PROGRAMMABLE PROTECTION FEATURES

How Advanced Drives & Controllers Are Offering More Machine Protection through Programmable Features & Quick-Stop Capabilities

MOOG ANIMATICS

Summary

Protection of the automated machine and related equipment/materials is an important design consideration for any motion control system. This protection can be provided through the use of various hardware components. However, looking beyond the protective capabilities of hardware components, what about the supplemental, programmable protection capabilities offered by the motion control system?

To assist with this, today's motion control systems provide programmable protection capabilities that can supplement the hardware components, simplify the design and reduce system costs. This white paper identifies and explains some of the newest innovations in programmable equipment/machine protection features for advanced controllers. It also provides several examples of their use and implementation.

Note that this information is not meant to address, supplement or replace knowledge or implementation of equipment protection regulations or operator safety regulations. The programmable protection discussed here is strictly meant for equipment/machine protection only.

Warning: Any machinery can be dangerous. Therefore, it is the responsibility of the machine designer/builder to perform a full risk assessment and implement all industry-required protection for equipment, operator safety protection and error handling as part of the motion control system and machine design.

Overtravel Limits

Hardware Overtravel Limits vs. Programmable Software Overtravel Limits

Hardware overtravel limits are frequently used to limit movement in motion control systems. A hardware limit is typically an external sensor or switch in a fixed position that limits motion and causes a motion fault if exceeded. In some advanced controllers, such as the SmartMotor™ by Moog Animatics, hardware overtravel limit inputs are enabled by default to protect the motor during startup and must be cleared before proceeding. Positive and negative hardware limits may be set individually or together. Hardware limits and limit switches are often used to control the movement of a machine, as control interlocks or to count objects passing a point.

An example of a limit switch would be a sensor that detects a lever arm moving beyond a certain point. If the operator unknowingly moves the lever arm past the limit switch (where it could potentially damage product or other parts of the machine), the limit switch will cause a motion fault within the motor and motion will be stopped. The hardware limit is a normally closed input to the motion control system, meaning that if a cable happened to come lose or there was a malfunction in the hardware limit, a motion fault would occur and motion would stop.

Software overtravel limits often exist in motion control systems in addition to the hardware limits and offer their own distinct advantages. Software overtravel limits are virtual limit switches that can interrupt motion with a fault in the event the actual position of the motor exceeds the desired region of operation. Similar to hardware overtravel limits, software overtravel limits are also directionally sensitive, so any motion commanded beyond the exceeded limit will cause a fault.

One distinct advantage of software overtravel limits is their lack of hardware components. Because software limits don't rely on hardware components, using software limits in machine design will lower the total machine build cost. This savings will increase linearly with each additional limit established (as compared to using hardware limits).

Because software limits don't rely on hardware components, using software limits in machine design will lower the total machine build cost.

Another distinct advantage of software overtravel limits is their flexibility. Positive and/or negative limits can be changed at the discretion of the user or machine builder without having to rearrange any hardware. If the machine designer requires a significantly wider range of motion, the software overtravel limits allow this without hardware changes or additional cabling.

Some controllers offer other software limits such as position error limits and velocity limits. More advanced controllers, like the Moog Animatics' SmartMotor, also offer derivative error limits (or rate-of-change error limits).

Nonprogrammable Protection Features

Peak Overcurrent

Peak overcurrent is a hardware-based, nonprogrammable protection feature included on most industrial drives. It limits the absolute maximum current to protect the drive and the motor. Peak overcurrent limits are set higher than continuous overcurrent limits to provide an instantaneous overcurrent protection when high current cannot be maintained over long periods of time.

Continuous Overcurrent

Continuous overcurrent is the maximum sustained current allowable for the protection of the motor and the drive. Its setpoint typically includes a thermal-based algorithm for the heating effect caused by continuous current over time. In both continuous and peak overcurrent, levels are set by the drive manufacturer to protect against accidental current overloading of the system. This feature is included on almost all industrial drives.

Bus Overvoltage and Undervoltage

Bus overvoltage sets the maximum upper limit of voltage to protect against damage to the insulation and components on the circuit board. Bus undervoltage sets the minimum lower limit of voltage in the case that power drops out. As voltage decreases, the current increases. Therefore, a low voltage could damage equipment with a surge of current. Typically, these limits are based within the hardware circuit board. However, some products, such as the Moog Animatics' SmartMotor, have the settings stored in EEPROM so that custom values may be used. For example, custom values could be used in battery-powered applications to prevent damage to the battery packs.

Thermal Limits

In most drives, thermal limits are established by the component manufacturer and cannot be changed. Upper thermal limits protect against physically overheating components that could be damaged such as the insulation and motor windings.

In some drives, such as the integrated SmartMotor, while upper thermal limits cannot be exceeded without creating a fault, they may be programmed to a lower value at the discretion of the end user or machine designer for more accurate, application-specific protection.

In some drives, such as the integrated SmartMotor, while upper thermal limits cannot be exceeded without creating a fault, they may be programmed to a lower value at the discretion of the end user or machine designer. This provides more accurate, application-specific protection.

Software Programmable Protection Features

Position Error Limits (or Following Error Limits)

Position error limits can be set to guard against any actual motion that is outside the range of programmed trajectory based on the position counter inside the drive/controller.

Most drives and controllers include some type of position error limit. Position error is the difference between where the actual position is compared to where it should be. Position error limits can be set to guard against any actual motion that is outside the range of programmed trajectory based on the position counter inside the drive/controller. All servos will increase torque to compensate for falling behind the expected position. The increased torque is proportional to the amplitude of the position error.

While position error limits have some advantages, they are not the optimal choice for all applications. If a routine is being performed where product could get caught in the wrong part of the machine (such as accumulating at the end of a conveyor line) causing the product to be pinched between machine components, the following error would increase until the position error limit is reached. However, before that limit is reached the motor will build up torque to compensate for falling behind in position. As demanded torque increases, supplied torque increases until the position error limit is reached. This increased torque could damage the product caught in the machine process. In the case of accidental personnel interference (for example, if someone's hand was to get caught in the process where position error limit is established), that increased torque could injure the person's hand before the position error limit is reached.

Therefore, to give an added level of protection, some advanced controllers (including the integrated SmartMotor) offer a derivative error limit. For details, see ***Derivative Error Limit*** on page 7.

Velocity Limits

Another type of software limit is the maximum allowable velocity limit. Velocity limit, like position limit, is a simple programmable feature that's common to most drives/controllers. Velocity limits are used to restrict maximum speeds of equipment in motion to help protect the machine from possible damage and are especially useful in vertical load applications.

Consider the example of a long, ball screw actuator. If the critical speed rating of the ball screw actuator is exceeded, it will exhibit a whipping or lashing vibration until it finally fails. This can occur in both closed loop and open loop systems. In this case, a simple velocity limit would prevent the failure.

In most vertical load applications, velocity limits can protect the moving load and equipment. For example, assume a velocity limit of 2000 RPM has been set within the controller/drive and the vertical load starts falling. If the speed of the falling load exceeds the 2000 RPM velocity limit, a motion fault will occur and a predetermined response will take place. In the case of the Moog Animatics' SmartMotor, if the velocity limit is exceeded, the motor windings will dynamically short out to bring the load to a stop by default.

Derivative Error Limit

Some advanced motion controllers have derivative error limits, i.e., the rate of change of following error in units of velocity. Unlike the previous example of torque increase related to position error limit, the derivative error limit can be implemented for an almost immediate stop while increased torque is reduced to near zero.

For example, suppose an actuator is moving from left to right at a rate of one inch per second. If its path is blocked, it would push against the blockage in the attempt to catch up to the commanded position. This could cause equipment damage because the motor would begin to supply more torque as the following error increased. Torque would increase until it reached the standard following error limit.

Because a derivative error limit monitors how quickly the rate of following error increases over time, it catches the fault much faster than a traditional position error limit, which saves time and protects the products and equipment.

A common and beneficial use of derivative error limit is in parts press applications. For example, consider the operation of a small parts press. In this case, derivative error limit can be used to capture the amount of applied torque before the breakage point is exceeded. The derivative error limit can also be set to a very low value, so once contact is made, the limit will be reached and the motor will stop.

Because a derivative error limit monitors how quickly the rate of following error increases over time, it catches the fault much faster than a traditional position error limit, which saves time and protects the products and equipment.

For a detailed example of derivative error limit, see ***Application Note: Derivative Error Limit*** on page 11.

Fault Stop Action

> *The ability to specify the fault stop action for the application offers the machine designer flexibility, as well as added protection for the equipment and products in process, that would not be available with simple drives.*

Some drives have the option of specifying the automatic action that will take place when a specific condition (such as a limit) is reached. Most drives will specify in the product manual what the default action is, such as the default action for exceeding overtravel limits. The ability to specify the fault stop action for the application offers the machine designer flexibility, as well as added protection for the equipment and products in process, that would not be available with simple drives. The following are the three most common actions that can be specified in the case of a fault.

Dynamic Braking or Mode Torque Brake – Dynamic braking is common to the motion control industry but is not always associated with servo motors. Dynamic braking shorts out the windings of the motor to immediately stop any motion of the motor shaft. In some cases dynamic braking is the default action for following error limits, derivative error limits, overtravel limits, thermal limits and overcurrent limits. Dynamic braking uses the motor's own back EMF to bring it to a sudden stop. Note, however, that it may slowly drift because nothing is actively holding it in place after the motion is stopped. For an application example of dynamic braking (mode torque brake), see ***Application Note: Dynamic Braking (Mode Torque Brake), Trajectory Overshoot Braking & Freewheel*** on page 12.

Decelerate to Stop – For certain applications, stopping abruptly (as with dynamic braking) is not an ideal situation. For example, a centrifuge application that is quickly stopped may cause a sudden shift in the load or liquid to spill. In this case, decelerating to a stop would be the preferred fault action. (While decelerating to a stop, the machine keeps moving until it goes through its specified deceleration move and then holds the end position.) Vertical load applications are another example where decelerating to a stop may be the ideal action because holding a load after deceleration may prevent further damage to the machine or products. However, decelerating to a stop is typically not the ideal solution for products that are being pressed or squeezed.

Freewheel – This method releases the load so the shaft can freely spin. After the fault occurs, the moment of inertia of the load will keep it moving until friction or gravity slows the load to a stop (similar to placing a moving car into neutral instead of using the brakes to stop). Note that freewheeling is not recommended for vertical applications. For an application example using the freewheel method, see ***Application Note: Dynamic Braking (Mode Torque Brake), Trajectory Overshoot Braking & Freewheel*** on page 12.

Trajectory Overshoot Braking

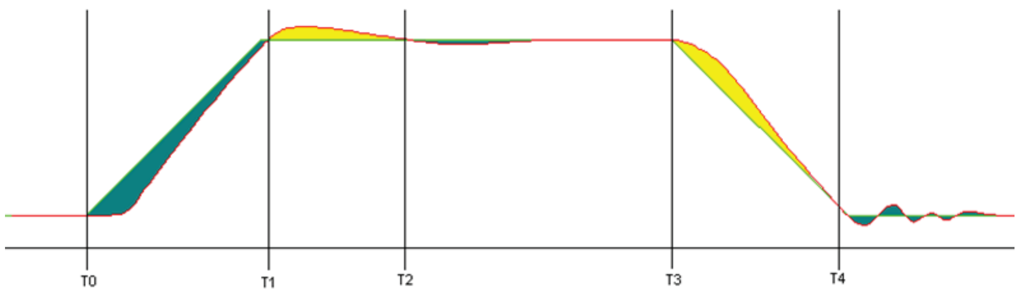
Trajectory overshoot braking (TOB) is a special method of commutation drive control that detects and corrects shaft movement only at points where the actual motion exceeds the commanded trajectory path. In most motor drive systems, when the speed of the shaft exceeds the commanded trajectory, the system will drive current in the opposite direction to correct the overshoot, or the speed will exceed the set velocity limit and the system will fault.

With TOB, when dynamic position exceeds the commanded trajectory, the drive switches from normal commutation mode to dynamic braking mode. For example, the Moog Animatics' SmartMotor TOB update rate is equal to the PID scan rate and applies to the entire trajectory path in any closed-loop operating mode.

TOB gives the system 60% more stopping power than standard deceleration methods. Even in applications with a high moment of inertia mismatch, it provides a smooth but powerful deceleration to a stop. When compared to velocity or following error limits, which fault and stop the system when there's a velocity overshoot, TOB decreases the chance that the motor will reach that fault condition, thereby increasing throughput and productivity. Also, because the system uses the motor's own power to stop versus power from the power supply, this method is more energy efficient.

TOB gives the system 60% more stopping power than standard deceleration methods. Even in applications with a high moment of inertia mismatch, it provides a smooth but powerful deceleration to a stop.

The following graph shows a typical Trapezoidal velocity move profile. The green line depicts the calculated trajectory; the red line depicts an exaggerated, but more realistic, servo move.



- At time T0, the command is given to start the move. From T0 to T1, the shaft begins to accelerate but lags behind the commanded trajectory. The motor speed increases to catch up with the commanded trajectory, eventually reaching a higher velocity than the commanded velocity (which could exceed the velocity limit and cause a fault).
- At time T1, the speed of the shaft exceeds the commanded speed. With TOB, the drive cycles between normal mode and dynamic braking mode until the shaft speed is in line with the commanded speed.
- At time T2, the motor speed is closing in on the command velocity and exactly matches it at the midpoint between T2 and T3.
- At time T3, there is a commanded deceleration and the shaft overshoots the velocity once again. However, with TOB (through the same dynamic braking method), the result is a very fast correction to the overshoot, which provides exceptionally smooth and quiet deceleration to zero.

For an application example of trajectory overshoot braking, see ***Application Note: Dynamic Braking, Trajectory Overshoot Braking & Freewheel*** on page 12.

Application Note

Derivative Error Limit

Industry:	Sheet Metal Fabrication
Application:	Electric Resistance Welding
Challenge:	Improve the quality of the welding process by ensuring equal pressure of the resistive welding clamp across two metal pieces being bonded together.

Situation

Resistive welding clamps are used to spot weld two pieces of sheet metal together (such as on a car body). The quality of the weld must be held to strict tolerances to avoid fatigue failure in the field.

Problem

As the resistive welding clamps tighten down on the two pieces of sheet metal, pressure should be equally applied to both surfaces. Equalizing actuators try to center the clamping force between the two pieces of metal. As soon as one half of the clamp meets the metal, that half (the equalizing half) stops moving until the other half of the clamp contacts the opposite side. Optimally, the clamp is attempting to avoid any bending of the metal in either direction. If either side of the clamp applies more force than the other, the sheet metal will be stressed, and the weld could later fail or the sheet metal could crack around the weld.

Solution

There are two possible solutions for this application: position limit and derivative error limit. For position limit, if the following error is set to 500 encoder counts and it's moving at 500 revolutions per second, it will take one second before the position limit is reached. Unfortunately, that could be enough time to deform the sheet metal.

Therefore, the preferred solution would be to use derivative error limit (such as with the Moog Animatics' SmartMotor). Instead of monitoring how far the encoder count has fallen behind the expected position, the controller looks at how fast the following error is increasing. So, instead of building 500 encoder counts of error and possibly pressing into the sheet metal with too much force, derivative error limit reaches the fault and stops motion within one to two servo samples (a matter of microseconds).

In this reduced time, the clamps will not dent or deform the metal. This improves the quality of the weld and ultimately increases productivity by increasing throughput and lessening the chance of a fault, which would cause a stop in production.

Application Note

Dynamic Braking, Trajectory Overshoot Braking & Freewheel

Industry:	Architecture
Application:	Automated Sliding Glass Doors
Challenge:	Protect the store patrons, employees, and the large (and expensive) glass doors; provide smooth, safe, quick-stop motion capabilities; deliver high torque with quiet operation for a retail environment.

Situation

An architect specified very large glass doors for the retail entrance of a well-known consumer product technology leader headquartered in Silicon Valley. Each door was 10 ft. by 8 ft. and weighed about 2000 pounds. The doors were programmed to be controlled either by a manual switch or by sensors for automated operation during the store's business hours.

Problem

The large doors required a significant amount of torque to open and close. However, to protect the doors (as well as any store patrons or employees), the motion control systems operating the doors needed to respond properly and quickly when operated manually or in emergency exit situations. For aesthetic reasons, the company didn't want an external controller to operate the doors, despite the complex requirements.

Solution

The architect and door manufacturer chose the Moog Animatics' SmartMotor because of its programmable protection features and integrated design (i.e., no external controller is needed). The customers wanted the doors to detect if something or someone was obstructing the path of motion, which was accomplished using following error and derivative error limit to stop in the quickest manner possible. In the case that the derivative error limit was reached because of an obstruction, the SmartMotor on each door was programmed to use dynamic braking (also known in the SmartMotor as Mode Torque Brake). Dynamic braking ensured that no more active force would be applied to the obstruction once it was detected.

However, dynamic braking normally applies a counter force once motion has stopped. In an emergency situation the doors needed to have the option of being manually pushed open. Therefore, the SmartMotor was programmed to turn off dynamic braking after coming to a stop and then to freewheel.

In the situation where the door was in motion but needed to stop and move in the opposite direction (for example, if commanded by the wall switch), the doors used trajectory overshoot braking to quickly and efficiently decelerate to a stop and accelerate in the opposite direction without exceeding any following error or velocity limits.

The combination of these three programmable protection features allowed the SmartMotor to operate effectively and safely with no external controller and minimal cabling. Further, it accomplished this while maintaining the sleek, aesthetic appearance required by the technology company's brand.

To learn more about how Moog Animatics can offer programmable protection in your application, please call 408.965.3320 or email us at sales@animatics.com.

About Moog Animatics

Since 1987, Moog Animatics has been designing, manufacturing and marketing motion control products. We bring total automation solutions to numerous industries, including semiconductor, defense, automotive, aerospace, biomedical, textile, security, marine sciences, packaging and many more.

When you need an innovative solution, you need Moog Animatics. We pride ourselves on offering the most creative and complete answers to your motion control questions.

The Moog Animatics headquarters is located in the heart of Silicon Valley, with international offices in Germany and Japan, and a vast network of Moog Animatics-trained Automation Solution Providers around the world.

For more information on Moog Animatics or to discuss your application requirements, please use the contact information provided below.

Moog Animatics
1421 McCarthy Boulevard
Milpitas, CA 95035
Tel: (408) 965 3320   Fax: (408) 965 3319
Email: sales@animatics.com
www.animatics.com