Fundamentals of Motion Control

Publish Date: Mar 18, 2013 | 91 Ratings | **3.75** out of 5

Overview

This tutorial is part of the National Instruments Measurement Fundamentals series. Each tutorial in this series teaches you a specific topic of common measurement applications by explaining theoretical concepts and providing practical examples. In this tutorial, learn the fundamentals of a motion control system including software, motion controller, drive, motor, feedback devices, and I/O.

For more information, return to the NI Measurement Fundamentals Main Page.

Table of Contents

1. Components of a Motion Control System

Figure 1 shows the different components of a motion control system.

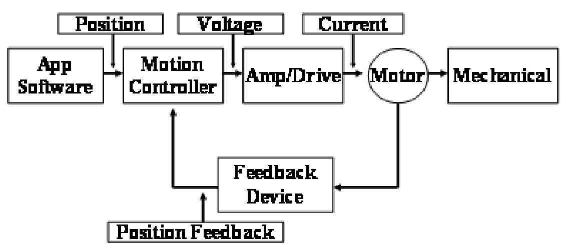

Figure 1. *Components of a Motion Control System*

Application software – You can use application software to command target positions and motion control profiles.

Motion controller – The motion controller acts as the brain of the system by taking the desired target positions and motion profiles and creating the trajectories for the motors to follow but outputting a ±10 V signal for servo motors, or a step and direction pulses for stepper motors.

Amplifier or drive – Amplifiers (also called drives) take the commands from the controller and generate the current required to drive or turn the motor.

Motor – Motors turn electrical energy into mechanical energy and produce the torque required to move to the desired target position.

Mechanical elements – Motors are designed to provide torque to some mechanics. These include linear slides, robotic arms, and special actuators.

Feedback device or position sensor – A position feedback device is not required for some motion control applications (such as controlling stepper motors) but is vital for servo motors. The feedback device, usually a quadrature encoder, senses the motor position and reports the result to the controller, thereby closing the loop to the motion controller.

2. Software for Configuration, Prototyping, Development

Application software is divided into three main categories: configuration, prototype, and application development environment (ADE). Figure 2 illustrates the motion control system programming process and the corresponding National Instruments product designed for the process:

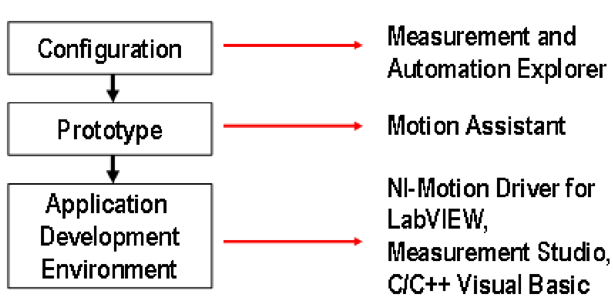

Figure 2. *The Motion Control System Development Process*

Configuration
One of the first things to do is configure your system. For this, National Instruments offers Measurement & Automation Explorer (MAX), an interactive tool for configuring not only motion control but all other National Instruments hardware. For motion control, MAX offers interactive testing and tuning panels that help you verify your system functionality before you program.

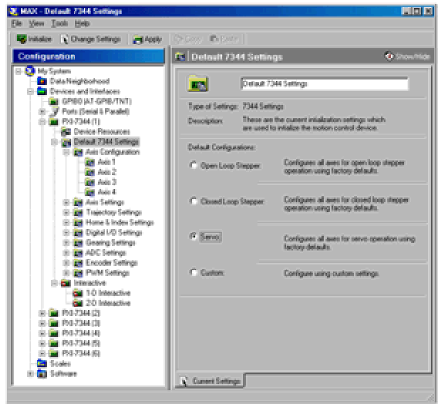

Figure 3. *NI MAX is an interactive tool for configuring and tuning your motion control system.*

Application Notes
Understanding Servo Tune
Using the 1D Interactive Environment to Test Motor Functionality
Axis Configuration for Motion Controllers
Axis Settings for Motion Controllers
Encoder Settings for Motion Controllers
Reference Settings for Motion Controllers
Digital I/O Settings for Motion Controllers

Prototyping
When you have configured your system, you can start prototyping and developing your application. In this phase, you create your motion control profiles and test them on your system to make sure they are what you intended. For prototyping, National Instruments offers an interactive tool called the NI Motion Assistant, which you can use to configure moves using a point-and-click environment and generate NI LabVIEW code based on the moves you configure. The key benefit of the NI Motion Assistant lies in the difference between configurable and programmable environments. With configurable environments, you can start your development without programming. You can think of the tasks in the NI Motion Assistant as prewritten blocks of code that you simply configure to meet your needs. Programmable environments, on the other hand, require you to use standard programming languages such as LabVIEW, C, or Visual Basic to accomplish your tasks. Unfortunately, many configurable environments may be limited in functionality or in the ability to integrate with other I/O outside motion. The NI Motion Assistant bridges the gap between programmable and configurable environments by offering all configurable system features as well as LabVIEW code generation.

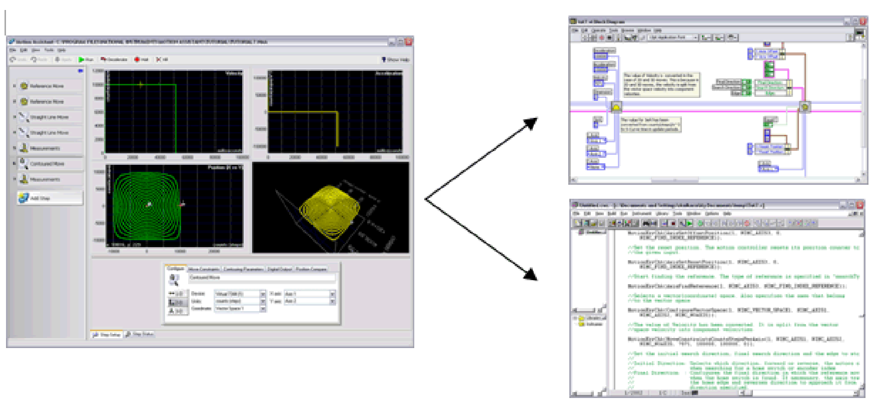

Figure 4. *The NI Motion Assistant helps you quickly prototype your application and then convert your project into LabVIEW VIs or C code for further development*.

Evaluation Software

Development
After the prototyping phase, the next step is to develop the final application code. For this, you use driver-level software in an ADE such as LabVIEW, C, or Visual Basic. For a National Instruments motion controller, you use NI-Motion driver software.

The NI-Motion driver software contains functions you can use to communicate with NI motion controllers in the Windows or LabVIEW Real-Time OS. NI-Motion also includes MAX to help you easily configure and tune your motion system.

For non-Windows systems, you can develop your own driver using the NI Motion Control Hardware DDK manual. It explains how to communicate on a low level with NI motion controllers. If you do not have the expertise or time to develop your own driver, National Instruments Alliance Partner Sensing Systems offers a Linux and VxWorks driver, and can create drivers for other OSs, such as Mac OS X or RTX.

Application Notes
Understanding Input and Return Vectors in Onboard Programming
Understanding Loop and Conditional Structures in Onboard Programming
Understanding Variable Arithmetic in Onboard Programming
Advanced Object Management in Onboard Programming
Controlling an X-Y Stage with a Joystick

3. Motion Controller

A motion controller acts as the brain of the motion control system and calculates each commanded move trajectory. Because this task is vital, it often takes place on a digital signal processor (DSP) on the board itself to prevent host-computer interference (you would not want your motion to stop because your antivirus software starts running). The motion controller uses the trajectories it calculates to determine the proper torque command to send to the motor amplifier and actually cause motion.

The motion controller must also close the PID control loop. Because this requires a high level of determinism and is vital to consistent operation, the control loop typically closes on the board itself. Along with closing the control loop, the motion controller manages supervisory control by monitoring the limits and emergency stops to ensure safe operation. Directing each of these operations to occur on the board or in a real-time system ensures the high reliability, determinism, stability, and safety necessary to create a working motion control system.

Learn more about the FlexMotion architecture of National Instruments DSP-based motion controllers.

Calculating the Trajectory
The motion trajectory describes the motion controller board control or command signal output to the driver/amplifier, resulting in a motor/motion action that follows the profile. The typical motion controller calculates the motion profile trajectory segments based on the parameter values you program. The motion controller uses the desired target position, maximum target velocity, and acceleration values you give it to determine how much time it spends in the three primary move segments (which include acceleration, constant velocity, and deceleration).

For the acceleration segment of a typical trapezoidal profile, motion begins from a stopped position or previous move and follows a prescribed acceleration ramp until the speed reaches the target velocity for the move.

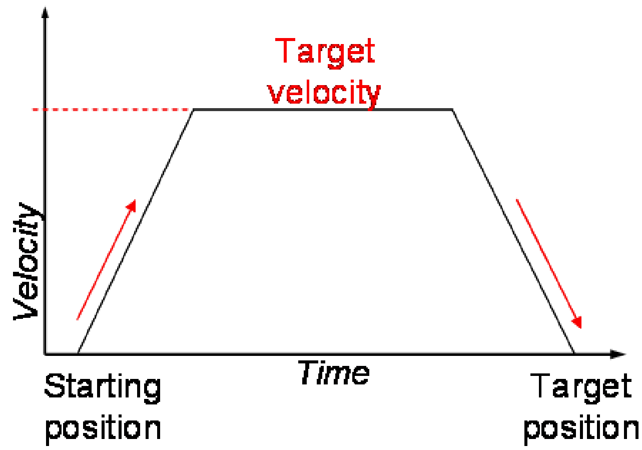

Figure 5. *A Typical Trapezoidal Velocity Profile*

Motion continues at the target velocity for a prescribed period until the controller determines that it is time to begin the deceleration segment and slows the motion to a stop exactly at the desired target position.

If a move is short enough that the deceleration beginning point occurs before the acceleration has completed, then the profile appears triangular instead of trapezoidal and the actual velocity attained may fall short of the desired target velocity. S-curve acceleration/deceleration is a basic trapezoidal trajectory enhancement where the acceleration and deceleration ramps are modified into a nonlinear, curved profile. This fine control over ramp shape is useful for tailoring motion trajectory performance based on the inertial, frictional forces, motor dynamics, and other mechanical motion system limitations.

Application Notes
Trajectory Settings for Motion Controllers
S-Curve Acceleration and Deceleration
Velocity Profiling

Selecting the Right Motion Controller
NI offers three main families of DSP-based motion controllers, including the low-cost NI 733x series, the mid-range NI 734x series, and the high-performance NI 735x series. The NI 733x low-cost controllers offer four-axis stepper motor control and most of the basic functions you need for a wide variety of applications, including single and multiaxis point-to-point motion. The NI 734x series is the mid-range series that offers up to four axes of both stepper and servo control, as well as some higher-performance features such as contouring and electronic gearing. The NI 735x series is the most advanced series that offers up to eight axes of stepper and servo control, extra I/O, and many powerful features including sinusoidal commutation for brushless motors and 4 MHz periodic breakpoints (or position triggers) for high-speed integration.

Selection Guides
View the Complete Motion Control Product Selection Guide

Creating Custom Motion Controllers
While current motion controllers with DSPs are suitable for many applications, when it comes to high-precision motion control with servo update rates as fast as 200 kHz, machine builders turn to designing their own motion controllers on a custom printed circuit board (PCB). Not only is the development expensive in terms of time and cost, but the fixed personality of the motion controller makes the system inflexible for future redesigns or for accommodating variations in the motion control algorithms at run time. Some applications that need such a high level of precision and flexibility include wafer processing machines in the semiconductor industry or the inline vehicle sequencing (ILVS) reconfigurable-at-run-time assembly line for the automotive industry. National Instruments reconfigurable I/O (RIO) technology combined with NI SoftMotion technology provides the right tools for machine builders who want high-precision customized motion control with the complete flexibility of a field-programmable gate array (FPGA). In addition to high-precision applications, machine builders and OEMs can use the LabVIEW NI SoftMotion Module to implement multiaxis coordinated motion control using LabVIEW on a variety of platforms – from plug-in NI M Series data acquisition (DAQ) devices for industrial PCs and PXI to rugged systems using NI CompactRIO and Compact FieldPoint programmable automation controllers (PACs).

Tutorials:
White Paper: Create Your Custom Motion Controller on Any Platform with LabVIEW

4. Move Types

Single-Axis, Point-to-Point Motion
One of the most commonly used profiles is the simple, single-axis, point-to-point move, which requires the position to which the axis needs to move. Often it also requires the velocity and acceleration (usually supplied by a default setting) at which you want the motion to move. Figure 6 shows how to move a single axis in LabVIEW using the default velocity and acceleration.

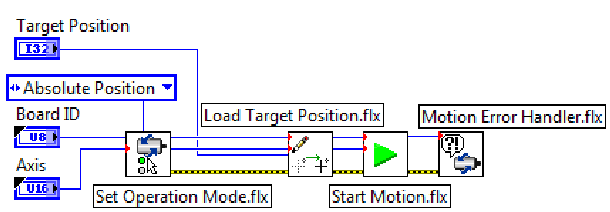

Figure 6. *Single-Axis, Point-to-Point Motion in LabVIEW*

Coordinated Multiaxis Motion

Another type of motion is coordinated multiaxis motion, or vector motion. This move is often point-to-point motion but in 2D or 3D space. Vector moves require the final positions on the X, Y, and/or Z axes. Your motion controller also requires some type of vector velocity and acceleration. This motion profile is commonly found in XY-type applications such as scanning or automated microscopy. Figure 7 shows how to accomplish a three-axis move using LabVIEW. For more information on coordinated motion, view the examples in the LabVIEW Multiaxis.llb library in NI-Motion driver software.

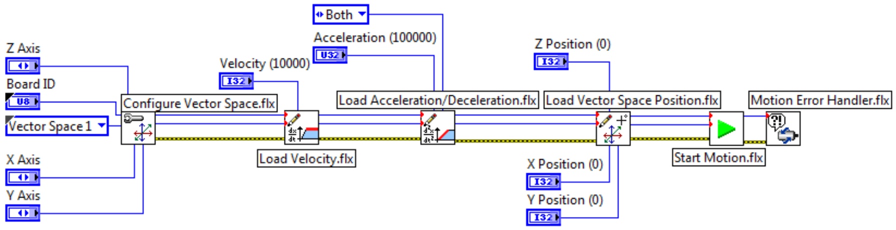

Figure 7. *Coordinated Multiaxis Motion in LabVIEW*

Blended Motion

Blended motion involves two moves fused together by a blend that causes the moves to act as one. Blended moves require two moves and a blend factor that specifies the blend size. Blending is useful for applications requiring continuous motion between two different moves. However, in blended motion, your system does not pass through all of the points in your original trajectory. If the specific position along the path is important to you, consider a contouring motion.

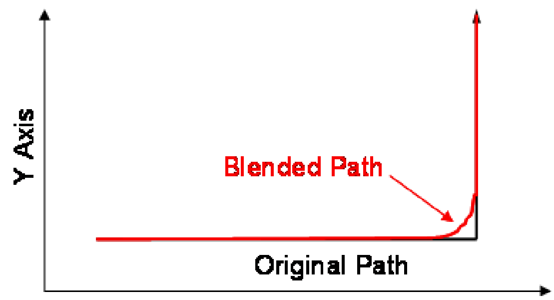

Figure 8. *Blended Motion*

Figure 8 explains the blending between two vector moves in LabVIEW. For more information on blending, view the Sequence of Blended Vector Moves example program in NI-Motion driver software.

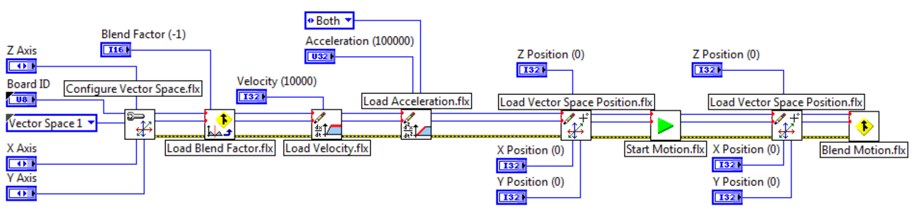

Figure 9. *Blended Motion in LabVIEW*

Contoured Motion

With contouring, you can supply a position buffer and create a smooth path or spline through them. Contouring holds an advantage over blending in that it guarantees that the system passes through each position.

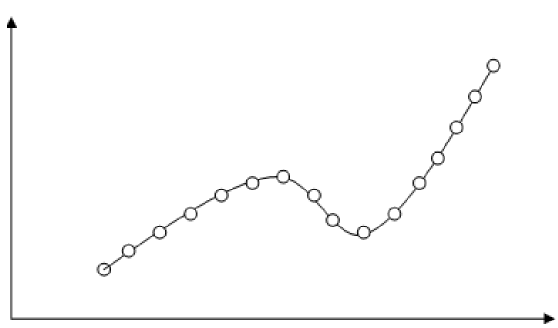

Figure 10. *Contoured Motion*

Figure 11 explains a contoured move using LabVIEW. For more information on contouring, view the examples in the Countouring.llb example library found in NI-Motion driver software.

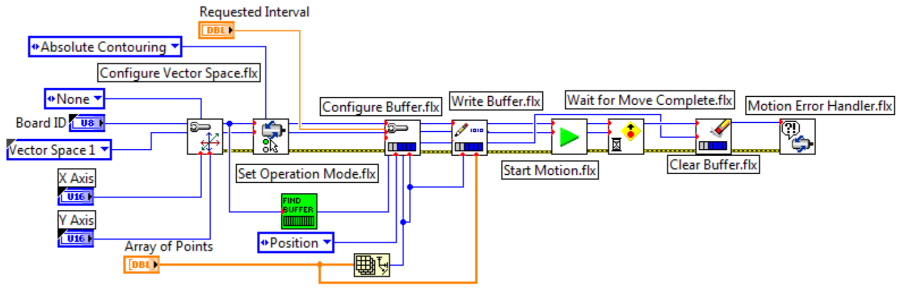

Figure 11. *Contoured Motion in LabVIEW*

Electronic Gearing
With electronic gearing, you can simulate the motion that would occur between two mating gears without using real gears. You use electronic gearing by supplying a gear ratio between a slave axis and a master axis, encoder, or ADC channel.

Figure 12 shows how to configure a slave axis to follow a master axis. For more information on electronic gearing, view the Gearing.llb example library found in NI-Motion driver software.

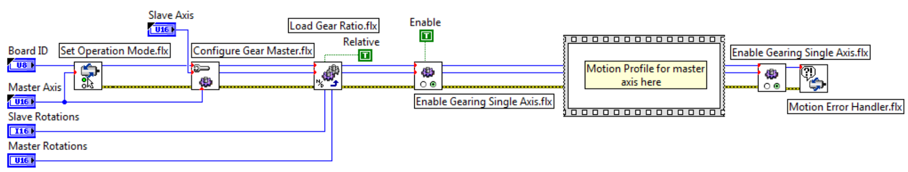

Figure 12. *Electronic Gearing in LabVIEW*

Application Notes
Helical Interpolation with FlexMotion
Stopping Modes
Spherical Interpolation
Circular Interpolation
Learn More about Electronic Gearing
Gearing Settings for Motion Controllers

5. Motor Amplifiers and Drives

The motor amplifier or drive is the part of the system that takes commands from the motion controller in the form of analog voltage signals with low current and converts them into signals with high current to drive the motor. Motor drives come in many different varieties and are matched to the specific type of motor they drive. For example, a stepper motor drive connects to stepper motors and not servo motors. Along with matching the motor technology, the drive must provide the correct peak current, continuous current, and voltage to drive the motor. If your drive supplies too much current, you risk damaging your motor. If your drive supplies too little current, your motor does not reach full torque capacity. If your voltage is too low, your motor cannot run at its full speed.

You should also consider how to connect your amplifier to your controller. Some motor companies sell drives that easily connect to the motors they offer. National Instruments offers drives for both two-phase stepper motors and DC brushed servo motors. These drives have screw terminals with which you can connect to many different motors. Table 1 describes the difference between NI motor drives.

For connecting to third-party drives and amplifiers, National Instruments offers the universal motion interface (UMI) – the standard UMI-7764 with screw terminal connectivity – and the industrial UMI-7774 with 24 V logic digital I/O and D-Sub connectivity.

Selection Guides
NI Drive Selection Chart

Application Notes
Simple Servo Amplifiers
DC Servo Amplifiers
Changing the Voltage Output on a FlexMotion Controller for Drives That Do Not Accept ±10 V
How to Connect the NI 73xx Inhibit and Command Signal Outputs to Third-Party Drives

6. Motors and Mechanical Elements

Motor selection and mechanical design is a critical part of designing your motion control system. Many motor companies offer assistance in choosing the right motor, but it is helpful to know some basics about motors before you start looking. Table 1 describes different motor technologies.

	Pros	Cons	Applications
Stepper Motors	Inexpensive, can be run open loop, good low-end torque, clean rooms	Noisy and resonant, poor high-speed torque, not for hot environments, not for variable loads	Positioning, micromovement
Brushed DC Servo Motors	Inexpensive, moderate speed, good high-end torque, simple drives	Maintenance required, no clean rooms, brush sparking causes EMI and danger in explosive environments	Velocity control, high-speed position control
Brushless Servo Motors	Maintenance-free, long lifetime, no sparking, high speeds, clean rooms, quiet, run cool	Expensive and complicated drives	Robotics, pick-and-place, high-torque applications

Table 1. *NI Motor Drives*

After determining which technology you want to use, you need to determine the torque and inertia at the motor shaft. For more information on calculating system torque, read the Motor Fundamentals article at ***zone.ni.com***.

Something else to consider when selecting your motor and other mechanics is whether an off-the-shelf actuator (such as a stage) might work for your application. Stages offer the power transmission to obtain useful rotary or linear motion without designing it yourself.

Selection Guides
Selecting the Correct Servo Motor for Your Application
Selecting the Correct Stepper Motor for Your Application
Selecting the Correct Stage for Your Application

Application Notes
Motor Fundamentals
Servo Motor Basics
Servo Motor Applications
Stepper Motor Basics
Stepper Motor Types
Linear Stepper Motors
Stepper Motor Applications
Stepper Motor Switching Sequence

7. Feedback Devices and Motion I/O

Feedback Devices
Feedback devices help the motion controller know the motor location. The most common position feedback device is the quadrature encoder, which gives positions relative to the starting point. Most motion controllers are designed to work with these types of encoders. Other feedback devices include potentiometers that give analog position feedback, tachometers that provide velocity feedback, absolute encoders for absolute position measurements, and resolvers that also give absolute position measurements. When using National Instruments motion controllers, you can use quadrature encoders and potentiometers.

Application Notes
Basics of Feedback
Encoders
Linear and Rotary Encoders
Absolute Encoders
Resolvers
Magnetic Encoders
Optical Encoders
Quadrature Encoders
Displacement and Position Feedback Devices
How to Choose Among LVDT, RVDT, Potentiometer, Optical Encoder, Ultrasonic, Magnetostrictive, and Other Technologies
Motion I/O
Other I/O that is important in motion control includes limit switches, home switches, position triggers, and position capture inputs. Limit switches provide information about the end of travel to help you avoid damaging your system. When a motion system hits a limit switch, it typically stops moving. Home switches, on the other hand, indicate the system home position to help you define a reference point. This is important for applications such as pick-and-place.

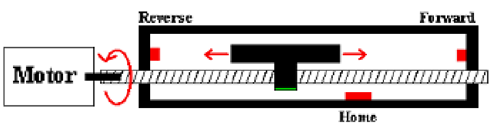

Figure 13. *Limit and Home Switches in a Motion Control System*

Triggers such as position trigger outputs or position capture inputs help when integrating with other devices. With position trigger outputs (also called breakpoints and position compare), you can set up a trigger that executes at a prescribed position. This type of action is useful in operations such as scanning, where you might want to trigger a system to take measurements at a series of prescribed positions. Position capture inputs, on the other hand, cause the motion controller to immediately capture an event occurrence position and store it in memory. This is useful if you have an external trigger and would like to know the position at which it occurs in your system.

Application Notes
Home and Index in Motion Control
Limit Switches in Motion Control
Integration with Motion Using RTSI
ADC Settings for Motion Controllers
Breakpoints/Triggers

8. Relevant NI Products

Customers interested in this topic were also interested in the following NI products:

- LabVIEW
- Data Acquisition (DAQ)
- Signal Conditioning

For more tutorials, return to the NI Measurement Fundamentals Main Page.