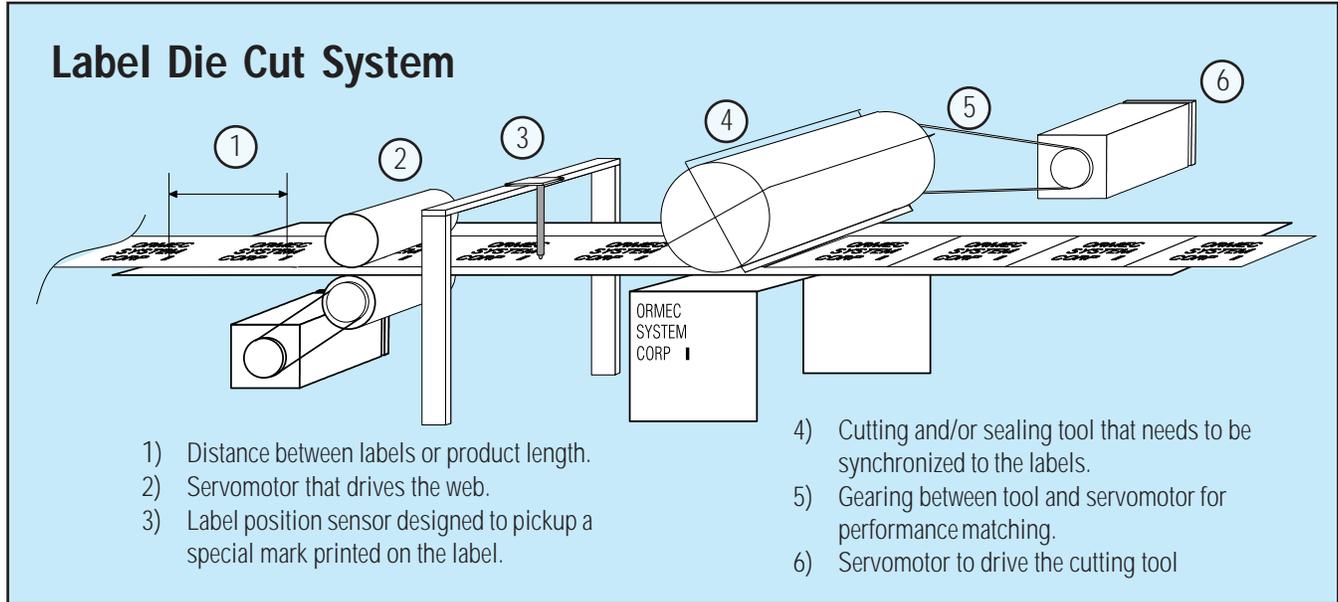




REGISTRATION FOR A CONTINUOUS WEB SYSTEM



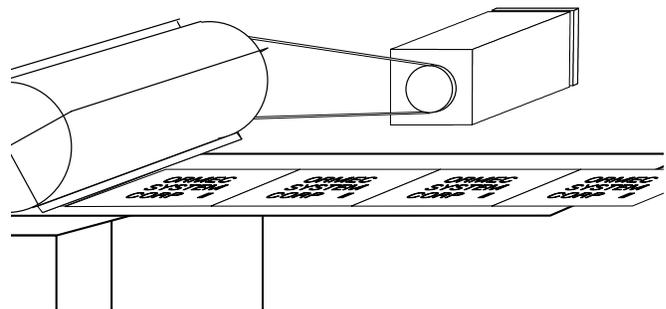
Overview

In a variety of packaging, cartoning, and converting machines, electronic registration techniques solve a multitude of problems arising from the fact that systems developed for automation must efficiently deal with inconsistencies in processing a product.

Some of these issues include:

- variations in material properties (thickness, width)
- environment conditions that create variations in the raw material (heat, humidity)
- dynamic conditions in the machine which affect the material (stretching, compressing)

For example, labels that are printed on material that stretches and shrinks due to temperature or humidity changes can be a problem. The print will not be placed consistently enough for repetitious cutting at a fixed distance. Unless the machine is designed to



Servomotor coupling ratio and tool circumference define the relationship between tool position and the material on the web.

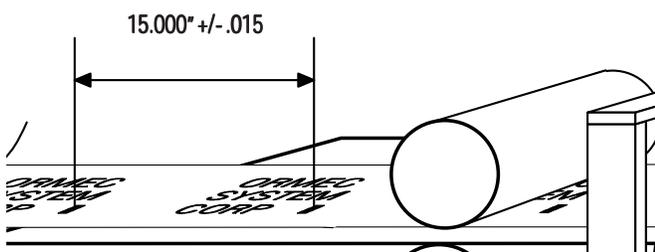
compensate for the inconsistent label placement, the printed information will not match to the label's edge after the cut. In fact it is more likely that the machine will cut right through the printed information rendering the label as useless.

Advantages of Servo-Driven Electronic Registration

The combination of computer controls and servo system performance provides effective tools for designing automation systems.

Benefits which ORMEC can provide in these applications are:

- precise servo-driven electronic gearing which greatly improves gearing performance compared to previous electromechanical designs;
- flexible software commands which allow



Label placement on the web can vary enough to present a problem.

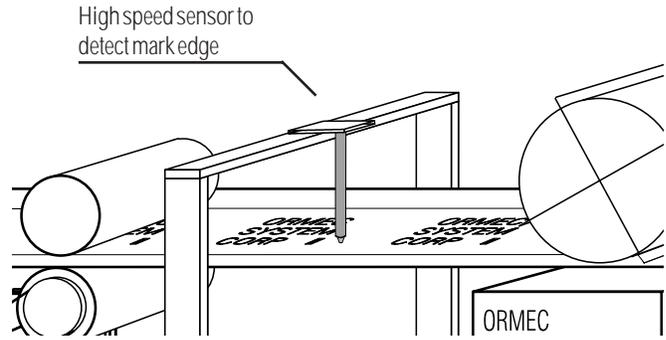
machine parameters to be dynamically adjusted and controlled for better machine design and operation

- Ability to use software to adjust machine operation dynamically and compensate for inconsistencies in the raw material
- tightly coupled, fast response to machine I/O inputs and outputs (including programmable limit switches and high-speed latches interfaced to sensor inputs) resulting in tighter control of critical operations

This application note provides an overview of electronic registration techniques designed and developed by ORMEC to deal with these problems.

Phasing

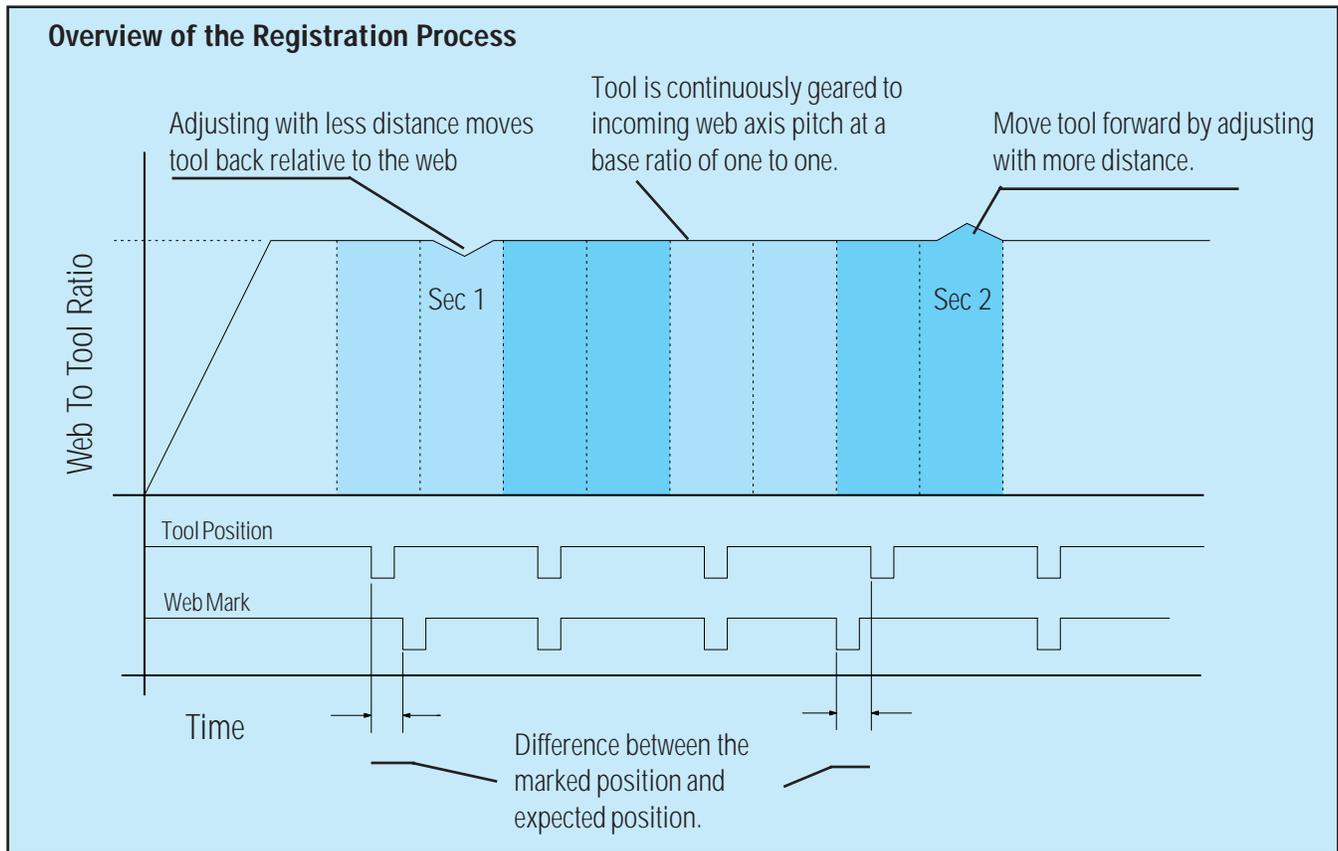
Electronic registration techniques allow the realignment of the cutting tool to assure that the printed information is in the proper position before the cut occurs. This realignment process involves an adjustment called *phasing* to be performed on the mechanical shaft holding the cutting tool. The cutting tool axis is designed with a definite relationship to the moving material. The distance required between cuts and how much the tool axis must travel, expressed as a gear ratio, defines this relationship. The ability to vary this relationship by increasing and decreasing the



Sensor to detect the mark on the material is wired directly to the DSP card.

gear ratio electronically allows changes in the phase relationship of the tool relative to the material being cut.

During the registration process, the correct amount of phasing is determined by having the system detect the position of a mark printed on the material. By constantly comparing the position of the mark to the position of the cutting tool, the phasing process compensates for any position difference by temporally adjusting the gear ratio. A temporary increase of the gear ratio advances the speed of the tool allowing the tool to catch up to the mark. Decreasing the gear ratio retards the speed of the tool allowing the mark to catch up to it.



(Sec 1) The web mark appears later than expected. System must move the tooling back, so the system executes a move using the gearing statement there by subtracting from the tool distance during the next cycle.

(Sec 2) The web mark appears sooner than expected. System must move the tooling forward, so the system executes a move using the gearing statement there by adding to the tool distance during the next cycle.

Superimposition

The adjustment of an running gear ratio to accomplish phasing is implemented by a software technique called *superimposition*. To define *superimposition* we must first define the electronic gear software statement. The electronic gear statement to set up a fundamental base ratio command is:

GEAR AT xxx TO yyy

where 'xxx' and 'yyy' are simply the numerator (output) and denominator (input), respectively, of the required ratio. This statement creates a constant relationship between the amount of distance the cutting tool will move to the amount of label material moving under the tool.

The software statement to perform intermittent gear ratio commands is:

GEAR FOR www IN zzz

where the parameter 'www' is the precise distance to move the cutting tool for the amount of 'zzz' material that travels under the cutting tool.

Once the system is running with the 'GEAR AT' statement, the 'GEAR FOR' is applied to the tool axis while moving at web speed. As the amount 'zzz' of the material passes under the tool, the final distance traveled by the tool is equal to the distance normally

traveled by the tool due to the 'GEAR AT' plus the distance specified in the 'www' parameter of the GEAR FOR statement. The final result of this action *superimposes* the GEAR FOR onto the GEAR AT command

If the distance parameter 'www' specified in the GEAR FOR statement is a negative value, the cutting tool phase is adjusted in the opposite direction. This effectively decreases the base ratio for a short travel of the web distance 'zzz', slowing the tool.

Since parameters in the gearing statements are specified as distance to be traveled, both the tool and the material are controlled and synchronized with respect to their relative positions, creating a system allowing for accurate and repeatable adjustments.

In electronic registration, precise measurements of the printed mark and the tool position is important. High speed sensors interfaced to fast reacting hardware coordinated with a industry standard digital signal processor (DSP) becomes the performance foundation needed for successful registration designs. The following section of this app note reveals one of the many possible ways servomotors, DSP axis modules, ORION PC-based controllers and the MotionBASIC language can be used to implement a practical registration system for a variety of situations.

Tutorial on MotionBASIC language syntax for simple gearing statements

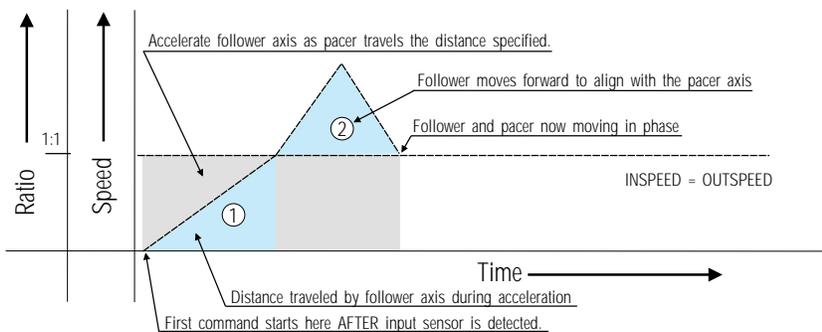
The MotionBASIC language includes the standard BASIC language commands with the addition of special commands for the purpose of implementing routines to control ORION servomotor positioning systems. The extent of these additional commands range from simple point-to-point moves up to complex gearing statements

involving high speed input and output transactions. Syntax for these commands allow expressions to cover multiple axes with variations on when and how the command is performed. The example below illustrates how a geared axis synchronizes itself to the pacer or master axis. To analyze the first GEAR statement (1) it is

(1) GEAR axis AT outspeed TO inspeed IN pacer dist AFTER input

(2) GEAR axis FOR pacer dist/2* IN pacer dist

best to start at the end and then work back. *After input is sensed, while pacer covers pacer dist, reach a ratio of outspeed to inspeed with axis and maintain that ratio.* The GEAR statement (2) will advance the geared axis the amount of distance it fell behind during the acceleration which turns out to be 1/2 the distance travelled in the first statement. To accomplish this the second statement superimposes a move to advance the geared axis by 1/2 pacer dist 'IN' the distance specified as pacer dist. This distance depends on how much room on the pacer there is to catch up and the top speed the geared axis has to reach to complete the synchronization.



Using the superimposing capability of the DSP axis module the GEAR FOR command will execute on top of the GEAR AT to align the follower shaft to the pacer shaft allowing phase adjustments.

* Once the GEAR AT command is complete (1), follower is behind the pacer by 1/2 the pacer distance travelled when the ratio is one to one.

Registration System Implementation

Design Overview

Two types of design considerations must be taken into account when a registration system is built:

- (1) the pitch is not always constant and;
- (2) the tool may not always lay at the desired point on the web of material.

Also, adjustments to the tool (in this case the knife) must be made within the cycle time of the machine to be effective and without causing undue stress on the mechanical parts of the machine.

To minimize stress due to large jerky motion in the mechanical system, pitch differences and tool-offsets are executed as smooth adjustments and at the same time achieving maximum performance. Effective performance, measured in speed and accuracy, is accomplished using high speed sensors interfaced to the hardware based motion I/O built into the axis module with microsecond response times to capture the precise position of the driven motor shaft at the point the sensor detects the physical mark.

Smooth adjustments are the result of the ability to sample the state of the machine at high repeatious rates using a digital signal processor on the axis control module and exploiting the fast arithmetic capability of the main processor executing registration algorithms programmed in MotionBASIC.

The knife axis position relative to the web mark, obtained from the action of the sensor, is used to calculate proper tool-offsets. These offsets are then applied as adjustments to the position of the servomotor driving the knife in-between cuts. Pitch length

is acquired by taking two web measurements, one at MARK A and one at MARK B, forming the difference, and comparing it to the expected range. Pitch adjustments are made to the base gear ratio of the servomotor driving the knife over a specified number of marks to lessen the overall effects of a change. With a gradual adjustments to the pitch, repetitious tool-offset adjustments are kept to small incremental moves.

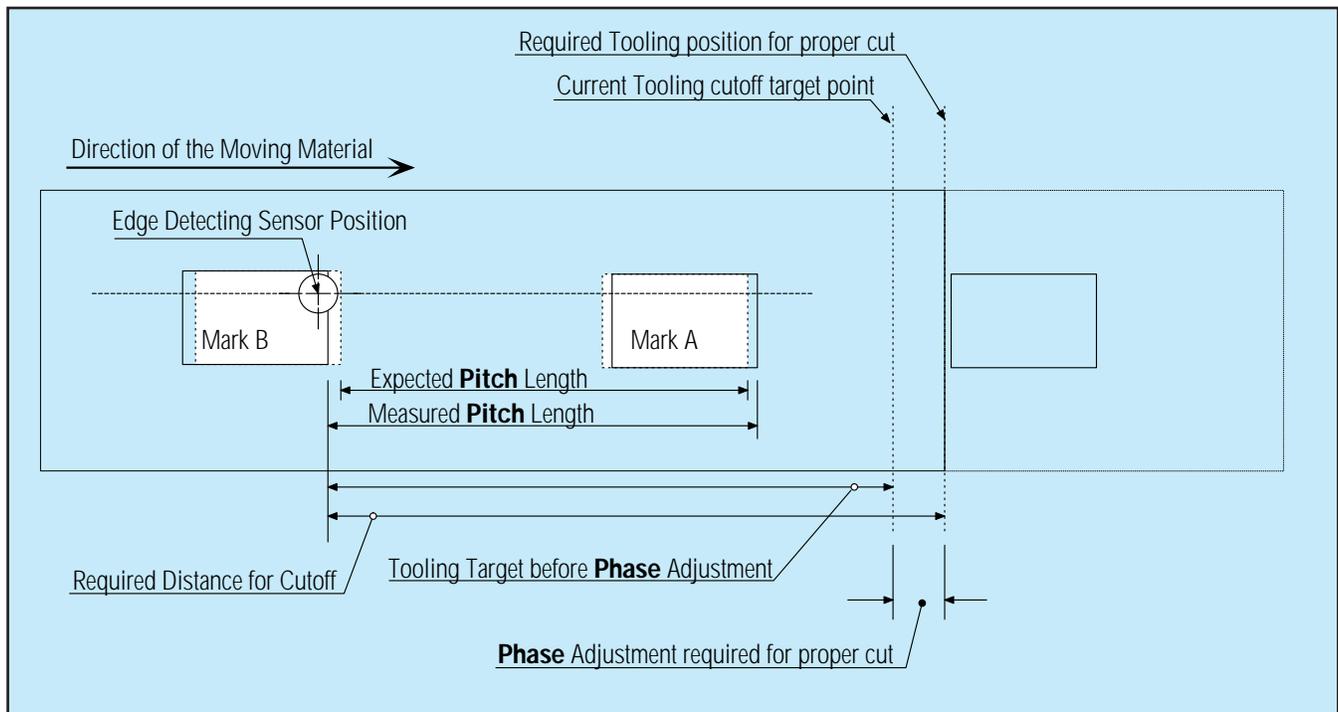
Position Capture Registers

Adjustments to the pitch length and tool-offset are referred to as trending and instantaneous registrations. Both use the position capture register information from their respective servomotor's feedback data. *Trending registration* adjustments vary the pitch using GEAR AT statements and are filtered to smooth out transitions on the web axis. *Instantaneous registration* adjustments vary the phase of the knife axis using GEAR FOR statements and are performed on the knife axis in between cuts, providing immediate correction of the knife's position.

The following sections will guide an endeavor to understand the interaction between hardware and software components of an ORION system that is designed to implement registration.

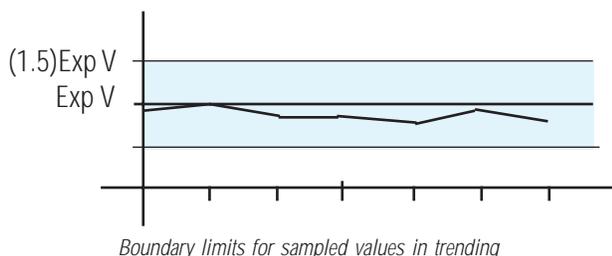
Trending Registration

Each time the mark is sensed, the position of the web is captured. The captured mark position is compared to the previous mark position and the difference



Details of mark position distances for pitch length measurement and tool position for knife cut points on web material.

is checked against boundary limits. Limits are set according to a rule that states; if the new pitch value is greater than one half the expected value and less than one and a half times the expected value, it is valid therefore use it. If the value is out of range then ignore it. This rule will prevent a phenomena called aliasing which can make recovering from false pitch length readings difficult for the system.



After the pitch sample is checked, the value is put through a filter to allow large fluctuations to be applied gradually to the gear ratio between the tool axis and the web axis. The filter will bring a change in the pitch up to 90% of its applied value within a specified number of samples. The number of samples is used to determine the coefficient for the filter algorithm. The basic filter algorithm is:

$$FILTER_{OUT_{new}} = (1-coeff)FILTER_{OUT_{old}} + coeff (FILTER_{INPUT})$$

This is a simple recursive filter to which the coefficient is determined by:

$$coeff = 1-(0.1)^{(1/n)}$$

where n is the number of pitch samples taken. The output of the filter $FILTER_{OUT_{new}}$ is the new pitch length to be applied to the ratio between the tool and the web. For the case of when the tool is mounted on a rotary knife and one turn on the knife is matched up to one product length for the pitch, the following gearing statement would be executed;

GEAR knife AT 3600 TO pitch_length IN T_dist

where 3600 represents 360.0 degrees per turn of the knife, pitch_length is in inches, and T_dist is the distance, in inches, the web will travel to make this trending adjustment.

Instantaneous Registration

In instantaneous registration we are concerned that the tool is in the proper position to make the cut. To meet this objective the adjustment must be made as soon as it is possible. Also the adjustment to be made, must be completed in the shortest distance possible which, in the case of the rotary knife, is less than one half revolution of knife travel. Finally, a check is performed on the magnitude of the adjustment to assure it is acceptable to the system.

Once all of the above criteria checks and modifications are made, the adjustment is applied to the system with the follow MotionBASIC statement:

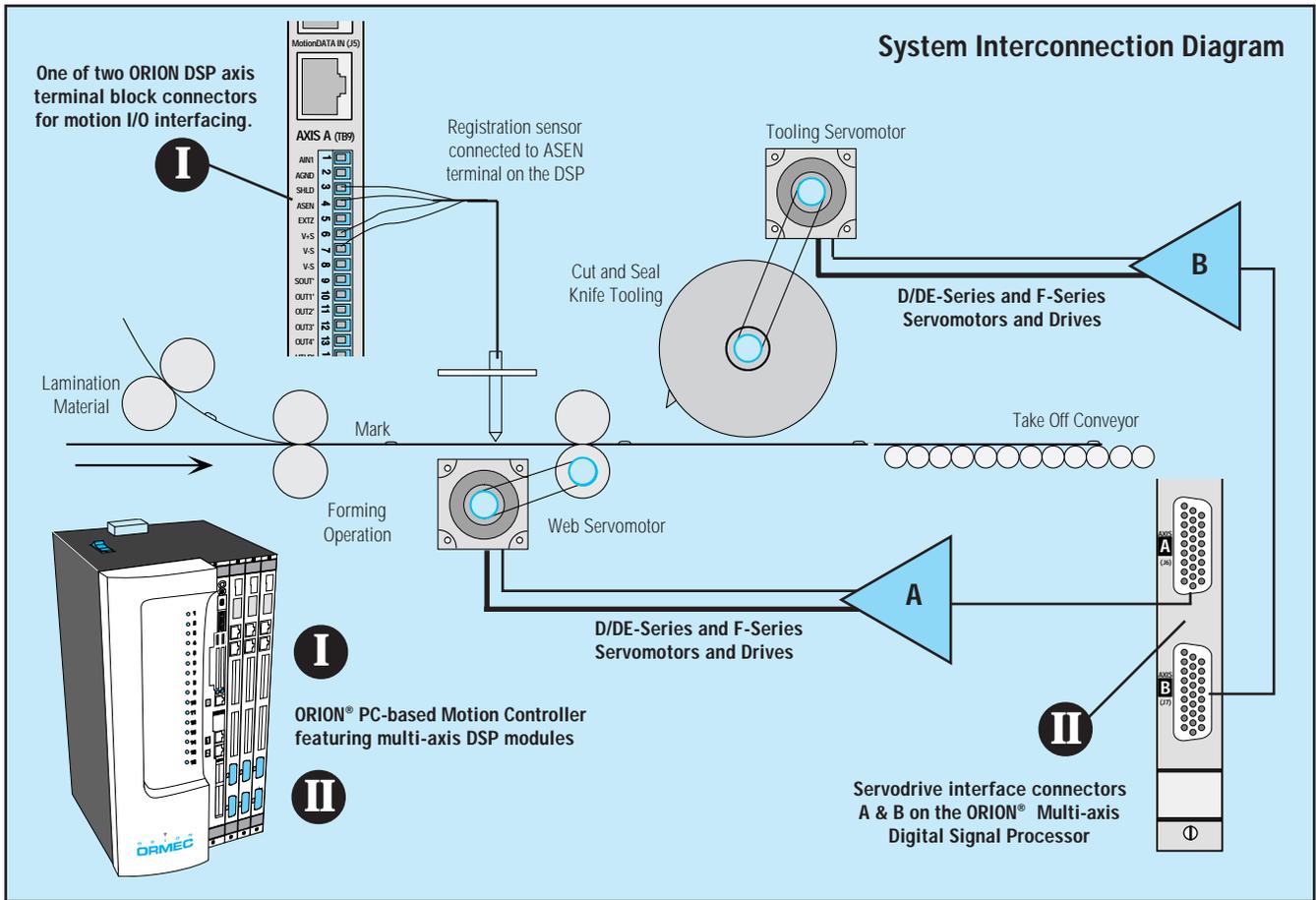
GEAR knife~ FOR corr& IN I_dist

where the correction corr& will be made in distance I_dist travelled by the web axis.

The technique of using distance as parameters for the GEAR statement assures that the control of the web is

Trending Subroutine Example	
<pre>TREND: IF OLD.POS.ASEN& <> 0 THEN PLEN.TMP& = POS.ASEN@(web~) - OLD.POS.ASEN& IF PLEN.TMP& > TREND.SET&\2 AND PLEN.TMP& < 3*TREND.SET&\2 THEN PLEN& = ((10000 - FCOEF&)*PLEN& + FCOEF&*PLEN.TMP&)\10000 ENDIF ENDIF OLD.POS.ASEN& = POS.ASEN@(web~) IF DSP.DONE@(knife~) THEN GEAR knife~ AT 3600 TO PLEN& IN T_DIST& ENDIF RETURN</pre>	<p>NOTE: This subroutine is designed to be executed every time the sensor detects a mark on the web. The MAIN routine calling TREND is outlined on page 7.</p>
<p>POS . ASEN@ (web~) is an ORMEC variable that holds the latched position of the web after the sensor, wired to ASEN input on the DSP card, makes a transition.</p> <p>DSP . DONE@ (knife~) is an ORMEC variable that holds the status of the DSP card activity.</p> <p>NOTE: Any variable name that has the symbol & as a suffix represents a 32 bit long integer variable and the operation denotes integer division.</p>	<p>TREND . SET& holds the expected pitch length,</p> <p>PLEN . TMP& holds the measured pitch length,</p> <p>PLEN& holds the applied pitch length,</p> <p>FCOEF& = 10000 * [1 - 0 . 1 ^ (1 / num)]; scaled to 10000 for accurate integer arithmetic.</p> <p>T_DIST& holds the web distance used to adjust the pitch length,</p>

MotionBASIC code for the trending registration portion of an ORION system designed to control a continuous web process.



independent of time. Therefore, the motion of the knife will be synchronized to the web even as the web axis accelerates and decelerates. The ability to stay synchronized at varying speeds is valuable during the initial alignment procedure and can also be used to

regulate the machine speed to meet overall process requirements without changing major software parameters.

Instantaneous Registration Subroutine Example

NOTE: This subroutine is designed to be executed every time the sensor detects a mark on the web. The MAIN routine calling INST is outlined on page 7.

```

INST:
  DIFF& = INST.SET& - POS.ASEN(knife~)
  IF DIFF& > POS.MOD@(knife~)\2 THEN
    DIFF& = DIFF& - POS.MOD(knife~)
  ELSE
    IF DIFF& <= -(POS.MOD@(knife~)\2) THEN
      DIFF& = DIFF& + POS.MOD(knife~)
    ENDIF
  ENDIF
  IF ABS(DIFF&) > MAX.DIFF& THEN
    DIFF& = SGN(DIFF&)* MAX.DIFF&
  ENDIF
  CORR& = (DIFF& * INST.GAIN&) \ 10000
  GEAR knife~ FOR CORR& IN I_DIST&
RETURN

```

INST.SET& holds the desired position of the knife relative to the mark point on the web; can be used to align knife to the cut point,

DIFF& holds the measured difference between the desired position and the actual position of the knife,

CORR& holds the correction distance to be applied to the knife to perform the phase correction.

POS.ASEN@(knife~) is an ORMEC function that returns the latched position of the knife after the sensor, wired to ASEN input on the DSP card, makes a transition.

POS.MOD@(knife~) is an ORMEC variable that defines the modulo user distance defined at configuration time. For this example the variable is loaded with 3600 representing 360.0 degrees based on its user units.

INST.GAIN& hold a percentage (0 to 99) to provide a way to control how the correction is applied when the distance from the sensor to the knife is greater than one pitch length. Otherwise use 100% is the distance is less than one pitch length.

I_DIST& holds the distance the web travels during the toll phasing adjustment

NOTE: Any variable name that has the symbol & as a suffix represents a 32 bit long integer variable, and the operation \ denotes integer division.

MotionBASIC® code for the instantaneous registration portion of an ORION system designed to control a continuous web process.

Continuous Web Registration Program Example

MAIN:

```

' Set axis values into appropriate variables
web~ = {1}
knife~ = {2}

' Set up all values for motors specified including user
' unit scaling values and MotionData links between the
' web axis and knife axis.
MP.CONFIG1

' Set the registration default values
REG.CONFIG2

' Implement initial ratio between web and knife
GEAR knife~ AT 3600 TO TREND.SET&

' Start web axis; run to 1000 RPM in 1.000 seconds
MOVE web~ AT 1000 IN 1000

' Clear sensor latch
ASEN@({web~, knife~}) = 0

' Loop to perform registration action
WHILETRUE
' Check sensor
IF ASEN({web~, knife~}) = TRUE THEN
' Do trending registration subroutine
TREND
' Do instantaneous registration subroutine
INST
' Clear sensor latch
ASEN@({web~, knife~}) = 0
ENDIF
WEND
END
    
```

¹ NOTE: The subroutine MP.CONFIG is produced by completing setup menus for the servomotor and is loaded with the main program on power up. An examination of the code produced for MP.CONFIG goes beyond the scope of this document. Details are available upon request.

The MotionBASIC code to the left is an example program for a simple continuous web implementation. Subroutines **TREND** and **INST** were outlined in the previous pages. Variables used for registration are explained along side of the subroutine code. The figure below gives an overview of the specific DSP hardware used in this implementation. An additional feature is outlined on the back page which demonstrates how to configure DSP Motion I/O to “blind” the mark sensor input for a portion of the cycle. Blinding is used to specify a window in the machine cycle where an sensor input signal can only be allowed to be used by the system.

²Registration Configuration Subroutine

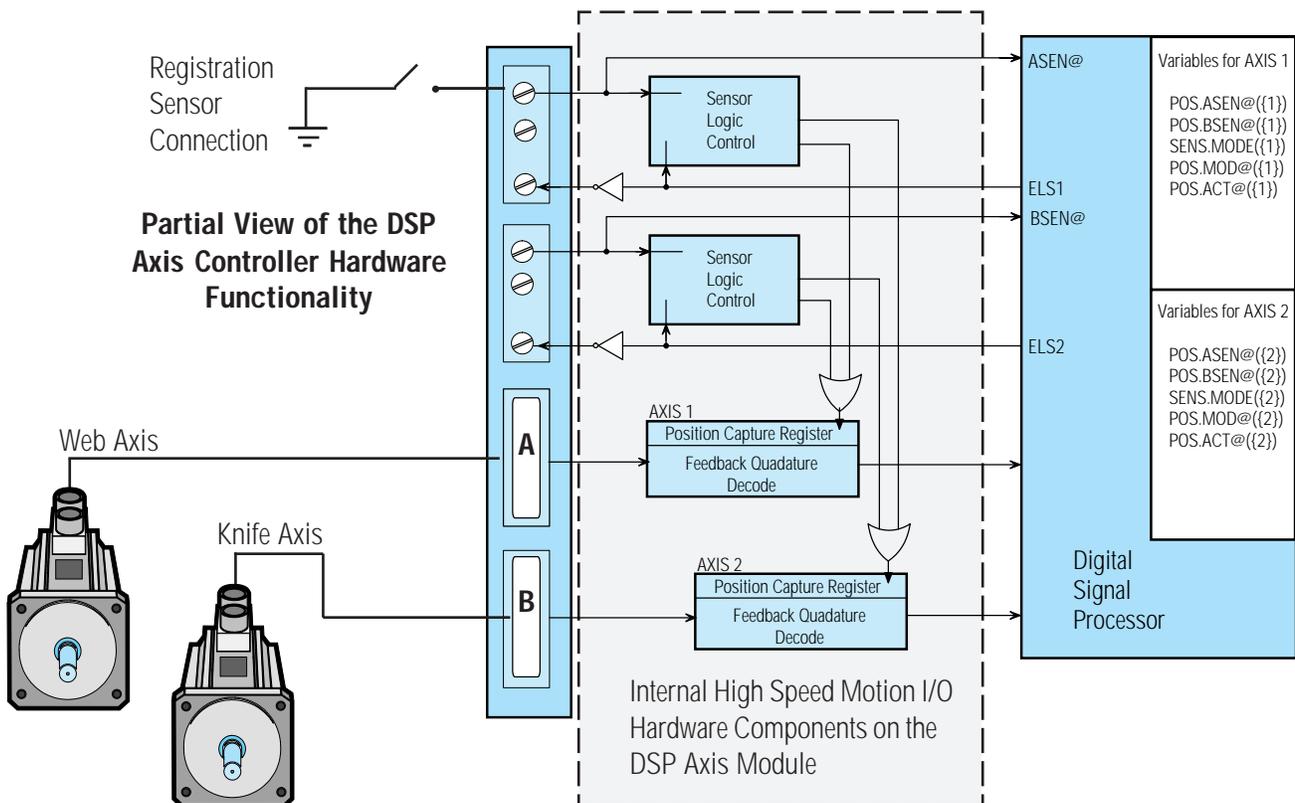
```

REG.CONFIG
INST.SET& = 0           ' 0.0 degrees
MAX.DIFF& = 300        ' 30.0 degrees for allowed error
L_DIST& = 500           ' 0.5 ins of web travel for tool adj.
INST.GAIN& = 2500       ' Correction gain set to 25.00%
TREND.SET& = 15000      ' Set point of 15.000 ins for pitch
T_DIST& = 500           ' 0.5 ins of web for pitch adj.
PLEN& = TREND.SET&     ' Initial pitch length value set.
NUM = 5                 ' Samples to spread pitch length adj.

' Sets up the mark sensor interface on DSP Axis Motion I/O
SENS.MODE@({web~, knife~}) = "RRRNN"

' Filter coefficient floating point calculation scaled to 10000
' and converted to a long integer.
FCOEF& = 10000 * [1 - 0.1^(1/NUM)]

' Force two register reads before calculating a new pitch.
OLD.POS.ASEN& = 0
RETURN
    
```



Blinding Mark Sensor Using Programmable Limit Switches

Blinding Requirements

There are times when a mark needs to be ignored by the system during part of a cycle. To conveniently provide this capability the ORION DSP axis controller is designed with the ability to logically block the sensor input. The sensor input signal for an axis can be logically blocked based on the position of the axis it is associated with, or the position of an alternate axis. The motor position is linked to one of three output points called programmable limit switches (PLS). The limit switch is set by software to turn-on and turn-off at precise positions of a defined cycle.

The cycle is defined by the MotionBASIC `PLS.modulo` function simply establishes a turnover point that ends and begins the cycle.

Modulo PLS Function

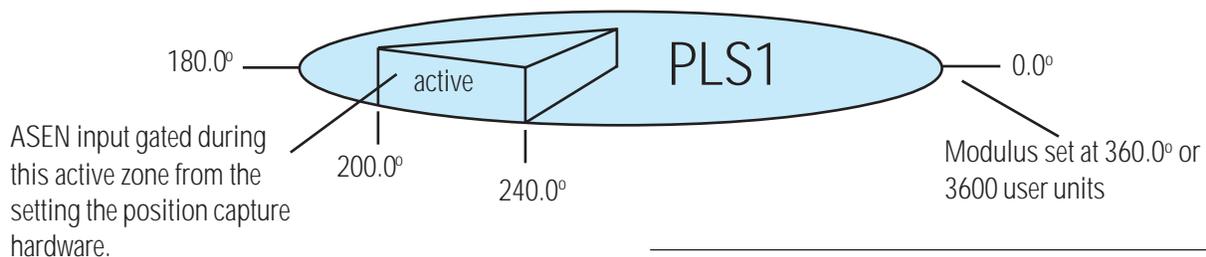
The PLS modulo function is controlled by an ORMEC variable which is set with a cycle length of the PLS, defined in user units. For example; when the cycle needs to be one revolution of the knife and which has a user unit of 0.1 degree per count, the variable would be set as follows:

```
PLS.MOD@{knife~} = 3600
```

Now that the axis cycle is established, the limit switch action is programmed into the DSP card to repeat for every revolution of the PLS.MOD@ cycle. The turn-on and turn-off points are loaded into the PLSX.yyy@ variable where xxx specifies the low and the high point zone for limit switch number x. The value of the action point is loaded as a user unit value. For example:

```
PLS1.LOW@{knife~} = 2000 ' 200.0 degrees  
PLS1.HIGH@{knife~} = 2400 ' 240.0 degrees
```

The definition of 'LOW' in the case of the limit switch



means that the signal is an active signal. The physical limit switch is an open collector circuit with a normally open switch configuration. The 'ON' condition of the switch is a low voltage potential while it is conducting current. The 'OFF' condition of the switch is a high voltage potential while it conducts very little current. To provide blinding control the PLS outputs are internally wired to the sensor logic circuits. Action of the PLS on the sensor is configured by setting the SEN.MODE@ variable.

Sensor Logic Control

The DSP controls action of the sensor signal and how it latches the position of the servomotor into the capture register. The ORMEC variable that controls the action of the DSP input sensors is:

```
SENS.MODE@{axis~} = "RRRGN"
```

The gating control of the ASEN input to its own axis position is dedicated to the PLS1 output. The PLS2 output controls the action of the BSEN input on the alternate axis position. The string expression in the above statement defines the action of all the sensors for one axis;

```
ZREF, ASEN, BSEN, PLS_GATE_A, PLS_GATE_B
```

where: Rising edge = R / Falling edge = F
Gate enabled = G / Gate disabled = N

Each character in the string is a place holder for the above inputs and controls. Sensor input ZREV is not gated. The string expression configures the sensors to be 'rising edge' for ZSEN, ASEN, and BSEN along with gating on ASEN and no gating on BSEN. Configurations can be modified while system is running. Mark sensor blinding can be set anywhere within the operating cycle of the system.

ORMEC



ORMEC Systems Corp.
19 Linden Park, Rochester NY 14625
(585) 385-3520 FAX: (585) 385-5999

Internet E-mail: sales@ormec.com or support@ormec.com
Internet WWW: <http://www.ormec.com/>