C H A P T E R ④

Servo Tuning

In a Hurry?

You should tune the 6270 before attempting to execute any motion functions. At a minimum, complete this chapter's *Tuning Setup Procedure* and *Controller Tuning Procedures* until you have found a proportional feedback gain that can give a stable response for your system. (The *Drive Tuning Procedure* below is for use with velocity drive systems only, not for servo valve systems.) Then you can proceed to execute your motion functions. To gain a full understanding of tuning, you should read through this entire chapter and follow its procedures to ensure your system is properly tuned.

Servo Tuning Software Available

To effectively tune the 6270 (and any velocity drives you may be using), use the interactive tuning features in the *Servo Tuner* add-on module for Motion Architect. It greatly improves your efficiency and gives you powerful graphical tools to measure the performance of the system. Instructions for using Servo Tuner to tune the 6270 are provided in the ***Servo Tuner User Guide***.

The Servo Tuner option is an add-on module and does not automatically come with the basic Motion Architect software package. To order your copy of Servo Tuner, which is provided on a separate disk, contact your local Automation Technology Center.

Servo System Terminology

This section gives you with an overall understanding of the principles and the terminology used in tuning the 6270.

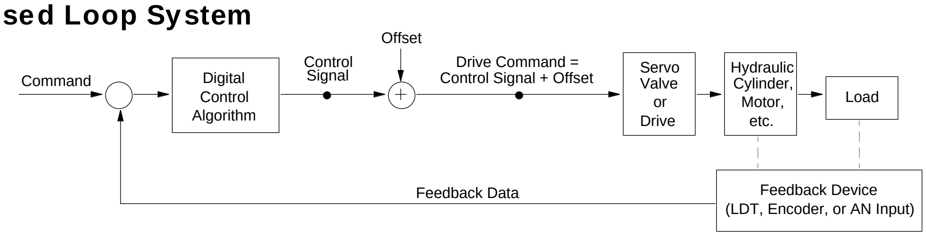
Servo Tuning Terminology

The 6270 uses a digital control algorithm to control and maintain the position and velocity. The digital control algorithm consists of a set of numerical equations used to periodically (once every **servo sampling period**) calculate the value of the **control signal** output. The numerical terms of the equations consist of the current commanded and actual position values (plus a few from the past sampling period) and a set of control parameters. Each control parameter, commonly called a **gain**, has a specific function (see *Servo Control Techniques* later in this chapter). Groups of gains may be saved to specified **gain sets** that can be invoked to affect motion under varied conditions at different times. **Tuning** is the process of selecting and adjusting these gains and gain sets to achieve optimal servo performance.

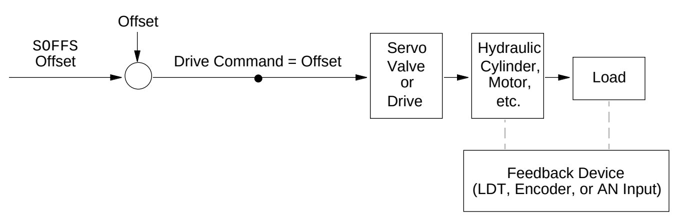
When this control algorithm is used, the whole servo system is a **closed-loop** system (see diagram below). It is called closed loop because the control algorithm accounts for both the **command** (position, velocity, tension, etc.) and the **feedback data** (from the LDT, encoder, or ANI input); therefore, it forms a *closed loop* of information flow.

When all gains are set to zero, the digital control algorithm is essentially disabled and the system becomes an **open loop** system (see diagram below). During system setup or troubleshooting, it is desirable to run the system in open loop so that you can independently test the drive/motor or valve operation (further details are provided in the *Open Loop Operation* section later in this chapter).

Closed Loop System



Open Loop System



The 6270 has the capability of providing an analog voltage output of ±10V or a current of ±20mA, ±50mA, ±60mA, ±80mA, ±100mA, or ±150mA for commanding the valve or drive. After the digital control algorithm has calculated the digital control signal, this digital value is sent out from the DSP (digital signal processor) to the Digital-to-Analog converter (DAC). The DAC has an analog output range of -10V to +10V or maximum current. It is often possible that the digital control signal calculated by the control algorithm can exceed this limit. When this happens, the analog output would just stay, or *saturate*, at the maximum limit until the position error changes such that the control algorithm would calculate a control signal less than the limit. This phenomenon of reaching the output limit is called **controller output saturation**. When saturation occurs, increasing the gains does not help improve performance since the DAC is already operating at its maximum level.

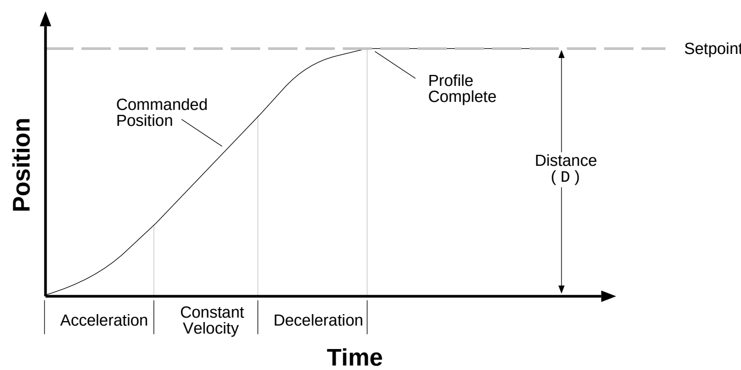
Position Variable Terminology

In a servo system, there are two types of **time-varying** (value changes with time) position information used by the controller for control purposes: commanded position and actual position. You can use this information to determine if the system is positioning as you expect.

Commanded Position

The **commanded position** is calculated by the motion profile routine based on the acceleration (A, AA), deceleration (AD, ADA), velocity (V) and distance (D) command values and it is updated every servo sampling period. Therefore, the commanded position is the intended position at any given point of time. To view the commanded position, use the TPC (Transfer Commanded Position) command; the response represents the commanded position at the instant the command is received.

When this user guide refers to the *commanded position*, it means this calculated time-varying commanded position, not the distance (D) command. Conversely, when this user guide refers to the **position setpoint**, it means the final intended distance specified with the distance (D) command. The following plot is a typical profile of the commanded position in preset (MCØ) mode.



Actual Position

The other type of time-varying position information is the **actual position**; that is, the actual position of the motor (or load or cylinder, etc.) measured with the feedback device (LDT, encoder, or ANI input). Since this is the position achieved when the drive/valve responds to the commanded position, we call the overall picture of the actual position over time the **position response** (see further discussion under *Servo Response Terminology*).

To view the actual position, use the TFB (Transfer Position of Feedback Device) command; the response represents the actual position at the instant the command is received. The goal of tuning the servo system is to get the actual position to track the commanded position as closely as possible.

The difference between the commanded position and actual position is the **position error**. To view the position error, use the TPER (Transfer Position Error) command; the response represents the position error at the instant the command is received. When the motor/valve is not moving, the position error at that time is called the **steady-state position error** (see definition of steady-state under *Servo Response Terminology*). If a position error occurs when the motor/valve is moving, it is called the **position tracking error**.

In some cases, even when the system is properly tuned, the position error can still be quite significant due to a combination of factors such as the desired profile, the servo mechanism's limitation, the dynamic characteristics of the system, etc. For example, if the value of the velocity (V) command is higher than the maximum velocity the hydraulic cylinder (or motor, etc.) can physically achieve, then when it is commanded to travel at this velocity, the actual position will always lag behind the commanded position and a position error will accumulate, no matter how high the gains are.

Servo Response Terminology

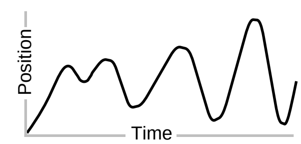
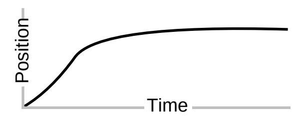
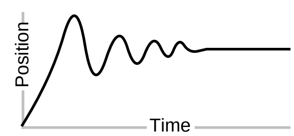
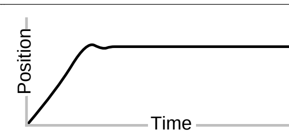
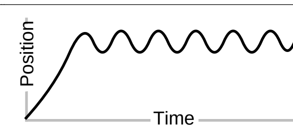
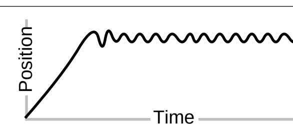
Stability

The first objective of tuning is to stabilize the system. The formal definition of **system stability** is that when a bounded input is introduced to the system, the output of the system is also bounded. What this means to a motion control system is that if the system is **stable**, then when the position setpoint is a finite value, the final actual position of the system is also a finite value.

On the other hand, if the system is **unstable**, then no matter how small the position setpoint or how little a disturbance (motor torque variation, load change, noise from the feedback device, etc.) the system receives, the position error will increase continuously, and exponentially in almost all cases. In practice, when the system experiences instability, the actual position will oscillate in an exponentially diverging fashion as shown in the drawing below. The definition here might contradict what some might perceive. One common perception shared by many is that whenever there is oscillation, the system is unstable. However, if the oscillation finally diminishes (damps out), even if it takes a long time, the system is still considered stable. The reason for this clarification is to avoid misinterpretation of what this user guide describes in the following sections.

Position Response Types

The following table lists, describes, and illustrates the six basic types of position responses. The primary difference among these responses is due to **damping**, which is the suppression (or cancellation) of oscillation.

Response	**Description**	**Profile (position/time)**
Unstable	Instability causes the position to oscillate in an exponentially diverging fashion.	
Over-damped	A highly damped, or *over-damped*, system gives a smooth but slower response.	
Under-damped	A slightly damped, or *under-damped*, system gives a slightly oscillatory response.	
Critically damped	A critically-damped response is the most desirable because it optimizes the trade-off between damping and speed of response.	
Oscillatory	An oscillatory response is characterized by a sustained position oscillation of equal amplitude.	
Chattering	Chattering is a high-frequency, low-amplitude oscillation which is usually audible.	

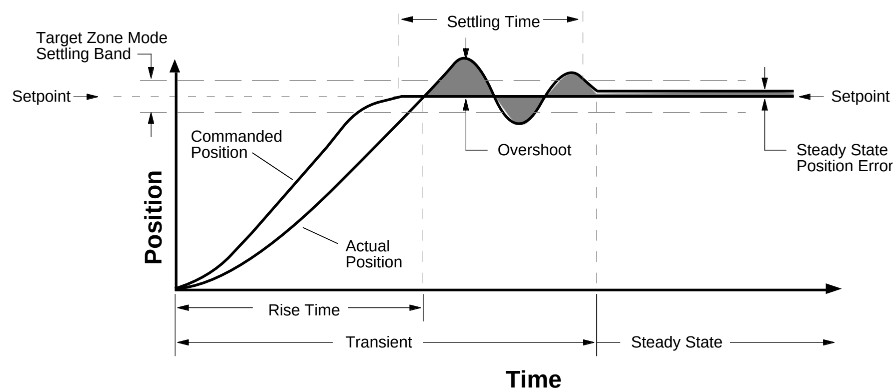
Performance Measurements

When we investigate the plot of the position response versus time, there are a few measurements that you can make to quantitatively assess the performance of the servo:

- **Overshoot**—the measurement of the maximum magnitude that the actual position exceeds the position setpoint. It is usually measured in terms of the percentage of the setpoint value.
- **Rise Time**—the time it takes the actual position to pass the setpoint.
- **Settling Time**—the time between when the commanded position reaches the setpoint and the actual position settles within a certain percentage of the position setpoint. (Note the settling time definition here is different from that of a control engineering text book, but the goal of the performance measurement is still intact.)

These three measurements are made before or shortly after the hydraulic cylinder (or motor, etc.) stops moving. When it is moving to reach and settle to the setpoint, we call such period of time the **transient**. When it is not moving, it is defined as in **steady-state**.

A typical stable position response plot in preset mode (MCØ) is shown below.



6000 Series Servo Commands

NOTE

The following list briefly describes each servo-related 6000 Series command. More detailed information can be found in the rest of this chapter and within each command's description in the ***6000 Series Software Reference Guide***.

Command	Title	Brief Description (detailed descriptions in ***6000 Series Software Reference Guide***)
LDTUPD	*LDT Position Update Rate*	Use LDTUPD to select the rate at which the LDT position is sampled. Decreasing the LDTUPD command value (speeding up the update rate)will improve the quality of the dynamic response. However, if the update rate is too fast, the LDT will not have enough time to read the position, resulting in read errors. The occurrence of read errors can be monitored with the TER and [ER] commands if you enable ERROR bit #15, and you can check the LDT status with bit #27 of TAS and AS.
SFB	*Select Servo Feedback Source*	Selects the servo feedback transducer (options: LDT, encoder, or ANI analog input). The SFBØ command sets all the gains to zero so that the controller runs in open loop mode, and disables the Setpoint Window feature (equivalent to SSWGØ)
SGAF & SGAFN *	*Acceleration Feedforward Gain*	Sets the acceleration feedforward gain in the PIV&$\mathbf{F_a}$ servo algorithm.
SGENB	*Servo Gain Set Enable*	Enables a previously-saved set of PIV&F gains. A set of gains (specific to the current feedback source selected with the SFB command) is saved using the SGSET command.
SGI & SGIN *	*Set Integral Feedback Gain*	Sets the integral gain in the P**I**V&F servo algorithm.
SGILIM	*Set Integral Windup Limit*	Sets a limit on the correctional control signal that results from the integral gain action trying to compensate for a position error that persists too long.
SGP & SGPN *	*Proportional Feedback Gain*	Sets the proportional gain in the **P**IV&F servo algorithm.
SGSET	*Save a Set of Servo Gains*	Saves the presently-defined set of PIV&F gains as a particular *gain set* (specific to the current feedback source on each axis). Up to 5 gain sets can be saved and enabled at any point in a move profile, allowing different gains at different points in the profile.
SGV & SGVN *	*Set Velocity Feedback Gain*	Sets the velocity gain in the PI**V**&F servo algorithm.
SGVF & SGVFN *	*Velocity Feedforward Gain*	Sets the velocity feedforward gain in the PIV&$\mathbf{F_v}$ servo algorithm.
SMPER	*Maximum Allowable Position Error*	Sets the maximum allowable error between the commanded position and the actual position as indicated by the feedback device. If the error exceeds this limit, the 6270 activates the Shutdown output and sets the DAC output to zero (plus any SOFFS offset). If there is no offset, a valve will return to the null position and a rotary motor will freewheel to a stop. You can enable the ERROR command to continually check for this error condition, and when it occurs to branch to a programmed response defined in the ERRORP program.
SOFFS & SOFFSN	*Servo Control Signal Offset*	Sets an offset to the commanded analog output voltage, which is sent to the drive system. The SOFFSN command allows you to set an offset voltage when the position error is negative. The SOFFSN value tracks the SOFFS value until a separate SOFFSN value is entered. To return SOFFSN to the default mode in which it track SOFFS, issue the SOFFSN command with a minus sign (-) in the command field for the affected axis (e.g., SOFFSN,- restores axis 2 to the default mode).

SSFR	*Servo Frequency Ratio*	Sets the ratio between the update rate of the move trajectory and the update rate of the servo action. The intermediate position setpoints calculated by the trajectory generator is updated at a slower rate then the servo position correction. This command allows you to optimize this for your application. The default setting (SSF4) is sufficient for most applications.
SSWD & SSWG	*Setpoint Window Gains*	You may now activate a specified set of gains to be used when with a defined region either side of the position setpoint ("setpoint window"). The SSWD command is used to specify the distance on both sides of the position setpoint ("setpoint window") in which the gain set specified with the SSWG command is used. Specifically, the gain set is automatically invoked by the controller *after* the commanded move profile is complete and the actual position is within the setpoint window. The setpoint window includes a hysteresis loop equal to 25% of the value used in the SSWD command. NOTE: The gain set to be used must first be defined with the SGSET command. For more information, refer to the *Setpoint Window Gains* section later in this chapter.
STRGTD STRGTE STRGTT STRGTV	*Target Zone Distance* *Target Zone Mode Enable* *Target Zone Timeout Period* *Target Zone Velocity*	When using the Target Zone Mode, enabled with the STRGTE command, the actual position and actual velocity must be within the *target zone* (that is, within the distance zone defined by STRGTD and within the velocity zone defined by STRGTV). If the motor/load does not settle into the target zone before the timeout period set by STRGTT, the 6270 detects an error. To prevent subsequent commands/moves from being executed when this error condition occurs, you must enable the ERROR command to continually check for this error condition, and when it occurs to branch to a programmed response defined in the ERRORP program. Otherwise, subsequent commands/moves can be executed regardless of the actual position and velocity. This feature is explained in greater detail later in the *Target Zone* section.
TDAC & DAC	*Value of DAC Output*	Transfers [or assigns/compares] the output from the 6270's digital-to-analog converter. This is the analog control signal output at the 6270's CMD terminal.
TFB & FB	*Position of Servo Feedback Devices*	Transfers [or assigns/compares] the actual position of the transducers selected for feedback (see SFB).
TGAIN	*Transfer Servo Gains*	Transfers the currently active set of PIV&F gains. The servo gain set reported represents the last gain values specified with the individual servo gain commands (SGI, SGP, SGV, SGAF, and SGVF), or the last gain set enabled with the SGSET command.
TPC & PC	*Position Commanded*	Transfers [or assigns/compares] the commanded position (intermediate position setpoint) to the drive or valve.
TPER & PER	*Position Error*	Transfers [or assigns/compares] the error between the commanded position (TPC) and the actual position (TFB, or TPE, TLDT, TANI) as measured by the feedback device.
TSGSET	*Transfer Servo Gain Set*	Transfers a previously-saved set of servo gain parameters. A gain set is saved with the SGSET command.
TSTLT	*Transfer Servo Settling Time*	Transfers the time it took the last move to settle within the *target zone* (that is, within the distance zone defined by STRGTD and within the velocity zone defined by STRGTV). The Target Zone Mode does not need to be enabled to use this command.

* The negative gain commands (SGAFN, SGIN, SGPN, SGVN, and SGVFN) allow you to establish gains to be used when the position error is negative. The negative gain value will track the positive gain value until a separate negative gain value is entered. To re-establish the default mode where the negative gain tracks the positive gain, issue the negative gain command with a minus sign (-) in the command field for the affected axis (e.g., SGPN,- restores axis 2 to the default mode).

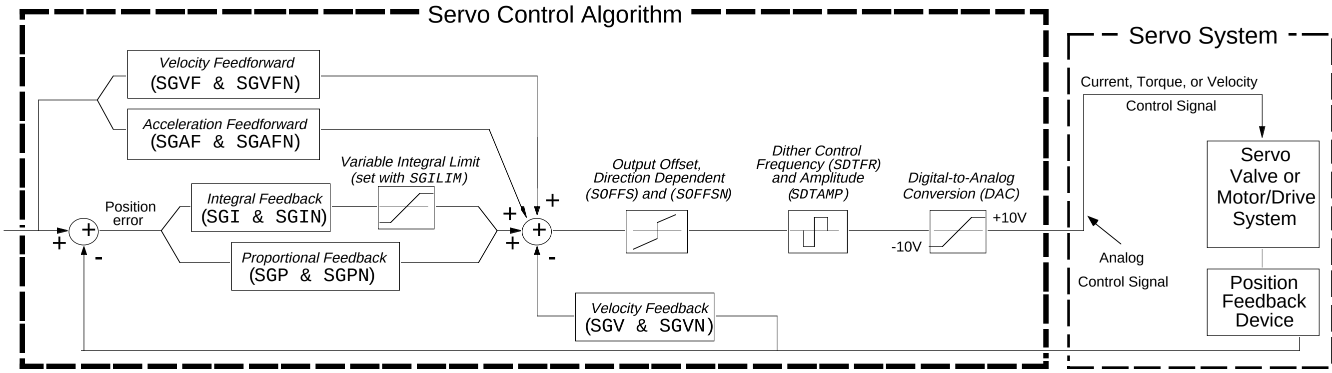
Servo Control Techniques

To ensure that you are tuning your servo system properly, you should understand the tuning techniques described in this section.

The 6270 employs a *PIV&F* servo control algorithm. The control techniques available in this system are as follows:

P.....Proportional Feedback (controlled with the SGP and SGPN commands)
I......Integral Feedback (controlled with the SGI and SGIN commands)
V.....Velocity Feedback (controlled with the SGV and SGVN commands)
F.....Velocity and Acceleration Feedforward (controlled by the SGVF and SGAF and SGVFN and SGAFN commands, respectively)

The block diagram below shows these control techniques in relation to the servo control algorithm configuration. The following table presents a condensed summary of each control's effect on the servo system.



Gain	Stability	Damping	Disturbance Rejection	Steady State Error	Tracking Error
Proportional (SGP and SGPN)	Improve	Improve	Improve	Improve	Improve
Integral (SGI and SGIN)	Degrade	Degrade	Improve	Improve	Improve
Velocity Feedback (SGV and SGVN)	Improve	Improve	-------------	-------------	Degrade
Velocity Feedforward (SGVF and SGVFN)	-------------	-------------	-------------	-------------	Improve
Acceleration Feedforward (SGAF and SGAFN)	-------------	-------------	-------------	-------------	Improve

Proportional Feedback Control (SGP)

NOTE: *The proportional feedback gain (SGP) should never be set to zero, except when open-loop operation is desired.*

***Proportional feedback* is the most important feedback for stabilizing a servo system.** When the 6270 uses *proportional feedback*, the control signal is linearly proportional to the position error (the difference between the commanded position and the actual position—see TPER command). The proportional gain is set by the Servo Gain Proportional (SGP) command. Proportional feedback can be used to make the servo system more responsive, as well as reduce the steady state position error.

Since the control is proportional to the position error, whenever there is any disturbance (such as torque ripple or a spring load) forcing the load away from its commanded position, the proportional control can immediately output a signal to move it back toward the commanded position. This function is called *disturbance rejection*.

If you tune your system using only the proportional feedback, increasing the proportional feedback gain (SGP value) too much will cause the system response to be oscillatory, underdamped, or in some cases unstable.

Integral Feedback Control (SGI)

Using *integral feedback control*, the value of the control signal is integrated at a rate proportional to the feedback device position error. The rate of integration is set by the Servo Gain Integral (SGI) command.

In most hydraulic servo applications, it is best to set the SGI gain to zero (SGIØ,Ø) during the move. Then, after motion has stopped, use the proper SGI gain to hold the position.

The primary function of the integral control is to overcome friction and/or gravity and to reject disturbances so that steady state position error can be minimized or eliminated. This control action is important for achieving high system accuracy. *However, if you can achieve acceptable position accuracy by using only the proportional feedback (SGP), then there is no need to use the integral feedback control.*

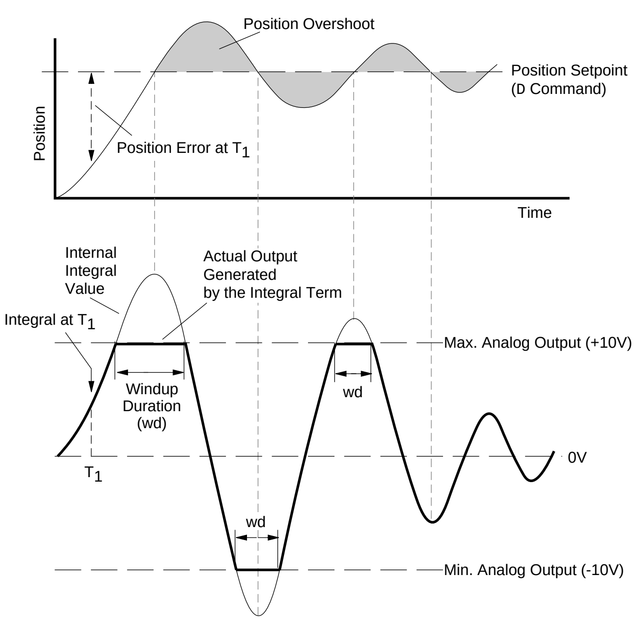
In the task of reducing position error, the integral gain (SGI) works differently than the proportional gain (SGP); this is because the magnitude of its control signal is not dependent on the magnitude of the position error as in the case of proportional feedback. If any position error persists, then the output of the integral term will ramp up over time until it is high enough to drive the error back to zero. Therefore, even a very small position error can be eliminated by the integral feedback control. By the same principle, integral feedback control can also reduce the tracking error when the system is commanded to cruise at constant velocity.

Controlling Integral Windup

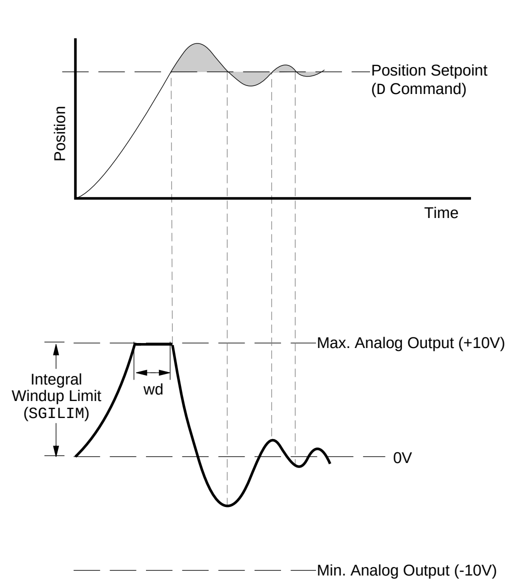
If integral control (SGI) is used and an appreciable position error has persisted long enough during the transient period (time taken to reach the setpoint), the control signal generated by the integral action can end up too high and saturate to the maximum level of the controller's analog control signal output. This phenomenon is called *integrator windup*.

After windup occurs, it will take a while before the integrator output returns to a level within the limit of the controller's output. Such a delay causes excessive position overshoot and oscillation. Therefore, the integral windup limit (SGILIM) command is provided for you to set the absolute limit of the integral and, in essence, turn off the integral action as soon as it reaches the limit; thus, position overshoot and oscillation can be reduced (see illustration below). The application of this feature is demonstrated in Step 4 of the *Tuning Procedure* below.

Without SGILIM



With SGILIM



Velocity Feedback Control (SGV)

When *velocity feedback control* is used, the control signal is proportional to the feedback device's velocity (rate of change of the actual position). The Servo Gain Velocity (SGV) command sets the gain, which is in turn multiplied by the feedback device's velocity to produce the control signal. Since the velocity feedback acts upon the feedback device's velocity, its control action essentially anticipates the position error and corrects it before it becomes too large. Such control tends to increase damping and improve the stability of the system.

A high velocity feedback gain (SGV) can also increase the position tracking error when traveling at constant velocity. In addition, setting the velocity feedback gain too high tends to slow down (*overdamp*) the response to a commanded position change. If a high velocity feedback gain is needed for adequate damping, you can balance the tracking error by applying velocity feedforward control (increasing the SGVF value—discussed below).

Since the feedback device's velocity is derived by differentiating the feedback device's position with a finite resolution, the finite word truncation effect and any fluctuation of the feedback device's position would be highly magnified in the velocity value, and even more so when multiplied by a high velocity feedback gain. When the value of the velocity feedback gain has reached such a limit, the motor (or hydraulic cylinder, etc.) will *chatter* (high-frequency, low-amplitude oscillation) at steady state.

Velocity Feedforward Control (SGVF)

The purpose of velocity feedforward control is to improve *tracking performance*; that is, reduce the position error when the system is commanded to move at constant velocity. The tracking error is mainly attributed to three sources—friction, torque load, and velocity feedback control (SGV).

Velocity feedforward control is directed by the Servo Gain Velocity Feedforward (SGVF) setting, which is in turn multiplied by the rate of change (velocity) of the commanded position to produce the control signal. Consequently, because the control signal is now proportional to the velocity of the commanded position, the 6270 essentially anticipates the commanded position and initiates a control signal ahead of time to more closely follow (*track*) the commanded position.

Applications requiring linear interpolation can benefit from improved tracking performance; however, *if your application only requires short, point-to-point moves, velocity feedforward control is not necessary*.

Because velocity feedforward control is not in the servo feedback loop (see *Servo Control Algorithm* drawing above), it does not affect the servo system's stability. Therefore, there is no limit on how high the velocity feedforward gain (SGVF) can be set, except when it *saturates the control output* (tries to exceed the 6270's analog control signal range).

Acceleration Feedforward Control (SGAF)

The purpose of acceleration feedforward control is to improve position tracking performance when the system is commanded to accelerate or decelerate.

Acceleration feedforward control is directed by the Servo Gain Acceleration Feedforward (SGAF) setting, which is in turn multiplied by the acceleration of the commanded position to produce the control signal. Consequently, because the control signal is now proportional to the acceleration of the commanded position, the 6270 essentially anticipates the velocity of the commanded position and initiates a control signal ahead of time to more closely follow (*track*) the commanded position.

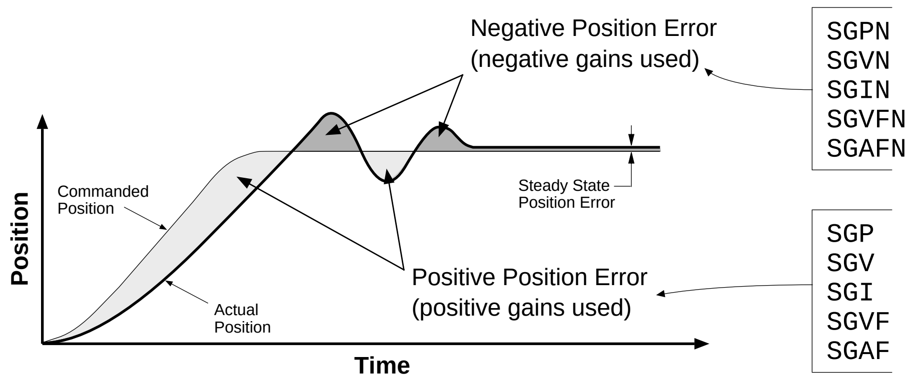
Same as velocity feedforward control, this control action can improve the performance of linear interpolation applications. In addition, it also reduces the time required to reach the commanded velocity. *However, if your application only requires short, point-to-point moves, acceleration feedforward control is not necessary.*

Acceleration feedforward control does not affect the servo system's stability, nor does it have any effect at constant velocity or at steady state.

Negative Gains

The negative gain commands (see list below) allow you to establish gains to be used when the position error is negative.

SGPN........Proportional Gain Negative
SGVN........Velocity Gain Negative
SGIN........Integral Gain Negative
SGVFN ......Velocity Feedforward Gain Negative
SGAFN ......Acceleration Feedforward Gain Negative



The 6270 automatically switches between the positive and negative gains to correlate with the positive or negative position error. This is particularly useful when controlling hydraulic cylinders in which the different surface areas on each side of the piston react differently with the same gain settings.

Each negative gain changes with (tracks) the corresponding positive gain value until a negative gain command is executed. For example, the SGPN value automatically tracks the SGP value until an SGPN command is executed with a value other than the current SGP command value. After the negative gain command is executed, separate positive and negative gain commands must be used.

To re-establish the default mode where the negative gain tracks the positive gain, issue the negative gain command with a minus sign (-) in the command field for the affected axis (e.g., SGPN,- restores axis 2 to the default mode).

Gain Sets

An added dimension to the control techniques discussed in the previous section is to group the gains into “gains sets” that can be invoked to affect motion under certain conditions. Gain sets may be useful for applications in which you would like to invoke different gains a different portions of a move profile, or at rest, or based on an external process, etc.

The SGSET command allows you to save the currently active gains, control signal offsets (SOFFS and SOFFSN), and maximum position error (SMPER) setting, to a specified gain set (see list below).

SGPN........Proportional Gain Negative
SGVN........Velocity Gain Negative

SGIN .... Integral Gain Negative
SGVFN...... Velocity Feedforward Gain Negative
SGAFN...... Acceleration Feedforward Gain Negative
SGILIM.... Integral Windup Limit
SOFFS...... Servo Control Signal Offset
SOFFSN.... Servo Control Signal Offset Negative
SMPER...... Maximum Allowable Position Error

The gain set saved with the SGSET command can be enabled/recalled later with the SGENB command or the SSWG command. Using the SGENB command, the gains can be enabled during motion at any specified point in the profile, or when not in motion (see programming example below). If using the SSWG command, the gain set is referred to as the "setpoint window gain set" and is invoked after the commanded profile is complete (see *Setpoint Window Gains* below for details).

NOTE

The tuning gains saved to a given gain are specific to the current feedback source (selected with the last SFB command) at the time the gains were saved with the SGSET command. Later, when you enable the saved gain set, **make sure that the gain set you enable is appropriate to the feedback source you are using at the time**.

To display the gain values currently in effect, use the TGAIN command. To display the contents of a particular gain set, use the TSGSET command.

Setpoint Window Gains

You may activate a specified set of gains to be used when with a defined region either side of the position setpoint ("setpoint window").

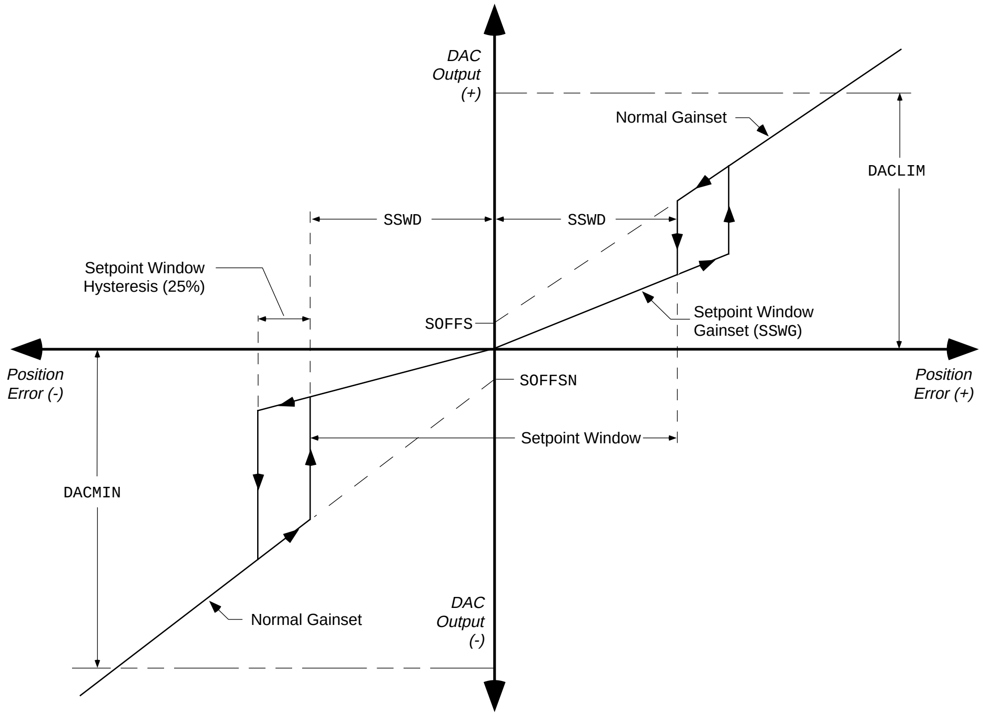
The setpoint window is defined by the SSWD command, which establishes the distance on both sides of the position setpoint ("setpoint window") in which the gain set specified with the SSWG command is used. Specifically, the gain set is automatically invoked by the controller *after* the commanded move profile is complete and the position error is within the setpoint window. The SSWG gain set accommodates different proportional, integral, and velocity gains and offsets for each direction.

The setpoint window includes a hysteresis loop equal to 25% of the value used in the SSWD command.

If you want to disable the servo control loop when the position error is within the setpoint window, set all of the gain values to zero—you can do this by executing all the individual gain commands or by executing the SFBØ command. If you want to disable the offsets (SOFFS & SOFFSN) when the position error is within the setpoint window, set them to zero volts.

To disable the setpoint window gain feature, use the SSWGØ command.

To assign a gain set as the setpoint window gain set with the SSWG command, you must first define/save the gain set with the SGSET command (see programming example below).



The diagram above makes two assumptions. First, for simplicity, only a proportional gain is being used. Second, related to the SSWG gain set, the proportional gains (SGP and SGPN) are lower than those used in the normal gain set and the offsets (SOFFS and SOFFSN) are set to zero.

The arrows on the above diagram illustrate the hysteresis loop. The SSWG gains are used until the position error increases to a value of 25% greater than the number specified in the SSWD command. At this point, the normal gain set is automatically substituted until the position error falls below the value in the SSWD command when the SSWG gains are returned.

Finally, the maximum positive DAC voltage is determined by the value in the DACLIM command and the maximum negative value is determined by the DACMIN command.

Target Zone — An alternative to SSWD

You can use the target zone settling mode to override the setpoint window distance (SSWD) and introduce distance and velocity end-of-move criteria to define when the controller switches to the setpoint window gains. When using the target zone settling mode (enabled with the STRGTE command), after completion of the commanded move profile, the actual position and actual velocity must be within the "target zone" (i.e., within the position band defined by STRGTD and within the velocity band defined by STRGTV) before motion can be determined complete. After that point, the controller will switch to the SSWG gains. For more information on the target zone mode, refer to the *Target Zone* section later in this chapter.

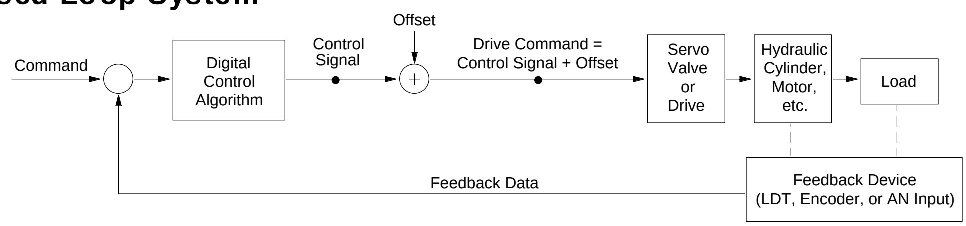
Programming Example

Example	Description
> SGP35	Set proportional gain to 35
> SGI3	Set integral gain to 3
> SGSET3	Save current gains as gain set #3 (to use when within the setpoint window)
> SSWG3	Assign gain set #3 as the setpoint window gain set
> SGP5Ø	While moving, use higher proportional gain
> SGIØ	While moving, use no integral gain,
> SGV4	While moving, use introduce velocity gain
> SSWD1ØØ	After the commanded move profile is complete, the controller will use gain set #3 if within 100 counts (125 counts including hysteresis) of the setpoint position

Open Loop Operation

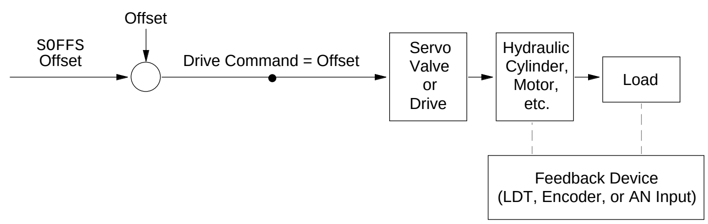
When the control algorithm is used, the whole servo system is a closed-loop system (see diagram below). It is called closed loop because the control algorithm accounts for both the command (position, velocity, tension, etc.) and the feedback data (from the LDT, encoder, or ANI input); therefore, it forms a *closed loop* of information flow.

Closed Loop System



When all gains are set to zero, the digital control algorithm is essentially disabled and the system becomes an open loop system (see diagram below). Open loop operation could be used during system setup or troubleshooting so that you can independently test the drive/motor or valve operation (the *Tuning Setup Procedure* below demonstrates open-loop operation).

Open Loop System



There are methods of entering the open loop mode:

• Disable each individual positive and negative gain value. This allows the controller to monitor the position error and signal a fault if the maximum limit is exceeded.

• Issue the SFBØ command. This automatically invokes the following conditions:

- **WARNING** — The hardware and software end-of-travel limits are disabled. Make sure that it is safe to operate without end-of-travel limits before using the open-loop function.
- Gain values (SGILIM, SGAF, SGAFN, SGI, SGIN, SGP, etc.) set to zero (open-loop operation).
- SMPER value set to zero (position error is allowed to increase without causing a fault).
- Subsequent attempts to change gain values or SMPER will cause an error message ("NOT ALLOWED IF SFBØ").
- SOFFS and SOFFSN set to zero, but allows subsequent servo offset changes to affect motion.
- SSWG set to zero (disables the Setpoint Window Gains feature).
- Disables output-on-position (OUTPA - OUTPD) functions.
- Any subsequent changes to PSET, PSETCLR, SCLD, SCLA, SCLV, SOFFS, and SOFFN are lost when another feedback source is selected.

Recommendation — Use the Disable Drive On Kill mode, enabled with the KDRIVE command, so that the 6270 will shut down the valve/drive if a kill command (e.g., !K) is executed or if a kill input is activated. **CAUTION**: Shutting down a valve/cylinder system returns the valve to the null position; shutting down a rotary drive system allows the load to freewheel if there is no brake installed.

Tuning Setup Procedure

Use the following procedure to set up your servo system before completing the tuning procedures. You can perform this procedure for both axes simultaneously.

Before you set up for tuning:

Do not begin this procedure unless you are sure you have successfully completed these system connection, test, and configuration procedures provided in Chapter 3:

- Connect the valve or the drive (especially the drive's shutdown output).
- Connect and test the feedback devices.
- Connect and test the end-of-travel limits.
- Test the 6270's analog output.
- Attach the load and the feedback devices as required for your application.
- Configure the number of axes in use, drive fault level (if using a rotary drive), and feedback device resolution.
- Select the appropriate feedback source per axis with the SFB command (tuning parameters for each axis are specific to the currently selected feedback source).

⚠ WARNING ⚠

The tuning process requires operation of your system's electrical and mechanical components. Therefore, you should test your system for safety under all potential conditions. Failure to do so can result in damage to equipment and/or serious injury to personnel.

EMERGENCY SHUTDOWN: You should be prepared to shut down the valve or drive during the tuning process (for instance, if the system becomes unstable or experiences a runaway). You can use the **ENBL** input (disconnect it from ground) to disable the 6270's analog output signal. An alternative is to issue the @DRIVEØ command to the 6270 over the communication interface, but this requires connecting a shutdown output to the drive. If the drive does not have a shutdown input, use a manual emergency stop switch to disable the valve's/drive's power supply.

Step 1 **Drive Users Only**: If you are using a rotary drive, make sure the power to the drive is off.

Step 2 Apply power to the 6270 only and issue the DRIVE11 command. Measure the 6270's analog output between the **CMD+** and **CMD-** terminals on the **DRIVE** connector with both an oscilloscope to check for noise and a digital volt-meter (DVM) to monitor the analog output. Both readings should be very close to zero. If an offset exists, ignore it for now; it will be taken care of later in step 8.

Step 3 If your system has mechanical stops, manually move the load to a position mid-way between them.

Step 4 Enter these commands to zero all the gains and run the system in open loop:

Open-Loop Operation ☞

Command	Description
> SGPØ,Ø	Set the proportional feedback gain to zero
> SGVØ,Ø	Set the velocity feedback gain to zero
> SGIØ,Ø	Set the integral feedback gain to zero
> SGVFØ,Ø	Set the velocity feedforward gain to zero
> SGAFØ,Ø	Set the acceleration feedforward gain to zero

Step 5 **Drive Users Only**: Apply power to the drive. The motor shaft should be stationary or perhaps turning very slowly (velocity drives only). *A small voltage to a torque drive, with little or no load attached, will cause it to accelerate to its maximum velocity. Since the torque demand at such a low voltage is very small, you can prevent the shaft from moving by holding it.*

Step 6 Observe the 6270's analog output noise level on the oscilloscope. Typically, the ideal noise level should be below 3.0mV, but inevitably you must determine the acceptable noise level for your application.

☞ *The control method (voltage or current output) is selected with internal jumpers. The default is voltage output. If you required current output, refer to the instructions in Chapter 6 to change the appropriate jumpers.*

If the noise level is acceptable, proceed to Step 7. If the noise level is too high:

a. Turn all the power off and tie the earth grounds of all the electrical components of your system to a single common earth ground point.

b. Shield the drive or valves properly and shield all the wiring that interconnect the components.

c. After you have completed a and b above, turn on the controller only and start over from Step 2. If the noise level is still unacceptable, consult the noise suppression techniques described in Appendix A.

Step 7 The purpose of this step is to ensure that a positive voltage on the 6270's analog control signal output (from the **CMD+** and **CMD-** terminals) results in the feedback device counting in the positive direction.

a. Using the `SMPER` command, set the maximum allowable position error to 1 inch by entering the `SMPER1,1` command. This assumes the default scaling factor (1 distance value = 432 counts, or 1 inch) is still in effect.

b. Enter the `TFB` command to check the current position of the feedback devices. Record this number for later use.

c.

> **CAUTION**
>
> The offset introduced in this step may cause an acceleration to a high speed, if there is little or no load.

Enter the `SOFFSØ.2` command to introduce an offset DAC output value of 0.2V to make the servo mechanism move slowly in the positive (clockwise or extension) direction. (Motion will stop when the maximum allowable position error is exceeded.) *If the load has a large stiction component, you may need to use a larger offset (`SOFFS` command) to overcome stiction and affect motion.*

d. Use the `TFB` command again to observe the feedback device's position. The value should have increased from the value observed in Step 7.b.

If the position reading decreases when using a positive `SOFFS` setting, turn off the 6270 (and the drive, if using one) and swap the **CMD+** and **CMD-** connections either at the 6270 or at the valve/drive, whichever is more accessible (this will not work for servo valves/drives that do not accept differential input). Then turn on the 6270 again, enter the `DRIVE11` command, and repeat Steps 4 through 7.d. before proceeding to Step 8.

*As an alternative to swapping **CMD+** and **CMD-**, you could use the appropriate feedback polarity reversal command (`LDTPOL`, `ENCPOL`, or `ANIPOL`).*

e. Enter the `SOFFSØ` command to *stop* the motor, and enter the `DRIVE11` command to re-enable the drives.

Step 8 Having set the servo output offset to zero with the `SOFFSØ` command (see Step 7.e.), read the 6270's analog output with the DVM to determine if there is any offset caused by the electrical interconnections between the 6270 and the valve or drive.

☞ *If you are using current control, convert the offset from milliamps to volts (output range in mA ÷ 10V = mA/V) and enter the result in the `SOFFS` command.*

If the DVM reads anything other than zero, enter the DVM's reading (but with the opposite polarity) as the offset adjustment with the `SOFFS` command. For example, if the DVM reading is 0.015V, then enter `SOFFS-Ø.Ø15`. If, after doing this, the reading is still not zero, then fine-tune it by trying `SOFFS` entries of slightly different values until the DVM reading is between ±3.0mV.

Step 9 **Drive Users Only**: If you are using a velocity drive, motion may still be occurring due to the drive's balance/offset setting. If so, adjust the drive's balance/offset until motion stops. Consult the drive's user documentation for instructions.

Step 10 Proceed to the *Drive Tuning Procedure* section to tune the velocity drive (if you are using a torque drive or a valve, skip to the *Controller Tuning Procedure*).

Drive Tuning Procedure (Velocity Motor Drives Only)

The goals of the *Drive Tuning Procedure* are to:

1. Tune the drive to output the desired velocity at a given voltage from the 6270.
2. Tune the drive (iteratively) to achieve the desired response.

> **NOTE**
> Be sure to complete the *Tuning Setup Procedure* before proceeding with the following drive tuning procedure. Unlike the *Tuning Setup Procedure*, you must **tune one axis at a time**.

Step 1 **Tune the drive to output the desired velocity at a given voltage from the 6270**:

a. If your system has mechanical stops, manually move the load to a position mid-way between them.

b. Enter the `SOFFS` command to set the 6270's output voltage to its maximum level, 10.0 volts (`SOFFS1Ø` for axis 1, or `SOFFS,1Ø` for axis 2).

c. Adjust the drive gain factor (sometimes called the *tach gain*) such that when the 6270's command output is 10V, the velocity just reaches its maximum value (check the velocity with the `TVELA` command). Refer to your drive's user documentation if necessary.

> **EXAMPLE**
> Suppose your drive can run at a max. velocity of 7000 rpm (or 116.67 rps). If the drive gain factor is 20 rps/V, then the drive will reach the maximum velocity (116.67 rps) when the 6270's command output is only 5.833V. This means the full range of ±10V is not fully usable. To use the full range of ±10V, the gain factor has to be adjusted to 11.667 rps/V.
>
> Drive manufactures usually provide a potentiometer for adjusting this gain factor. Some manufacturers provide a few preset values selectable with jumpers or DIP switches.

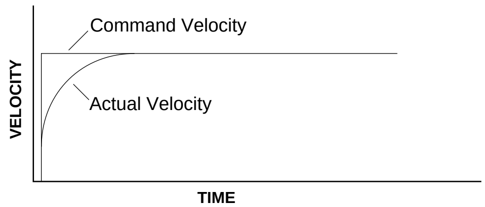
Step 2 **Tune the drive (iteratively) to achieve the desired response**:

a. Enter the following commands to create and execute a step velocity command:

Command	Description
> DEF STEP	Begin definition of the program called STEP
- @SGPØ	Set the SGP gain to zero
- @SGIØ	Set the SGI gain to zero
- @SGVØ	Set the SGV gain to zero
- @SGAFØ	Set the SGAF gain to zero
- @SGVFØ	Set the SGVF gain to zero
- @SMPERØ	Disable checking the maximum allowable position error
- @SOFFSØ.5	Set the command output to 0.5 volts
- T1	Wait for 1 second
- @SOFFSØ	Set the command output to zero volts (stopping the motor)
- @SMPER1	Re-enable checking the maximum allowable position error
- END	End definition of the program called STEP
> STEP	Execute the program called STEP (the motor will move for 1 second and then stop)

b. Observe the plot of the commanded velocity versus the actual velocity on the oscilloscope.

Using the tuning methods specified in the drive's user documentation, tune the drive to achieve a first-order response (no overshoot) as illustrated below—repeat Steps 2.a. and 3.b. as necessary.



Step 3 **Proceed to the *Controller Tuning Procedure* section to tune the 6270.**

Controller Tuning Procedure

The *Controller Tuning Procedure* leads you through the following steps:

1. Setup up for tuning.
2. Select the 6270's servo Sampling Frequency Ratios (SSFR).
3. Set the Maximum Position Error (SMPER).
4. Optimize the Proportional (SGP) and Velocity (SGV) gains.
5. Use the Integral Feedback Gain (SGI) to reduce steady state error.
6. Use the Velocity Feedforward Gain (SGVF) to reduce position error at constant velocity.
7. Use the Acceleration Feedforward Gain (SGAF) to reduce position error during acceleration and deceleration.

Before you tune the 6270:

Be sure to complete the *Tuning Setup Procedure* (and the *Drive Tuning Procedure*, if you are using a velocity drive) before proceeding with the following tuning procedure. Unlike the *Tuning Setup Procedure*, you must **tune one axis at a time**.

If your application requires switching between feedback sources on the same axis, then for each feedback source on each axis you must select the feedback source with the SFB command and repeat steps 3-7.

Step 1 **Set up for tuning**:

Use a computer (with a terminal emulator) or a dumb terminal to enter the commands noted in the steps below. To monitor system performance, you may use visual inspection, or use an analog type position transducer (potentiometer, LVDT, RVDT, etc.) to pick up the load's or motor's position displacement and monitor the transducer output on a digital storage oscilloscope.

Step 2 **Select the sampling frequency ratios (SSFR) and max. position error (SMPER)**

The 6270's control signal is computed by the digital signal processor (DSP). The velocity of the commanded position, the velocity of the feedback device's position, and the integral of the position error are used for various control actions. These measurements are derived by the DSP from the position values sampled periodically at a fixed rate; this sampling rate is called the *servo sampling frequency* (samples/second).

NOTE

The SSFR setting affects the dither frequency ratio (SDTFR setting) and the LDT position update rate (LDTUPD setting). If the sampling rate is too fast for the LDT, position errors or bad LDT reads will occur. The occurrence of read errors can be monitored with the TAS and [AS] command bit #27. Refer to the SSFR, SDTFR, and LDTUPD command descriptions in the ***6000 Series Software Reference Guide*** for more details.

Higher sampling frequencies improve the accuracy of the velocity and integral values derived. A higher sampling frequency can also improve the tracking of a rapidly changing or oscillating position. Therefore, the servo sampling frequency is a key parameter that influences the servo system's stability and closed loop bandwidth.

In addition to computing the 6270's control signal, the DSP also computes the commanded position trajectory. When the servo sampling frequency is increased, the motion trajectory update rate has to be decreased, and vice versa. The ratio between the servo sampling frequency and the trajectory update rate, called the *sampling frequency ratio*, depends on the requirements of your application and/or the dynamic characteristics of the system. The Servo Sampling Frequency Ratio (SSFR) command offers four selectable ratio settings. These four ratios and the actual sampling frequencies and sampling periods (reciprocal of sampling frequency) are shown below.

> **NOTE**
> Changing the active axes with the `INDAX` command will change the SSFR ratio.

	# of Axes Active (`INDAX`)	SSFR Command Setting	Servo Sampling Update		Motion Trajectory Update		System Update	
			Frequency (samples/sec.)	Period (μsec)	Frequency (samples/sec.)	Period (μsec)	Frequency (samples/sec.)	Period (μsec)
	INDAX1	SSFR1	3030	330	3030	330	757	1320
	INDAX1	SSFR2	4000	250	2000	500	500	2000
	INDAX1	SSFR4	4651	215	1163	860	581	1720
	INDAX1	SSFR8	4878	205	610	1640	610	1640
	INDAX2	SSFR1	1667	600	1667	600	417	2400
	INDAX2	SSFR2	2272	440	1136	880	568	1760
Default →	INDAX2	SSFR4	2500	400	625	1600	625	1600
	INDAX2	SSFR8	2667	375	333	3000	333	3000

The general rule to determining the proper SSFR value is to first select the slowest servo sampling frequency that is able to give a satisfactory response. This can be done by experiment or based on the closed-loop bandwidth requirement for your application. (Keep in mind that increasing the SSFR value allows for higher bandwidths, but produces a rougher motion profile; conversely, decreasing the SSFR value provides a smoother profile, but makes the servo system less stable and slower to respond.)

As an example, if your application requires a closed-loop bandwidth of 300 Hz, you can determine the minimum servo sampling frequency by using the rule of thumb—setting the servo sampling frequency at least 8 times higher than the bandwidth frequency—the required minimum servo sampling frequency would be 2400 Hz. If two axes are running (INDAX2), then you should try using the SSFR4 setting.

The table below provides guidelines for various application requirements.

Application Requirement	SSFR1	SSFR2	SSFR4	SSFR8
XY Linear Interpolation	✔	✔		
Fast point-to-point motion			✔	✔
Regulation (speed, torque, etc.)			✔	✔
High natural frequency system				✔

> **Setting the Sampling Frequency Ratio**
> Select a sampling ratio (with the SSFR command) appropriate to your system now, before you proceed to tune each gain.
> If you change the sampling frequency ratios (SSFR) after the tuning is complete and the new servo sampling frequency is lower than the previous one, the response may change (if your system bandwidth is quite high) and you may have to re-tune the system.

Step 3 **Set the Maximum Position Error (SMPER)**:

The SMPER command allows you to set the maximum position error allowed before an error condition occurs. The position error, monitored once per system update period, is the difference between the commanded position and the actual position as read by the feedback device selected with the last SFB command. *Larger values allow greater oscillations/motion when unstable; therefore, smaller SMPER values are safer.*

When the position error exceeds the value entered by the SMPER command, an error condition is latched (see TAS or AS bit #23) and the 6000 controller issues a shutdown to the faulted axis and sets its analog output command to zero volts. To enable the system again, the appropriate DRIVE1 command must be issued, which also sets the commanded position equal to the actual feedback device position (incremental devices will be zeroed).

If the SMPER value is set to zero, the position error condition is not monitored, allowing the position error to accumulate without causing a fault.

Step 4 **Optimize the Proportional (SGP) and Velocity (SGV) gains (*see illustration for tuning process*)**:

a. Enter the following commands to create a step input profile *(use a comma in the first data field when tuning axis 2—e.g.,* D,1ØØ*)*:

Command	Description
> A999	Set acceleration to 999 units/sec^2
> AD999	Set deceleration to 999 units/sec^2
> V3Ø	Set velocity to 30 units/sec
> D1ØØ	Set distance to 100 units

b. Start with an SGP command value of 0.5 (SGPØ.5 or SGP,Ø.5).

c. Enter the GO1 or GO,1 command depending on which axis is being tuned at the time.

d. Observe the plot of the commanded position versus the actual position on the oscilloscope. If the response is already very oscillatory, lower the gain (SGP); if it is *sluggish* (overdamped), increase the SGP gain.

Repeat Steps 4.c. and 4.d. until the response is slightly under-damped.

e. Start with an SGV command value of 0.1 (SGVØ.1 or SGV,Ø.1).

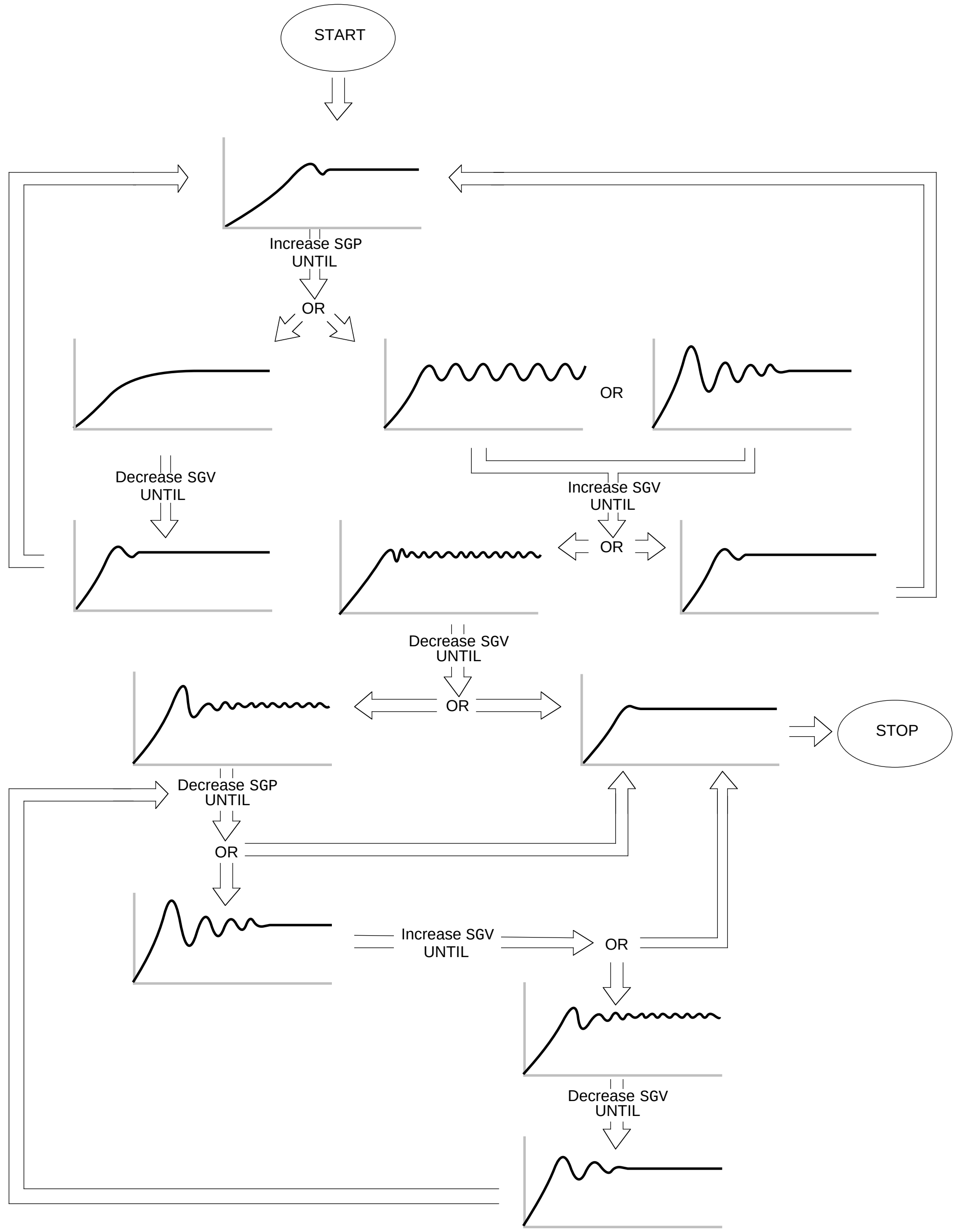
f. As you did in Step 4.c., enter GO1 or GO,1.

g. Observe the plot on the oscilloscope. If the response is *sluggish* (overdamped), reduce the SGV gain. *Repeat Steps 4.f. and 4.g. until the response is slightly under-damped.*

☞ *Refer to the* Tuning Scenario *section later in this chapter for a case example.*

h. The flow diagram below shows you how to get the values of the proportional and velocity feedback gains for the fastest, well-damped response in a step-by-step fashion. The tuning principle here is based on these four characteristics:

- Increasing the proportional gain (SGP) can speed up the response time and increase the damping.
- Increasing the velocity feedback gain (SGV) can increase the damping more so than the proportional gain can, but also may slow down the response time.
- When the SGP gain is too high, it can cause instability.
- When the SGV gain is too high, it can cause the motor (or valve, hydraulic cylinder, etc.) to chatter.



Step 5 **Use the Integral Feedback Gain (`SGI`) to reduce steady state error**:

☞ *Steady state position error is described earlier in the* Performance Measurements *section.*

a. Determine the steady state position error (the difference between the commanded position and the actual position). You can determine this error value by the `TPER` command when the load is not moving.

> **NOTE**
>
> If the steady state position error is zero or so small that it is acceptable for your application, **you do not need to use the integral gain**. For hydraulic applications, it is usually best to use a small `SGI` value, or use `SGIØ` while moving and use `SGIn` when stopped. The use of the Target Zone Settling Mode (`STRGTE`) is recommended.

b. If you have to enter the integral feedback gain to reduce the steady error, start out with a small value (e.g., SGIØ.1). After the gain is entered, observe two things from the response:

- Whether or not the magnitude of steady state error reduces
- Whether or not the steady state error reduces to zero at a faster rate

c. Keep increasing the gain to further improve these two measurements until the overshoot starts to increase and the response becomes oscillatory.

d. There are three things you can do at this point (If these three things do not work, that means the integral gain is too high and you have to lower it.):

1st Lower the integral gain (SGI) value to reduce the overshoot.

2nd Check whether the 6270's analog output saturates the ±10V limit; you can do this by observing the signal from a digital oscilloscope. If it saturates, then lower the integral output limit by using the SGILIM command. This should help reduce the overshoot and shorten the settling time. Sometimes, even if the analog output is not saturated, you can still reduce the overshoot by lowering SGILIM to a value less than the maximum output value. *However, lowering it too much can impair the effectiveness of the integral feedback.*

☞ *If you are using current control, convert the offset from milliamps to volts and enter the result in the* SGILIM *command.*

3rd You can still increase the velocity feedback gain (SGV value) further, provided that it is not already at the highest possible setting (causing the motor or valve to chatter).

Step 6 **Use the Velocity Feedforward Gain (SGVF) to reduce position error at constant speed**:

a. Execute a continuous (MC1 command) move, setting the acceleration, deceleration and velocity values appropriate to your application. Set the SGVF value to be the product of SGP ∗ SGV (if SGV = zero, set SGVF equal to SGP).

b. Check the position error at constant velocity by issuing the TPER command.

c. Increase SGVF to reduce the position error (repeat steps a and b as necessary).

Step 7 **Use the Acceleration Feedforward Gain (SGAF) to reduce position error during acceleration**:

a. Execute a continuous (MC1 command) move, setting the acceleration, deceleration and velocity values appropriate to your application. Set SGAF to 0.01 (SGAFØ.Ø1).

b. Check the position error during acceleration by issuing the TPER command.

c. Increase SGAF to reduce the position error (repeat steps a and b as necessary).

Tuning Scenario

This example shows how to obtain the highest possible proportional feedback (SGP) and velocity feedback (SGV) gains experimentally by using the flow diagram illustrated earlier in Step 4 of the *Tuning Procedure*.

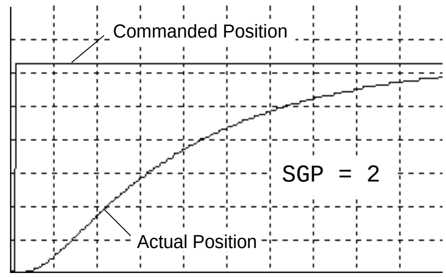
NOTE

The steps shown below (steps 1 - 11) represent the major steps of the process; the actual progression between these steps usually requires several iterations.

The motion command used for this example is a step command with a step size of 100. The plots shown are as they might appear on a scope (X axis = time, Y axis = position).

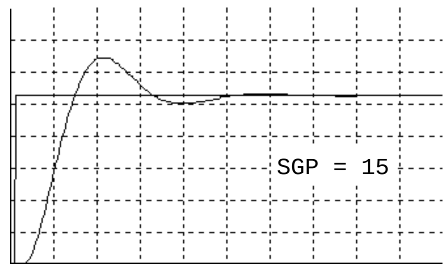
Step 1 For a starting trial, we set the proportional feedback gain (SGP) to 2. As you can see by the plot, the response is slow.

In the next step, we should increase SGP until the response is slightly underdamped.



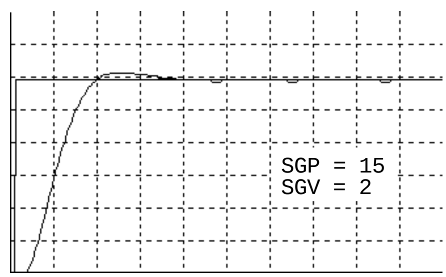
Step 2 With SGP equal to 15, the response becomes slightly underdamped (see plot).

Therefore, we should introduce the velocity feedback gain (SGV) to *damp out* the oscillation.



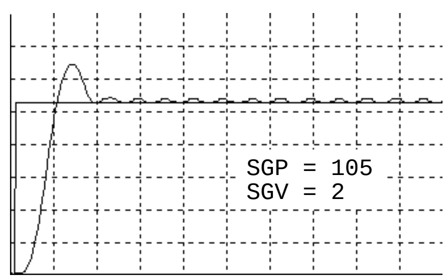
Step 3 With SGV equal to 2, the response turns out fairly well damped (see plot).

At this point, the SGP should be raised again until oscillation or excessive overshoot appears.



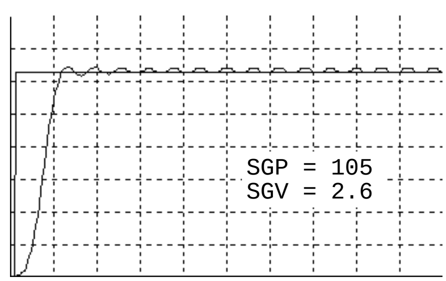
Step 4 As we iteratively increase SGP to 105, overshoot and chattering becomes significant (see plot). This means either the SGV gain is too low and/or the SGP is too high.

Next, we should try raising the SGV gain to see if it could damp out the overshoot and chattering.



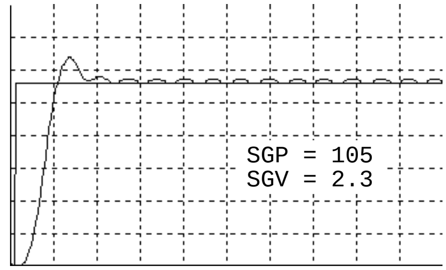
Step 5 After the SGV gain is raised to 2.6, the overshoot was reduced but chattering is still quite pronounced. This means either one or both of the gains is too high.

The next step should be to lower the SGV gain first.



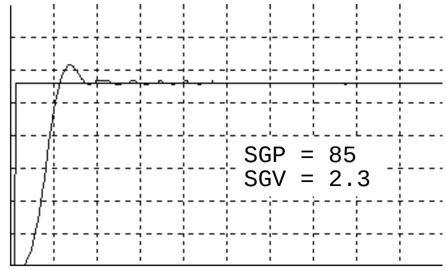
Step 6 Lowering the SGV gain to 2.3 does not help reduce the chattering by much.

Therefore, we should lower the SGP gain until chattering stops.



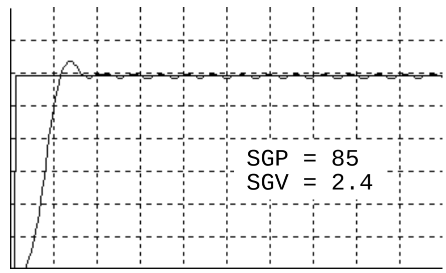
Step 7 Chattering stops after reducing the SGP gain to 85. However, the overshoot is still a little too high.

The next step should be to try raising the SGV to damp out the overshoot.



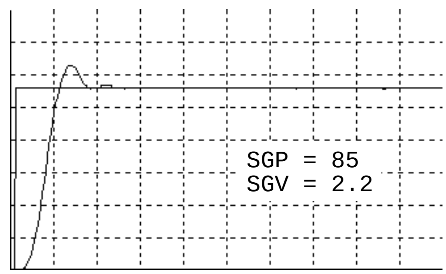
Step 8 After raising the SGV gain to 2.4, overshoot is reduced a little, but chattering reappears. This means the gains are still too high.

Next, we should lower the SGV gain until chattering stops.



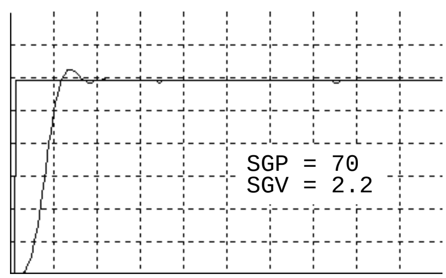
Step 9 After lowering the SGV gain to 2.2 (even less than in Step 7—2.3), chattering stops.

Next we should lower the SGP gain.

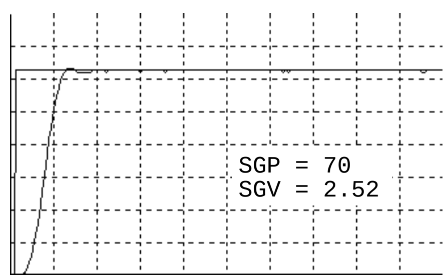


Step 10 Overshoot is reduced very little after lowering the SGP gain to 70. (The SGV gain might have been lowered too much in Step 9.)

Next, we should try raising the SGV gain again until the overshoot is gone.

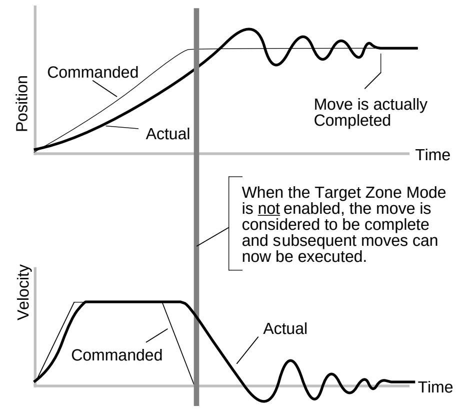


Step 11 When we raised the SGV gain to 2.52, the step response became fast and very stable.



Target Zone (Move Completion Criteria)

Under default operation (Target Zone Mode not enabled), the 6270's move completion criteria is simply derived from the move trajectory. The 6270 considers the current preset move to be complete when the commanded trajectory has reached the desired target position; after that, subsequent commands/moves can be executed for that same axis. Consequently, the next move or external operation can begin before the actual position has settled to the commanded position (see diagram).



To prevent premature command execution before the actual position settles into the commanded position, use the *Target Zone Mode*. In this mode, enabled with the STRGTE command, the move cannot be considered complete until the actual position and actual velocity are within the *target zone* (that is, within the distance zone defined by STRGTD and less than or equal to the velocity defined by STRGTV). If the load does not settle into the target zone before the timeout period set with the STRGTT command, the 6270 detects a *timeout error* (see illustration below).

Refer to the Error Handling *section in the* **6000 Series Software Reference Guide** *for error program examples*

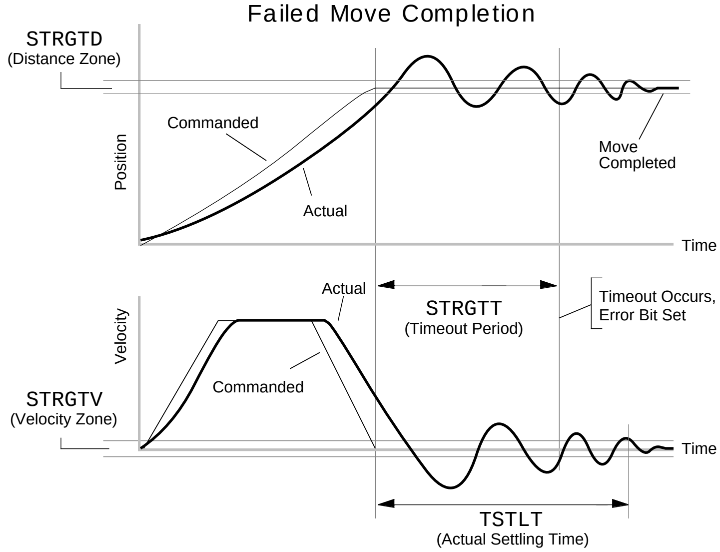
If the timeout error occurs, you can prevent subsequent command/move execution only if you enable the ERROR command to continually check for this error condition, and when it occurs to branch to a programmed response you can define in the ERRORP program.

As an example (assuming scaling is enabled and default scaling values are used), setting the distance zone to ±0.01 inches (STRGTD.Ø1), the velocity zone to ≤0.5 ips (STRGTVØ.5), and the timeout period to 1/2 second (STRGTT5ØØ), a move with a distance of 8 inches (D8) must end up between position 7.99 and 8.01 and settle down to ≤0.5 ips within 500 ms (1/2 second) after the commanded profile is complete.

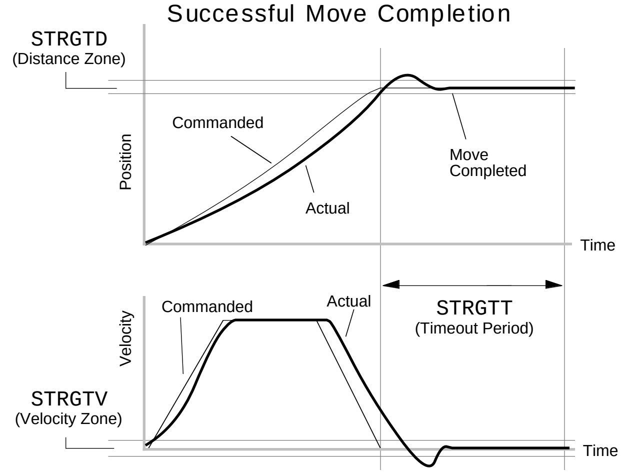
Damping is critical.

To ensure that a move settles within the distance zone, it must be damped to the point that it will not move out of the zone in an oscillatory manner. This helps ensure the actual velocity falls within the target velocity zone set with the STRGTV command (see illustration below).

Failed Move Completion



Successful Move Completion



Checking the Actual Settling Time

Using the TSTLT command, you can display the actual time it took the last move to settle into the target zone (that is, within the distance zone defined by STRGTD and less than or equal to the velocity defined by STRGTV). The reported value represents milliseconds. **This command is usable whether or not the Target Zone Settling Mode is enabled with the STRGTE command.**