TECH NOTE 120-1: Rotating Print/Cut Tool with Registration Applications

Web handling with registration applications include the following main components (Fig. 1, Fig. 2, Fig. 3):

1. HMI as the human machine interface device
2. PLC/Motion Controller as the machine controller
3. Master Encoder (or a Servo motor with encoder)
4. EtherCAT Slave Module for Master Encoder interface
5. Slave Servo Motor for the tool axis (print head or a cut head) which follows the Master Encoder in order to match the speed of the web.
6. Registration mark sensor

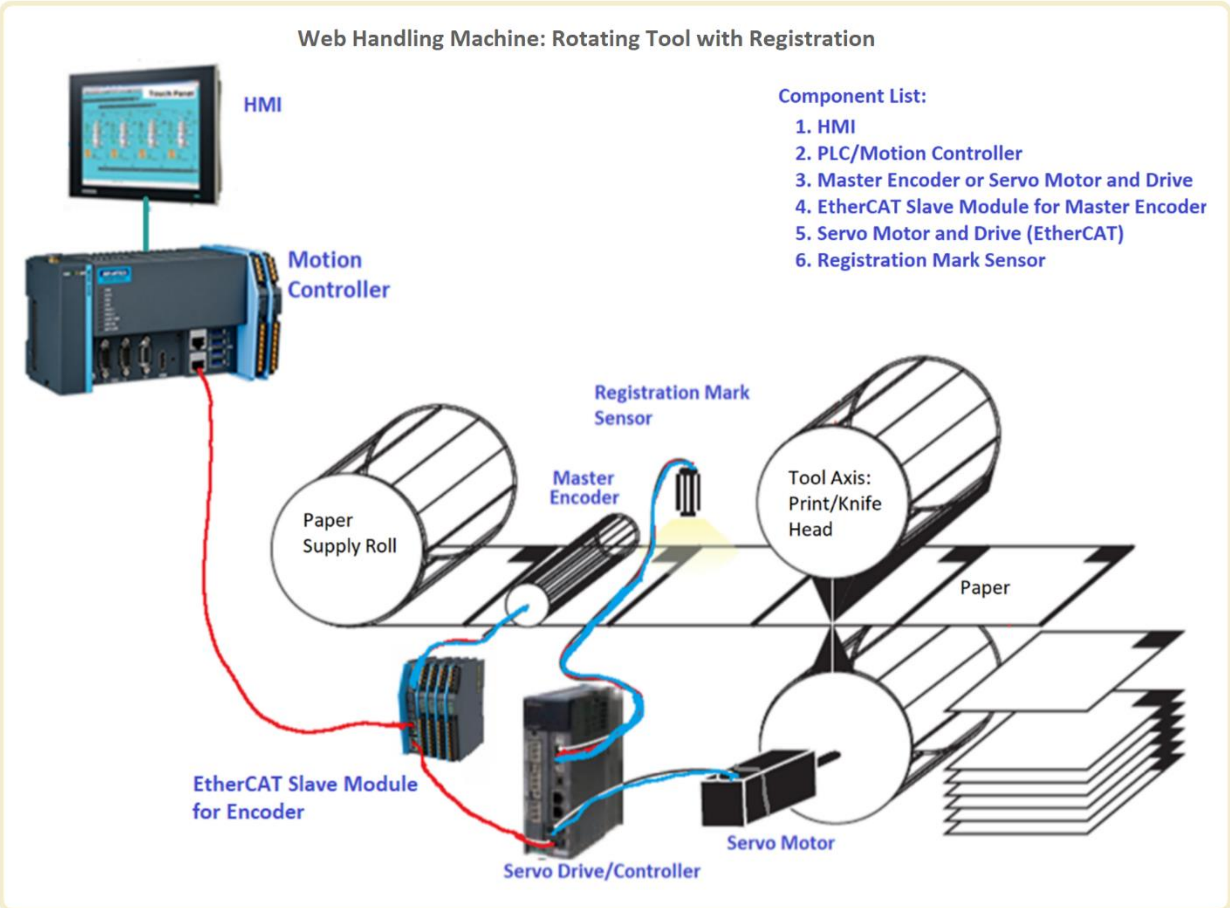


Fig. 1- Web handling machine with rotating knife: master axis is encoder which measures web speed and position. Slave axis is the knife axis which is electronically geared to Master axis and has registration mark sensor based motion correction.

The key motion coordination requirement in web handling applications is that during the period during which the web and tool are in contact (contact phase), the tool (cutting knife, print head, sealing head, etc.) must be moving at the same linear speed as the web. When there is no contact between the tool and the web (non-contact phase), they don't have to match the linear speeds. In addition, the application may require "registration", which means that during the contact-phase, the position of the tool with respect to web must be accurate relative to a mark on the web. For that purpose, there is a "registration mark sensor" which is used to make position corrections to make the relative positions accurate despite imperfections in the process.

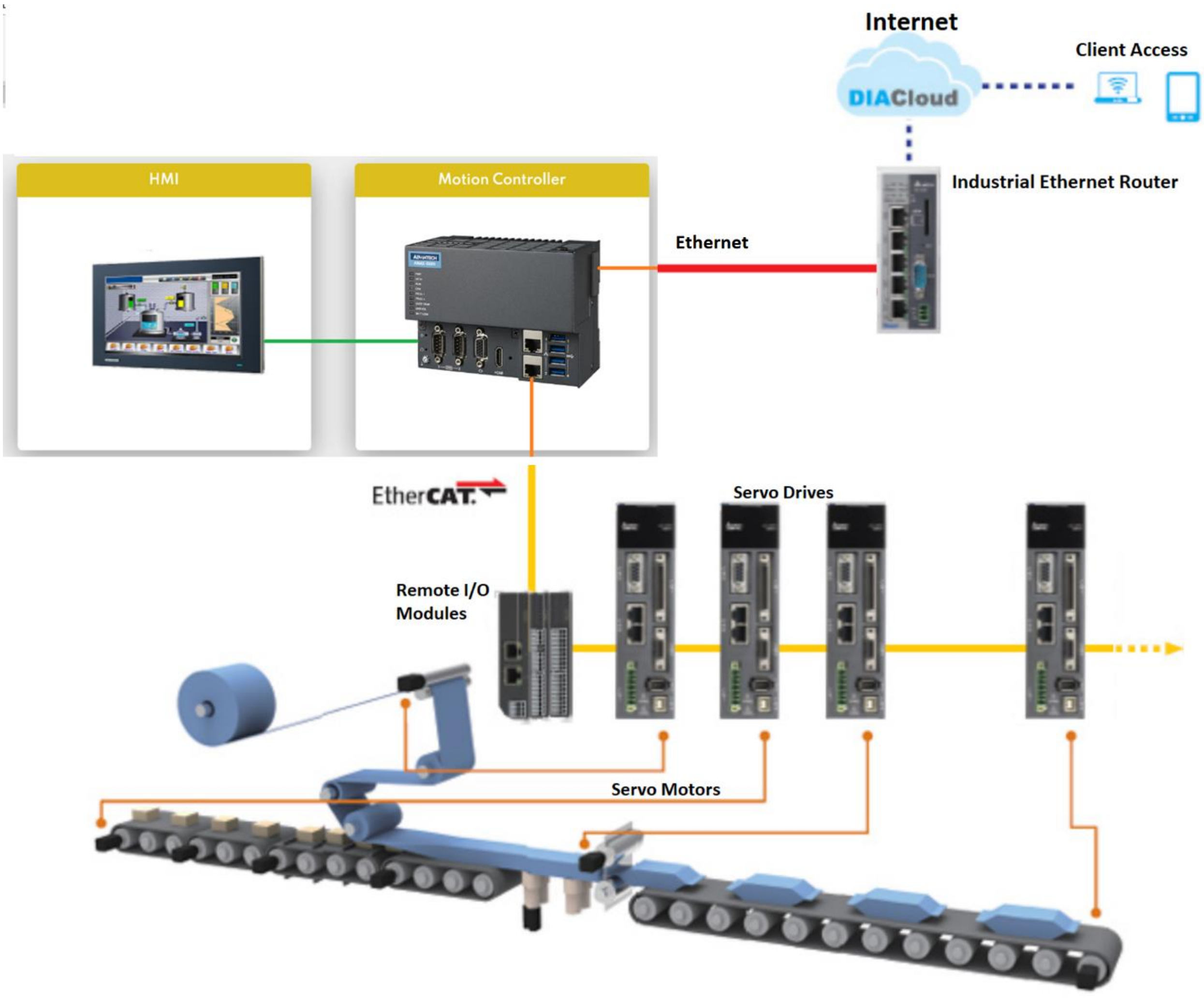


Fig. 2 - Packing machine application: web moves in electronically geared mode with the product on conveyor, including registration motion correction, to make a package around a product. The system involves four axis servo motion control: two servos (one for each) for two conveyors, one web motion (electronic gearing and tension) and one for sealing/cutting station. All are EtherCAT compatible servo drives and motors. Motion controller with EtherCAT Master role controls the servo motions, communicates with HMI and Internet thru Ethernet.

Fig. 3- Commercial printing machines have web as the master motion reference axis, and every station is electronically geared as Slave axis, some stations with registration sensor.

When the tool is in contact with the web, the linear speed of the tool and the web must be the same. When they are not in contact, they must move in a gear ratio dictated by the distance the two axes must move during the non-contact phase. Some printing heads print through the whole circumference of the print cylinder. In that case, the linear speeds must always be matched. The whole cycle is a contact phase and there is only one gear ratio.

For instance, in paper cutting machines, the flying knife can be geared to the master axis moving the paper (Fig. 4). The paper must be cut at certain length and at a certain offset distance from a registration mark. The paper may slip (or stretch due to temperature or tension variations) in the master axis drive. As a result, the master encoder reading of paper movement cannot always be perfect. To correct for such unavoidable problems, a registration mark is always sensed on the paper at a fixed distance from the cutting point. Regardless of the source of the error (i.e. printing error of the mark position or master encoder accuracy loss due to slippage), there has to be a fixed phase relationship between the cutting knife and the position of the registration mark on the web.

When the tool is not in contact with the web, the tool speed can be adjusted in order to let the proper amount of web distance pass. If the design was not a programmable motion control system and it was mechanically geared, a rotating knife machine could only cut one paper length only which is the length of the knife circumference. The ability to program the motion of the knife so that while it is in contact with the web it matches the linear speed for a correct straight cut, and speed up or slow down while it is not in contact with the web, provides us with the capability to cut many different lengths of web with one rotary knife. The same discussion applies to print heads and sealing head applications where the cycle length for which a new print or seal is to be made can be programmed.

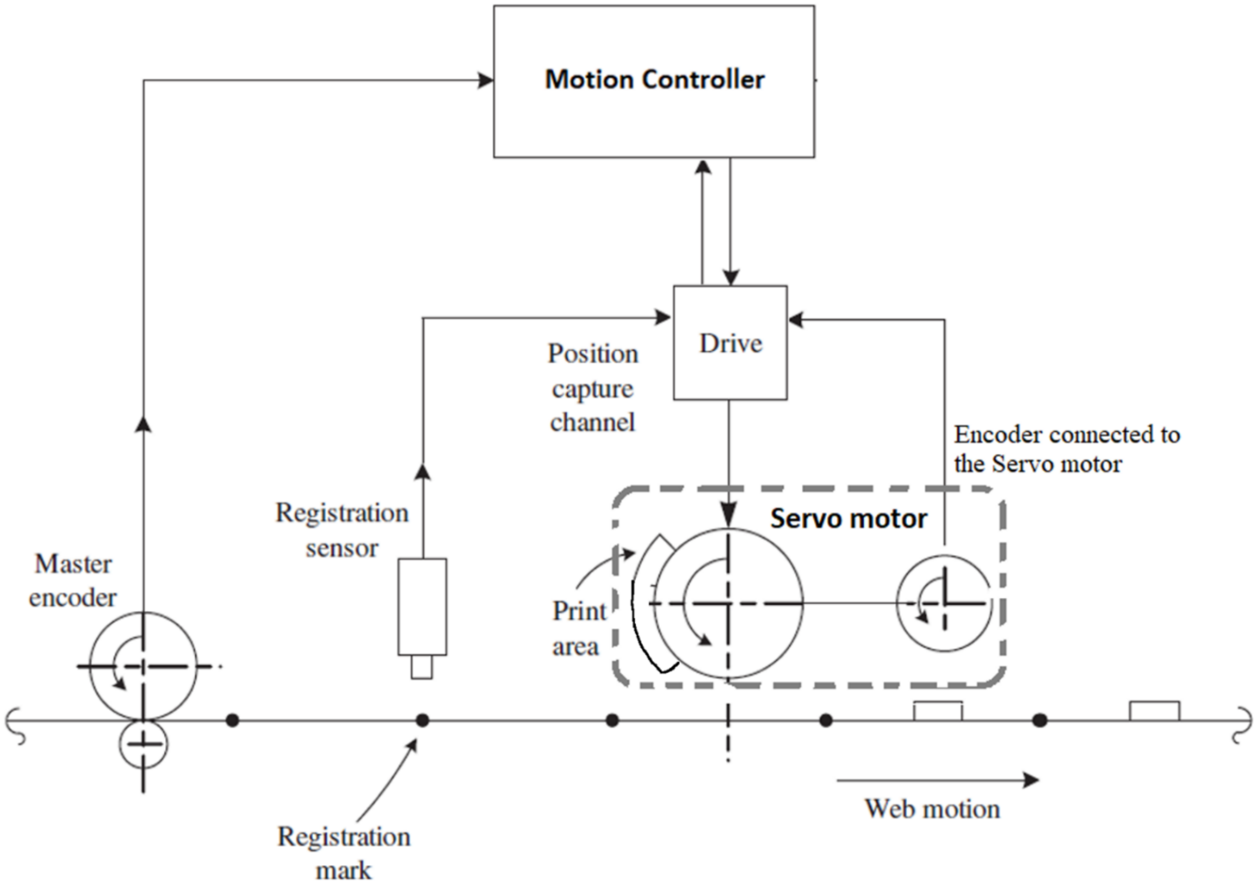
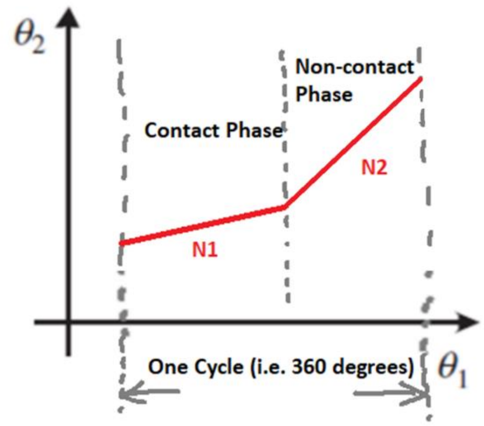


Fig. 4 – Printing machine motion with electronic gearing to the master axis (web motion, measured by “Master Encoder”) and registration mark sensor based motion correction.

There are only two phases to the coordination per cycle (Fig. 5 and Fig. 6):

1. Contact phase: This is a fixed gear ratio relationship as shown in the Axis 1 and Axis 2 position relationship in the figures below. When the web and tool are in contact, their linear speed must match.
2. Non-contact phase: tool and web are not in contact: gear ratio 2. When the web and tool are not in contact, the motion of the two axes can be independent yet during that phase each one must complete a certain distance to match the product length requirement. That leads to a second gear ratio in the motion coordination.



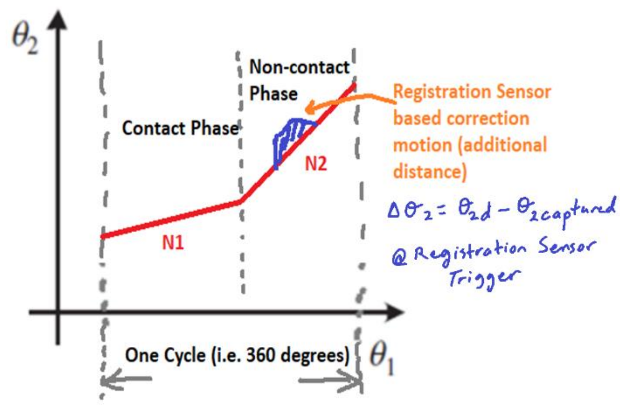


Fig. 5- Two gear ratios per cycle (on the left) and same with registration sensor based position correction motion (on the left).

3. A variation on this concept is **motion coordination using a "registration sensor" signal from an external sensor (also called *registration application*)**. In addition to implementing electronic gear ratio, "registration application" does the following (Fig. 7)

   a. on a "registration sensor" signal state change, position of the Slave axis is captured (typically at the servo drive and communicated to the motion controller). This position capture must be accomplished within a few microseconds of variation.
   b. the captured position is compared to the desired position (ideally where the Slave axis position should be when the "registration sensor" is ON), and
   c. corrective motion is commanded for the position error in addition to the electronic gearing command.

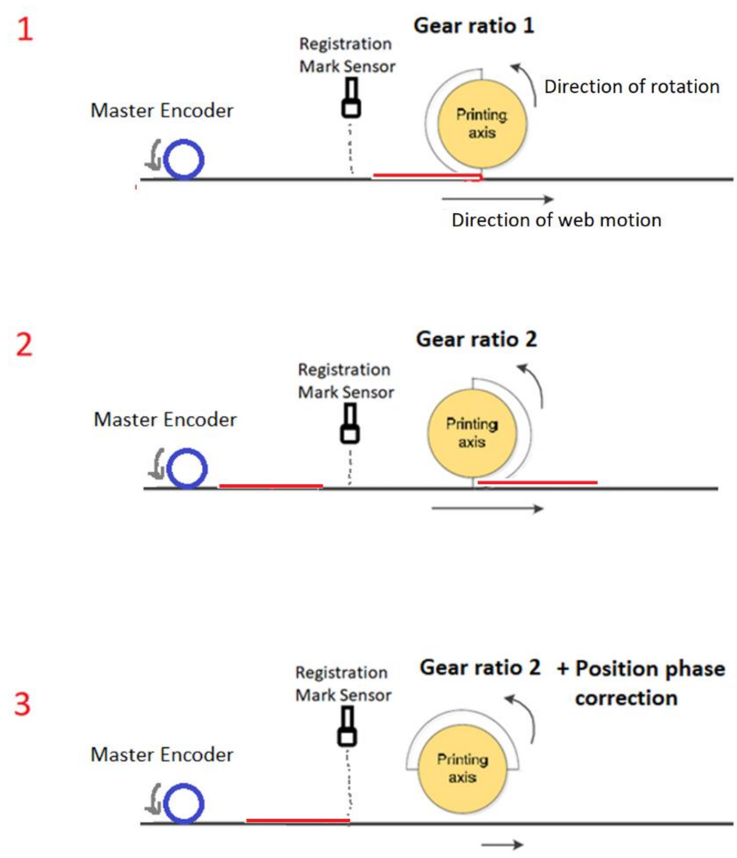


Fig. 6 – Printing with electronic gearing and registration.

There are many applications where per motion cycle of the machine, there is two different gear ratios. Clearly, there is no limitation in electronic gearing in that the gear ratio does not have to be a constant. It can be two or more different values. The gear ratio value to use at any given time can be decided based on a logic. In many automation applications, there are two gear ratios and which one to use when is decided based on the position of the master axis: in one cycle, there are two position ranges; during range 1 we use gear ratio 1 (N1) and during phase 2 (the rest of the cycle), we use gear ratio 2 (N2). In printing or cutting web applications, the phase 1 is the phase when print head is in contact with the web, and phase 2 is when the print-head is not in contact with the web (non-contact) phase. Based on monitoring the position of the master axis, the motion control logic selects the active gear ratio.

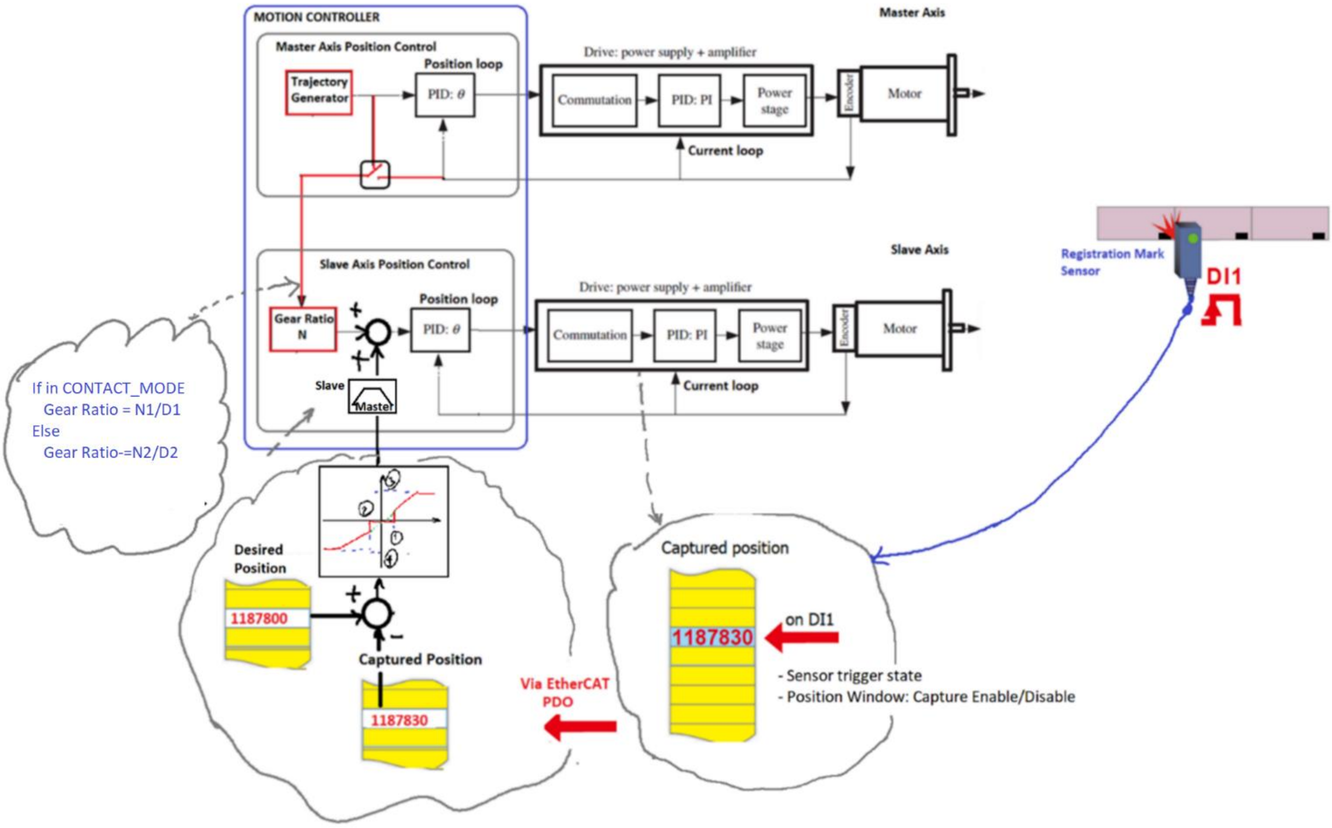


Fig. 7 – Two axis servo motion control system: one is Master axis and the other Slave axis. Slave axis commanded motion is generated by 1. Electronic gearing to Master axis, plus 2. Corrective index motion superimposed on the motion based on registration mark sensor data.

Before auto cycle begins, the two axes must be **"homed"** with correct relative position relationship to each other as shown below (Fig. 8).

Home: Master and Slave "homed" axes such that once the cycle begins, the print head and mark meet at the correct position

First, Home Master Axis to a defined Home position.
Then, Home Slave Axis with a corresponing Home

Homeing Parameters:

Master Axis; HomeOffset
Slave Axis; Home Offset

Desired Position Phasing Between Master and Slave Axes

Example Homing:

Master Axis: Home to Registration Mark Sensor
Slave Axis: a corresponding Home Offset

Example Homing:

Master Axis: Home to Registration Mark Sensor plus Home Offset
Slave Axis: a corresponding Home Offset

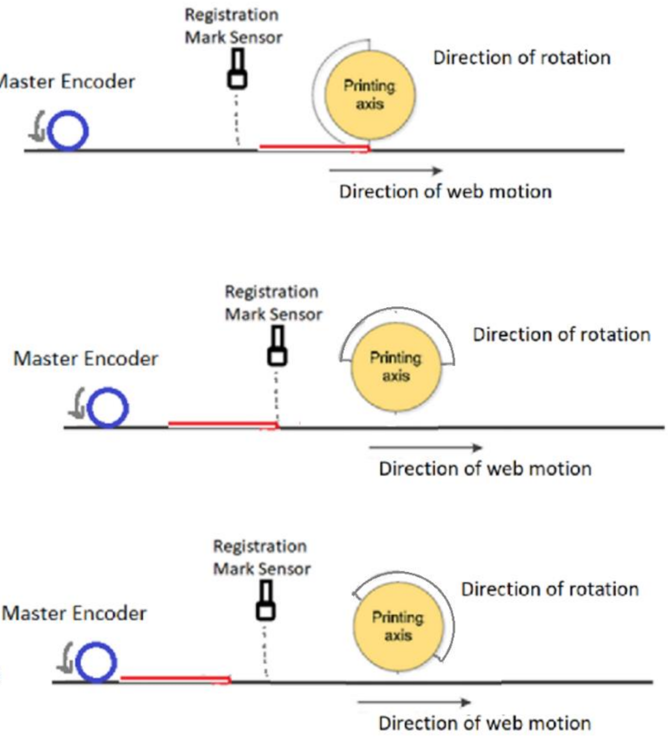


Fig. 8 – Homing of Slave axis relative to the Master axis and Registration Mark Sensor.

The web axis motion is always the master since it is difficult to make the web do whatever we want it to do. Rather, we need to follow and track the web with the tool axis. In contact phase, the tool axis gear ratio needs to be adjusted so that the linear speed between the web and tool are equal. If the gear ratio is programmed in linear motion units, it will be 1 to 1. During the non-contact phase, the gear ratio will be different than 1 to 1 linear ratio. The gear ratio will be such that the remaining length of web will pass during the time that the tool moves over the non-contact phase. If the web cycle length happens to be equal to the circumference of the rotary tool, then the gear ratio will be 1 : 1 during non-contact phase in linear distance units as well. More often, a cut must be made relative to a registration mark. A print must be made repeatedly but at a certain offset distance from a registration mark every cycle. Similar needs exists for sealing type applications. In order words, the tool will cut/print/seal the web every cycle where the web length per cycle is programmable, but it must be done relative to a registration mark.

From an electronic gearing motion coordination point of view, the registration mark requirement adds one more motion coordination problem. That is, the tool must be in correct positional phase relative to the mark in order to cut it at the right offset location. This motion is done as a superimposed incremental index motion on top of the current electronic gearing motion.

The Slave Servo Motor is electronically geared (with gear ratio #1 or #2) to the Master Encoder (or Master Servo Axis). In addition, it makes phase correction to its position based on a registration sensor, where the difference between the captured Slave servo position and desired Slave servo position is used as the position phase correction.

The registration mark sensor should be placed as close as possible to the Slave Servo Motor axis. The longer the distance between them the higher the chance of positioning error. If the sensor is too far from the Slave Servo Motor, by the time sensed mark position passes thru the Slave Motor drive mechanism, there could be position errors introduced that goes unnoticed. We want to measure the positioning error and correct that as soon as possible after the measurement.

The position of the print head must be captured very accurately when the registration sensor triggers ON. The registration sensor must have good repeatability and fast response in order to have a good positioning accuracy for the print head. The “position capture” on “registration sensor trigger” is done using special input line for high-speed trigger input to the Slave Servo Drive to capture the position in hardware. This is referred as “capture position input” in microcontroller terminology, or “registration sensor input” or “touch probe input” in servo drive terminology.

Typical response time delay from trigger signal to captured position (“latched position”) should be about 5 microseconds.

Let us consider an application where web speed is 3000 ft/min and required print/cut positioning accuracy is 3/1,000 in.

The web line is moving at 3000 ft/min, which is 3000 * 12 / 60 inches/sec, which is 3000 * (12/60)/ 1,000,000 microsecond.
Then the variation in distance in 1 microseconds is 0.6/1000 in.
Then 3/1,000 inch web movement takes 5 microseconds.
Hence, if there is 5 microsecond variation in the position capture sensor and microcontroller operation, there will be 3/1,000 inch measurement error.

Notice that the position capture can not be done with a general purpose digital input and software interrupt; in such a case 1 millisecond variation in the response is typical. In 1 millisecond, there can be 0.6 inches (600/1,000 inches) of web position variation in measurement accuracy which is unacceptable.

Notice that the correction of the position error does not (and cannot) need to be made within 5 microsecond, rather the position correction need to be made within the non-contact phase remaining in the cycle which is typically in the order of tens of milliseconds.

The acceleration and deceleration rate of the slave axis can be defined as a function of the master axis position change or as a function of time. However, it is much better design to define them as function of master axis position change so that the slave axis speeds up/slows down as the master axis speeds up/slows down.

The accuracy of the electronic gear tracking is as good as the tracking accuracy of each slave axis whatever method of control is used. Therefore, for the electronic gearing to be successful, the servo loops should be properly tuned. Otherwise, the tracking errors may deteriorate the performance to the point that it does not resemble the performance that would be achieved had the shafts been connected to each other by mechanical gears.

Remark: In some applications, the registration mark sensor-based position correction is accomplished by modifying the gear ratio instead of super-imposed incremental index motion. In this case,

- the product linear pitch length is measured by calculating the difference between two last position captures,
- that calculated value is optionally passed thru an averaging filter,
- then the output of the filter is used to calculate the new gear ratio: product linear distance measured ratio to the Slave Axis position movement.

Controller Inputs:

Controller Outputs:

Control Algorithm Parameters:

Control Algorithm Logic:

Parameters:

Master Axis
Slave Axis

Master_Contact_Range
Master_Non_Contact_Range
Slave_Contact_Range
Slave_Non_Contact_Range

Master Axis: Contact Phase Start Position
Master Axis: Contact Phase End Position

RegistrationEnable
Desired Position at Capture: PosCapD
Err_mmax, Err_mmin, Err_pmin, Err_pmax
Master Distance to Complete Registration Move

BlindWindow Enable
BlindWindow Begin Master Position
BlindWindow End Master Position

Master axis position: in contact or non-contact mode

```
Get Master_Axis_Pos :  X_pos

if ((X_pos >= Contact_Phase_Start)  AND (X_pos <  Contact_Phase_End  ))  AND (Current_Mode != Contact_Mode)
    Current_Mode = Contact_Mode
    MOVE Slave_Axis  FOR  Slave_Contact_Range  IN   Master_Contact_Range

elseif  (NOT ((X_pos >= Contact_Phase_Start)  AND (X_pos <    Contact_Phase_End  )) )  AND (Current_Mode != Non_Contact_Mode))
    Current_Mode = Non_Contact_Mode
    MOVE Slave_Axis FOR Slave_Non_Contact_Range IN Master_Non_Contact_Range

endif
```

Registration Motion: Position correction based on registration sensor position capture

Executed if Registration is Enabled, once on each "Position Captured" on Registration Sensor Signal

```
If Blind_Window_Enabled
    Check if  Position Capture occured in Blind_Window
        If TRUE  return  ; // ignore it

Get captured position:  PosCapA
err= PosCapD- PosCapA

if err < err_m_max
    err= err_m_max
elseif  ((err > err_m_min) AND (err < err_p_min ) )
    err = 0
elseif err > err_p_max
    err = err_p_max

if (err != 0)
  MOVE Slave FOR err IN Master_Distance
```

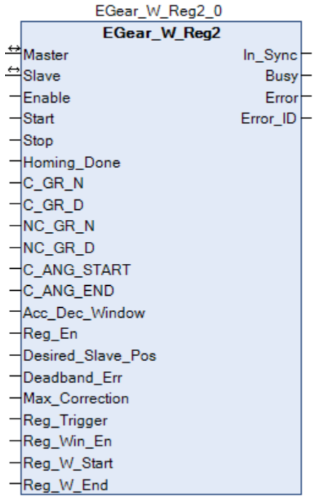
Motion Coordination Algorithm: Rotating Tool (Cutting Knife, Print Head, Seal Head) with Registration Mark Application:

```
Main Program
{

  Initialize the rotating knife with registration mark application algorithm

  Verify HOME motion sequence is done

  Check Task 1

  Check Task 2

   .....

 while (TRUE)
 {
    Check Print Phase and Registration Mark
   {
      When Enterered Contact Phase, execute
            Contact_Phase Function once
      When Entered Noncontact Phase, execute
            Non_Contact_Phase Function once
      When Registration mark is received, excecute
            Registration_Mark_Phase_Adjustment Function once
    }
  }

Return
}
```

CODESYS Application Software in PLC/Motion Controller using Function Blocks:

Function Block: EGear_W_Reg2

Electronic Gearing with Registration Function Block – Two Gear Ratio Per Cycle



LOGIC:
This function block implements electronic gearing with registration; it has two different gear ratios per cycle; two phases of motion per cycle is defined as a position based window; registration capture valid window is defined with a position based window (referred as "Blind Window"); actual commanded position correction based on registration error has deadband and saturation limit.

VAR IN_OUT:
Master:SM3_Basic.AXIS_REF_SM3;
Slave:SM3_Basic.AXIS_REF_SM3;

INPUTS:
Enable: BOOL :Rising edge Enables execution of the block.
Start: BOOL; Rising edge Starts the operation
Stop: BOOL; Rising edge stops the operation and motion of Knife axis.
Homing_Done: BOOL; Connect Output Done of Homing. block
C_GR_N: DINT; Contact Gear ratio numerator
C_GR_D: UDINT; Contact Gear ratio denominator
NC_GR_N: DINT; Non-Contact Gear ratio numerator
NC_GR_D: UDINT; Non-Contact Gear ratio denominator
C_ANG_START: The start of contact motion marked as angle on knife axis.
C_ANG_END: The end of contact motion marked as angle on knife axis.

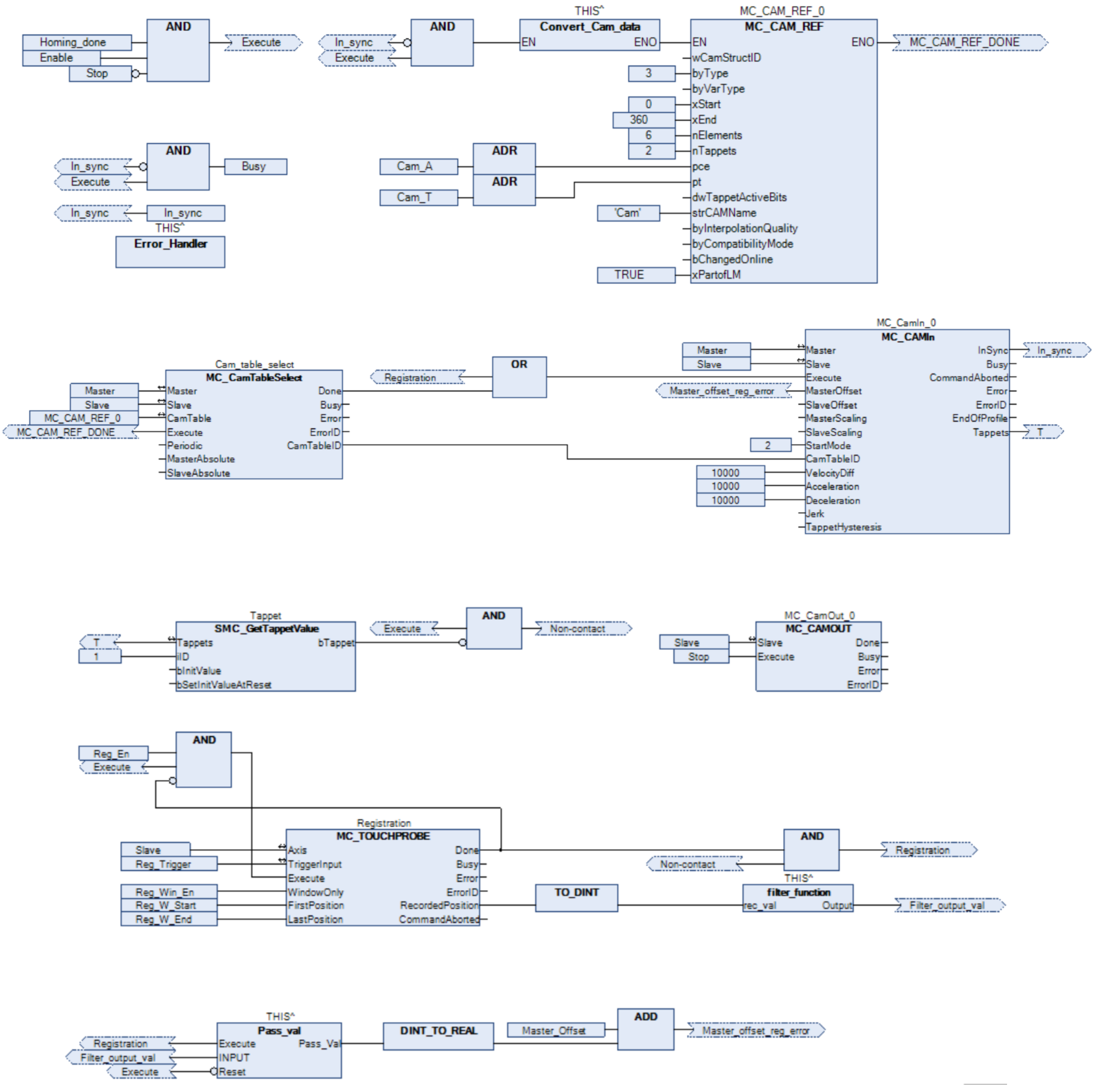
Reg_En: BOOL; Enable Registration and registration correction
Desired_Slave_Position: DINT; The expected slave position at registration
Deadband_Err: DINT; permissible error in degrees of knife axis rotation where no correction action is taken
Max_Correction: DINT; Max correction beyond which error values are ignored
Reg_Trigger: SM3_Basic.TRIGGER_REF; Specify registration probe trigger mapping
Reg_Win_En: BOOL; Enable registration sensor only in a short window of the cycle

Reg_W_Start: LREAL; Define start of the window in which registration sensor is enabled
Reg_W_End: LREAL; Define End of the window in which registration sensor is enabled

OUTPUTS:

In_Sync: BOOL; True if two axes are in synchronized motion as per the set gear ratios.
Busy: BOOL; Performing initial routines not in full operational mode.
Error: BOOL; Sets true if any error occurred.
ErrorID: STRING; Displays the error ID with error Description. Refer the link below for documentation.
https://help.codesys.com/webapp/ESyhuZmlO2jU2jrW7hYhBPXo2bc%2FSMC_ERROR;product=SM3_Error;version=4.7.0.0
application specific ErrorID 1090 refers to: NC_GR or C_GR incorrect.
C_GR*(percentage_cutting_part_of_Master_cycle/100) + NC_GR*(1-(percentage_cutting_part_of Master cycle/100)) should be equal to 1.

CFC Logic for the Function Block:



Electronic gearing with Registration Description:

CAM:

Convert_Cam_Data is a method of Parent function block created using structured text. It takes values of Electronic gearing ratio calculation values and converts them to Arrays: Cam_A: This array contains information regarding points on the Cam curve.
Cam_T: This array contains information regarding points for tappet switching.

Cam Tappet uses a servo drive internal latching to capture time at which servo crosses a particular position 10 times faster than normal PLC program loop.

The OR logic with Registration and Master Offset is explained in the Registration part.

MC_CAM_REF: Creates an object of Cam using the Cam_A and Cam_T. The other inputs are not relevant

MC_CamTableSelect: Creates Cam Table Id using Cam_Ref object

MC_CAMIN function block performs actual Camming operation between two axes.

Start Mode 2 to provide Ramp_in with Vel,Acc and Dec values provided to the block

The OR logic with Registration and Master_Offset_Reg_error is explained in the Registration part.

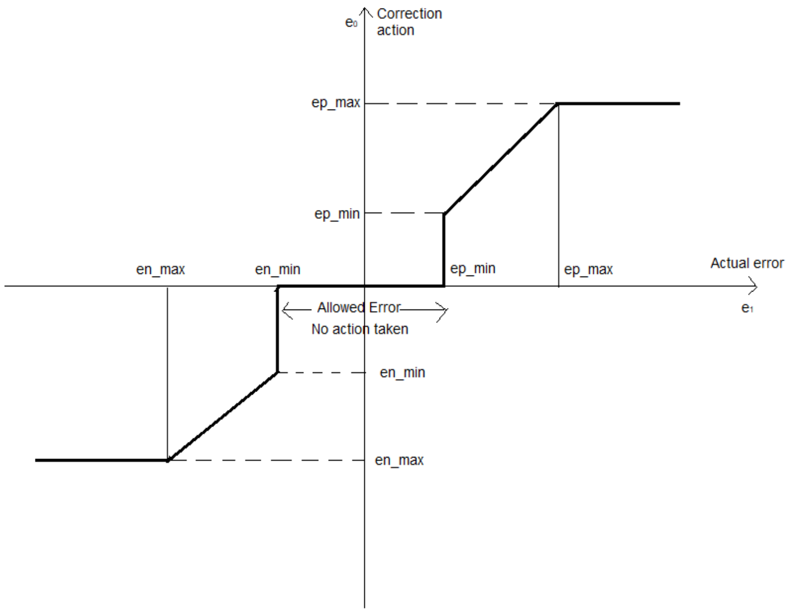
SMC_GetTappet_Value is used to set a bit for contact and non contact part based on values generated in Cam_T so that registration correction doesn't happen while Contact motion is taking place.

MC_CamOUT: Essential for Error free disengaging of synchronized motion. Set’s the Slave axis in continuous motion even after CAM_OUT, additional Stop block required to Stop the Slave Axis.

MC_Touch_probe block is used for registration. It records Slave position at the rising edge and has inbuilt inputs for detecting input only in a certain window which is taken input in the function block.

The recorded position is then compared with the desired position passed through the filter function.

Deadband and Saturation on Registration Based Position Correction Logic:




```
METHOD Filter_function
VAR_INPUT
    rec_val: DINT;
END_VAR
VAR
    err_val:DINT;
END_VAR
VAR_OUTPUT
    output:DINT;
END_VAR
```

```
IF rec_val>180 THEN
    rec_val:=-(360-rec_val);
END_IF
err_val:=Desired_Slave_Pos-rec_val;
IF err_val < -Max_Correction THEN
    output:=-Max_Correction;
ELSIF err_val<-Deadband_Err THEN
    output:=err_val;
ELSIF err_val<Deadband_Err THEN
    output:=0;
ELSIF err_val<Max_Correction THEN
    output:=err_val;
ELSE
    output:=Max_Correction;
END_IF
```

Pass_Val Method: Pass_Val method created in structured text increments

the Passed value with the value or registration correction only on the rising edge of Registration. When Execute is False the accumulated registration correction value is set to zero.

The additional Master_Offset variable is implemented in cases where the Contact part does not have a center at 180 degree. That is added to the Passed value from Pass_Val Method.

This Added value is Fed to Master_Offset input variable of MC_Cam_IN function block but the new Master_Offset value is only implemented on rising edge of Execute Block and that is why the Execute input of MC_CamIn block has Registration in OR logic.

Note: TheEN or Execute bits of Convert_Cam_Data Method, MC_CAM_REF, MC_Cam_Table_Select, MC_CamIN blocks get set to true when MC_CamIN.InSync bit is true This is to save computational overhead in calculation on each iteration.

Whenever Execute has a rising edge all the values are calculated again, Cam_Table is created again and MC_CamIn is executed.

References:

Cetinkunt, S., Mechatronics with Experiments, John Wiley and Sons, 2012, Second Edition, pp. 717-748.

Conveyor tracking by Yaskawa: https://www.youtube.com/watch?v=cWa8evbwSkI