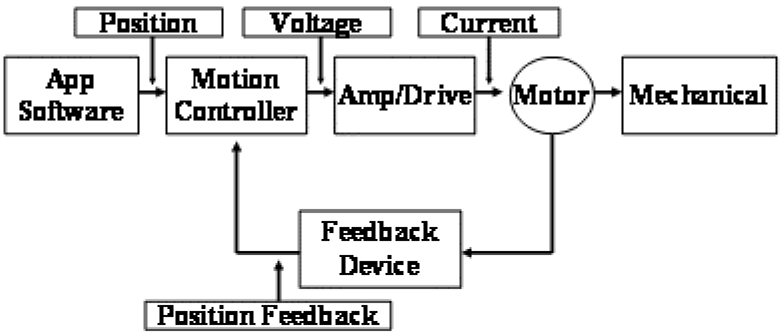
Fundamentals of Motion Control

Components of a Motion Control System

The figure below shows the different 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 brain of the 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.

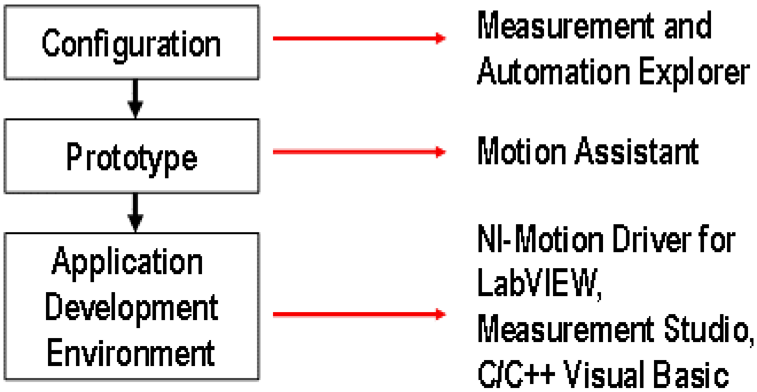
Motor – Motors turn electrical energy into mechanical energy and produce the torque required to move to the desired target position.

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.

Application Software for Configuration, Prototyping, and Development

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



Configuration
One of the first things to do is configure your system for all your motion control and other hardware. (Note: Measurement and Automation Explorer is the product of National Instruments)

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. (Note: Motion Assistant is the product of National Instruments)

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. The NI-Motion driver software contains functions with which you can communicate with NI motion controllers in Windows or LabVIEW Real-Time. NI-Motion also includes Measurement and Automation Explorer to help you easily configure and tune your motion system. (Note: NI-Motion driver is the product of National Instruments)

For non-Windows systems, you can develop your own driver using the 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.

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 also 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.

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.

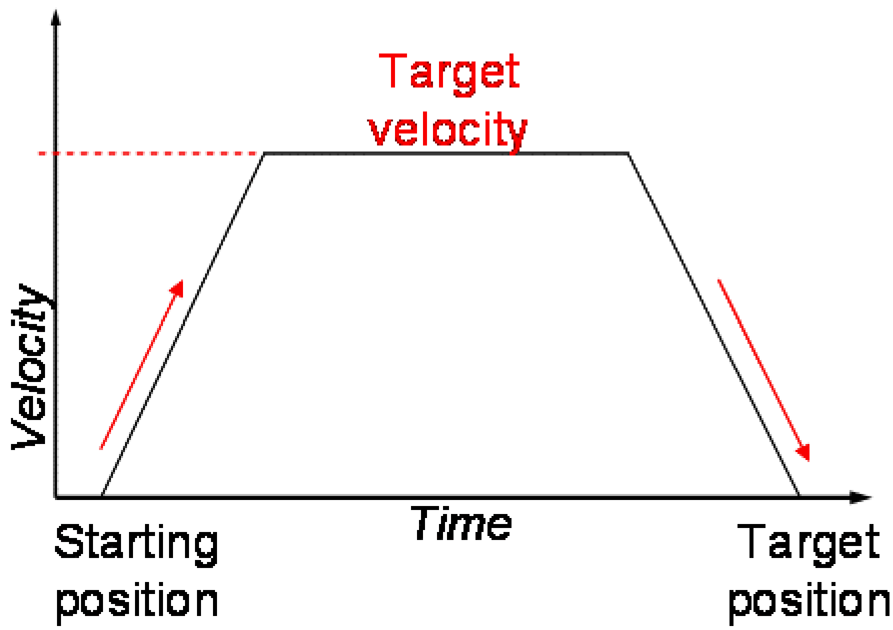


Figure 5: A typical trapezoidal velocity profile

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 very useful for tailoring motion trajectory performance based upon the inertial, frictional forces, motor dynamics, and other mechanical motion system limitations.

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 also 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.

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. The figure below describes different motor technologies.

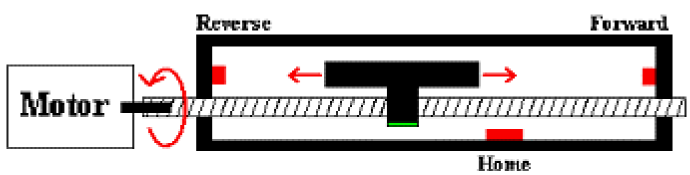
	Pros	Cons	Applications
Stepper Motors	Inexpensive, can be run open loop, good low-end torque, clean rooms	Noisy and resonant, poor high-speed torque, not for hot environments, not for variable loads	Positioning, micro movement
Brushed DC Servo Motors	Inexpensive, moderate speed, good high-end torque, 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

After determining which technology you want to use, you need to determine the torque and inertia at the motor shaft.

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.

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 very important for applications such as pick-and-place.



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 very 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.