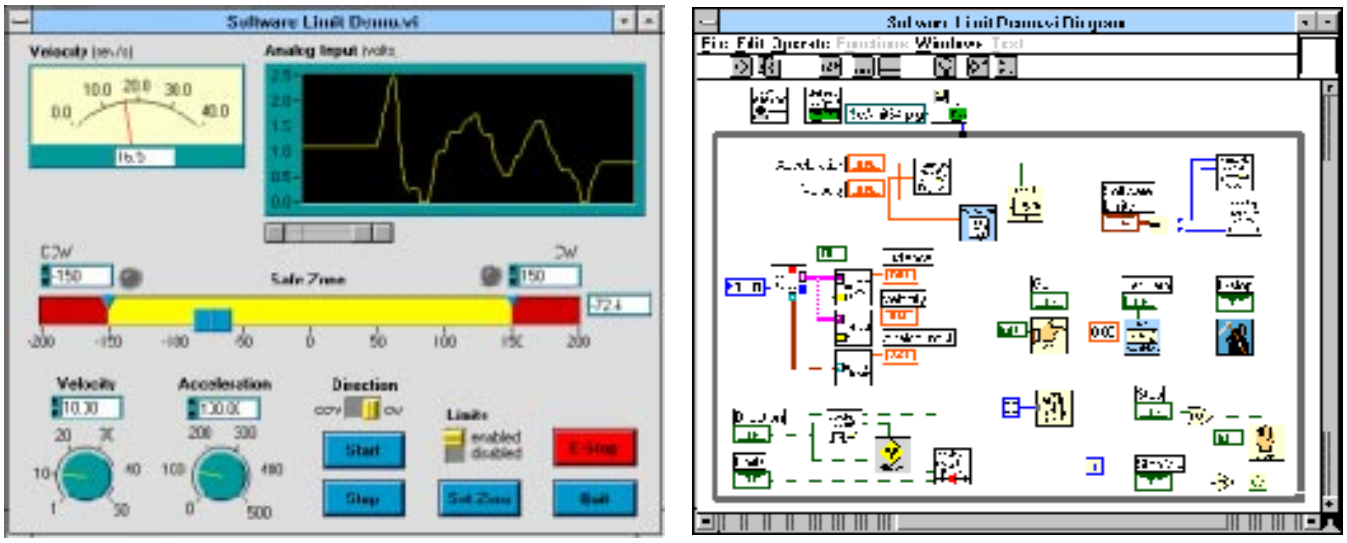
Motion Toolbox is an extensive software library of LabVIEW virtual instruments (VIs) for icon-based programming of Compumotor's 6000 Series motion controllers.

When using Motion Toolbox with LabVIEW, applications are developed by linking graphic icons (VIs) together to form a block diagram. Motion Toolbox's library of more than 150 command, status and example VIs, accentuates the power of the 6000 Series including VIs for the following capabilities and more:

- Downloading setup and control programs to the 6000 controller's memory for later execution
- Motion control (Go, Stop, velocity, acceleration, etc.)
- Input/output setup and function configuration
- Home configuration and control
- Hardware limit and soft limit configuration
- Indexer configuration of jogging, joystick, limits, encoders, drives, etc.
- Fast status querying of I/O, limit, home, motor and encoder position, velocity, etc.
- Debugging capabilities include command "snooping" where developers can view commands sent to the 6000 Controller and communications tracing where command and query information is streamed to disk

All command and status VIs include LabVIEW source diagrams so you can modify them, if necessary, to suit your particular needs. Motion Toolbox also comes with a Windows-based installer and a comprehensive user manual to help you get up and running quickly.

Motion Toolbox: The definitive answer in motion control programming.



A LabVIEW user-interface and an associated block diagram showing how Motion Toolbox VIs are used to control the 6000 series.

Parker Automation

Parker Hannifin Corporation
Compumotor Division
Rohnert Park, California

The LabVIEW Graphical Environment

LabVIEW is a graphical programming environment for data acquisition, data analysis, and instrument control that runs on several platforms including Windows-based PCs, Sun SPARC stations, and HP computers. This programming environment allows you to rapidly build, test, and modify application programs without the syntactical knowledge of a conventional programming language.

LabVIEW programmers can use Motion Toolbox to develop motion control systems for a wide range of applications including automated test and manufacturing, medical, biotech, metering and dispensing, machine control, and laboratory automation. Using LabVIEW in conjunction with Motion Toolbox and the 6000 Series allows you to integrate motion control into data acquisition, process control, and image processing systems. LabVIEW includes knobs, slides, switches, charts, and many other controls and indicators allowing programmers to quickly create custom user-interface panels for motion control applications.

Capture the Power of Icon-Based Motion Control Programming

The combination of LabVIEW and Motion Toolbox simplifies the development of any application that requires motion control. Motion Toolbox can directly control motion or act in a supervisory capacity. In the supervisory mode, complex motion programs and paths can be downloaded to the 6000 controller and be orchestrated by the Motion Toolbox application. The supervisory approach provides parallel execution and performance advantages required by demanding motion control applications.

- Develop applications faster with less debug time
- Reduce operator training by providing intuitive user interfaces.
- Maximize your development time by reusing the code from project to project.
- Reduce application development by leveraging off powerful Motion VIs designed to take full advantage of the 6000 Series products
- Utilize rapid prototype techniques

Motion Toolbox was developed by Snider Integration Group through an alliance with Parker Compumotor. Snider Integration Group is an NI Alliance member specializing in the development of LabVIEW-based systems.

Partial List of VIs

Counters and Timers
- Start 6000 Timer
- Stop 6000 Timer
- Reset Hardware Counter

Configuration
- Motion Scaling
- Path Scaling
- Participating Axes
- Set Continuous/Preset Mode
- Set Absolute/Incremental Mode
- Enable Drive
- Set Drive Resolution
- Configure Feedrate Override
- Set Encoder/Motor Step Mode

Device Communication
- Set Default Address/Port
- Set Error Action
- Send 6000 Block
- Receive 6000 Block
- Download 6000 File

Fast Status
- Motor Position
- Encoder Position
- Motor Velocity
- Captured Position Status
- Command Error
- Limit Status
- Motion Status
- System Status
- Analog Input Status
- Joystick Status
- Digital Output Status
- Digital Input Status

I/O & Limits
- Set 6000 Input Active Level
- Set 6000 Output States
- Set 6000 Output Active Level
- Configure Limits

Joystick & Jogging
- Set Jog Velocity
- Set Jog Acceleration
- Set Joystick Acceleration
- Set Joystick Velocity

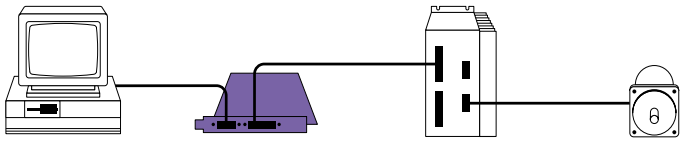
Motion
- Initiate Motion
- Stop Motion
- Kill Motion
- Set Distance
- Set Velocity
- Set Acceleration
- Set Deceleration
- Set Direction
- Set Position
- Initiate Lin. Interpolated Motion
- Run Path
- Set Path Velocity/ Acceleration
- Wait for Move
- Run Program
- Go Home

RP240 Display
- Write Text to RP240
- Display Variable on RP240
- Clear RP240 Display
- Set RP240 LED States

Variable & Transfer
- Set Numeric Variable
- Set String Variable
- Transfer Numeric Variable
- Transfer String Variable

6000 Series Controllers

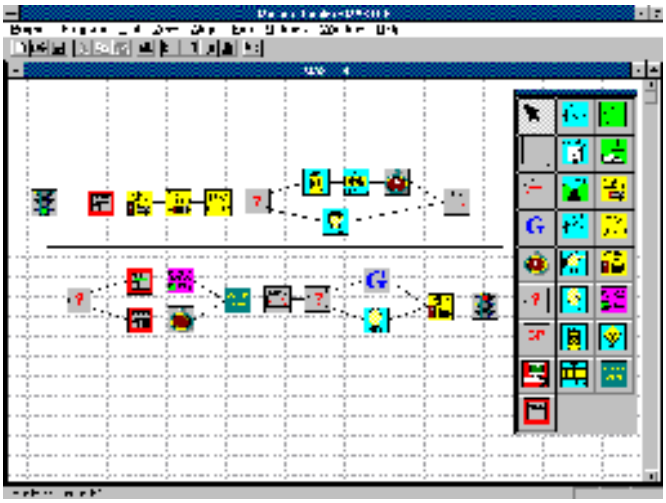
- Single and multiple axes of control
- Stepper, servo and hydraulic control
- ISA bus-based, stand-alone, and packaged drive/ controller systems



Motion Builder: Making Motion Control Programming Easy

Motion Builder revolutionizes motion control programming by providing a software tool which decreases the learning curve of motion control programming. Motion Builder allows users to design and program their motion control application in a way they are already familiar with—a flow chart style method. This innovative tool shortens your system design time and reduces your overall system cost by decreasing programming and troubleshooting cost.

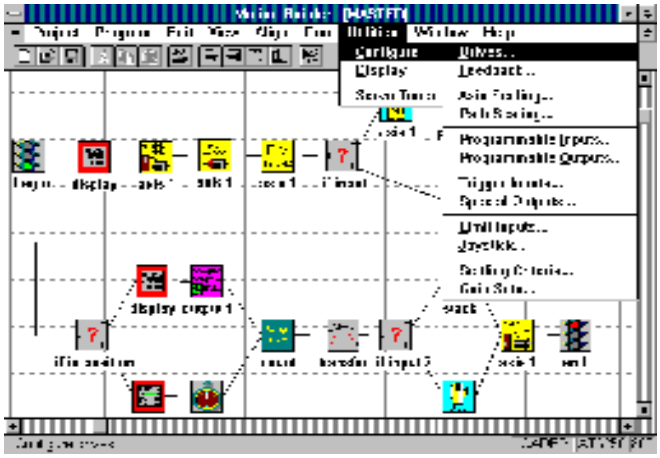
Motion Builder, a Microsoft Windows-based graphical-development environment, allows expert and novice programmers to easily program Compumotor's 6000 Series products without learning a new programming language or syntax. Visual icons, representing the motion function you want to perform, are simply dragged and dropped.



- Graphical motion–development environment
- Visual programming
- "Fill-in-the-form" process details
- Motion icons with label and comment fields
- Real-time display environment
- Debug environment
- On-line Help system

Motion Builder provides a user-friendly graphical environment for designing motion control applications. Visual icons representing standard motion and process control operations are selected from a floating palette. In addition to visually programming Compumotor's 6000 Series products, it can completely configure the motion controller; compile, run and debug the program—all from the same tool!

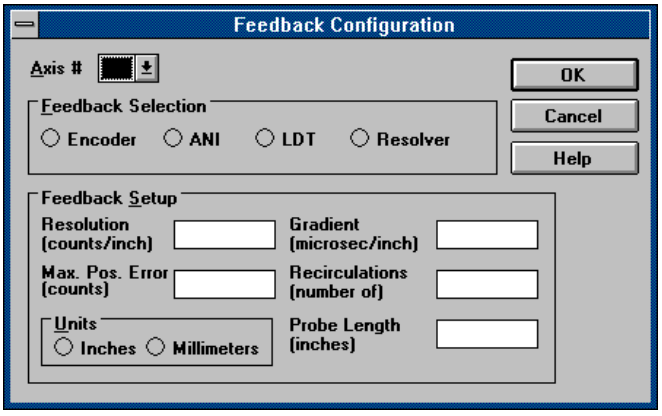
You simply select the 6000 Series product you are using and Motion Builder then configures the on-line help for steppers and servos from one to four axes, communication protocol for serial or PC ISA bus communication, and graphical icons to reflect your product.



Application parameters are specified by answering a series of questions contained within a series of drop-down dialog boxes. The topics for configuration include:

- Drive and feedback type
- Axis and path scaling
- Programmable inputs and outputs
- Home and end-of-travel limits
- Joystick and analog I/O
- Servo gain sets
- Settling criteria for move complete
- Performance criteria for steppers

When you select one of these parameters, a dialog box appears. Shown here is an example of the feedback configuration dialog box.

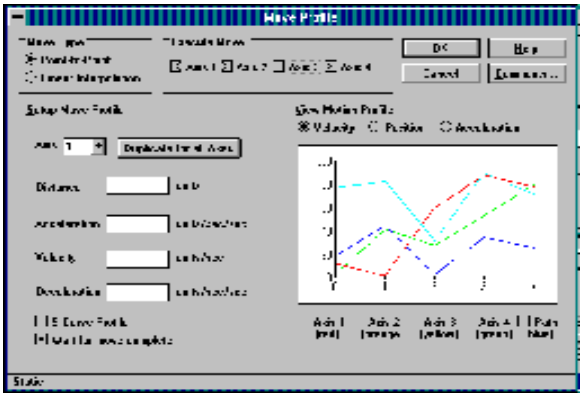


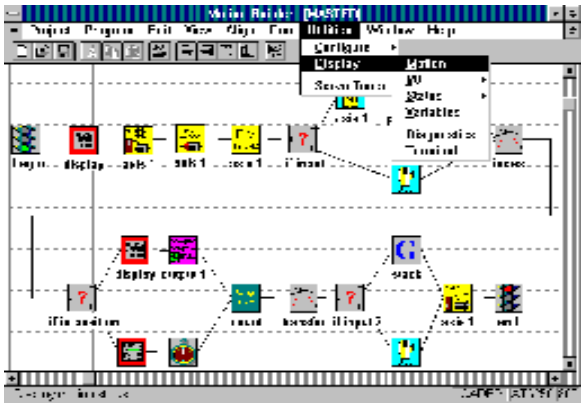
Parker Automation
Parker Hannifin Corporation
Compumotor Division
Rohnert Park, California

Programming is accomplished by selecting the motion or process functions from the floating palette and dropping them, one by one, onto the chart. Each function's action is set up by a Windows dialog box associated with the icon.

For example, double clicking on the Move icon brings up the dialog box where values for move parameters are entered.

When all functions are on the screen, a special tool connects the icons in a manner indicating the program flow. When the program diagram is complete, the program is compiled and can be executed.





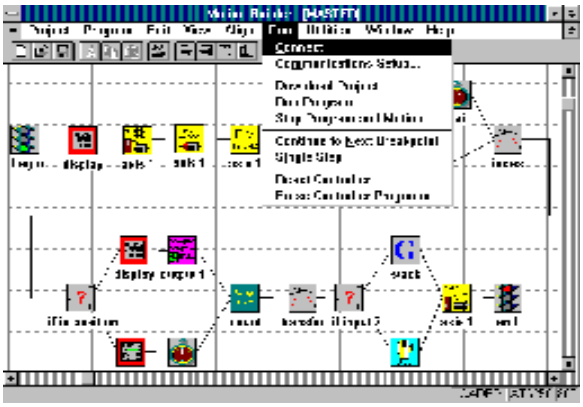
You can even display real-time controller information while your program is running:

- Motion (motor and encoder position, velocity)
- I/O (programmable I/O, analog I/O, limits)
- Status (axis, system)
- Variables
- Diagnostics
- Terminal (communications interface)

A debug environment is also included in Motion Builder. The user can set program break points by selecting the break point icon from the palette and dropping it into the area (beginning and end) of the program that needs to be debugged. Another debugging aid is the single-step mode. This allows the user to control and step through the program, executing the icons one at a time.

An on-line help system is also included in Motion Builder. The on-line help system includes:

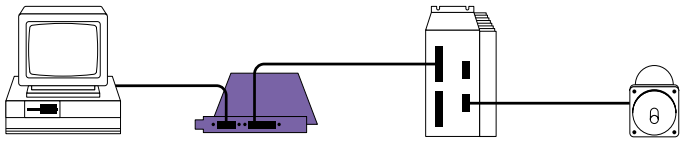
- Context sensitive help
- Help index with search
- Tooltip help
- Status bar help



Benefits

- Easy programming
- Shorten your system design time
- Cut your cost
- Eliminate syntax errors

C Step Motor Systems



Compumotor's new Motion OCX Toolkit

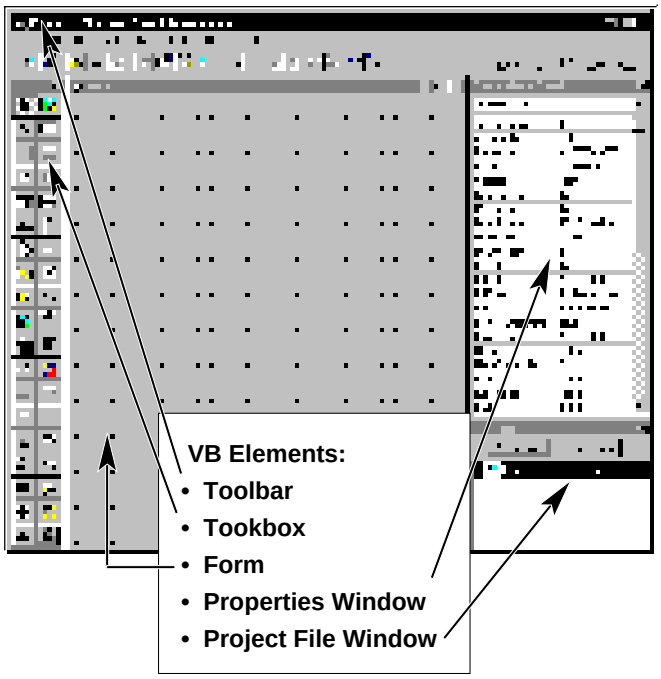
Compumotor's new Motion OCX Toolkit is a powerful standalone software package that allows the programmer to easily and effectively develop virtual instrument interfaces for the AT6000 Series Bus-based controllers (i.e., AT6200/AT6400 and AT6250/AT6450). The Motion OCX Toolkit provides programmers with an easy way to incorporate advanced functionality into their applications, with little or no programming. The Motion OCX Toolkit includes three OCX custom controls, which are loaded directly into the development environment's toolbox palette. With this application development tool, an operator interface can be built by simply clicking on the palette and placing the pre-designed object on a form.

The programmer does not have to bother with the details of syntax associated with implementing Dynamic Link Libraries (DLLs) into their code. With this burden removed, programmers can concentrate solely on their applications. Novice and advanced virtual instrument programmers will appreciate the flexibility and functionality of Compumotor's new Motion OCX Toolkit.

Compumotor's Motion OCX Toolkit provides a 32-bit OLE Custom Control Extension (OCX) designed to run under Windows 95 or Windows NT and the following development environments.

- Visual Basic 4.0 or later
- Delphi 2.0 or later
- Visual C++
- Any other 32-bit development environment containing OCX controls.

OCX controls, also known as an OLE (Object Linking and Embedding) Custom Controls, are the latest addition to Microsoft®'s OLE family. The controls provide unprecedented compatibility with almost any virtual instrument application. The Motion OCX Toolkit makes DLLs a thing of the past.



The Motion OCX Toolkit facilitates the quick development of your custom operator interface to Compumotor's 6000 Series bus-based controllers. Motion OCX Toolkit includes these controls:

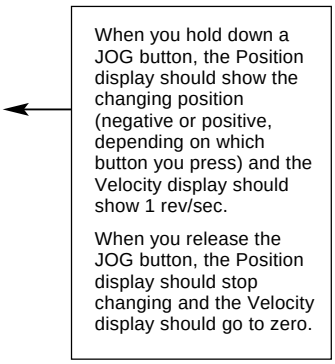
- Communications Shell (OCX (Comm6000)

  Use Comm6000 to control basic communication with the 6000 product, including interrupt handling and sending/ receiving files.
- Terminal OCX (Term6000)

  Use Term6000's active communication interface to the 6000 product for executing 6000 commands, checking program responses and status reps, and viewing error messages.
- Fast-Status Polling OCX (Poll6000)

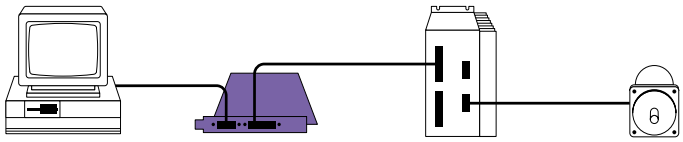
  Use Poll6000 to poll the 6000 product's fast status register and display information such as position, velocity, axis status, system status, etc.

Each control has a custom properties dialog box for easy set up. There is also a registration utility that can check for 6000 controller registration information in the windows registry and enter or change parameters.



Parker Automation

Parker Hannifin Corporation
Compumotor Division
Rohnert Park, California



6000 Position-Based Following

Position following is a standard feature on all 6000 series servo controllers. Position following is commonly used in applications such as:

- Electronic gearbox
- Flying Cutoff
- Random Timing Infeed
- Web Processing
- Product Spacing
- Cut-to-Length

Positioning following can include continuous, preset, and registration-like moves in which the velocity is replaced with a ratio.

Position Following allows for these capabilities and more:

- The slave may follow in either direction and change ratio while moving.
- Phase shifts are allowed during motion.
- Ratio changes or new moves may be dependent on master position or based on receipt of a trigger input.
- A slave axis may perform following moves or normal time-based moves in the same application because following can be enabled and disabled at will.

In position following, acceleration ramps between ratios are dependent upon a specified master distance. Product cycles can be easily specified with the “master cycle concept.”

Continuous Process Automation

In any continuous process, throughput can often be increased if familiar motion functions are done on moving targets as opposed to stopping the process. For example, a conveyor belt carries trays of parts which are to be unloaded. If a controller could detect and track the motion of the tray and then perform pick and place operations on those parts without stopping the process, overall efficiency would be dramatically increased.

The Position Following feature of the 6000 servo controllers has the capability to solve continuous automation process applications. Controller capability requirements range from simple concepts, such as electronic gearbox, track ball and slave feed-to-length, to complex changes of ratio based on master position. Common applications include packaging, printing, continuous cut-to-length, and coil winding.

The following examples represent common applications in packaging and printing. Each application demonstrates the advance and retard capabilities or the use of the Periodic Master/Slave Synchronization features of the 6000 servo controllers. The latter feature is important for applications in which periodic operations must occur in intervals which are not perfectly repeatable. For these, the master and slave must be re-synchronized every cycle.

The features of the 6000 servo controller’s Following package that are used here are not specific to these applications. They can be combined in a variety of ways to solve almost any application which requires coordination and synchronization of multiple axes. Even if you do not recognize your exact requirements in these examples, there is a very good chance that your synchronization application could be solved using these features.

Electronic Gearbox

Two or more axes are “electronically geared together to maintain an exact relationship between the different axes and yet allow any mechaical gearing or linkages to be removed. Mechanical backlash and wear are eliminated.

Random Timing Infeed

Randomly spaced product on a conveyor is adjusted so that it is placed precisely on another conveyor for a later process. See following description.

Flying Cutoff (cut to length)

A continuous bar sheet or other extgrusion of material is cut to specific lengths while the material is in motion. Production times are reduced as product does not have to be stopped to be cut.

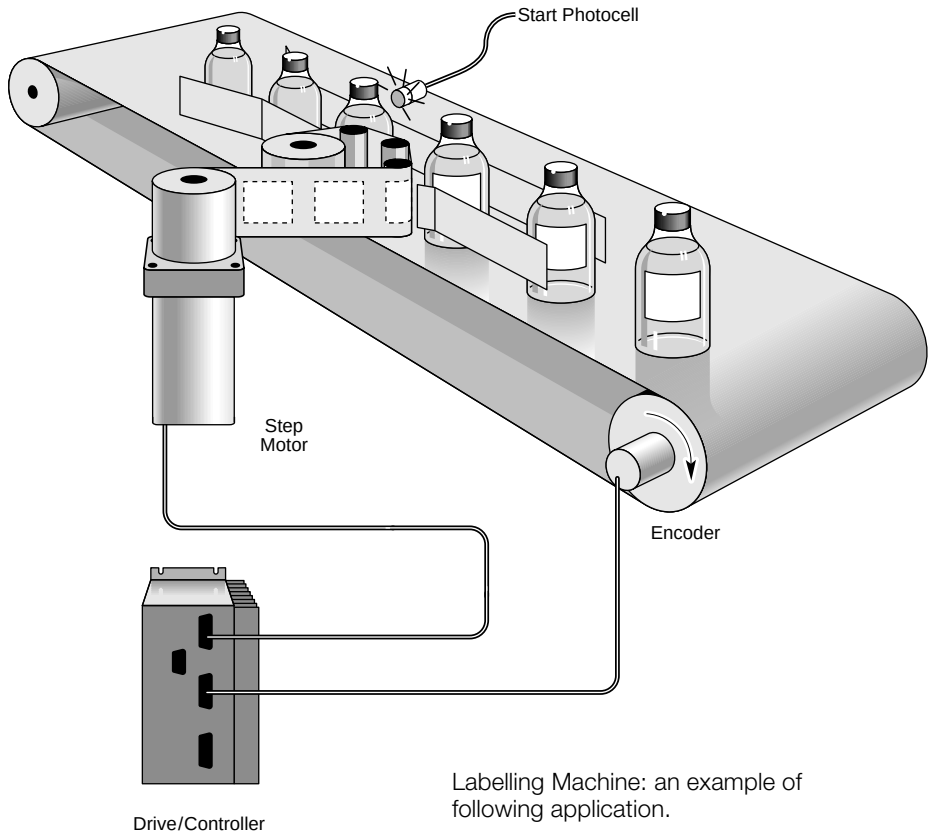
Product Spacing

A method of spacing out product that may have been cut in a flying cutoff process without stopping the product. Production times are reduced as the product does not have to be stopped to be separated.

Web Processing

Any process that must be performed on a continuous web of product can be accomplished on the fly with 6000 following. See following example on next page.

Parker Automation
Parker Hannifin Corporation
Compumotor Division
Rohnert Park, California



Labelling Machine: an example of following application.

Application Description

Bottles on a conveyor run through a labelling mechanism that applies a label to the bottle. The spacing of the bottles on the conveyor is not regulated and the conveyor can slow down, speed up, or stop at any time.

Machine Requirements:

- Accurately apply labels to bottles in motion
- Allow for variable conveyor speed
- Allow for inconsistent distance between bottles
- Pull label web through dispenser
- Smooth, consistent labelling at all speeds

Motion Control Requirements:

- Synchronization to conveyor axis
- Electronic gearbox function
- Registration control
- High torque to overcome high friction
- High resolution
- Open-loop stepper if possible

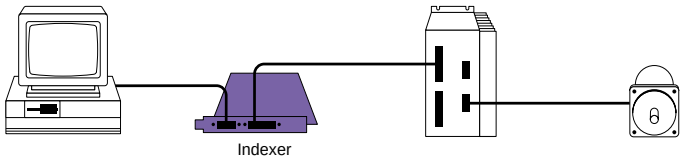
Application Solution

A motion controller that can accept input from an encoder mounted to the conveyor and reference all of the speeds and distances of the label roll to the encoder is required for this application. A servo system is also required to provide the torque and speed to overcome the friction of the dispensing head and the inertia of the large roll of labels. A photosensor connected to a programmable input on the controller monitors the bottles' positions on the conveyor. The controller commands the label motor to accelerate to line speed by the time the first edge of the label contacts the bottle. The label motor moves at line speed until the complete label is applied, and then decelerates to a stop and waits for the next bottle.

Product Solutions:

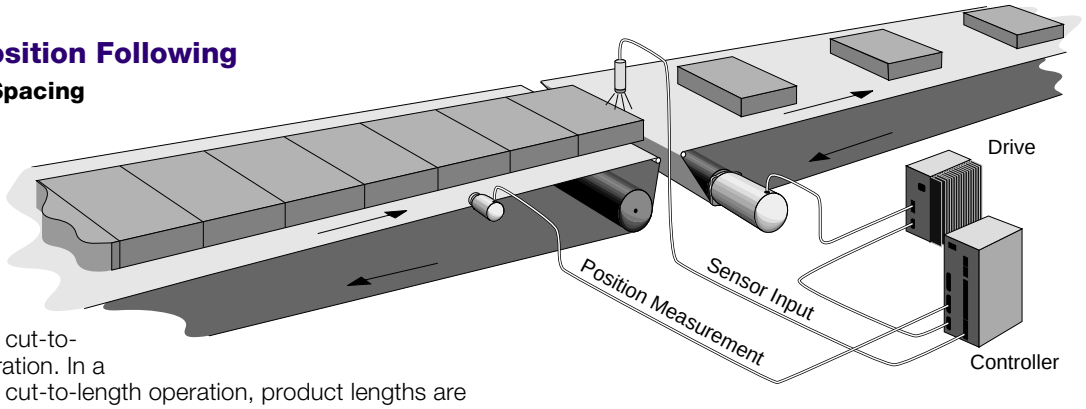
Controller	Motor
APEX6152*	APEX604

* The ZXF single-axis servo controller has also been used in these types of applications.



6000 Position Following

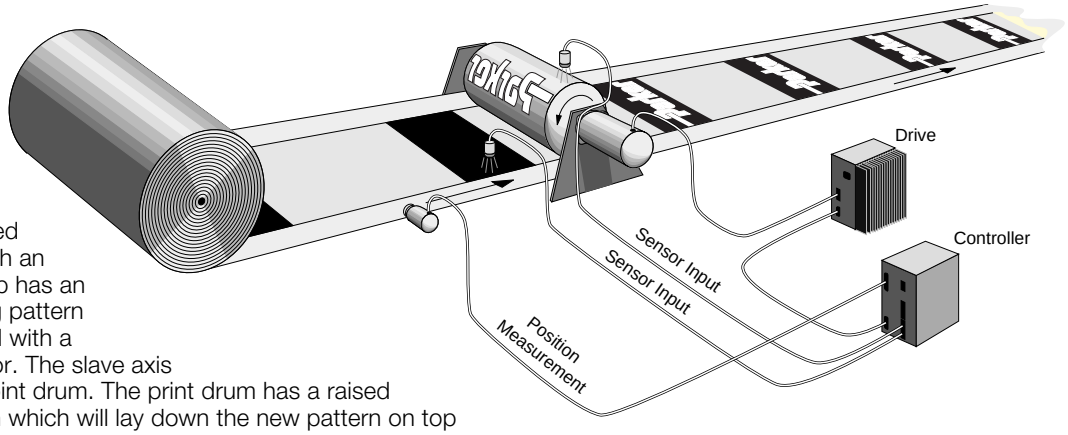
Product Spacing



This example shows a possible down-stream operation from a continuous cut-to-length operation. In a continuous cut-to-length operation, product lengths are separated from the feed stock, but are not spaced apart from each other. In order to wrap the individual products, they must be spaced at specific intervals.

The master conveyor moves the product away from the cutting operation where newly cut product is spaced close together. The slave conveyor (moving at a velocity proportional to the master) takes the correctly spaced products to a wrapping station. A sensor at the entrance to the slave conveyor is wired into the controller. The front edge of the product is detected by the sensor as it moves off the master conveyor. The controller waits for the entire product length to leave the master conveyor before it superimposes a distance-shift (i.e., a short fast advance) to introduce uniform product spacing on the slave conveyor. This advances each product a specific distance ahead of the next product as they move onto the second (or following) conveyor.

Web Processing



Rotary printing on a moving web is a typical web processing application. In this example, the web is the master, and its position and speed are measured with an encoder. The web has an existing repeating pattern which is detected with a registration sensor. The slave axis drives a rotary point drum. The print drum has a raised and inked portion which will lay down the new pattern on top of the existing pattern. During the print segment of its rotation, the print drum surface must match the position and speed of the existing print on the web in order to print without smearing, and maintain the proper registration. This requires that the surface speeds match at a 1:1 ratio during the actual print portion. The length of the existing pattern on the web, however, will generally not match the circumference of the drum. This requires that the print drum change to a different ratio during the non-print portion of the cycle. In other words, it requires the construction of a periodic ratio profile. Such a profile allows the drum to match the web speed during printing, yet rotate to correct position at the start of the next web pattern. Registration errors could arise if the web slips, stretches, or the original pattern is not perfectly regular. This problem is solved by using the registration sensor. Each new edge of each repeated web pattern is detected and the drum position is noted automatically. If the drum is not in the required position at the instant the web pattern is detected, a corrective advance or retard is superimposed on the regular cycle. This allows continuous perfect print alignment, even if the existing pattern is slightly irregular. Because the control of the print drum is based on the measurement of the web positions and speed, registration is maintained regardless of web speed changes.

Parker Automation
Parker Hannifin Corporation
Compumotor Division
Rohnert Park, California

Random Timing Infeed

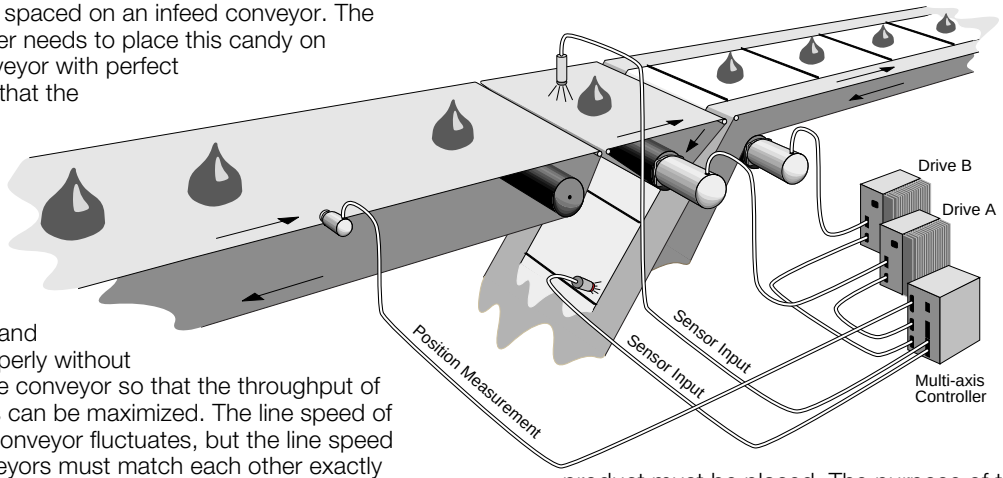
Random timing infeed refers to operations in which a product, at a particular point in a process, enters a conveyor with non-repeatable, or random timing, yet must leave the conveyor with perfect spacing. Typically, there is an infeed conveyor on which products are randomly spaced, a short conveyor on which the correction is made, and an exit conveyor with dividers on which the product must be placed. Line speed is critical when the product moves from one conveyor to another.

Two sensors, one located on the junction between the infeed conveyor and correction conveyor, and the other sensor on the exit conveyor must have a fixed phase relationship. The first sensor senses the product and the other sensor detects the locations of the dividers. The phase relationship is simply the distance the correction conveyor has traveled between the activation of the sensor on the exit conveyor and the activation of the product sensor.

The goal of the correction conveyor is to correct for variations in this phase relationship. Both the correction and exit conveyors are normally moving, or slaving, at a 1:1 relationship with the infeed conveyor, or master axis. When the product enters the correction conveyor, the 6000 servo controller can determine how much correction is needed by looking a the phase relationship between sensors. After the product has completely left the infeed conveyor, the 69099 servo controller adds an advance to the correction conveyor so that the calculated adjustment is made in time for the product to be placed on the exit conveyor. The controller can also be programmed to adapt to changes in mechanical alignment of the conveyors.

Random Timing Infeed Application: Problem

A food products manufacturer produces candy that is randomly spaced on an infeed conveyor. The manufacturer needs to place this candy on an exit conveyor with perfect spacing so that the candy can be wrapped and packaged accurately. The candy must be transferred and spaced properly without stopping the conveyor so that the throughput of the process can be maximized. The line speed of the infeed conveyor fluctuates, but the line speed of the conveyors must match each other exactly when the product moves from one conveyor to another in order to maintain product quality.



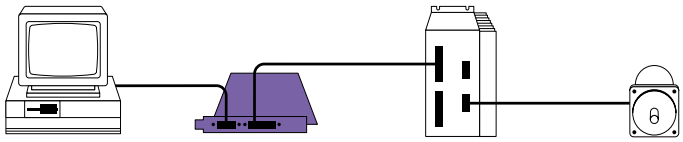
I/O Requirements

- 2 high-speed sensor inputs
- encoder feedback from the infeed conveyor

Solution

The solution of this application uses the Following and Random Timing Infeed functionality of the 6000 servo controller to control a short correction conveyor as well as the exit conveyor onto which the product is placed. Two sensors, one located on the junction between the infeed conveyor and the correction conveyor and the other on the exit conveyor, are placed with a fixed phase relationship. The first sensor recognizes the product and the other sensor detects the locations of the dividers between which the product must be placed. The purpose of the correction conveyor is to correct for variations in this phase relationship. The phase relationship is simply the distance the correction conveyor has traveled between activation of the exit conveyor sensor and activation of the product sensor. Both the correction and exit conveyors are moving, or slaving, at a 1:1 relationship with the infeed conveyor, or master axis. The conveyors are following the feedback from the encoder on the infeed conveyor.

When the product enters the correction conveyor, the 6000 servo controller can determine how much correction is needed by looking at the phase relationship between sensors. After the product has completely left the infeed conveyor, the correction conveyor makes the required shift and then matches the speed of the exit conveyor for a smooth transfer. The result is a quality product that is perfectly spaced on the exit conveyor.



Contouring (Circular Interpolation)

Contouring in the 6000 Series indexers makes it easy to follow a two-dimensional path consisting of multiple line and arc segments. Circular interpolation is useful for making arcs and circles, especially when a constant path velocity must be maintained along a two-dimensional path. One such example would be in dispensing and engraving.

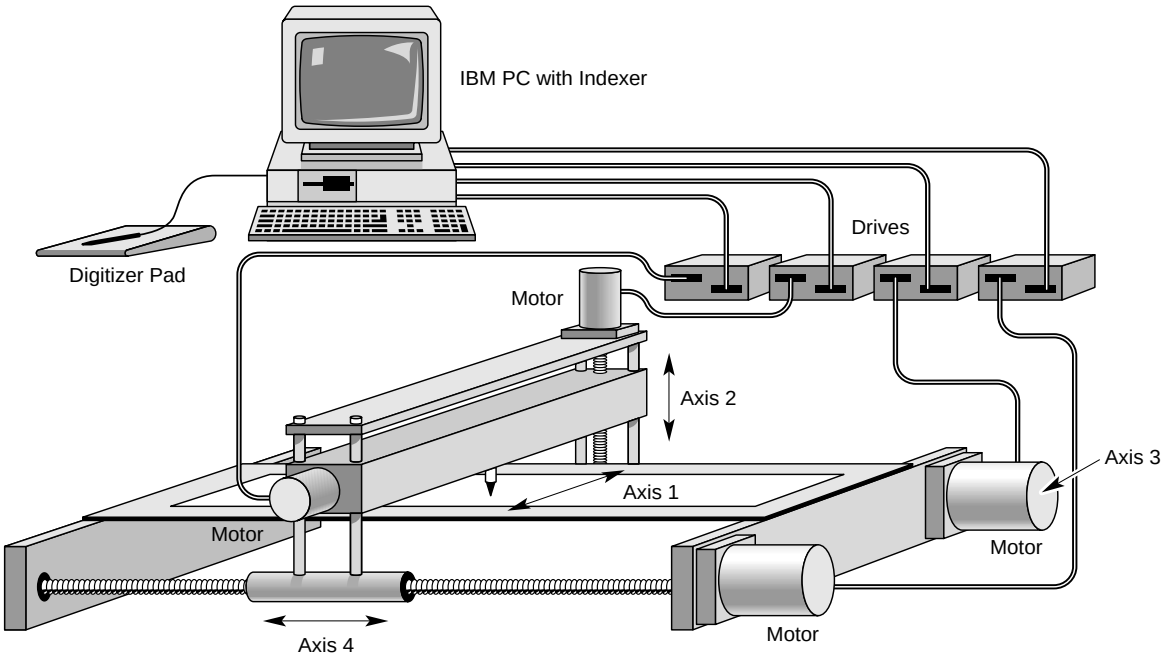
In addition, a tangential axis is available to keep an angular position which changes linearly with the path direction. The tangential axis would be used in applications that require a work piece or tool to remain tangent or perpendicular to the path direction. A typical example that requires a tangential axis is a knife always pointing into the cut or a welding head staying normal to the weld.

A proportional axis may also be included to keep a position that is proportional to the distance traveled along the path described by X and Y. This allows helical interpolation.

The 6000 Series contouring option allows users to create complex move profiles on multiple axes. The easy programming language enables the user to generate arcs by only specifying the endpoint, radius or origin of the arc, and the direction of travel. 6000 Series contouring also provides the user with the ability to control I/O while in the contouring mode. The 6000 Series indexers allows the user to define and execute up to 300 two-dimensional motion paths with expanded memory. A path refers to the path traveled by the load in an X-Y plane, and must be defined before any motion takes place along the path.

Coordinated systems allow the assignment of an arbitrary X-Y position as a reference position for subsequent absolute end point specifications.

Example of Contouring Application—
Engraving Machine



Parker Hannifin Corporation
Compumotor Division
Rohnert Park, California