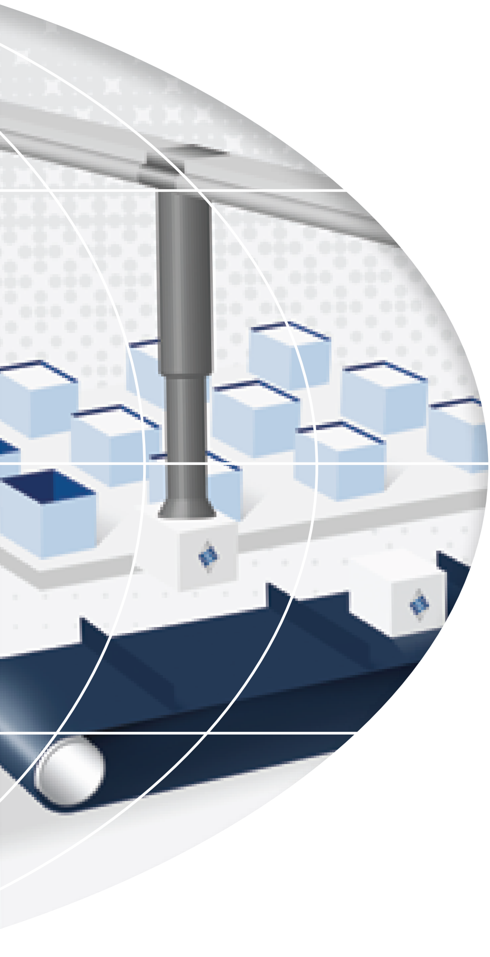


All for dreams

A Guide to
Motion Control Technology
Systems & Programming

Contents

1 Introduction to motion control

Motion control is an exciting automation technology that has become the cornerstone of modern industrial machinery design. It is about making a mechanism move under complete control, how you want, when you want. It incorporates the finer elements of motor control and requires careful mechanical design.

Motion technology is advancing to provide improved performance and increased ease of use, enabling servo and motion controls to be applied more widely than ever before. Motion control can provide greater accuracy, performance and efficiency in many production machine applications. Motion control technology can be applied to rotary servo motors, asynchronous AC and DC motors and various linear motor technologies, to provide ever more flexible and dynamic production systems.

The result is that every automation engineer should now have some background knowledge of motion control systems and their mechanical requirements. This document aims to explain the fundamentals behind the technology, and the terminology.

2 Common industrial motion applications

Motion applications call upon a varied array of engineering skills to thoroughly understand the mechanical and control system. The following three examples show how motion is used in very different applications, each requiring a different method to achieve the goal.

2.1 Indexing

Figure 2-1 shows the simplest form of indexing, a rotary table. The motor rotates the table a preset angle, known as an ‘index’, allowing the drill to operate efficiently. Figure 2-2 shows a conveyor carrying boxes that must be filled with a free flowing product. In this application the conveyor is moved forward by a set distance. When the index is completed and the conveyor has stopped, the box is filled. When the box has been filled the sequence is repeated, indefinitely.

Another similar application is for the production of stamped products from a roll of material such as sheet metal (press feeder) as shown in Figure 2-3. In applications such as this, the unwound product is often referred to as a web, as it passes through the machine. In this application, the web is moved forward a set distance, and then while stopped, a press is used to stamp the product; the process is again then repeated indefinitely.

Figure 2-1 Rotary table requiring absolute positioning

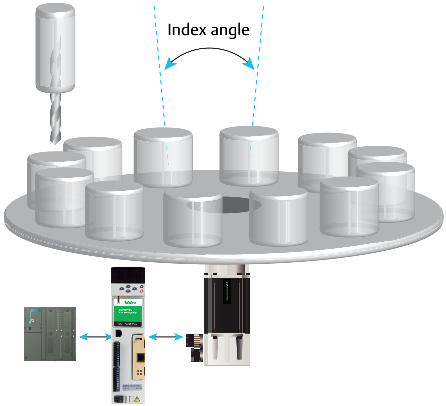


Figure 2-2 Conveyor indexing

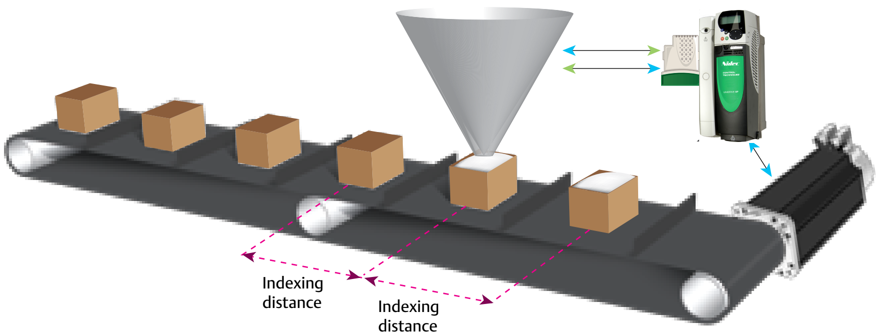
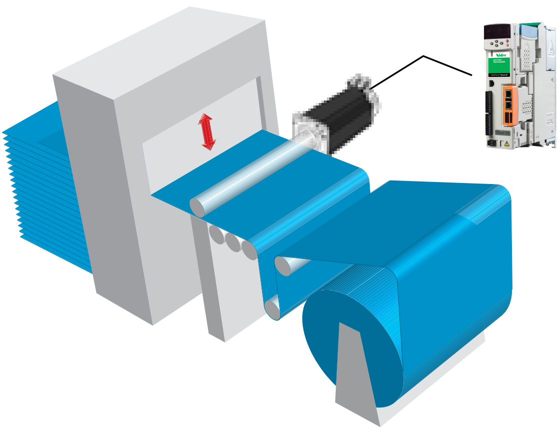
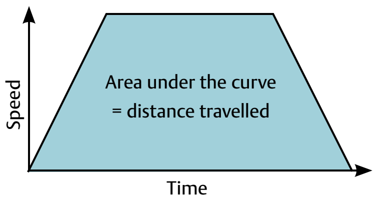


Figure 2-3 Sheet metal press feeder



For indexing applications, the key is to move the exact specified distance in a given time. Motion can be made smoother by following an acceleration and deceleration profile. The motion must be controlled to ensure that the correct distance is travelled for each index. This is usually achieved with a trapezoidal motion profile, where the acceleration, deceleration and maximum speed are controlled by a motion profiler.

Figure 2-4 Single axis motion profile



This type of application is known as a single axis motion, or discrete motion with time.

2.2 Flying shear

A flying shear is an application for cutting a continuously moving product to a specific length. This type of application is known as a reference/follower system; in this case, the whole cutting mechanism (the follower) is accelerated to match the speed of the product (the reference). Once the cut is executed, the shear is then decelerated and reverses to the start position ready to begin again.

The speed and position of the product being cut is measured by a reference encoder, this tells the motion controller and drive the speed of the reference so that the follower can synchronise to it. If the product feed changes speed, the cutting mechanism must also change. This application is commonly used for cutting materials such as steel sections and wood.

Figure 2-5 Flying shear application

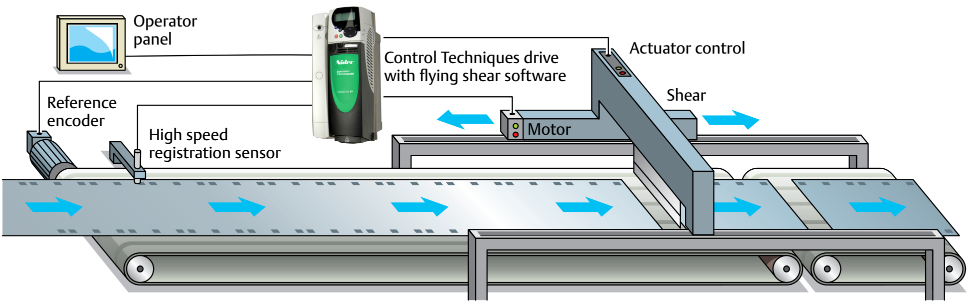
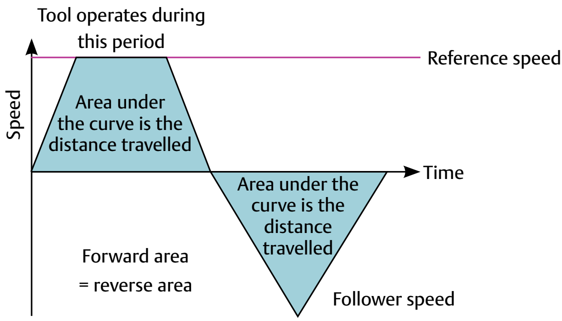
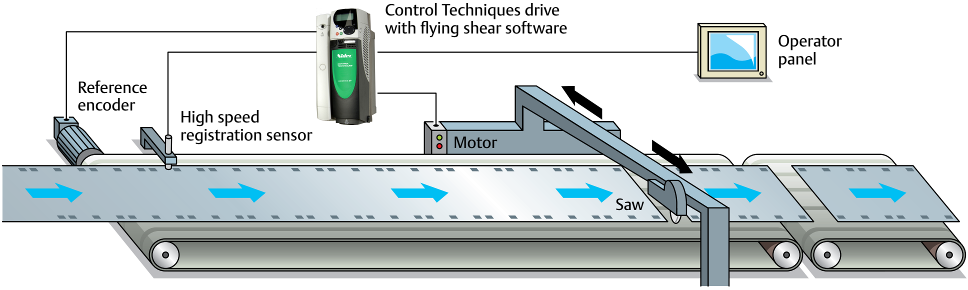


Figure 2-6 Flying shear profile



An alternative method of cutting the material uses a tool such as a circular saw that cuts across the product at an angle. The angle allows the saw to be synchronised with the product to achieve a straight line cut over a long distance. The greater the angle the faster the saw will have to travel to maintain a straight line cut. This is commonly used for cutting materials such as glass, mineral wool and fibre board; where the material web is wide but slower moving.

Figure 2-7 Flying shear saw application

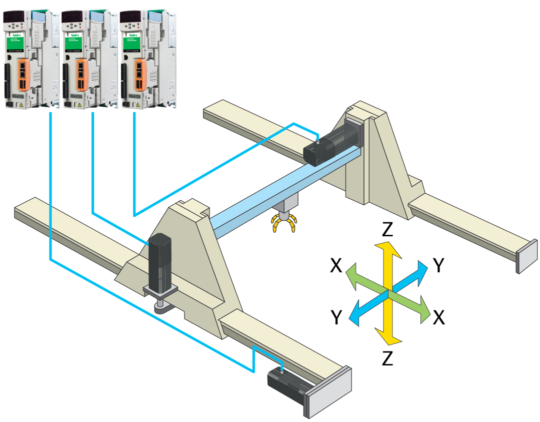


In these applications, the system needs to accurately synchronise with the rest of the production machine and so is connected to a reference encoder that provides process position. This is known as 1.5 axis interpolated motion, or synchronised discrete motion.

2.3 Pick and place

Pick and place machines are common for handling all types of processes where an item must be picked up and placed in a different location. These machines are often called X-Y tables because they are able to operate across two dimensions; the example below also has a Z axis for full three dimensional positioning.

Figure 2-8 Pick and place machines



Some pick and place applications require all of the axes are coordinated so that the interpolated motion of the manipulator (the device for picking up and releasing the item) travels along a defined path such as a straight line from one position to the next. This is desirable to increase the efficiency and performance of any multi-axis system and is essential in applications where a process such as cutting occurs as the system travels along the motion path.

In a distributed control system, this is achieved by common time synchronised clocks, such as what can be done with a virtual reference distributed to both drives. The movements of both the x and y axis are started at the exact same time and with a common time reference they can define a unique trajectory.

Figure 2-9 Fully interpolated motion pick and place application

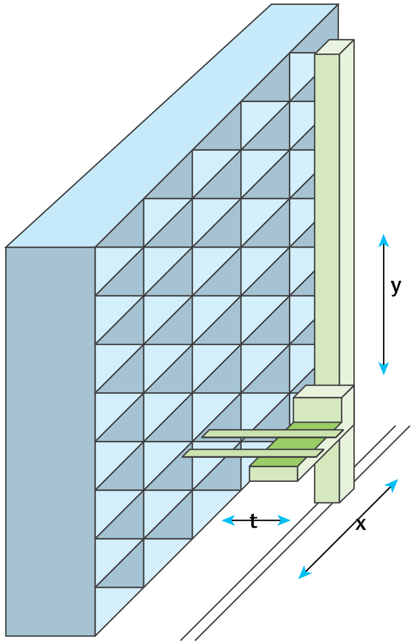
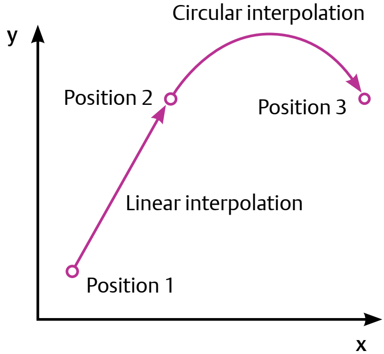


Figure 2-10 Linear and circular interpolation



This type of application shown in Figure 2-9 is known as fully interpolated motion, or multi-axis synchronised motion. Different types of interpolation can be used, Figure 2-10 shows linear and circular interpolation with 2 axes.

Note

Multi-axis synchronised motion should not be confused with a multi-axis system. Multi-axis system only refers to the fact that there are 2 or more motor axes, and does not define how those axes are controlled.

3 Motion system layout

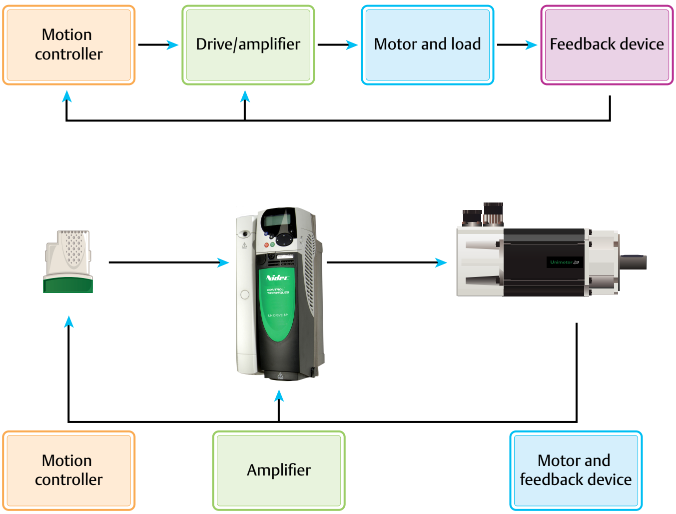
There are two typical motion system configurations

1) Multi-axis controllers where the motion control, trajectory and machine logic is defined by a single device
2) Decentralised where the motion controller resides within a single axis system

3.1 Single axis

A motion control system for a single motor axis generally consists of the following physical components:

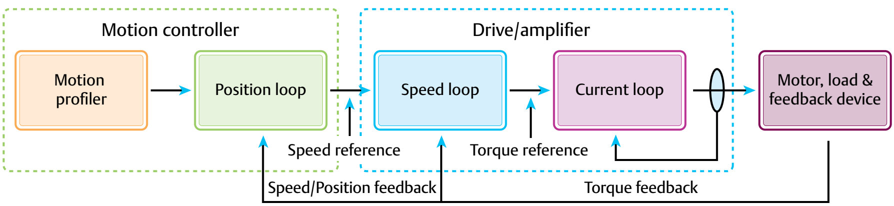
Figure 3-1 Typical motion control system



The system feedback encoder does not necessarily have to be on the back of the motor although in the majority of applications it is.

Breaking this down further, there are three closed loop control systems that are at work within the motion system, these are the position loop, speed loop and current loop.

Figure 3-2 Closed loop control systems



3.1.1 The motion profiler

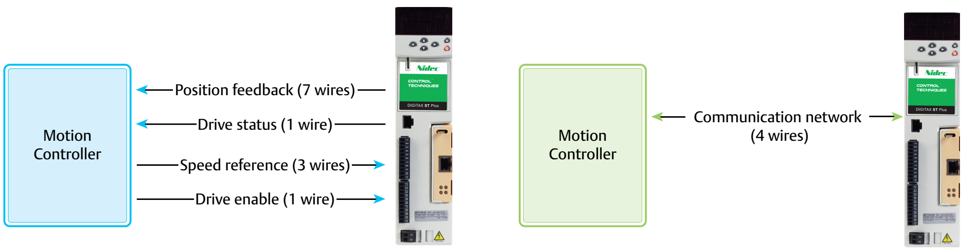
The method that is used to generate the motion profile is an important part of any system as it defines the motion requirements of the system. For this reason it is discussed in greater detail in Chapter 14 *Motion profiler features* and Chapter 15 *Motion references*. The control and generation of the position reference is usually handled within the motion controller.

3.1.2 Position loop

In a traditional motion control system, the position loop usually resides within the motion controller and compares position reference with the position feedback to determine the position error. This is used to generate a speed demand, also known as a speed reference, to regulate the position. Traditionally this was transferred from the motion controller to the amplifier via an analog signal, however this is prone to error and electrical noise. Today this is commonly achieved via a motion communication network such as SERCOS, EtherCAT, PROFINET IRT; or in the case of single axis with integrated motion control it is applied as a direct digital reference to the drive.

Typical update rates for position loop is 0.25ms to 2ms. 1ms update rates will give sufficient performance to control even the most dynamic servo applications, due to the long time constant of the mechanical systems being controlled. Faster update rates usually result in negligible performance improvements.

Figure 3-3 Traditional analog vs digital communications systems



3.1.3 Speed loop

The speed loop is usually integrated within the drive/amplifier. This compares the speed demand from the position loop to the speed feedback from the motor, and generates a torque reference. For the motion control system to function correctly, it is important that any ramps within the drive are fully disabled, otherwise the system will be unstable.

Typical update rate for the speed loop is usually between 250µs to 500µs.

3.1.4 Torque loop

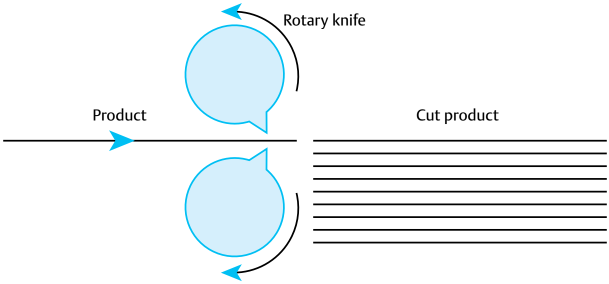
The current loop is integrated within the drive/amplifier, and is designed to regulate the torque producing output current to the motor. It takes the torque demand from the speed loop, and current feedback from current sensors within the amplifier to generate the control signals to produce a power output from the drive.

A typical update rate for a current loop is around 62.5µs.

3.2 1.5 axis interpolation

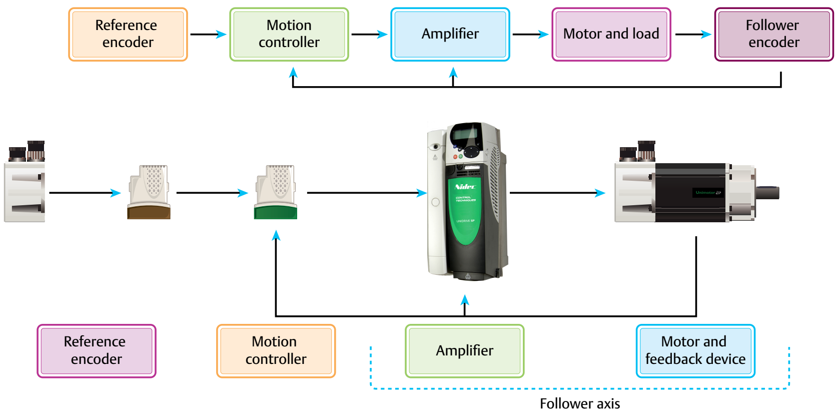
1.5 axis or reference/follower applications are where the motion of an axis is directly linked to the motion of another axis. Simple applications for this are pulse following more and electronic gearing. The reference encoder source can be another encoder port or a digital network reference. In a 1:1 gear ratio, as described in Figure 3-4, the follower motor will move one feedback pulse for every reference pulse received. Simply stated, the reference and follower motor will move the same distance and velocity at the same time.

Figure 3-4 An example of a reference/follower application is a rotary knife controller



A rotary knife is used to cut a continuous moving product to set lengths by synchronising the speed of the rotating knife with the speed of the production line. In this application, the position of the axis is a function of the position of the product. As the production line increases or decreases in speed the knife must change speed accordingly to maintain the same length between cuts. To achieve this there must be a reference encoder input to the motion controller to indicate the position of the product. In this system the reference encoder forms the reference axis as this will determine the rotary knife position. Therefore the rotary knife is the follower axis, and also requires a feedback or follower encoder.

Figure 3-5 Simple reference/follower configuration



Common references include:

➜ Digital lock or electronic gearbox (EGB) ➜ Cubic spline ➜ Cam profiling

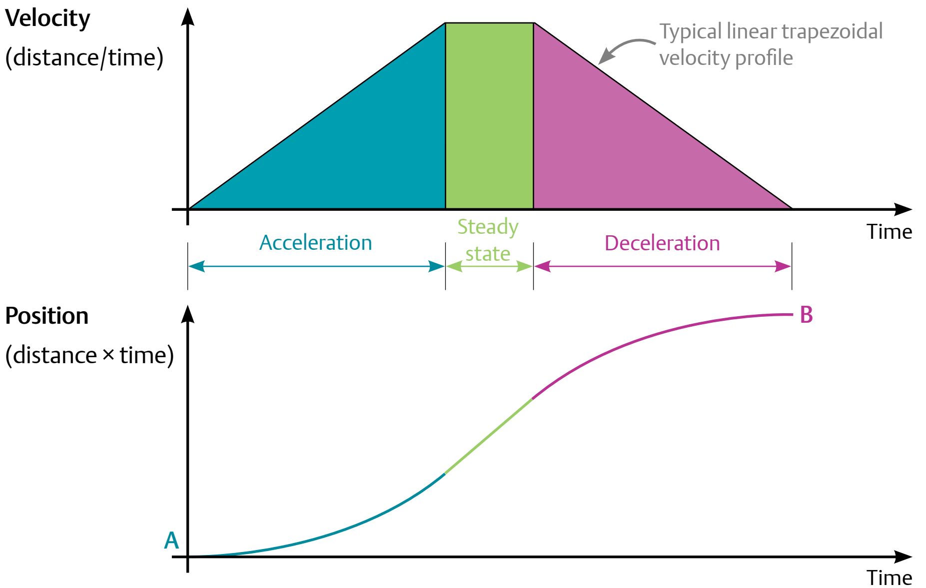
4 Relationships of motion

There are four basic types of motion parameters that are important when dealing with motion systems. These are distance, velocity, acceleration, and jerk. The relationships between these parameters are as follows:

Distance(θ)	**= velocity × time**
	= $\int v \cdot dt$ (integral of velocity × time)
Velocity(v)	**= distance / time**
	= $d\theta/dt$ (rate of change of distance)
	= $\int \alpha \cdot dt$ (integral of acceleration × time)
Acceleration(α)	**= velocity / time**
	= dv/dt (rate of change of velocity)
	= $\int \gamma \cdot dt$ (integral of jerk × time)
Jerk(γ)	**= acceleration / time**
	= $d\alpha/dt$ (rate of change of acceleration)

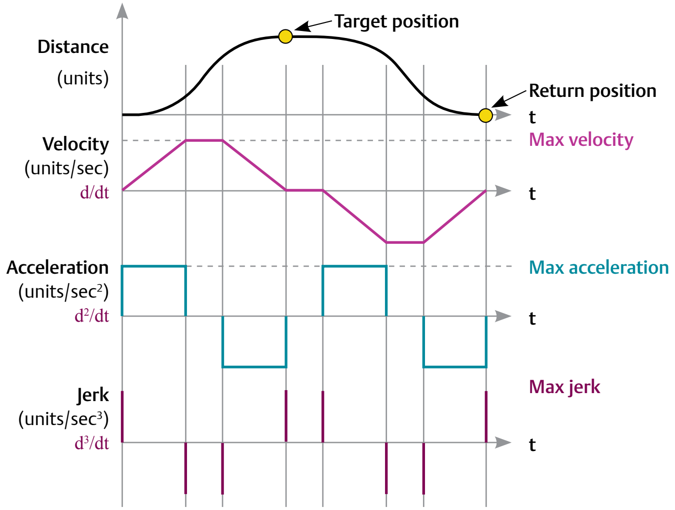
The following simple trapezoidal velocity profile has a constant rate of acceleration and deceleration; the equivalent position profile is also shown.

Figure 4-1 Trapezoidal position/velocity profiles



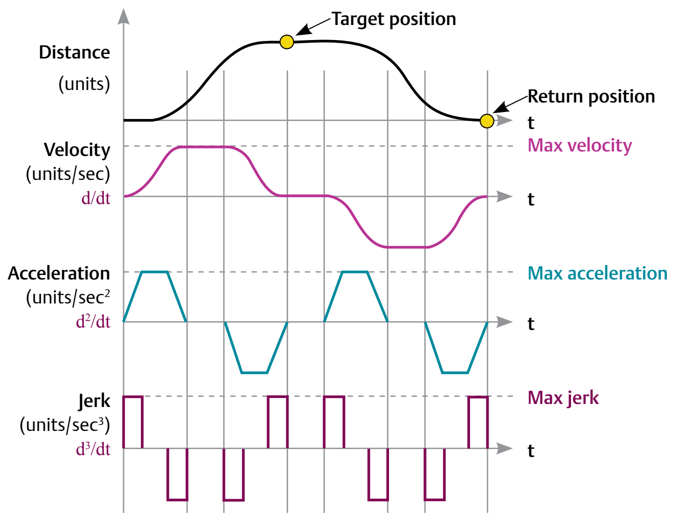
If we analyse the motion profile with respect to acceleration and jerk (jerk being very important for a smooth transition from one velocity to another; critical in applications like elevators), the results are seen in Figure 4-2.

Figure 4-2 Distance, velocity, acceleration and jerk for a trapezoidal profile



From this it can be seen that with a constant rate of acceleration and deceleration, the level of jerk is high when starting to accelerate and decelerate, and stopping acceleration and deceleration. The levels of jerk can be reduced by changing the acceleration rate. The introduction of a velocity S-ramp can alter the levels of jerk as shown in Figure 4-3.

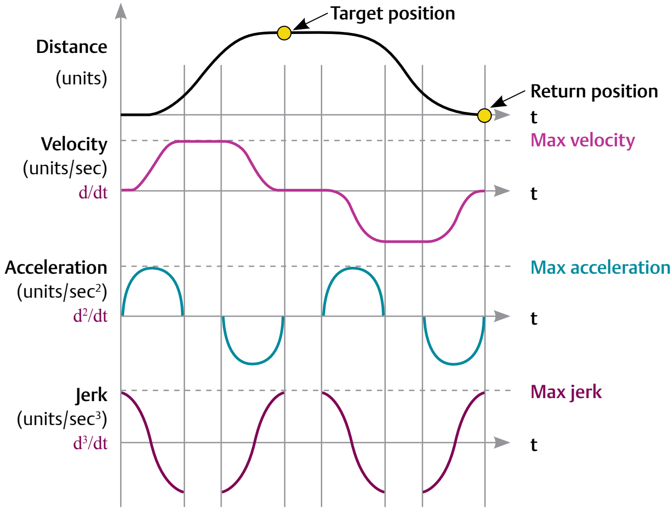
Figure 4-3 Distance, velocity, acceleration and jerk for an S-ramp profile



From this we can see that by applying S-ramp velocity, acceleration and deceleration are trapezoidal profiles. This in turn applies a square law profile to the velocity when starting to accelerate or decelerate, and finishing accelerating or decelerating, giving a smoother transition from one velocity to another.

Further improvements can be made by introducing more complex speed ramps. Figure 4-4 shows the effect of implementing a 'sine' ramp.

Figure 4-4 Distance, velocity, acceleration and jerk for a sine ramp profile



From this it can be seen that by applying the sine ramp, acceleration and deceleration is given a sinusoidal profile. This in turn applies a sinusoidal profile to the velocity when accelerating or decelerating, giving a very smooth transition from one velocity to another, which is ideal for high precision servo mechanics.

The downside is that the motor current demand is proportional to the change in acceleration; so the steeper the acceleration rate, the more current is required from the drive/motor system. Hence a bigger system is required.

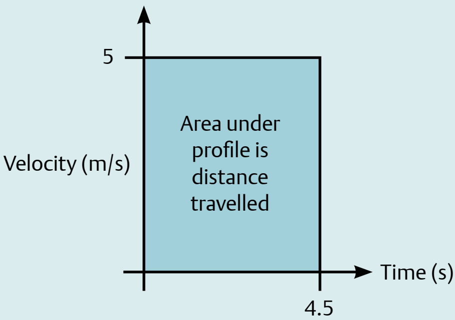
The sine ramp peak torque will increase by T½ compared to the equivalent linear ramp. The RMS torque for the same profile will also increase.

5 Worked examples using motion equations

The aim of this section is to show some worked motion problems in standard industrial units, to demonstrate how easily standard trapezoidal motion profiles can be solved.

5.1 Constant velocity

If a conveyor moves at a constant speed of 5m/s for 4.5s, what distance will be travelled?



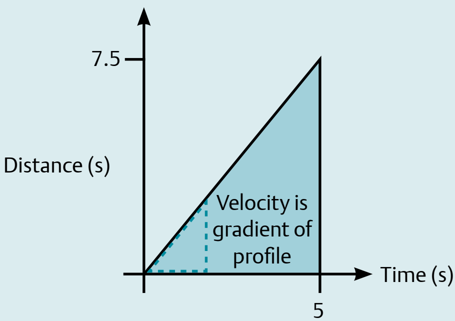
Distance = Velocity × Time

(or area of a rectangle base x height)

= 5m/s × 4.5s

= 22.5m

If a conveyor covers 7.5m in 5 seconds at a constant speed, what speed was used?



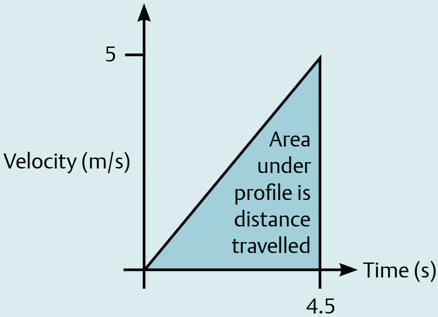
Velocity = Distance/Time

= 7.5m/5s

= 1.5m/s

5.2 Constant acceleration

If a crane picks up a load from the ground, and accelerates it up to 5m/s in 4.5 seconds, what distance will be covered?



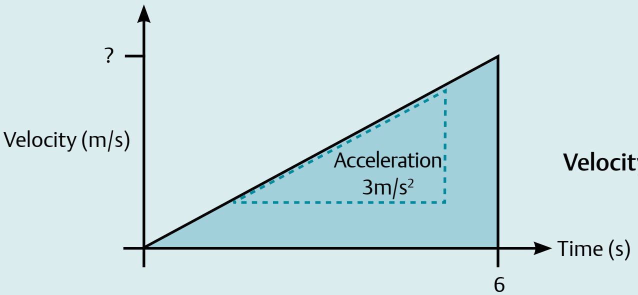
Under linear acceleration the area under the profile is halved compared to a profile at a constant speed for the same amount of time. Therefore distance used when accelerating = (velocity × time) / 2 or the area of a triangle.

Distance = (Velocity × Time)/2

= (5ms × 4.5s)/2

= 11.25m

If a crane picks up a load from the ground, and accelerates it at 3m/s/s or $3m/s^2$ for 6s, what speed was reached?

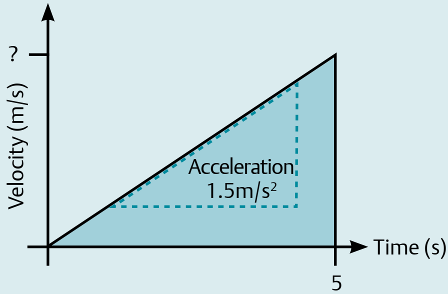


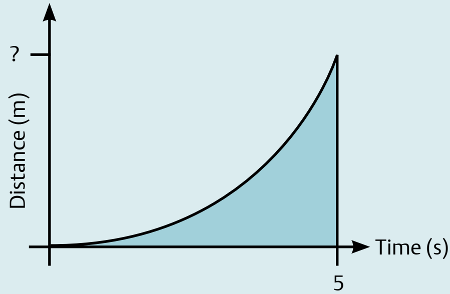
Velocity = Acceleration × Time

= $3m/s^2 \times 6s$

= 18m/s

If a crane picks up a load from the ground, and accelerates it at $1.5m/s^2$ for 5s, how far will the load have travelled?





Velocity = Acceleration × Time, however if we re-arrange the equation to make acceleration the subject of the equation we get:

Equation 1: Acceleration = Velocity/Time

Distance (Linear acceleration) = Velocity × Time/2; however if we re-arrange the equation to make velocity the subject of the equation we get:

Equation 2: Velocity = 2 × Distance/Time

If we substitute velocity in equation 1 for 2 × Distance/Time, we get:

Equation 3: Acceleration = 2 × Distance/$Time^2$

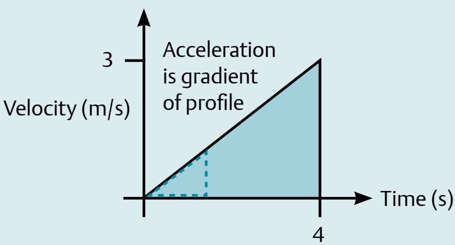
If we re-arrange this to make distance the subject of the equation we get:

Equation 4: Distance = Acceleration × $Time^2$/2

From this we can see that under linear acceleration, distance changes by a square function.

Distance = Acceleration × $Time^2$/2
= 1.5m/s × $5s^2$/2
= 18.75m

If a crane picks up a load from the ground, and takes 4s to accelerate up to 3m/s, what was the rate of acceleration?



Velocity = Acceleration × Time
Acceleration = Velocity/Time
= 3m/s/4s
= $0.75m/s^2$

5.3 Motion equation reference

5.3.1 Steady state

Basic equations

Velocity	= Distance / Time
Time	= Distance / Velocity
Distance	= Velocity × Time

Acceleration	= Velocity / Time
Time	= Velocity / Acceleration
Velocity	= Acceleration × Time

Where time is not known

Acceleration	= Velocity^2 / Distance
Distance	= Velocity^2 / Acceleration
Velocity	= √(Distance × Acceleration)

Where velocity is not known

Acceleration	= Distance / Time
Distance	= Time^2 / Acceleration
Time	= √(Distance / Acceleration)

5.3.2 Ramp

Basic equations

Velocity	= (2 × Distance) / Time
Time	= (2 × Distance) / Velocity
Distance	= (Velocity × Time) / 2

Where time is not known

Acceleration	= Velocity^2 / (2 × Distance)
Time	= Velocity^2 / (2 × Acceleration)
Velocity	= √(2 × Distance × Acceleration)

Where velocity is not known

Acceleration	= (2 × Distance) / Time^2
Distance	= (Acceleration × Time^2) / 2
Time	= √((2 × Distance) / Acceleration)

6. Specifying feedback devices for motors

One of the most important requirements for feedback devices used on motors is the quality of the device itself. As with any component in a control system it is always a trade-off between cost and quality, and feedback devices are no different.

6.1 Degrees, minutes and seconds

Before considering what factors are important for selecting a suitable feedback device, it is useful to understand and consider the realm of angular units. 360° is equivalent to one complete revolution. For smaller angular graduations, minutes or arc minutes are used, which is a subdivision of a degree and like time, there are 60 minutes to a degree.

The symbol used for arc minutes is ', so for 0.5° there are 30' arc minutes. This analogy continues, as a subdivision of an arc minute is an arc second. The symbol used for arc second is ".

Note – always useful to remember

- ➜ 1' arc minute = 0.0166667° (1°/60')
- ➜ 1" arc second = 0.000277778° (1°/(60' × 60"))

6.2 Characteristics of feedback devices

There are many features of feedback devices that effect the overall performance of a control system that contains drives and motors. It is easy to confuse position accuracy with position resolution and vice versa.

6.2.1 Accuracy

Accuracy is the most important factor for the quality of a feedback device in a motor drive system. The accuracy of a feedback device is governed by many factors, some being related to the design of the encoder and the manufacturing process, others being the mounting arrangement. This is usually quoted in terms of worst-case position error over one complete mechanical revolution and given in arc seconds; sometimes referred to as single cycle error.

When considering a servo motor that has a feedback device mounted, if the velocity of the feedback device was scoped, a trace similar to Figure 6-1 would be seen.

The top trace (pink) shows the marker pulse of the feedback device.

The trace below (green) is a capture of the incremental channels of the feedback device.

The bottom trace (yellow) is the actual feedback device single cycle error.

Close observation of the trace can see that there is a repeating pattern every marker pulse.

Figure 6-1 Accuracy of a feedback device over one revolution

In practice, the bottom trace (yellow) shows a summary of the cogging torque of the servo motor (seen as the small higher frequency disturbances on the trace). Also it highlights the mounting misalignment of the feedback device to the servo motor plus the actual feedback device single cycle error.

Generally, for any given feedback technology, the more accurate the feedback device is, higher the costs. Also, the accuracy of the feedback device should not be confused with the positional resolution of a motor-drive system.

6.2.2 Resolution

Resolution of a positional system that consists of a servo motor with a feedback device and drive can be difficult to define. The reason for this is due partly down to the type of feedback device selected for the application. This section considers three identical systems with different feedback devices fitted.

System 1 – Resolver

A resolver feedback device is electro mechanical similar to a transformer and generates an analog output. It does not offer any discrete positional information on its own and needs to be converted to a digital signal. The conversion is done onboard the drive and the resolution is determined by the number of bits in the analog to digital converter. 14 bits are typical for a resolver input, which would give a resolution of 2^{12} or 4096 PPR. The drive will actually use the edges of the pulses for positioning (PPR multiplied by 4), so for one mechanical revolution of the motor 16,384 CPR (2^{14}) discrete positions are available. Resolvers are frequently used because of their robust construction and tolerance of high temperatures, but have low positional accuracy compared to other devices.

System 2 – Quadrature incremental encoder

A quadrature incremental encoder is the most common digital feedback device. Most encoders use optics to shine a light through a slot to indicate positional movement. Resolution of these devices is dependent on the number of slots that can be put on an encoder disk. 10,000 lines per revolution is the upper limit. On an encoder with 4096 PPR, then as with the resolver, the drive will actually use the edges of the pulses for positioning (PPR multiplied by 4). So for one mechanical revolution of the motor 16,384 CPR (2^{14}) discrete positions are available. Quadrature encoders are used due to their combination of low cost and reasonable performance, and linear positioned accuracy.

System 3 – SinCos incremental encoder

A SinCos incremental encoder is an analog feedback device, so does not offer any discrete positional information on its own and needs to be converted to a digital signal. This conversion is handled onboard the drive and is split into two items, one being the number of LPRs and the other being the number of bits in the analog to digital converter (number of interpolated bits). If the digital converter has 11 bits then for every SinCos cycle 2048 (2^{11}) discrete positions are available. This determines the positional resolution of the system. If we have an encoder with 4096 LPR (2^{12}) and assume the drive has 11 bits of interpolation available, then for one mechanical revolution of the motor 4096×2048 or $(2^{12})\times(2^{11})$ = 8,388,608 CPR (2^{23}) discrete positions are available.

Further information on PPR, LPR and CPR is given at the end of this section.

Table 6-1 Typical performance of different feedback types

Feedback type	Encoder supply voltage	SinCos cycles or incremental pulses per revolution	Resolution available to position loop	Feedback accuracy
Resolver	6 V rms Excitation 6kHz	1	Medium 16384 (14 bit)	Low +/- 720"
Incremental encoder	5Vdc	4096	Medium 16384 (14 bit)	High +/- 60"
Inductive absolute encoder EnDat 2.1	7 - 10Vdc	32	Medium Absolute position 524288 (19 bits)	Medium +/- 280"
SinCos optical encoder Hiperface	7 - 12Vdc	1024	Very high 1.04x10^6 (20 bits)	High For SinCos integral non-linearity +/- 45" For SinCos differential non-linearity +/- 7" (Total accuracy+/-52")
Optical absolute encoder EnDat 2.2	3.6 - 14Vdc	2048	Very High 2.08x10^6 (21 bits)	Very high +/-20" (Differential non linearity +/- 1% signal period)

6.2.3 Mechanical misalignment

The mounting arrangement of feedback devices is critical to the long-term success of the motor-drive system. Mounting is needed for aligning the position of the encoder to the position of the motor magnet relative to the motor windings. This alignment is called the commutation angle. If the feedback device is not correctly inline with motor shaft, then heat can be generated in the encoder bearings leading to premature failure. In severe cases, the glass disk in the encoder can rub against the optical sensing head.

6.2.4 Environment

The environment that the feedback device has to work in is critical to the long-term success of the system. There are two key factors that should be considered. The first factor is mechanical shock being transmitted from the load, through the motor shaft and into the feedback device. In severe cases the glass disk within the encoder device can shatter.

The second factor is ambient temperature. The feedback device is in a position where there is no airflow, so heat generated locally is not removed from the vicinity. In cases where the temperature of the feedback device is higher than specified (100°C), issues such as component failure or lost pulses could occur.

The solution to environmental problems is, where possible, to isolate the encoder from the problems. To achieve this consider where or how it is mounted. If this is not possible, fit more robust devices such as a resolver.

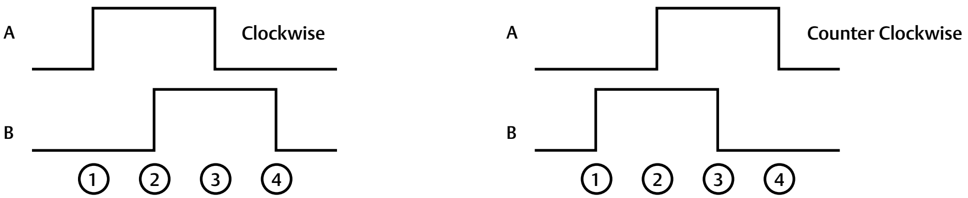
6.3 Output interface

This is the type of output circuit on the feedback device. EIA–422, EIA(RS)–485,TTL, HTL, open collector and 1V peak-to-peak are common. It is important to select a suitable output driver circuit that is compatible with drive.

6.4 Pulses per revolution (PPR)

Pulses per revolution relates to the number of cycles on the incremental channels of a digital encoder (Quadrature) when the feedback device is rotated 360°. It is common for feedback devices to have PPR to the power of 2, for example 512, 1024, 2048, 4096 etc.This allows the drive to process the information easier as it is a binary value. This is usually 4 times the number of lines per revolution.

Figure 6-2 Measurement of position using quadrature encoder signals



6.5 Encoder input specifications

The encoder input interface on motion controllers and drives usually specify a maximum encoder input frequency. It is important that your application does not exceed this value. To calculate the encoder frequency, the following formula is used:

Maximum frequency = (Maximum speed (rpm)/60) × Encoder PPR

6.6 Lines per revolution (LPR)

Lines per revolution relates to the number of Sine wave cycles on the incremental channels of an analog encoder (SinCos) when the feedback device is rotated 360°. It is common for feedback devices to have LPR to the power of 2, for example 512, 1024, 2048, 4096, etc. This allows the drive to process the information easier as it is a binary value.

6.7 Counts per revolution (CPR)

On digital encoders (quadrature), counts per revolution is usually the pulses per revolution multiplied by 4. This is performed by the drive and made available to the position loop. This is also referred to as encoder edges. Consider the following:

A servo motor has an incremental quadrature encoder with 4096 PPR (2^{12}). The counts per revolution become 4096 × 4 = 16,384 CPR (or $2^{12} + 2^{2} = 2^{14}$)

On analog encoders (SinCos), counts per revolution are usually the lines per revolution multiplied by the drive's interpolation number of bits. Consider the following:

A servo motor has a SinCos encoder with 4096 LPR (2^{12}) and a drive with 11 bits of interpolation. The counts per revolution become 4096 × 2048 = 8,388,608 CPR (or $2^{12} + 2^{11} = 2^{23}$).

7 Unit conversion from encoder counts

When controlling a motor in a motion system, at a low level the motion controller controls the position in feedback counts. Position maybe be controlled in encoder counts, speed in encoder counts per second (CPS) and acceleration in encoder counts per second^2 (CPS^2). However, in most systems the user interface uses more familiar rotational or linear units, e.g. revolutions per minute (rpm), or metres per minute (mpm).

7.1 Conversion to rotational units

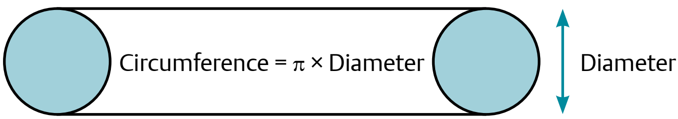
In order to convert CPS to some other rotational unit, the number of encoder CPR must be known for the system encoder. Once this is known, conversion is by a simple ratio:

$$\text{Units per second (UPS) out} = (\text{CPS in} / \text{CPR}) \times \text{Units per revolution (UPR)}$$
$$\text{CPS out} = (\text{UPS in}/\text{UPR}) \times \text{CPR}$$

7.2 Conversion to linear units

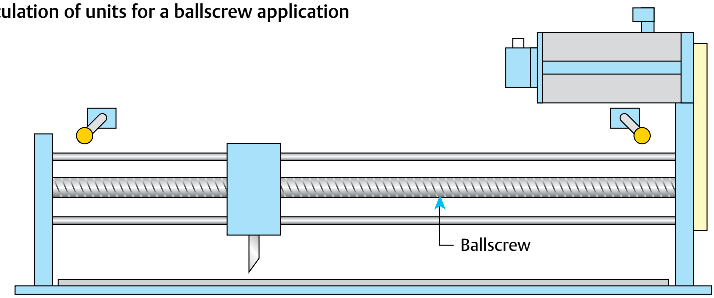
In order to convert CPS to a linear measurement of speed, e.g. CPS to m/min, the number of linear units per revolution must be known for the given system. For a simple conveyor this would be the circumference of the conveyor belt roller, e.g. 100mm diameter roller gives 314mm/rev.

Figure 7-1 Calculation of units for a conveyor application



For a ballscrew or linear slide this is the turn pitch e.g. 10mm/rev.

Figure 7-2 Calculation of units for a ballscrew application



Conversion from CPS to a linear unit measurement is by the same simple ratio as 2.1:

UPS out = (CPS in ⁄ CPR) × UPR

CPS out = (units/s in ⁄ UPR) × CPR

Example 1: A conveyor has a 65536 CPR encoder fitted, running at a speed of 30000 CPS, with a roller circumference of 200mm, what is the equivalent line speed in mm/s?

mm/s = (CPS in ⁄ CPR) × mm/rev
= (30000 ⁄ 65536) × 200
= 91.55mm/s

Example 2: A conveyor has a 65536 CPR encoder fitted, running at a speed of 250mm/s, with a roller circumference of 200mm, what is the equivalent speed in CPS?

CPS out = (mm/s in ⁄ mm/rev) × CPR
= (250 ⁄ 200) × 65536
= 81920CPS

Example 3: A conveyor has a 65536 CPR encoder fitted, running at a speed of 40000 CPS, with a roller circumference of 200mm, what is the equivalent line speed in mm/min?

mm/min = (CPS in ⁄ CPR) × mm/rev × 60
= (40000 ⁄ 65536) × 200 × 60
= 7324mm/min or 7.324m/min

Example 4: A conveyor has a 65536 CPR encoder fitted, running at a speed of 100000 CPS, with a roller circumference of 200mm, and a 3:1 output gear box, what is the equivalent line speed in mm/min?

mm/min = ((CPS in ⁄ CPR) × mm/rev × 60) × (Turns Out ⁄ Turns In)
= ((100000 ⁄ 65536) × 200 × 60) × (1 ⁄ 3)
= 6104mm/min or 6.104m/min

7.3 Data conversion accuracy

When converting from one data type to another, e.g. mm to encoder counts, it is important to consider that if the values in the calculations are not accurate, over time precision will be lost. e.g. a line encoder wheel has a 172.472mm circumference, and the attached encoder has 65536 CPR. If the circumference is rounded down to 172, the counts per unit conversion numerator and denominator would be 65536/172. If 10000mm is converted the resultant number of encoder counts is:

Encoder counts = 10000mm x 65536 / 172
= **3810232**

However, if we include the decimal information in to the conversion ratio we get:

Encoder counts = 10000mm 65536000 / 172472
= **3799805**

This is a total difference or error of 10427 counts or 27.44mm in only 10m, if a low accuracy conversion ratio is used.

7.4 Summary

Calculations in this section are valid when using motion systems such as RTL function blocks and APC in standard mode. However, if using PLCopen, SM-EZMotion/PowerTools Pro or Digitax ST Indexer, the conversion between user units and internal encoder count units is made automatically. See Chapter 9 *Control Techniques motion control solutions* for further details.

8 Tuning the speed loop for motion control

8.1 Speed loop gains

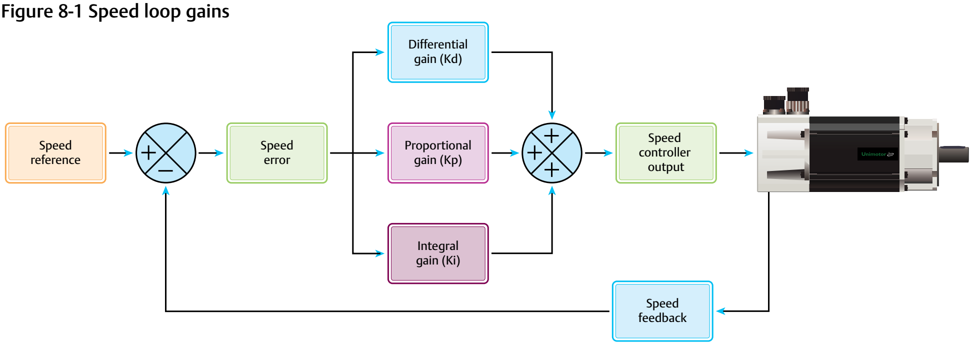
The speed loop maintains the speed of the application motor by varying the torque demand to the current loop, depending on the speed following error.

Note

The speed loop must be tuned before attempting to control position, otherwise the position loop may not be able to hold position correctly, or may even become unstable.

To help keep the speed error to a minimum, i.e. follow the reference as closely as possible, three speed loop gains are provided; proportional, integral and differential (PID) as shown in Figure 8-1.

Figure 8-1 Speed loop gains



8.1.1 Speed following error

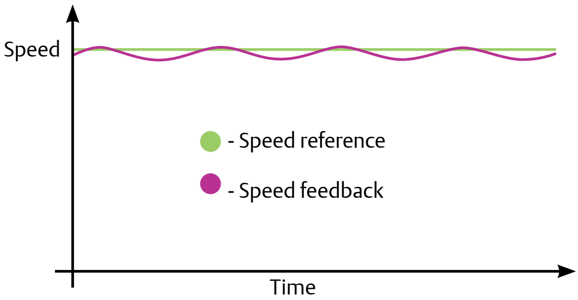
The speed error is the difference between the speed reference and the speed feedback i.e:

- ➜ Speed error = Speed reference - Speed feedback

8.1.2 Proportional gain

The proportional gain responds proportionally to the error, i.e. the proportional gain is a direct multiple of the speed following error and increases system response. This means that for the proportional term to produce an output there must be an error, and if proportional gain is used alone, the final speed demand will oscillate about the speed reference. This process is called regulation. Figure 8-2 shows the regulating effect of the proportional term on the speed feedback.

Figure 8-2 Regulating effect of the proportional term on the speed feedback



If the proportional gain is raised for a given load, the speed error will decrease. If the proportional gain is raised too high, either the acoustic noise generated by the motor from the proportional gain amplifying digital speed feedback quantisation becomes unacceptably high, or the closed loop stability limit is reached and the output will oscillate. A higher feedback resolution usually allows higher gains to be used.

8.1.3 Integral gain

The integral gain responds proportionally to the accumulation of speed error over a time period. It is provided to prevent speed regulation as described above. Raising this gain will decrease the amount of time taken to reach the speed set point, thereby increasing the system stiffness. Integral gain also reduces system dampening, which leads to overshoot after a transient. For a given integral gain the dampening can be improved by increasing proportional gain. In general a compromise between system response, stiffness and dampening must be reached for a given application.

8.1.4 Differential gain

The differential gain responds proportionally to the rate of change of error. It is provided to add system dampening under transient conditions. For 99% of applications differential gain is not required, as correct tuning of the proportional and integral gains will provide sufficient performance.

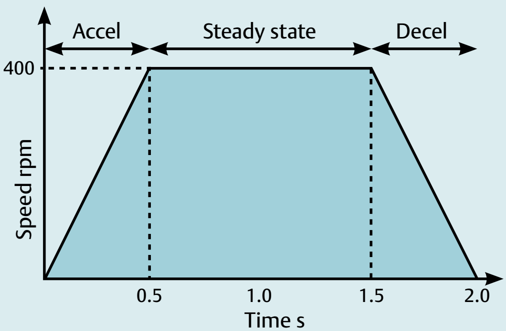
8.2 Practical speed loop tuning

In order to tune the speed loop for high performance dynamic position systems, the most demanding speed profile must be known, i.e. the maximum speed with the highest acceleration and deceleration rate. Once this is known, the drive can be set up to follow the profile using the drive's preset speeds and acceleration/deceleration rate as follows:

1. Set the drive's acceleration and deceleration times to match the most demanding profile. The acceleration/deceleration rates are set in s/1000rpm, to calculate the actual acceleration time for any speed use the following formula:

   Accel time for profile = (Time to reach application max speed/Application max speed) × 1000

   Example 1: An application profile takes 0.5s to reach 400rpm, and 0.5s to stop what is the acceleration time for this profile?



Accel/Decel rate for profile

Accel = Velocity/Time

= 400/0.5

Accel = 800

T_{Drive} = 1000/800

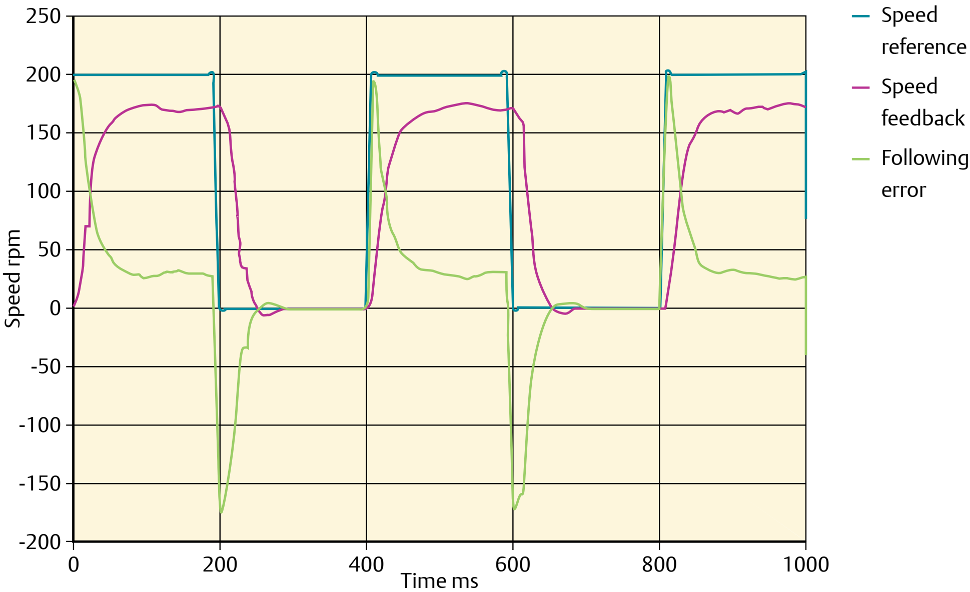
= **1.25s**

2. Program the drive so that it continually cycles through preset speeds, accelerating from zero to maximum speed, and then back to zero speed
3. While the profile is repeating, the speed loop gains can be altered to get the best performance. The gains should be raised in small steps till the optimum level is reached. To see the improvement in performance, monitor the following parameters using an oscilloscope or software:
   - ➜ The final speed reference
   - ➜ The speed feedback
   - ➜ The speed following error

It may also be beneficial to monitor the 'drive output at current limit' bit during this procedure, particularly with high inertia applications, to make sure that the drive's output current limit is not reached during acceleration or deceleration. The current limits should be set to match the motor or, if this is too high, the application requirements. Torque is proportional to the current, so if the current limits are reached, the torque produced will also be limited which will impact on system response and accuracy.

The plots in Figure 8-3 show the effect of the proportional and integral gains upon the speed reference, speed feedback and the following error with a step change in reference from 0 to 200rpm.

Figure 8-3 Insufficient proportional and integral gain



This plot shows the performance with low proportional and integral gain. Note that the speed feedback does not reach the peak speed of the final speed reference, and that the speed feedback profile is excessively rounded.

Figure 8-4 Sufficient proportional gain, insufficient integral gain

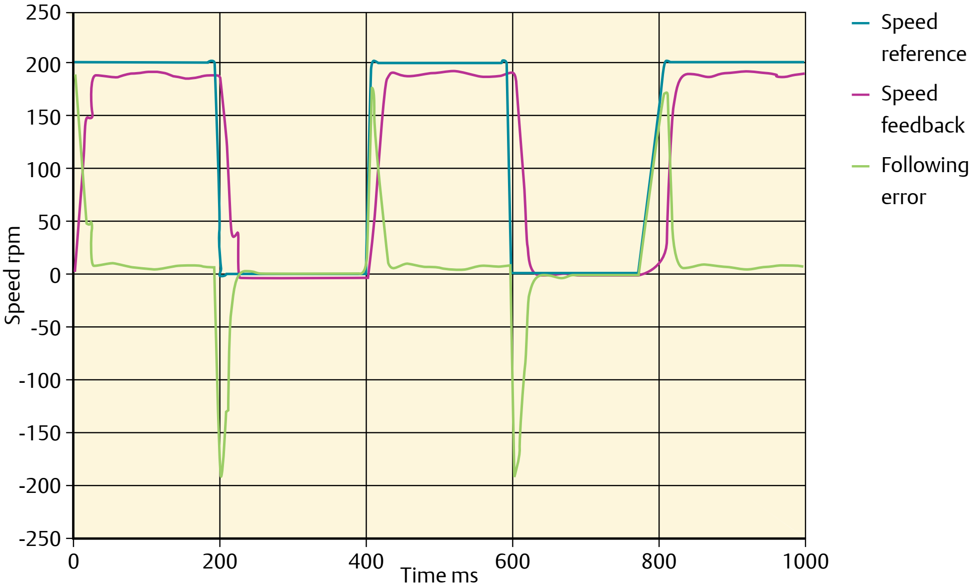


Figure 8-4 shows the performance with a low integral gain only. Note that while the speed feedback profile broadly matches the reference, the speed feedback still does not reach the peak speed of the final speed reference.

Figure 8-5 Sufficient Integral gain, insufficient proportional gain

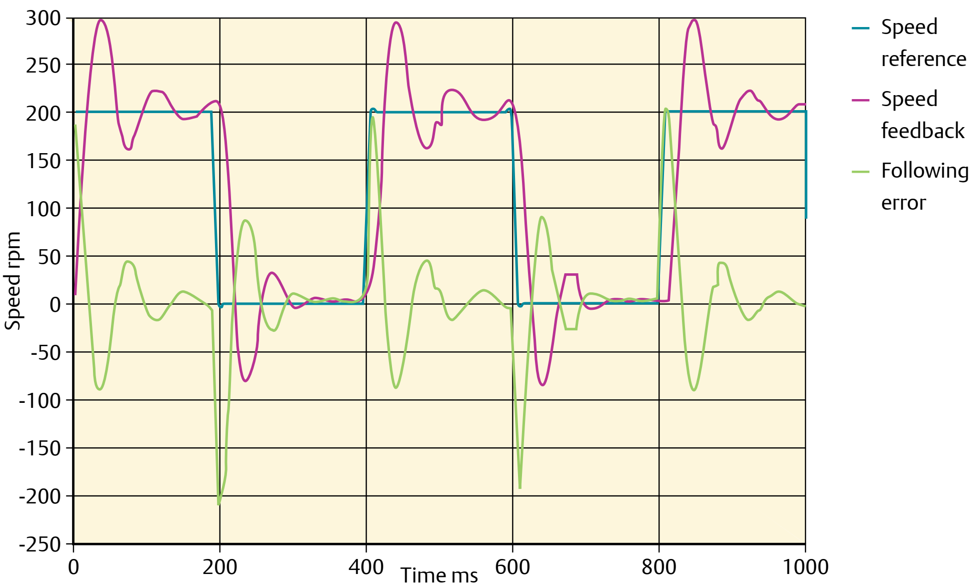


Figure 8-5 shows the performance with a low proportional gain only. Note that while the speed feedback profile reaches the peak speed of the final speed reference, the speed feedback overshoots badly due to the lack of proportional gain dampening. This type of wave form can also be caused by excessive proportional gain on a low inertia system, or insufficient proportional gain on a high inertia system.

Figure 8-6 Optimum proportional and integral gain

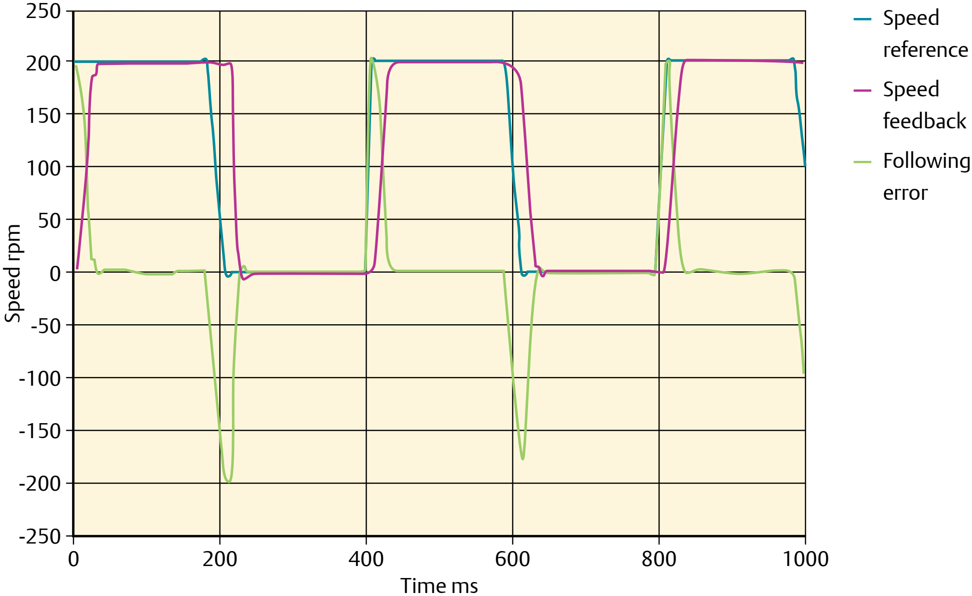


Figure 8-6 shows the performance with ideal gain settings for both the proportional and integral gains. Note that the feedback speed profile, closely matches the final speed reference with no overshoot, and that the final speed reference peak speed is reached.

8.2.1 Acoustic noise

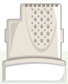
If acoustic noise becomes an issue when raising the proportional gain to get a good response, or a low resolution encoder is used, introducing a current loop filter of up to 1ms can significantly reduce noise level, and thereby allow higher gains to be set. Adding a filter inherently adds a small amount of system lag, but provided the filter applied is not excessive it can be very effective. An alternative to filtering would be to fit a better resolution encoder, since acoustic noise is often caused in this situation by relatively low resolution feedback devices.

9 Control Techniques motion control solutions

9.1 SM-Applications

1.5 axis programmable motion controller

The Unidrive SP has 3 different programmable SM-Applications modules, which are:



SM-Applications Lite V2

- High speed dedicated microprocessor
- SYPTLite 10kb flash memory
- SYPTPro 384kb flash memory, 80kb user program memory
- Dual-port RAM interface for communicating with the Unidrive SP and other option modules
- Task based programming system allowing for real-time control of drive and process

SM-Applications Plus

- High speed dedicated microprocessor up to 2x faster than SM-Applications and SM-Applications Lite
- 384kb flash memory for user program
- 80kb user program memory
- Dual-port RAM interface for communicating with the Unidrive SP and other option modules
- Task based programming system allowing for real-time control of drive and process
- EIA(RS)-485 port offering ANSI Modbus-RTU follower and reference, and Modbus-ASCII follower and reference protocols
- CTNet high speed network connection offering up to 5Mbit/s data rate
- Two 24V high speed digital inputs
- Two 24V high speed digital outputs

SM-Register

- High speed dedicated microprocessor
- 384kb Flash memory for user program
- 80kb user program memory
- Dual-port RAM interface for communicating with the Unidrive SP and other option modules
- Dedicated registration functions see Chapter 15 *Registration* for more information
- EIA(RS)-485 port offering ANSI Modbus-RTU follower and reference, and Modbus-ASCII follower and reference protocols
- CTNet high speed network connection offering up to 5Mbit/s data rate
- Two 24V high speed digital inputs
- Two 24V high speed digital outputs

A high speed digital input, on SM-Applications Plus and SM-Register can also be used to trigger a freeze position capture. When a 24V signal is received on terminal 1 of the SM-Applications Plus, the position of the feedback and/or the reference encoder is captured within 1μs, and is stored in hardware. This information can be used within a motion program for homing routines, registration etc. For more information regarding registration features refer to Chapter 15 *Registration*.

The EIA(RS)-485 port on the SM-Applications Plus has an additional feature called CTSync. CTSync performs 2 main functions:

- It synchronises the current loop, speed loop position tasks of all drives connected to the network (1 drive is CTSync producer, the rest are CTSync consumers)
- Provides 3 data channels which update synchronously with the drive speed and position loops, such that a position reference transmitted via CTSync will be responded to by all drives on the network at the same time

This is a useful feature for applications with >1 motion axis, e.g. printing applications.

9.2 SM-EZMotion

Simple to use distributed position control module

The module is equipped with four high speed digital inputs and two digital outputs for external control. SM-EZMotion is able to perform high-speed capture, queuing, profile summation, and program multi-tasking capabilities

Typical applications include:

- Indexing
- Random infeed control
- Rotary knife
- Phase synchronization
- Flying shear
- High speed labeling

PowerTools Pro programming software includes a simple yet powerful basic programming language and intuitive drag and drop interface.

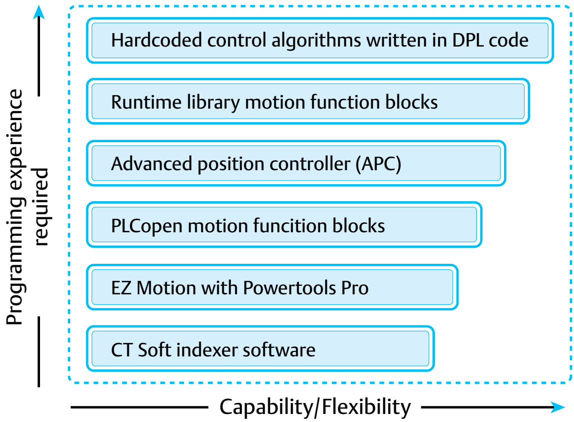
SM-EZMotion

- High speed dedicated microprocessor
- 8MB user programmable memory
- Dual port RAM interface for communicating with the Unidrive SP or Digitax ST and other option modules
- 4 digital inputs (All can be used for high speed capture)
- 2 digital outputs
- Programmed using PowerTools Pro
- 1.5 axis motion control

10 Levels of motion programming overview

There are 6 levels of motion currently available with Control Techniques drives:

Figure 10-1 Levels of motion available with Control Techniques drives



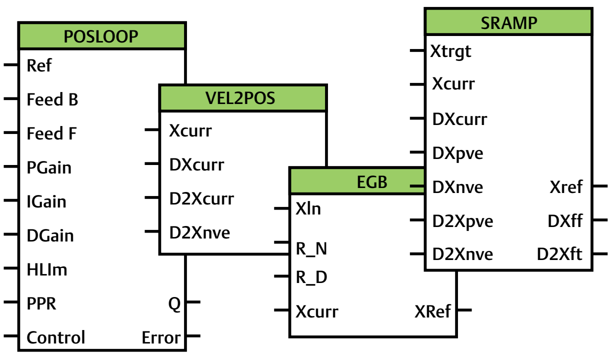
10.1 DPL code

The SyptPro drive programming language (DPL), is very similar to Basic, and supports standard programming structures, e.g. IF, THEN, ELSEIF, ENDIF, DO WHILE etc. From this coding level, any motion function/algorithm can be created giving maximum flexibility, however program development time is increased.

10.2 Run time library motion function blocks

The next stage in motion programming is to use the run time library (RTL) motion function blocks, which are a collection of pre-coded standard motion functions; e.g. position loop, electronic gearbox and camming. These can be combined with some interfacing DPL to create virtually any 1 or 1.5 axis system, and are platform independent; i.e. can be used on a wide range of drives, including Digitax ST, Mentor MP, Mentor II, Unidrive Classic and Unidrive SP.

Figure 10-2 Example motion function blocks



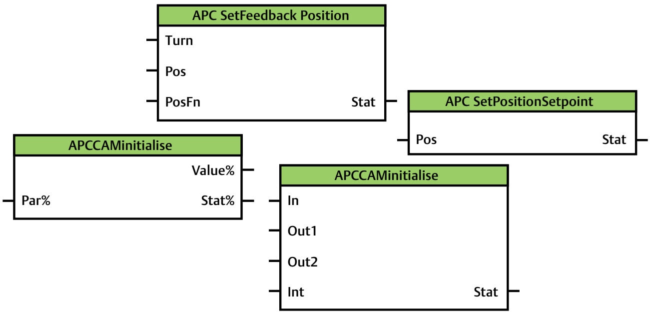
10.3 APC

The APC (Advanced Position Controller) is a comprehensive motion kernel comprising of:

- ➜ A feedback and reference encoder input counter, with selectable 32bit signed resolution
- ➜ Integrated position reset and offset functionality
- ➜ Stop, position, speed, CAM and digital lock motion references
- ➜ Bumpless transition between motion references (except in to CAM reference)
- ➜ Main and offset motion profiler
- ➜ Position loop
- ➜ Output channel synchronised to drive speed loops
- ➜ System can be run in encoder counts or user defined position units directly

The APC can be considered as a collection of all of the main motion functions, with all of the function block inter-connections, reset functionality, encoder counters, inter-motion reference profiling and user interface pre-written. Most 1 or 1.5 axis systems can be created using the APC, however several APC function calls may be required to perform a system function like jogging or homing etc.

Figure 10-3 Example APC function blocks



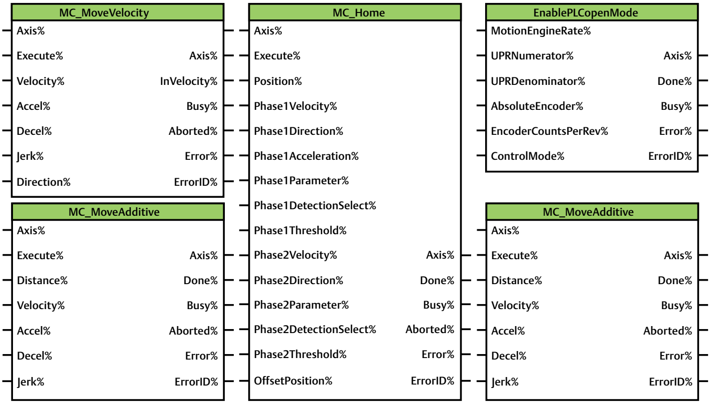
10.4 PLCopen

PLCopen is an international organisation which has published standards for a set of motion control function blocks for use in PLC programs. The standards define the function block user interface, the motion that results from their use and their interaction with each other. The PLCopen motion function blocks have been specifically designed to make motion control systems easy to create.

With SM-Applications, the PLCopen function blocks are essentially high level user interface layer, which controls the APC, but makes it easier to use by grouping all of the attributes required for a type of move in to one function block. The following PLCopen function blocks have been implemented:

Parameter	Resolution
DisablePLCopenMode	Used to disable PLCopen mode and restore parameter values that were changed by EnablePLCopenMode
EnablePLCopenMode	Used to configure the user units, control loop rates, etc. It must be called before other function blocks can be used
MC_AbortMotion	Used to perform a controlled stop, normally in the event of an error condition
MC_Home	Used to place the system at pre-defined position before commencing normal operation
MC_InitialiseAxis	Used to set the system limits e.g. overall maximum speed
MC_MoveAbsolute	Used to control movements about a fixed point; e.g. move to the tool loading position, 100mm from the home position
MC_MoveAdditive	Advanced function block. Commonly used for adjusting moves that are currently in progress
MC_MoveRelative	Used for incremental moves; e.g. move the conveyor forward another 200mm
MC_MoveSuperImposed	Advanced function block. Commonly used for adjusting moves that are currently in progress; e.g. seamlessly skipping a fill cycle when a carton is missing
MC_MoveVelocity	Used for moving at a constant velocity
MC_ReadParameter	Reads one of the PLCopen parameters
MC_ReadStatus	Used to check the current operating status
MC_Reset	Used to clear error conditions
MC_SetPositionCT	Re-calibrates the current position
MC_WriteParameter	Writes values to the PLCopen parameters
MC_DigLockIn	Engages following to reference encoder object (SM-Applications Plus only)
MC_DigLockOut	Disengages following of reference encoder object, and leaves the axis running at the present velocity (SM-Applications Plus only)

Figure 10-4 Example PLCopen function blocks



10.5 CTSoft indexer tool

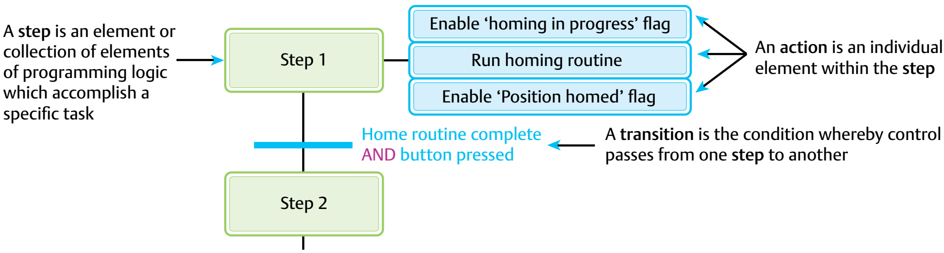
The CTSoft indexer tool is designed for simple stand-alone applications, using the onboard motion controller functionality within the Digitax ST Indexer and Digitax ST Plus servo drive variants.

The Indexer software allows the user to quickly and simply generate high speed point-to-point positioning applications using standard motion functions, such as homing and various types of motion index.

The indexer is programmed using Sequential Function Chart (SFC), a graphical programming language defined within the IEC 61131-3 global standard for industrial control programming.

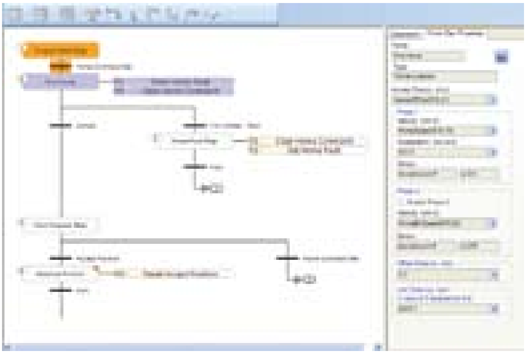
The 3 main elements of an SFC program are steps, actions and transitions.

Figure 10-5 SFC program elements



The SFC programming environment is part of the free drive commissioning software CTSoft.

Figure 10-6 Example indexer program screen



The indexer program is compiled and downloaded into the Digitax ST motion processor.

The indexing system provided by CTSoft and Digitax ST consists of the following:

- A graphical user interface for defining, controlling and monitoring an index system

  The graphical interface is designed to make development of a sequence of moves (homes or indexes) possible using a few button clicks

- A target runtime for executing the indexing system

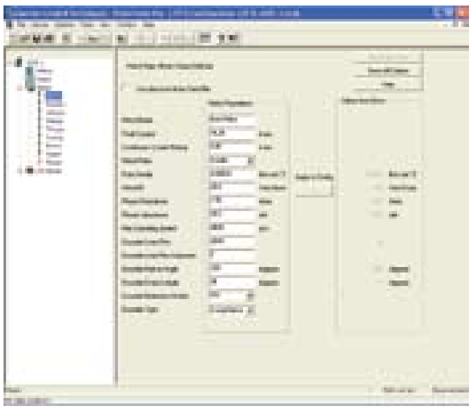
  The target runtime executes the indexer and provides access to drive and option module inputs and outputs

10.6 PowerTools Pro

PowerTools Pro is an advanced motion control programming software for Control Techniques servo drives; specifically Digitax ST-Z, Unidrive SP (with SM-EZMotion module), Epsilon EP, FM-3E and FM-4E modules.

Control Techniques' free PowerTools Pro software is aimed at users who want to create precise motion sequencing without the complexity of open software development environments. PowerTools Pro uses a familiar Microsoft® Windows™ interface with extensive help and intuitive functions to provide operators and machine builders with the tools needed for complete servo control. These include jogs, indexes, camming, electronic gearing, user programs, PLS, queueing, analog-in, user variables, high-speed capture, built-in user units, motor auto-tuning, and easy network set-up.

Figure 10-7 Typical PowerTools Pro screen



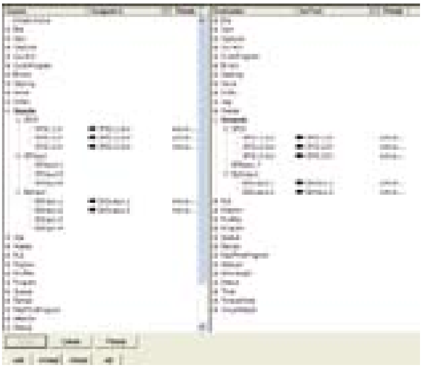
Developing motion applications with PowerTools Pro is a simple ‘five step, top-down process’. The five steps are displayed within an explorer bar that allows easy navigation. Each step is configured using simple check boxes, drop down selections and drag and drop functionality.

The five steps are:

- Hardware configuration
- Drive set-up
- I/O set-up
- Motion
- Programs

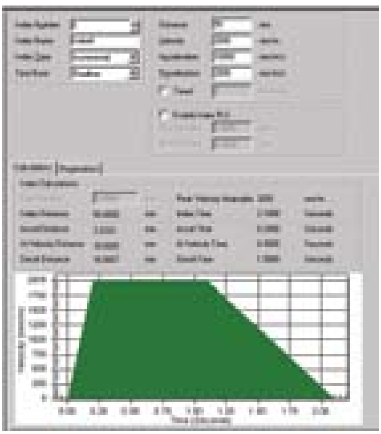
A ‘Basic-like’ programming language enables users to develop more complex applications and sequencing, with functions being selected by dragging and dropping onto the work area. Examples of the intuitive user interface are:

Figure 10-8 Virtual wiring



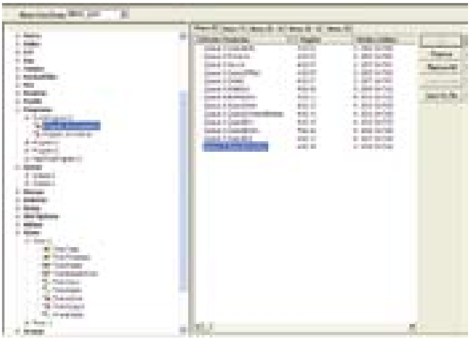
Assignments – ‘Virtual wiring’ can be used to create programs, without writing any code. For example, the assignment screen allows the user to drag-and-drop the desired machine function onto the digital inputs and outputs.

Figure 10-9 Index profile



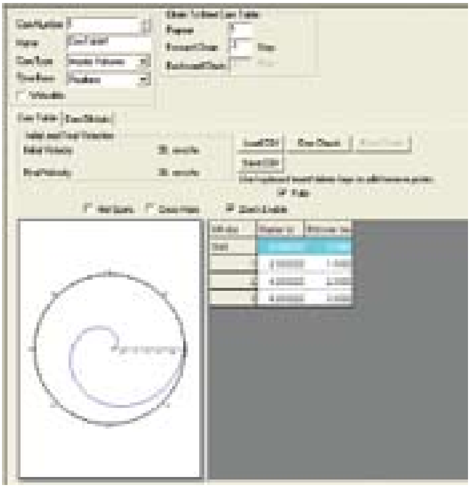
Indexes – Setting up index profiles is easily accomplished by filling in the blank field on-screen. Select from Incremental, Absolute, Registration, or Rotary Plus and Minus types. With position tracker, synchronization can be easily achieved by adjusting the time bar. Choose the time base of the index by selecting either real time or synchronized to a reference.

Figure 10-10 Setting up network communications



Network – Whatever fieldbus is used, setting up network communications is quick and easy. Fill-in-the-blank and drag-and-drop procedures are used to get drives communicating. PowerTools Pro diagnostics are intuitive, allowing real-time monitoring of the actual data being sent and received.

Figure 10-11 Cam graphing tool



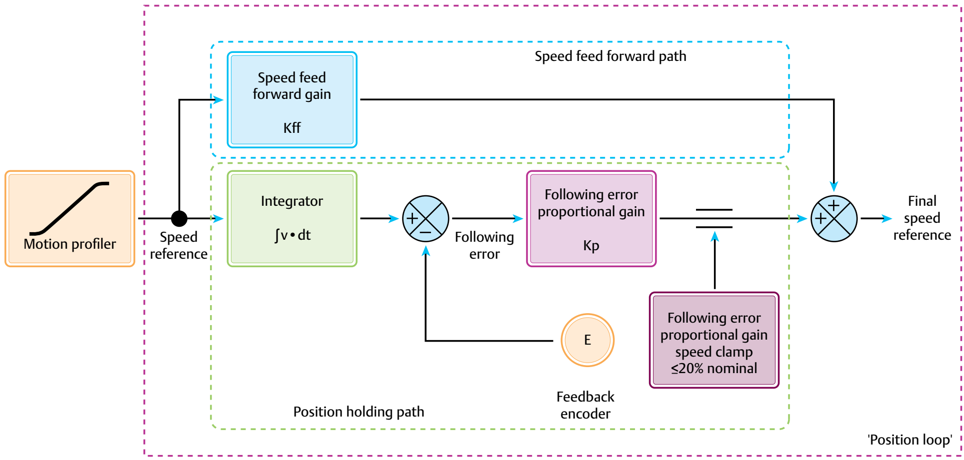
Camming – Cam data is easily entered within PowerTools Pro, incorporating an excellent Cam graphing tool; with reference/follower, reference/follower with interpolation, cubic spline, and jerk free types available.

PowerTools Pro is free of charge and can be downloaded from **www.controltechniques.com**

11 Position loop features

The position loop is a fundamental part in all motion control systems. Figure 11-1 illustrates a typical position loop.

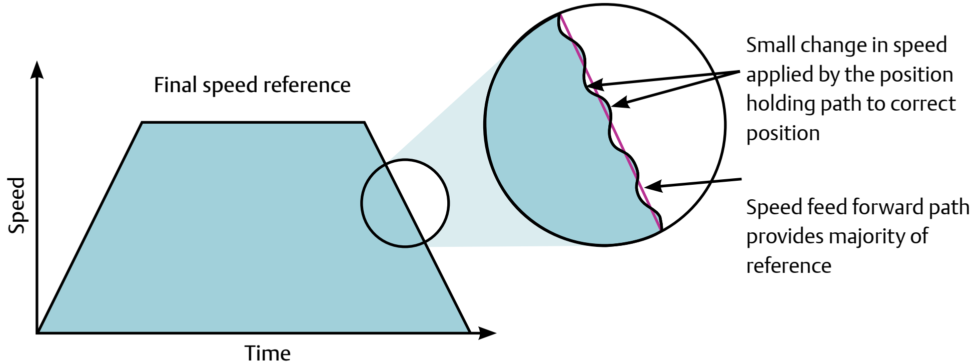
Figure 11-1 Typical position loop



In Figure 11-1 a speed reference is generated by a speed motion profiler, after which the reference splits two ways and is used by speed feed forward path and the position holding path.

The speed feed forward path provides the majority of the final speed reference sent to the drive, with the position holding path providing a trim reference signal. See Figure 11-2.

Figure 11-2 Speed reference generated from speed feed forward of proportional control functions



The speed feed forward gain (Kff) allows a proportion of the motion profile reference to correct the output of the position loop, to define the final speed reference. Normally this gain is set at 1, so that the position holding path only has to correct for slight positional differences. The speed feed forward value after gain is applied, is converted to a drive speed reference in rpm x1000 (for the Unidrive SP speed shortcuts), and is then added to the position holding path speed output.

Note

Where motion profilers provide a position output only, Kff $\propto$ d/dt of position (speed)

The position holding path integrator accumulates the motion profiler speed every sample, to build up a speed profile position. The following table shows how the integrator might accumulate position over 5 samples:

Sample	Speed in counts per sample	Integrator position accumulation in counts
1	0	0
2	300	300
3	300	600
4	300	900
5	300	1200

The position holding path integrator value is usually linked to the feedback encoder counter relative reset feature; so that if the feedback is reset the integrator value is also reset to the same value. This is done so that an instantaneous following error will not be created when a reset is actioned, which could cause unintended motion.

The feedback encoder position is then subtracted from the integrator accumulated profile position to create a following error, such that:

- If the feedback encoder position is greater than the integrator accumulated position, then the following error will be negative; which generates a negative position holding path speed to slow the motor/feedback encoder down
- If the feedback encoder position is less than the integrator accumulated position, then the following error will be positive; which generates a positive position holding path speed, to increase the speed of the motor/feedback encoder

The following error proportional gain (Kp) is then applied to the following error. In most systems, gain is applied in units per second per unit e.g. counts per second per count. A gain of 25 would mean that for an error of 1 count, a speed of 25 counts per second would be applied to correct the position error, i.e. the higher the gain value the faster the position correction is, the lower the gain value the slower the correction is.

The position holding path correction speed is then symmetrically clamped to a speed limit, usually set to 10% of the full speed range of the application; e.g. 3000rpm application = 300rpm speed clamp. This is done to prevent a large positional following error from applying a huge speed reference to the drive, e.g. if the motor has become jammed, or if the feedback encoder fails etc. In this way only a 'safe' speed is applied by the position holding path. The output of the following error speed clamp is then summed with the final speed feed forward path output to create the final speed reference to the drive.

12 Tuning the speed loop with a motion controller

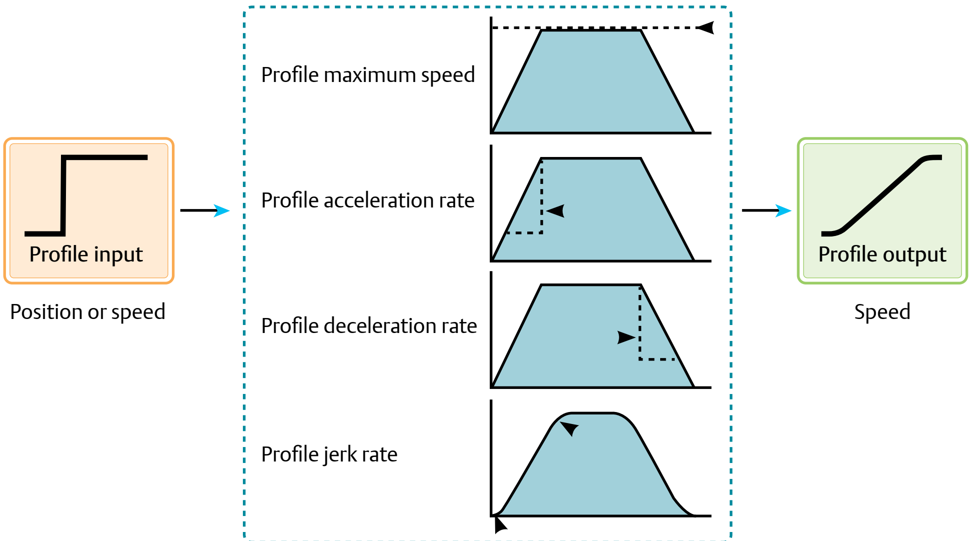
When tuning the drive speed loop on a complete application, rather than using the drive's preset speeds to create the speed profile to follow, if the position holding path proportional gain is set to 0, and the most demanding profile is activated in the motion controller, the speed loop can be tuned to the actual most demanding profile. Once the speed loop is tuned, the position loop proportional gain can be reinstated.

When tuning the speed following error in this way, the positional following error will build up since the position holding path is effectively removed. Consideration should be given as to whether or not the mechanical layout can cope with the positional following error building up, without damaging the application mechanics.

13 Motion profiler features

The motion profiler in a basic motion control system is used to apply speed, acceleration and deceleration rate limits (known as profile limits) to a speed or position setpoint, such that the output from the motion profiler will rise or fall to the new setpoint within the profile limits. Figure 13-1 illustrates a basic motion profiler configuration:

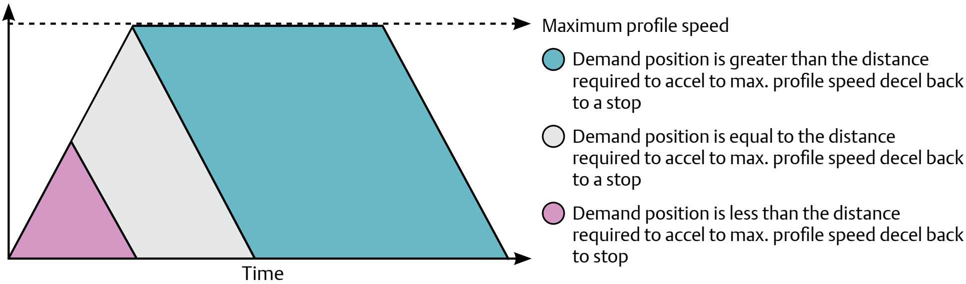
Figure 13-1 Basic motion profiler configuration



When operating in speed mode, the profile output speed will increment to the profile input speed or the profile maximum speed, whichever is the greater. The increments in speed used to ramp up to the reference speed are set by the acceleration or deceleration rate, depending on if a lower or higher speed is required. The profile output speed will never overshoot the profile input speed.

When operating in position mode, the speed output will ramp in increments defined by the acceleration or deceleration rate. If the difference between profile input position, and the profile output position is greater than the amount position required to accelerate up to the maximum speed and back to zero, the resultant speed profile will be trapezoidal; i.e. the profile output speed will ramp up to and stay at the profile maximum speed until the appropriate time to slow down and stop at the position set point. However, if the difference between profile input position and the profile output position is equal to or less than the amount of position required to accelerate up to the maximum speed and back to zero, the resultant speed profile will be triangular as shown in Figure 13-2.

Figure 13-2 Trapezoidal and triangular speed profiles



The internal algorithm of the motion controller will control the profile speed output so that when the profile finishes, demand position is always reached without overshooting in position or speed.

14 Motion references

It is common, in motion controllers available today, to have the following motion references:

- Position
- Speed
- Digital Lock (Synchronous to a reference encoder)
- CAM (Synchronous to a reference encoder)

The position and speed references are the basic motion profiler inputs as described in Chapter 13, however digital lock uses the motion profiler only some of the time, and CAM never uses it.

14.1 Position and speed functions

The position and speed reference may be used with some additional sequencing and logic to form motion functions; the more common motion functions are described in the following sub sections.

14.1.1 Jogging

Jogging produces rotation of the motor at controlled velocities in a positive or negative direction. When the jog input is turned on, jogging begins and continues jogging until the jog input is removed.

Jogs typically have their own acceleration and deceleration ramp, along with a specified velocity. Jogging has no distance parameter associated with it since underneath it is really a speed reference with some sequencing and logic to select direction.

14.1.2 Homing

The home routine is used in applications where the axis must be precisely aligned with some part of the machine. Homing to the motor's encoder marker will establish an accurate and repeatable home position, as will using a quality external sensor if homing to marker is not suitable; e.g. if there is a motor gear box and the output shaft of the gear box is to be aligned. Both methods will position the motor relative to the location of the rising edge of the marker pulse either edge of the homing sensor used; typically an input used for this purpose has selectable rising and falling edge triggering. Several parameters affect how the home function operates. Homing is typically the combination of both the speed and position references with some sequencing and logic.

Typical home sequence

1. Back off the sensor, if on the sensor. (This step is optional)
2. Move to the external home sensor to establish a home reference point
3. Next it will move to the Offset position
4. Then the command and feedback positions are set to the value entered into the end of home position.

Accuracy and repeatability

The amount of accuracy an application requires will determine the home reference option selected. Homing to an external sensor only will establish a repeatable home position based on the speed of the motor during homing, the triggering repeatability of the sensor, and the throughput delay of the sensor / input used to capture event. For example, a system with the motor traveling at 3000 rpm and having a 1.6ms sensor capture interval will result in a accuracy of 0.08 revolutions. The slower the motor moves, or the lower the sensor throughput delay and the faster I/O scan, the more accurate this will be. Where a high throughput delay sensor has been used, the repeatability and accuracy can be improved if the home routine can move to the sensor first and then to the motor encoder's marker position.

Home offset

The home offset is the distance from the reference position (indicated by the homing sensor) to the final stopping point at the end of the homing sequence.

14.1.3 Indexing

An index is a complete motion sequence that moves the motor a specific incremental distance. This motion sequence includes an acceleration ramp to a programmed velocity, a run at velocity, and a deceleration to a stop. Indexing is typically the combination of position reference with some sequencing and logic.

Indexes use acceleration and deceleration ramps that may or may not reach the specified velocity depending on the total distance and the ramp values. For example, a short move with long acceleration and deceleration ramps may not reach the target velocity entered.

Most devices support four types of index; absolute, incremental, registration and rotary which are described in the following subsections.

Absolute index

An absolute index will move the motor to a specified absolute position, the direction of the motor can be either positive or negative and will depending upon the starting position.

For example, if the motor starts at 1 revolution, and is required to go to an absolute position of 3 revolutions it will move in the positive direction 2 revolutions. If it is then asked to move to an absolute position of 1 revolution it will travel 2 revolutions in the negative direction.

Incremental index

An incremental index will move the motor a specified relative distance in the positive or negative direction regardless of the starting position. The direction of the incremental index motion may be determined by the sign of the index distance parameter value.

For example, the motor starts at 1 revolution, travels a distance of 2 revolutions and stops at 3 revolutions. If the same index is initiated a second time, the drive would move the motor another 2 revolutions to a position of 5 revolutions. If initiated a third time, the motor would travel another 2 revolutions to a final position of 7 revolutions.

Registration index

A registration index is used in applications where the motor must move until an object is detected and then move a specific distance from the point of detection, such as finding a registration mark and moving a distance beyond.

The registration index consists of two parts. The first part accelerates the motor to the target velocity and continues at this velocity until it receives a registration trigger (sensor or analog). Upon receipt of a registration trigger, the registration offset will be executed at the target velocity. The sensor limit distance hit source can be used to turn on an output, if a sensor input or analog limit is not received within the limit distance. A registration window can also be used to determine the validity of a registration trigger, where if a registration trigger is received outside of the registration window, it will be ignored.

Rotary index

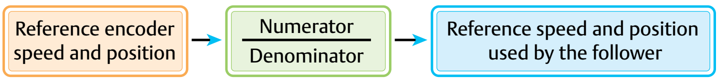
A Rotary index provides directional control of moves to absolute positions, typically within one revolution. Rotary indexes can be always forward direction (Rotary Plus), always reverse direction (Rotary Minus), or shortest route. The position entered must be within the rotary range, $0 \leq$ Distance < Rotary rollover point. If the roll over range is specified as 1 revolution you can specify a position within a single revolution to orientate to, however the total number of revolutions moved by is disregarded e.g. if the rotary index position is 100, and you are on turn 2 and you do a rotary plus index, you will move forwards to position 100, turn 3.

14.2 Digital lock

Digital lock, also known as 'electronic gearing' is a means of positionally linking the motion of follower axis via a ratio to the motion of a position reference. The leader position reference is typically in the form of a line encoder, or a virtual encoder commonly known as an 'electronic line shaft'. The user digital locking ratio is set up as a number of follower distance units over a number of leader distance units, with a typical resolution of 32bit. Digital lock references have following features:

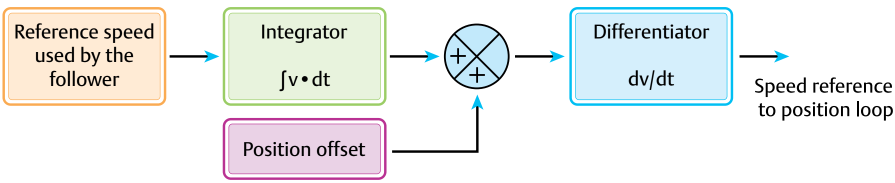
- **Electronic gearbox** – This allows the user to run the digitally locked 'follower' axis at higher or lower speed than the reference encoder while still remaining positionally locked. If integer maths is used in the ratio calculation, remainder must be accumulated during the division so that decimal information is not lost. This ratio can be used in real applications to apply draw control or tension control, or to compensate for motor gearbox ratios

Figure 14-1 Electronic gearbox ratio



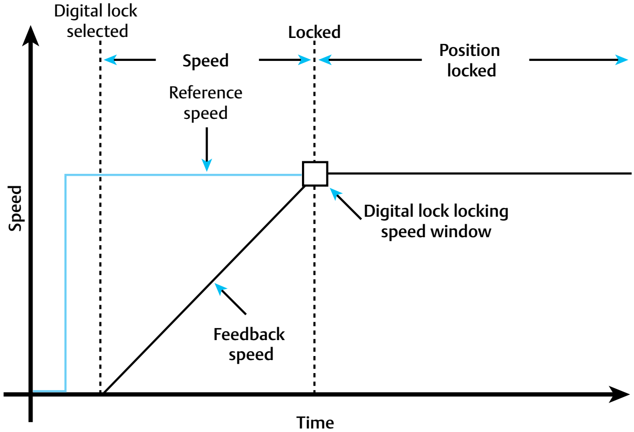
- **Position offset** – This allows the user to add or subtract position from the rotating digitally locked position, so that the phase of the reference and follower can be altered. In real applications this can be used to apply phase offsets, or registration correction. This is sometimes referred to as 'phase advance/retard'

Figure 14-2 Phase advance retard



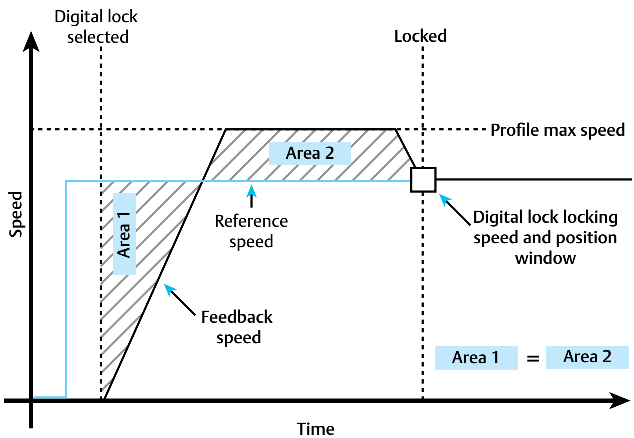
- **Rigid and non-rigid locking** – This feature uses the motion profiler to ramp up the follower axis speed to match the reference axis speed (non-rigid lock) or speed and relative position (rigid lock). A non-rigid lock application uses the following process:
  - The live reference axis may be running at some speed, and the follower may be stationary
  - Digital lock will be engaged on the follower, and its speed will ramp up towards the reference's speed using the motion profiler
  - When the reference speed is reached by the follower, i.e. the follower speed is within the speed locking window, the motion profiler will be bypassed so that the reference directly drives the follower, and the system will be considered locked

Figure 14-3 Non rigid digital locking



- A rigid lock application uses the following process:
  - The reference may be running at some speed, and the follower may be stationary
  - Digital lock is engaged on the follower, and its speed will ramp up towards the reference's speed using the motion profiler
  - During acceleration up to the reference's speed, position will be lost with respect to the reference, so the follower will accelerate to a faster speed than the reference to regain the lost position
  - When the reference speed and position is reached by the follower, i.e. the follower speed and position are within their locking windows, the motion profiler will be bypassed so that the reference directly drives the follower, and the system will be considered locked

Figure 14-4 Rigid digital locking



Digital lock is used in basic synchronisation applications like wire drawing, where the ratio between the reference and the follower speed is static.

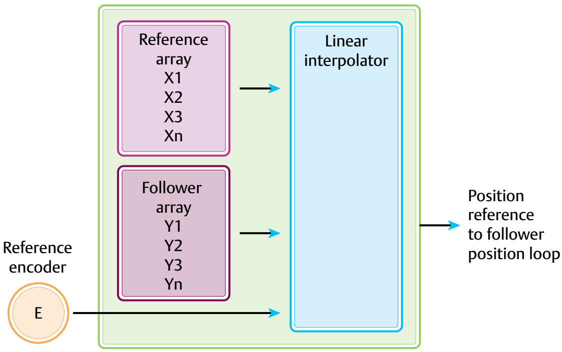
14.3 CAM

The CAM reference works in a similar way to a mechanical cam; where a mechanical cam creates linear or varied motion from rotational motion, an electronic CAM creates follower varied rotational motion from reference rotational motion.

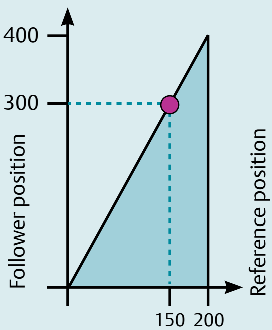
Basic electronic CAMs are defined by 2 data arrays, a reference position array, and a follower position array, where position is defined in encoder counts. Each array element is paired with the array element of the same number in the other array, e.g. reference array element 1 is paired with follower array element 1, these pairs of coordinates are known as CAM segments.

Motion occurs in the follower by turning the reference encoder, after which the position of the reference is fed into a linear interpolation calculation which calculates, for a given reference position, what the equivalent follower position will be, with respect to the CAM segment data.

Figure 14-5 Basic electronic CAM



If we consider a 1 segment CAM with reference segment value of 200 counts and a follower segment value of 400, and an input reference position of 150 counts, the equivalent follower position is derived from the following linear interpolation calculation:

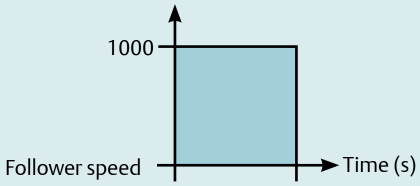


Follower position = Reference position counts × Follower segment / Reference segment
= 150 × 400 / 200
= **300 counts**

If this CAM is run cyclically, where when the position of the reference keep rising, and the CAM wraps over seamlessly, re-using the same reference and follower co ordinates over and over (cyclic mode), the behaviour would be similar to the digital lock EGB function, where the speed of the follower would be:

Follower speed = Reference speed × Follower segment / Reference segment

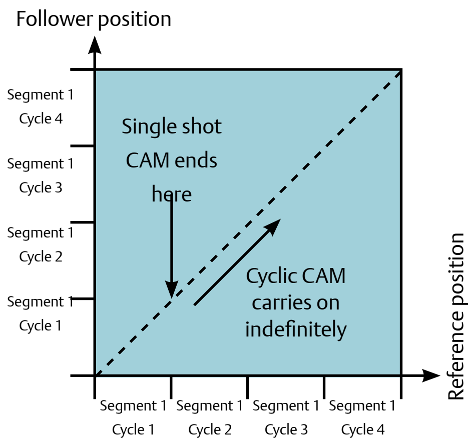
E.g. If in the previous example the speed of the reference was 500 rpm, the follower speed would be:



Follower speed = 500 × 400 / 200
= 1000 rpm

CAMs operate in either cyclic mode as described, or in single shot mode where once the reference position exceeds the final reference segment position, the follower will stop at its final segment position.

Figure 14-6 Single shot and cyclic CAMs



In CAMs where more than 1 segment is defined, the speed of the follower can be ramped up or down using a linear interpolation ramp. A linear interpolation ramp is a series of small jumps in speed to reach a final speed. The more steps there are in the ramp, the less granular the resulting follower speed response will be.

More advanced CAM interpolation methods are cubic spline and jerk free. Cubic spline uses the third order cubic polynomial splining to find smooth velocity and position curves from point-to-point. This table allows a different reference, follower, and interpolation for each point in the CAM. For a jerk free interpolation, the jerk starts and ends at zero. Jerk increases or decreases in smooth transition.

More advanced CAM function generators have the ability to link CAM tables together, to allow for dynamically changing one of the CAM table values or even the CAM table itself based on some external events such as IO values. Control Techniques supports several different types of CAMs such as absolute, relative, time based and cubic spline CAMs. Coordinate entry can be 'relative' or 'absolute'. Absolute entry means each point is an absolute distance from the start of the CAM table, which is an implied zero, although the starting value does not have to be zero. Relative CAMs have the entries as distance deltas from point-to-point, which means each segment is a relative move with respect to the previous segment.

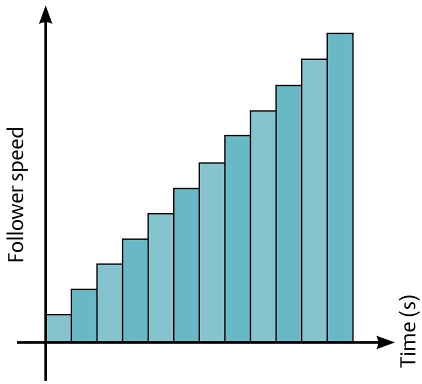
14.3.1 Interpolation

For relative, absolute and time based CAMs; the follower position is derived using interpolation types that are valid in this mode, such as:

- Linear; constant velocity across the complete point as shown previously
- Square; velocity increases or decreases linearly across the segment. This means that the position changes quadratic across the point
- S-curve; the velocity and position change along a sinusoidal shape across the point
- Cosine; the velocity starts and ends at the same velocity, but increases or decreases along a sinusoidal shape in order that the proper final position is achieved
- Jerk free; the jerk starts and ends at zero. Jerk increases or decreases in smooth transition

These interpolation methods, allow the user to quickly program sinusoidal and linear acceleration ramps over one CAM segment, where each speed step in the segment is only as wide (in time) as the motion task scheduling rate; e.g. a 1s speed ramp, with a POS task scheduling rate of 1ms will have 1000 x 1ms wide steps generating the ramp.

Figure 14-7 Motion task time slicing of a linear ramp



14.3.2 Cubic spline

Cubic spline uses the third order cubic polynomial splining to find smooth velocity and position curves from point-to-point. Like absolute, relative and time based CAMs, a table allows a different leader, follower and interpolation coordinate for each segment in the CAM.

14.4 CAM uses

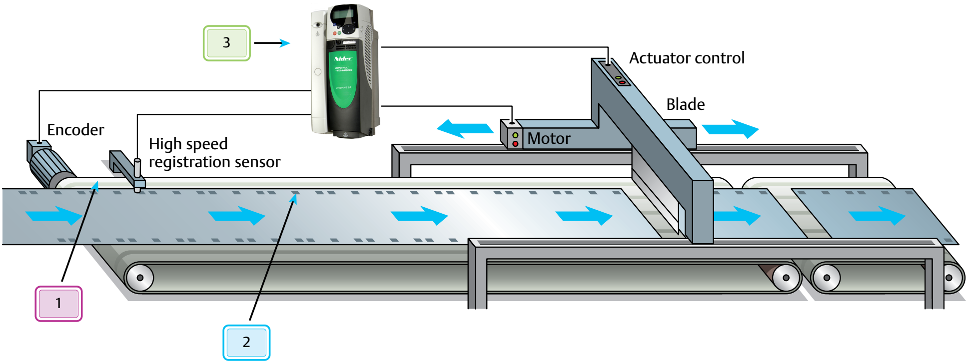
CAMs are used in high speed demanding applications such as rotary knives or flying shears, where the follower axis has to remain positionally locked to the reference, but the follower needs to speed up or slow down to gain or lose relative position with respect to the reference position.

15 Registration

Registration in its simplest form is a means by which one process can be automatically aligned with another, without stopping the flow of product, thereby maximising productivity.

Figure 15-1 shows an illustration of a machine that transforms a continuous strip of material into sheets. The strip is fed into the machine continuously and the saw is mounted on a sliding mechanism that can accelerate the saw to match its speed, allowing it to cut the product without stopping. In this application it is important that the saw cuts the material in a specific position relative to the marks on the product. The process or system to achieve this is called registration. To facilitate registration alignment 3 additional things are required:

Figure 15-1 Components of a registration system



1. A registration sensor. This must be capable of detecting either a mark on the produced item, or the item itself. The sensor must be capable of consistently detecting marks in the same way each time, and at a high speed
2. A suitable registration mark. The mark can be printed, stamped, cut, or punched depending on the item being manufactured, or could even be the product itself where height or length, or leading edge is detected
3. An intelligent system to interface the registration sensor detection signal to and action the physical motion required for the tool alignment

Figure 15-2 A range of registration marks that can be used in print applications.

Typical applications that use registration include:

- Rotary printing, where an initial color is applied to a material web and registration is used to align all subsequent colors applied to the material, so they are overlaid precisely above the initial print region
- Rotary cutting, where a material has to be cut in to sheets by a rotating knife blade at every registration mark, and registration is used to align the knife and mark
- Automated labelling machine, where a label applicator must be aligned with a particular point on a passing item and stick the label correctly. Registration is used to align the label applicator head with the appropriate place on the item
- Sorting machines, where a gate is used to sort different sized items. In this application the registration serves two purposes; recognition of a particular item, and the coordination of the motion required to sort the item correctly

Figure 15-3 Printing machine showing the multiple print sections, each applying a different color and accurately aligned with the previous sections.

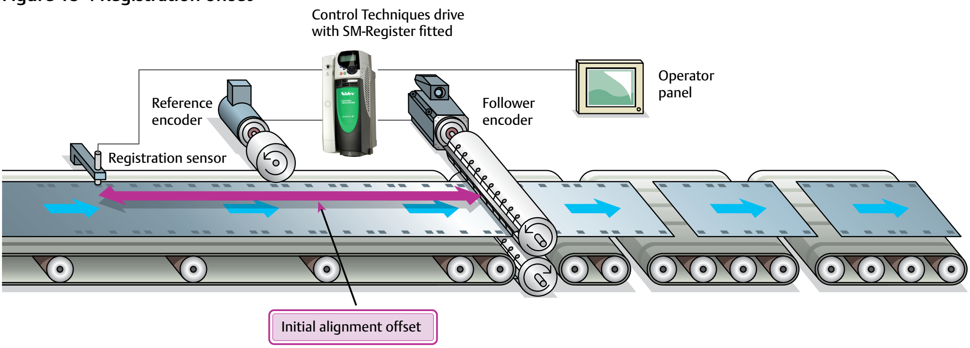
Typically, the intelligent system that interfaces with the registration sensor can be split in to 2 key areas; registration offset calculation and registration motion.

15.1 Registration offset calculation

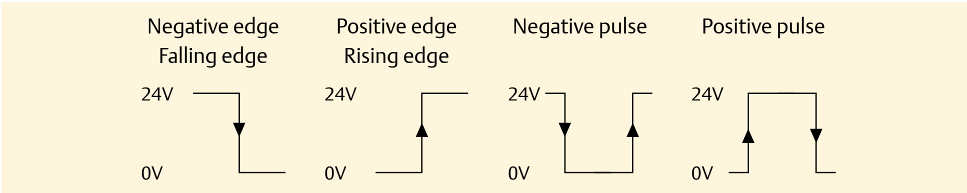
A registration offset is a distance by which a process must be moved in order to be aligned with the registration mark, or registered item being processed. In many systems there are two distinctive types of registration offset:

- Initial alignment offset; which can be a large value that is used bring the process to be aligned from standstill to alignment at the first registration mark or item seen by the registration sensor. In most basic systems this is all that is required as the motion system will come to rest after each item is processed. This offset is calculated from the distance from the registration sensor to the point where the process is supposed to happen (registration distance) and the distance required to ramp the process up to speed
- Continuous alignment offset; which is usually a very small value that makes very fine adjustments to compensate for things like material stretch, material slippage and registration mark variation. This offset is generally calculated by comparing the average item length (repeat length) against, the length measured between two contiguous marks

Figure 15-4 Registration offset



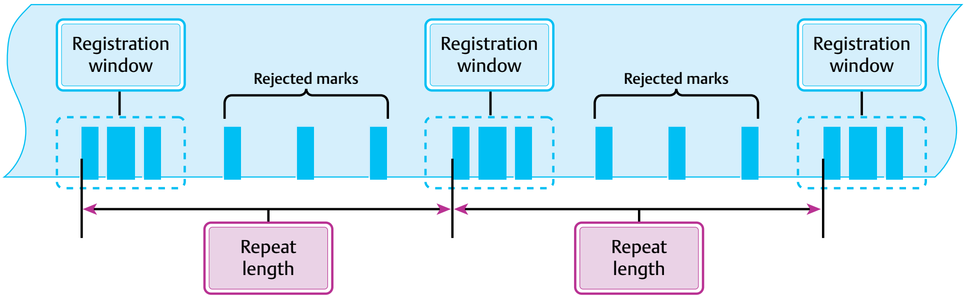
The intelligent system that generates the registration offset must be able to accurately sample the position of the motion axis upon a rising or falling edge of the registration sensor output. The accuracy of this captured position depends upon the registration sensor processing speed, output jitter, quality of the sensing electronics, and in the intelligent system, the latency of the receiving electronics, and the speed of the processor/controller used to sample the position.



In many registration applications, the item being processed will have several marks on it that may falsely trigger a registration position capture. Typically the intelligent system will be capable of filtering out any unwanted marks using one of the following techniques:

- Registration windowing; where the registration sensor output is only monitored within a positional acceptance window, such that marks detected in the window will be passed to the offset calculation. Those outside of the window will be rejected

Figure 15-5 Registration windowing



- Distance from last mark filter; where the last accepted registration mark position is compared against the current position, and only when a minimum length has passed by will new registration marks be accepted. This method is typically used in systems only capable of single rising or falling edge capture
- Pulse width filtering; where a registration mark of a particular width and tolerance will be accepted
- Pattern recognition; where a pattern of registration marks, with particular widths and tolerances will be accepted. This type of filtering is typical in high end printing application

15.2 Registration motion

The registration motion physically synchronises the process with the item being processed.

In basic systems this might be something simple like a linear slide pushing a box in to place at the right time. In this type of registration system, a registration offset distance may not be required to align the processes, and merely triggering the process directly will be sufficient. Useful programing objects are captures or queues.

On a more complex system, it is usually coordinated motion with a reference encoder or a centralised electronic line shaft which is followed by the processing axis running an electronic CAM or Digital Lock profile, and will be aligned specifically on the calculated registration offset distance.

Traditionally registration has been achieved using an external registration controller, as shown in Figure 15-6. Intelligent drives such as Control Techniques Unidrive SP and Digitax ST can achieve registration using option modules that fit within the drive itself.

Figure 15-6 Traditional registration system

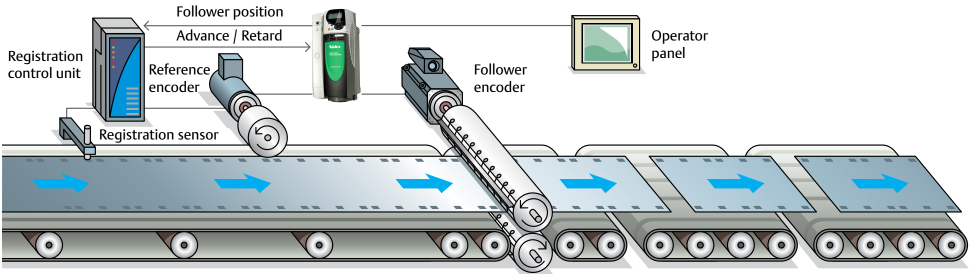
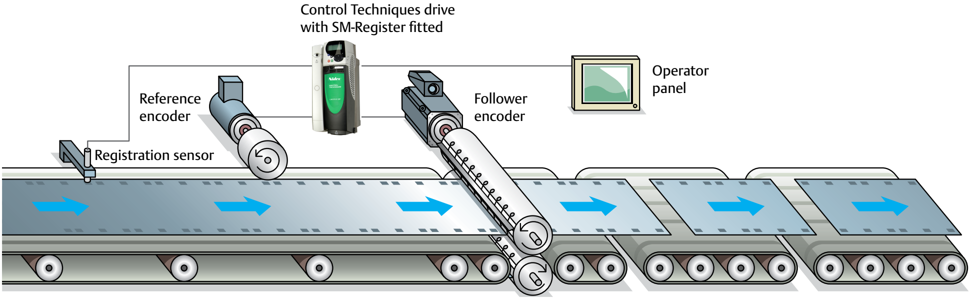


Figure 15-7 System using SM-Register



The SM-Applications, SM-Register and SM-EZMotion modules, used with Unidrive SP and Digitax ST, have features that can enable the user to develop complete intelligent registration systems. The following table show the capabilities of the two systems:

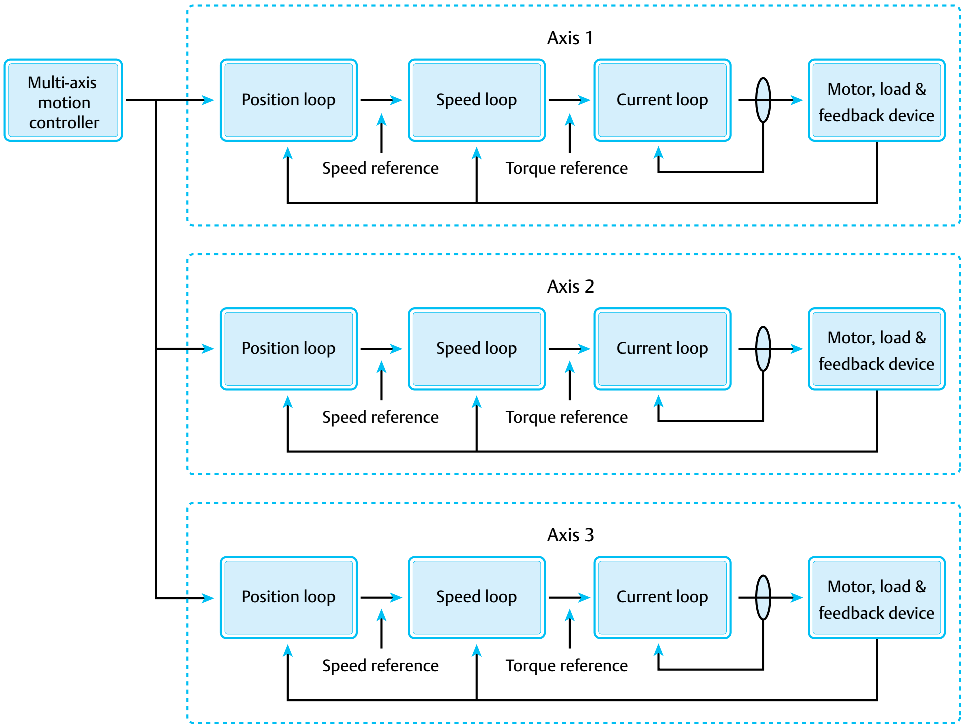
Feature	SM-Applications	SM-Register	SM-EZMotion
< 1μs capture speed	Yes	Yes	Yes
High resolution captured position	Yes	Yes	Yes
Single edge capture	Yes	Yes	Yes
Pulse capture	No	Yes	Yes
Windowing	User must write code	Dedicated feature, only requires setup	User must write code
Distance from last pulse filter	User must write code	Dedicated feature, only requires setup	User must write code
Pulse width filter	No	Dedicated feature, only requires setup	No
Pattern of pulses filter	No	Dedicated feature, only requires setup	No
Registration sensor throughput delay compensation	User must write code	Dedicated feature, only requires setup	User must write code

16 Communications systems for motion applications

A number of different networks are available that can be used for multi-axis synchronised motion. These include: SERCOS, EtherCAT and ProfiNET.

The primary applications for these fieldbuses are systems requiring centrally controlled multi-axis motion with a high degree of synchronisation and determinism ('centrally controlled'). They can also be used for applications with less stringent requirements for synchronisation and determinism, where the motion is handled more at the drive level and coordinated by a PLC as a network reference ('centrally coordinated').

Figure 16-1 Typical layout of a motion control system



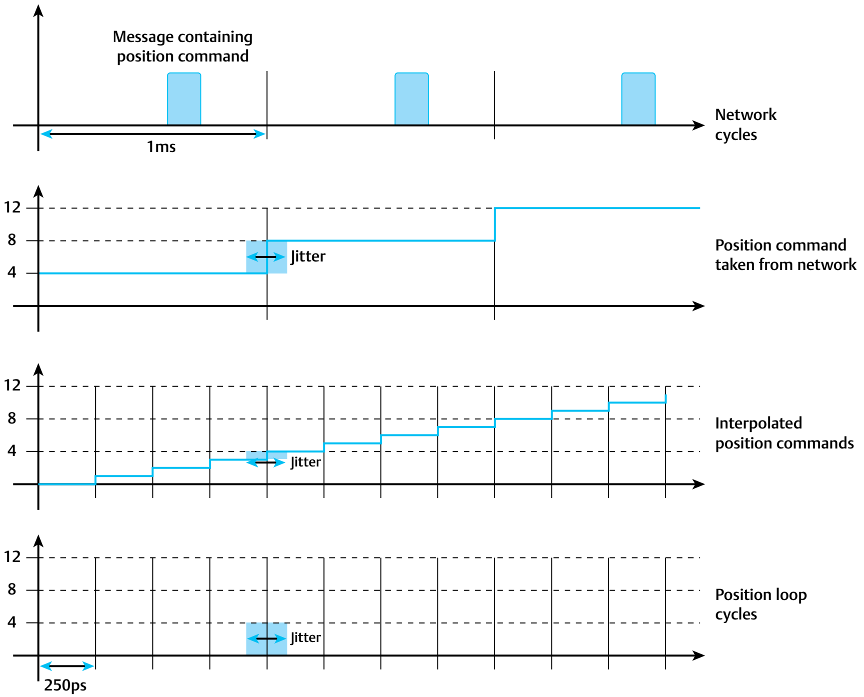
16.1 Centrally controlled systems

A common example of this type of system is where the motion coordinator is a CNC controller.

Traditionally, the standard network for this kind of application was SERCOS. Now, networks such as EtherCAT and ProfiNET IRT are also available and can be used for such applications. An important feature that all of these networks provide is the ability for the CNC controller to provide position commands over the network, and that the position control loop is closed on each of the drives. This provides a number of benefits:

- Distributed processing: As the reference no longer has to close a position loop, it just has to run an interpolator to generate position commands for all of its attached axes from a desired tool path. This means that a simpler CNC controller with a slower CPU can be used.
- Control loop delays: As the position loop is executed on each follower, it is possible to reduce the position update rate without introducing excessive delays in closing the loop. It is common for drives to close the position loop at 4kHz (i.e. every 250µs), while position commands are sent by the CNC reference at only 1kHz (i.e. every 1ms). Typically, the drives implement some form of simple interpolation between position commands to smooth the motion in this case. This can further reduce the CPU speed requirements of the CNC reference.

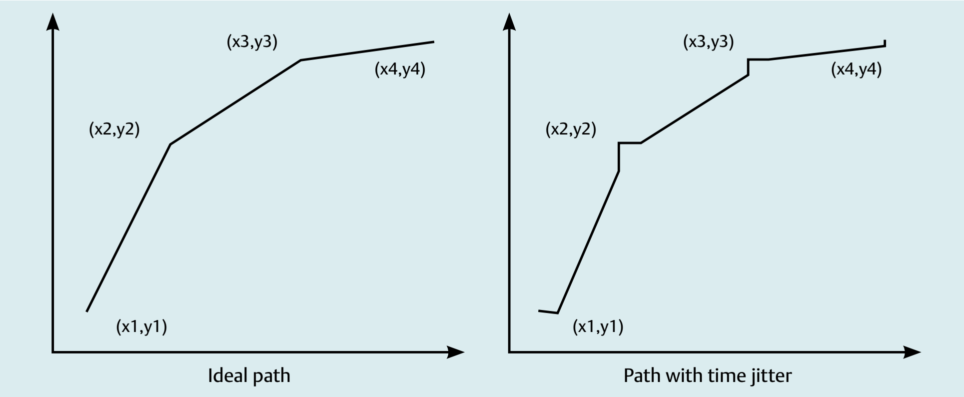
Figure 16.2 Timing diagram of a typical communication system



It is also possible, for the position loop to be closed at the controller, with it providing a velocity or current command to the follower, and the drive returning position and velocity feedback.

Systems of this type typically have very stringent synchronisation requirements. This is necessary because all of the drives' control loops must be executed at the same time. Actual positions must be sampled simultaneously, and command data transmitted over the network by the controller must be used by each of the respective drives at the same time. This is all necessary to ensure that actual physical moves occur simultaneously, so that each move in multiple dimensions appears as a straight line segment (see Figure 16-3). Time jitter between drives will result in a departure from this ideal. To some extent time jitter between axes will always occur, but synchronisation should minimise this.

Figure 16-3 Communication jitter in a motion system



In addition, when a position command is being periodically transmitted in this way, it is necessary to ensure that the control loop of an individual drive is synchronised to the CNC controller's position value generation (so that position commands are sent to the drive at the same time each and every cycle) to prevent 'phasing'. If this synchronisation is not present so that the drive control loops are 'drifting' compared with the drive cycles, problems can occur where the speed reference provided to the drive can momentarily glitch to zero causing discontinuities in the motion and acoustic noise from the motor.

This means that these networks all provide some means to synchronise axis control loops. For example, SERCOS synchronises control loops to the beginning of network telegrams (which means that the CNC controller must be able to send the telegrams with very little jitter), and EtherCAT provides a means to synchronise clocks on each of the drives and the CNC controller ('distributed clocks'). This means that more jitter can be tolerated in the telegram sending of the CNC controller.

16.2 Centrally coordinated systems

It is also possible that the drive axes implement some profiled positioning, so that the CNC controller transmits positions, and the drives move to them with pre-defined velocities and accelerations. Here the synchronisation requirements are less critical. This might be used in the pick-and-place example mentioned earlier where motion must be coordinated enough to ensure that movement is in straight lines, and defined accelerations/velocities are required to prevent item damage/loss. It makes sense for a CNC controller in this case to be as basic and low cost as possible, so that it provides drives with the locations to which they should move and ramp rates, etc. As well as providing the position loop, the drive implements a basic profile generator, providing an fieldbus interface for this functionality.

Both EtherCAT and SERCOS allow for such functionality in their respective standards. EtherCAT can use the profile position mode of operation in DS402, and SERCOS defines a system called 'drive controlled positioning'.

16.3 Distributed motion systems

Connecting drives together through a high-speed digital network with peer-to-peer or reference/follower capability in conjunction with a synchronization signal accomplishes many of the tasks a centralized control system can, but in a distributed control environment. This is advantageous in that the machine can be built without the cost and components of a PLC or multi-axis controller, saving time and money. Coordinated motion, such as circular interpolation or constant velocity cornering, can be done using a synchronization signal, like a 'virtual reference', to provide a common time reference to multiple axes. The motion of each axis can be relative to this signal. The digital network allows the motion to be initiated at precisely the same time, providing perfectly synchronized X, Y, and Z movements.

17 Glossary

1.5 Axis	A system that comprises of a single axis together with a follower/reference input to allow the motion to be synchronised with an external source
Acceleration feed forward	The acceleration feed forward term is used to compensate for the control lag during the acceleration and deceleration phases of a motion profile by increasing the demand speed during acceleration and reducing the demand speed during deceleration
Acceleration rate	The acceleration rate is the rate at which the motion will increase in speed, this is usually expresses as unit/s/s or units/s^2
Amplifier	Amplifier is used to refer to a servo drive, often where either limited functionality is available or is used
APC	Advanced Position Controller. A flexible position control kernel embedded within Control Techniques motion option modules
Axis	An axis is the name used to describe the combination of a servo motor and drive
Cam	A mechanical cam has a shaft with a shaped 'cam' object fixed to it. A follower is pushed against the cam and as the shaft turns the follower is pushed forwards and backwards. The movement of the follower is dependant upon the speed of the shaft and the shape of the follower. An electronic cam provides the same functionality to allow a follower axis to move as a function of a reference axis in a pre-determined way. The shape of the cam is defined by a look-up table that lists reference and follower positions. Cams are used to define complex movements such as in rotary knife cutting machines
CPR	Counts per revolution
Control loop update rate	This determines the frequency that a control algorithm that is implemented in a digital based system is executed
Coordinate transform	This is a mathematical function used in complex mechanical systems to transform the motor positions to the mechanical part being controlled. An example is a SCARA robot where multiple arms are moved to control the position of the robot's manipulator
Coordinated motion	See 'Interpolated motion'
Current loop	The current loop is a closed loop control algorithm for regulating the torque output of a motor. Current measurement devices within the drive are used for feedback information. The output of the current loop is used to control the output power devices within the drive
Deceleration rate	The deceleration rate is the rate at which the motion will decrease in speed. This is usually expressed as unit/s/s or units/s^2
Differential gain	The differential gain is a function within a closed loop control system that acts upon the rate of change of the error. If the error increases rapidly, the differential term provides rapid compensation

Digital lock	Digital lock is a simple reference follower application where the output of the follower is directly proportional to the input from the reference
Discrete motion	Discrete motion is a simple motion system used to provide point-to-point positioning
DPL	Drive programing language. A flexible structured text language that can be used to program Control Techniques drives
EIA(RS)-485	The EIA(RS)-485 is the most commonly used hardware layer for fieldbus systems. It defines a line with a maximum of 32 stations, terminated at both ends by bus-terminating resistors. It originally defined data rates of up to 500 kbit/s. The maximum distances can be up to 5km (hardware and data rate dependent). EIA(RS)-485 employs shielded twisted-conductor data transmission lines. This ensures a relatively high degree of noise immunity
Electronic gearbox	See 'Digital lock'
EnDAT	EnDAT is a communication system that was developed by Heidenhain primarily for transferring speed and position information from encoders
Error	Closed loop control system operation is based on the error between the desired and actual output. This can apply to the position, speed or current loop in a motion control system
EtherCAT	EtherCAT is a high speed Ethernet based communication system that is suitable for high performance motion control systems
Feed forward	Feed forward is usually used to refer to a speed feed forward term in a motion control system. This is a method used to calculate the desired speed required to achieve the desired position, based on the rate of change of the position reference. The feed forward term is used in conjunction with a position control loop
Follower	A follower axis has a motion output that is a function of a reference (also known as a master) source
Following error	Following error is the difference between the desired position and the actual position
HIPERFACE	HIPERFACE is a communication system that was developed by SICK Stegmann, primarily for transferring speed and position information from encoders
Index	An index is commonly used to define a movement of a mechanical system from one specific position to another
Integral gain	The integral gain is a function within a closed loop control system. The integrator acts on the error, increasing (ramping) its effect over time, so that if a steady state error exists it is compensated for
Interpolated motion	Interpolated motion is usually used to describe the coordination of axis to create defined motion paths in two or more axes. Various types of interpolation can be defined such as linear, circular or helical
Jerk	Jerk is a term used within profile generators to determine the rate of change of acceleration in a motion profile

Jitter	In a perfect multi-axis system, the control loops for all of the axes are synchronised together. Many systems try to achieve this such as EtherCAT. Jitter refers to the error in the synchronisation
Linear interpolation	Linear interpolation refers to two or more axes coordinated so that they operate together using the same profile to reach the desired final position at the same time. In a two or three axes system this would mean the motion being controlled by the axes moves is a straight line from one position to another
LPR	Lines per revolution
Master	A master is a reference source that is used adapt the position of a follower axis. The master source usually derives from a position measurement device such as an encoder
Multi-axis	A system that has two or more axes
PID	PID refers to a type of closed loop control algorithm incorporating proportional, integral and differential functions
PLCopen motion	PLCopen is a standard motion programming language, providing a simple interface for simple motion applications
Profile	A profile defines the shape of the velocity curve that an axis will follow when moving from one position to another. This usually includes the acceleration, deceleration, jerk and maximum speed. More complicated profiles can be defined
Profile generator	A profile generator produces a position reference that changes with time to achieve desired speed and position profiles. In multi-axis coordinated systems, the profile generator produces synchronised position references for multiple axes
Proportional gain	The proportional gain is a function within a closed loop control system that applies a direct multiplier to the error
PPR	Pulses per revolution
Quantization	Quantization noise is generated through errors between an actual value such as speed and the measured quantized digital value. In a motion system this is due to the sampling of the value and subsequent rounding or truncation In a motion system these effects can be minimised by reducing the sample rate of the motion controller, increasing the resolution of the encoder, or in a complete system by introducing filtering to the control loops
Reference	A reference is an input to a control loop that determines the desired outcome of the loop. This is also commonly known as a set-point. In motion systems a reference can also refer to an input, where a follower axis applies a function to the reference
Reference encoder	A reference encoder is used as a position reference that is used adapt the desired position of a follower axis. This is commonly used to allow two mechanical systems to synchronise
Registration	Registration is the dynamic alignment of two mechanical systems such as aligning the different colors in a printing system

SCARA robot	Selective Compliant Assembly Robot Arm
SERCOS	SERCOS is a communication system that is designed for high performance motion control systems
Slave	Alternative name for a follower. See ‘Follower’ for description
Speed loop	The speed loop is a closed loop control algorithm for regulating the speed of the motor. An encoder fitted to the motor is usually used to feedback speed information. The output of the speed loop is the current reference for the current loop
SSI	Serial Synchronous Interface. This is a communication system that was developed by Stegmann for transferring absolute position information from an encoder
Torque loop	See ‘Current loop’
Web	A web is a continuous flat material such as rolled aluminium or paper that is threaded through a machine for processing

DRIVING THE WORLD...

Control Techniques Drive & Application Centres

AUSTRALIA
Melbourne Application Centre
T: +613 973 81777
controltechniques.au@emerson.com

Sydney Drive Centre
T: +61 2 9838 7222
controltechniques.au@emerson.com

AUSTRIA
Linz Drive Centre
T: +43 7229 789480
controltechniques.at@emerson.com

BELGIUM
Brussels Drive Centre
T: +32 1574 0700
controltechniques.be@emerson.com

BRAZIL
São Paulo Application Centre
T: +55 11 3618 6688
controltechniques.br@emerson.com

CANADA
Toronto Drive Centre
T: +1 905 949 3402
controltechniques.ca@emerson.com

Calgary Drive Centre
T: +1 403 253 8738
controltechniques.ca@emerson.com

CHINA
Shanghai Drive Centre
T: +86 21 5426 0668
controltechniques.cn@emerson.com

Beijing Application Centre
T: +86 10 856 31122 ext 820
controltechniques.cn@emerson.com

CZECH REPUBLIC
Brno Drive Centre
T: +420 511 180111
controltechniques.cz@emerson.com

DENMARK
Copenhagen Drive Centre
T: +45 4369 6100
controltechniques.dk@emerson.com

FRANCE*
Angoulême Drive Centre
T: +33 5 4564 5454
controltechniques.fr@emerson.com

GERMANY
Bonn Drive Centre
T: +49 2242 8770
controltechniques.de@emerson.com

Chemnitz Drive Centre
T: +49 3722 52030
controltechniques.de@emerson.com

Darmstadt Drive Centre
T: +49 6251 17700
controltechniques.de@emerson.com

GREECE*
Athens Application Centre
T: +0030 210 57 86086/088
controltechniques.gr@emerson.com

HOLLAND
Rotterdam Drive Centre
T: +31 184 420555
controltechniques.nl@emerson.com

HONG KONG
Hong Kong Application Centre
T: +852 2979 5271
controltechniques.hk@emerson.com

INDIA
Chennai Drive Centre
T: +91 44 2496 1123/
2496 1130/2496 1083
controltechniques.in@emerson.com

Pune Application Centre
T: +91 20 2612 7956/2612 8415
controltechniques.in@emerson.com

New Delhi Application Centre
T: +91 112 2581 3166
controltechniques.in@emerson.com

IRELAND
Newbridge Drive Centre
T: +353 45 448200
controltechniques.ie@emerson.com

ITALY
Milan Drive Centre
T: +39 02575 751
controltechniques.it@emerson.com

Reggio Emilia Application Centre
T: +39 02575 751
controltechniques.it@emerson.com

Vicenza Drive Centre
T: +39 0444 933400
controltechniques.it@emerson.com

KOREA
Seoul Application Centre
T: +82 2 3483 1605
controltechniques.kr@emerson.com

MALAYSIA
Kuala Lumpur Drive Centre
T: +603 5634 9776
controltechniques.my@emerson.com

REPUBLIC OF SOUTH AFRICA
Johannesburg Drive Centre
T: +27 11 462 1740
controltechniques.za@emerson.com

Cape Town Application Centre
T: +27 21 556 0245
controltechniques.za@emerson.com

RUSSIA
Moscow Application Centre
T: +7 495 981 9811
controltechniques.ru@emerson.com

SINGAPORE
Singapore Drive Centre
T: +65 6891 7600
controltechniques.sg@emerson.com

SLOVAKIA
EMERSON A.S
T: +421 32 7700 369
controltechniques.sk@emerson.com

SPAIN
Barcelona Drive Centre
T: +34 93 680 1661
controltechniques.es@emerson.com

Bilbao Application Centre
T: +34 94 620 3646
controltechniques.es@emerson.com

Valencia Drive Centre
T: +34 96 154 2900
controltechniques.es@emerson.com

SWEDEN*
Stockholm Application Centre
T: +468 554 241 00
controltechniques.se@emerson.com

SWITZERLAND
Lausanne Application Centre
T: +41 21 637 7070
controltechniques.ch@emerson.com

Zurich Drive Centre
T: +41 56 201 4242
controltechniques.ch@emerson.com

TAIWAN
Taipei Application Centre
T: +886 2 8161 7695
controltechniques.tw@emerson.com

THAILAND
Bangkok Drive Centre
T: +66 2962 2092 99
controltechniques.th@emerson.com

TURKEY
Istanbul Drive Centre
T: +90 216 4182420
controltechniques.tr@emerson.com

UAE*
Emerson FZE
T: +971 4 8118100
ct.dubai@emerson.com

UNITED KINGDOM
Telford Drive Centre
T: +44 1952 213700
controltechniques.uk@emerson.com

USA
California Drive Centre
T: +1 562 943 0300
controltechniques.us@emerson.com

Charlotte Application Centre
T: +1 704 393 3366
controltechniques.us@emerson.com

Chicago Application Centre
T: +1 630 752 9090
controltechniques.us@emerson.com

Cleveland Drive Centre
T: +1 440 717 0123
controltechniques.us@emerson.com

Florida Drive Centre
T: +1 239 693 7200
controltechniques.us@emerson.com

Latin America Sales Office
T: +1 305 818 8897
controltechniques.us@emerson.com

Minneapolis US Headquarters
T: +1 952 995 8000
controltechniques.us@emerson.com

Oregon Drive Centre
T: +1 503 266 2094
controltechniques.us@emerson.com

Providence Drive Centre
T: +1 401 541 7277
controltechniques.us@emerson.com

Utah Drive Centre
T: +1 801 566 5521
controltechniques.us@emerson.com

Control Techniques Distributors

ARGENTINA
Euro Techniques SA
T: +54 11 4331 7820
eurotech@eurotechsa.com.ar

BAHRAIN
Emerson FZE
T: +971 4 8118100
ct.bahrain@emerson.com

BULGARIA
BLS - Automation Ltd
T: +359 32 968 007
info@blsautomation.com

CHILE
Ingeniería Y Desarrollo
Tecnológico S.A
T: +56 2 719 2200
rdunner@idt.cl

COLOMBIA
Sistronic LTDA
T: +57 2 555 60 00
luis.alvarez@sistronic.com.co

Redes Electricas S.A.
T: +57 1 364 7000
alvaro.rodriguez@redeselectricas.com

CROATIA
Zigg-Pro d.o.o
T: +385 1 3463 000
zigg-pro@zg.htnet.hr

CYPRUS
Acme Industrial Electronic
Services Ltd
T: +3572 5 332181
acme@cytanet.com.cy

EGYPT
Samiram
T: +202 29703868/+202 29703869
samiramz@samiram.com

EL SALVADOR
Servielectric Industrial S.A. de C.V.
T: +503 2278 1280
aeorellana@gruposervielectric.com

FINLAND
SKS Control
T: +358 207 6461
control@sks.fi

GUATEMALA
MICE, S.A.
T: +502 5510 2093
mice@itelgua.com

HONDURAS
Temtronics Honduras
T: +504 550 1801
temtronics@amnethn.com

HUNGARY
Control-VH Kft
T: +361 431 1160
info@controlvh.hu

ICELAND
Samey ehf
T: +354 510 5200
samey@samey.is

INDONESIA
Pt Apikon Indonesia
T: +65 6468 8979
info.my@controltechniques.com

Pt Yua Esa Sempurna Sejahtera
T: +65 6468 8979
info.my@controltechniques.com

ISRAEL
Dor Drives Systems Ltd
T: +972 3900 7595
info@dor1.co.il

KENYA
Kassam & Bros Co. Ltd
T: +254 2 556 418
kassambros@africaonline.co.ke

KUWAIT
Emerson FZE
T: +971 4 8118100
ct.kuwait@emerson.com

LATVIA
EMT
T: +371 760 2026
janis@emt.lv

LEBANON
Black Box Automation & Control
T: +961 1 443773
info@blackboxcontrol.com

LITHUANIA
Elinta UAB
T: +370 37 351 987
sales@elinta.lt

MALTA
Mekanika Limited
T: +35621 442 039
mfrancica@gasan.com

MEXICO
MELCSA S.A. de CV
T: +52 55 5561 1312
jcervera@melcsa.com

MOROCCO
Cietec
T: +212 22 354948
cietec@cietec.ma

NEW ZEALAND
Advanced Motor Control. Ph.
T: +64 (0) 274 363 067
info.au@controltechniques.com

PERU
Intech S.A.
T: +51 1 224 9493
artur.mujamed@intech-sa.com

PHILIPPINES
Control Techniques
Singapore Ltd
T: +65 6468 8979
info.my@controltechniques.com

POLAND
APATOR CONTROL Sp. z o.o
T: +48 56 6191 207
info@acontrol.com.pl

PORTUGAL
Harker Sumner S.A
T: +351 22 947 8090
drives.automation@harker.pt

PUERTO RICO
Motion Industries Inc.
T: +1 787 251 1550
roberto.diaz@motion-ind.com

QATAR
Emerson FZE
T: +971 4 8118100
ct.qatar@emerson.com

ROMANIA
C.I.T. Automatizari
T: +40212550543
office@citautomatizari.ro

SAUDI ARABIA
A. Abunayyan Electric Corp.
T: +9661 477 9111
aec-salesmarketing@
abunayyangroup.com

SERBIA & MONTENEGRO
Master Inzenjering d.o.o
T: +381 24 551 605
office@masterinzenjering.rs

SLOVENIA
PS Logatec
T: +386 1 750 8510
ps-log@ps-log.si

TUNISIA
SIA Ben Djemaa & CIE
T: +216 1 332 923
bendjemaa@planet.tn

URUGUAY
SECOIN S.A.
T: +5982 2093815
jose.barron@secoin.com.uy

VENEZUELA
Digimex Sistemas C.A.
T: +58 243 551 1634
digimex@digimex.com.ve

VIETNAM
N.Duc Thinh
T: +84 8 9490633
infotech@nducthinh.com.vn

* Operated by sister company

P.N. 0704-0007-02