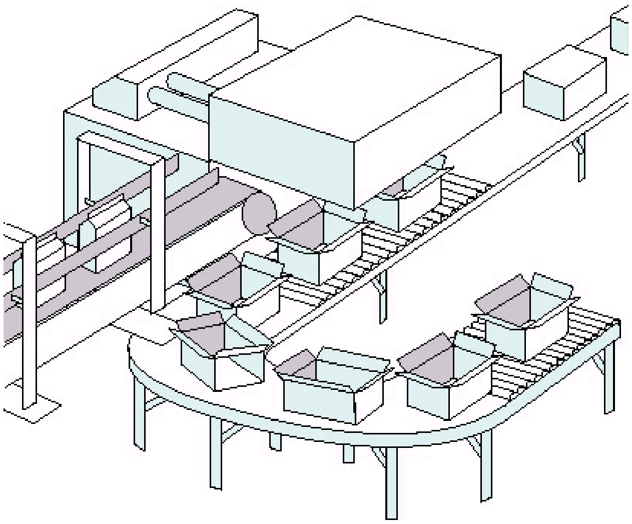
Designing a Smart Belt System that Coordinates Delivery of Product from a Conveyer to a Packaging Machine.



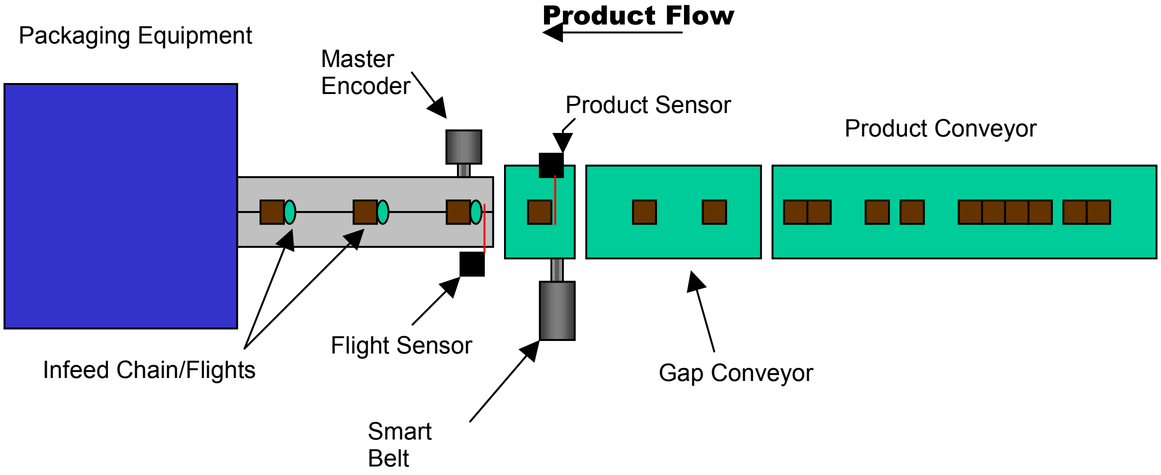
Purpose of this Document: This application note describes a method used to detect the relationship between a moving machine and an incoming product. The position of the product will be corrected via a smart belt, as necessary. This application note assumes that product enters the machine at the most desirable position and speed. The logic displayed in this application note was developed using the RSLogix 5000 development system designed for use with a ControlLogix hardware platform.

Application Description: In packaging applications, products are often transported in random fashion from a filling or wrapping operation to a final cartoning machine on a conveyor belt. Many packaging machines require these products to be positioned to a moving reference point (flights, overhead sweeps, etc.). A "smart servo belt" can provide uniform spacing to match the moving reference point. The servo-controlled belt follows the main machine drive (master), so that the motion is relative to the machine. The servo controller monitors the position of the flights. When a product is detected, a correction distance is calculated and the correction move is performed. The correction move is relative to the machine motion and is added to the speed required to follow the master. This ensures the product will be positioned correctly relative to the flight or other moving target.

Control System Description:
- ControlLogix controller
- Servo Motors
- RSLogix 5000 Programming Software

Control Challenges – As a machine designer, the main challenge is to develop a robust and predictable control system that allows maximum production with the widest variety of products For applications like this, and others involving registration, the details of programming the application often become the greatest challenge. This application note provides an example of how to capture and process registration positions in terms of a smart belt type application.

Control Solution: The following diagram shows the major components involved in a simple smart belt application.



Design Considerations

In addition to the normal considerations designers must take when designing any conveyor and servo system, they must also consider the dynamics of making positional corrections in a smart belt application.

Sensor Placement & Wiring

The sensor used to detect *products* should be placed so that products are detected only after the majority of product weight is on the correction belt. This will ensure that products do not slip.

The sensor used to detect *flights* should be placed so the flight being detected is the same flight to which current product will be positioned. This minimizes the effects of mechanical differences in flight spacing.

Mechanics

The servo motor and its connection to the belt must be chosen carefully to minimize reflected inertia to the motor. While gear reduction may be used to reduce reflected inertia, make sure motor top speed allows maximum correction belt speed. The system should be tuned to provide maximum response and stability for all ranges of product weights.

Smart Belt Length and Product Slippage

Nominal machine speed, maximum potential correction distance, and the frictional relationship of the product and the conveyor material will all have an impact on the design and number of smart belt sections needed for a particular application.

If the coefficient of friction between the product and the conveyor belt is known and predictable, you can calculate the maximum acceleration and deceleration a correction move can make without the product slipping backwards on acceleration or being thrown forward when decelerating.

The maximum correction distance is usually, ½ the product spacing of the packaging equipment. Using the maximum correction distance and the maximum acceleration and deceleration, you can calculate the minimum time needed to make the maximum correction.

Using the above minimum correction time, and the nominal speed of the machine, you can calculate the distance a product will travel at nominal speed making the maximum correction, at the maximum accel/decel rates in the direction of product. This distance will be the minimum length of the correction belt. Most applications will need a larger belt to accommodate lower accel/decel rates, lower top end speed, and distance to allow product to completely transfer to/from the smart belt, etc.

Gap Conveyor

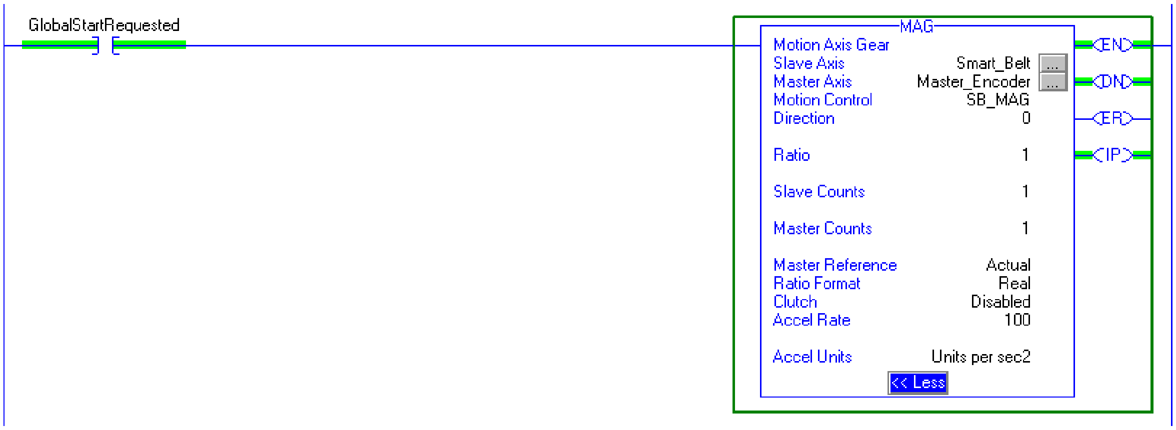
The gap conveyor can eliminate issues surrounding negative corrections. The purpose of the gap conveyor is to pull a gap between successive products such that a product can make a negative corrective move without contacting the next product. Without a gapping mechanism, the application may be limited to positive correction moves and will affect the smart belt length as the maximum correction limit will be the distance between flights.

Sample Application Programming

There are 2 main functional objectives that the smart belt must provide in this application:

1. Convey the product from the gap conveyor to the packaging machine.
2. Position the conveyed product between the packaging equipment infeed flights.

The first objective requires the smart belt to be geared to the master encoder such that as the packaging equipment moves 1 flight. The smart belt will move the distance between successive flights. The actual ratio used in the MAG command will depend on how the programmer chooses to scale each axis. For instance, if the master axis is scaled in inches, and the smart belt is scaled in inches, the ratio will be 1:1. The following rung of logic accomplishes the application's first objective.



The second objective (positioning the conveyed product between the packaging equipment infeed flights) is what separates this solution from a conventional conveyor application. In order to position the conveyed product between the packaging equipment infeed flights, the controller requires several pieces of information.

- Product position – Captured with product sensor
- Flight position – Referenced with flight sensor. Captured with product position
- Desired product position – Operator entered value. Represents the captured product position that requires no correction.
- Flight distance – Physical distance between flights
- Maximum acceleration – See *Design Considerations* above.
- Maximum deceleration – See *Design Considerations* above.
- Maximum correction speed – See *Design Considerations* above.

Process

As a product crosses the product sensor, the position of the master encoder is captured. This position represents the position of the product relative to the packaging machine infeed flight (i.e., product position with respect to flight position). The product position is then subtracted from the desired product position to determine the product error. The product error represents the correction distance the product must be moved to place the product in the relative position represented by the desired product position. If the product position equals the desired product position, no correction takes place.

Capturing the Data

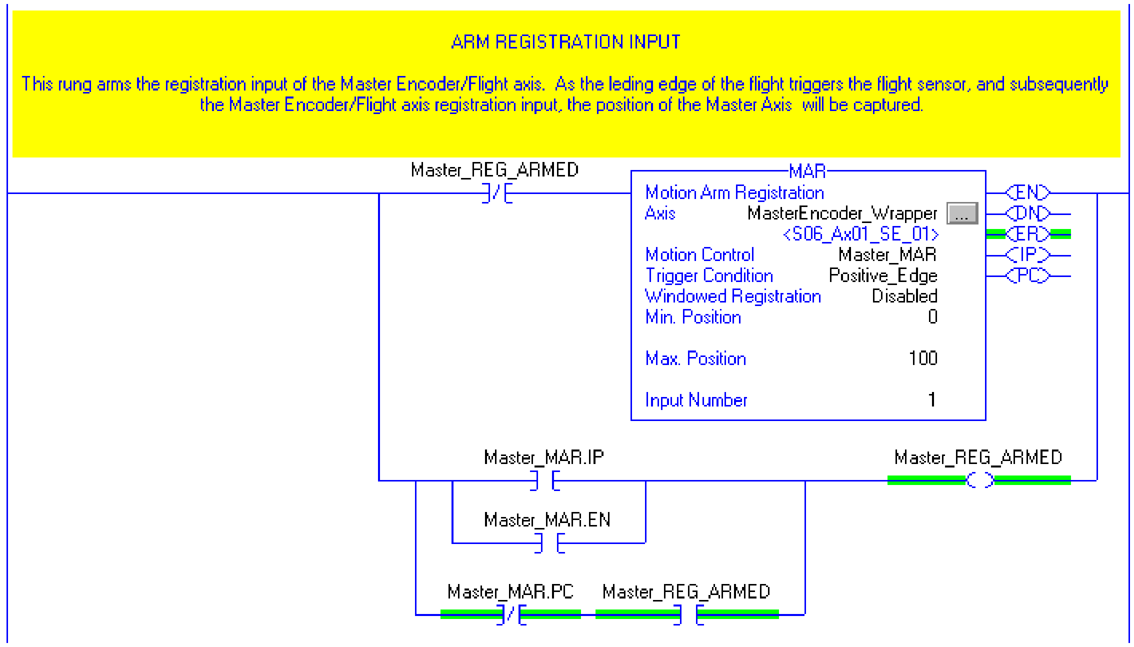
The product position and flight position are the two critical pieces of data that need to be captured as the machine runs.

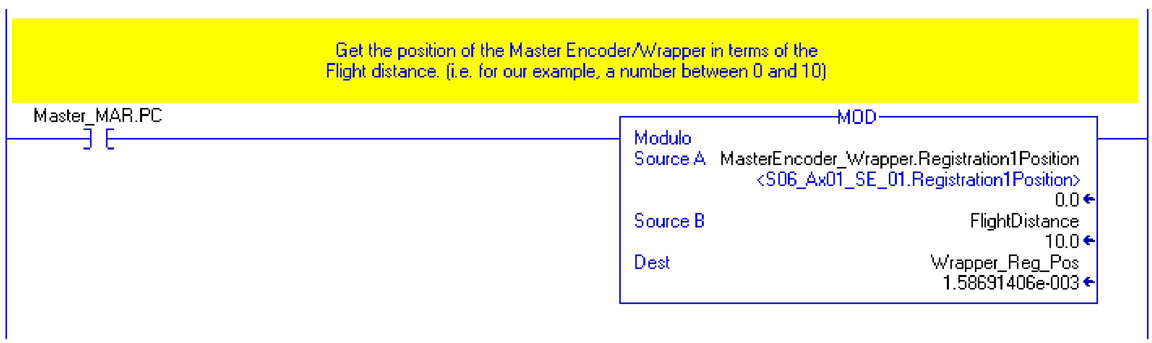
For the flight position, the program should be set up such that the master encoder is in rotary mode with a conversion constant that gives the position units of the flights in inches. The unwind constant should be set such that the position of the master encoder will rollover to zero as the distance between each flight passes. Ideally, the mechanics would allow an integer number of encoder counts for the rollover value. However, in a majority of the applications, circumstances happen that require the rollover value to be rounded to the nearest integer. In this case, to prevent accumulation error, the application needs the master encoder position to be synchronized to the actual flight position at regular intervals.

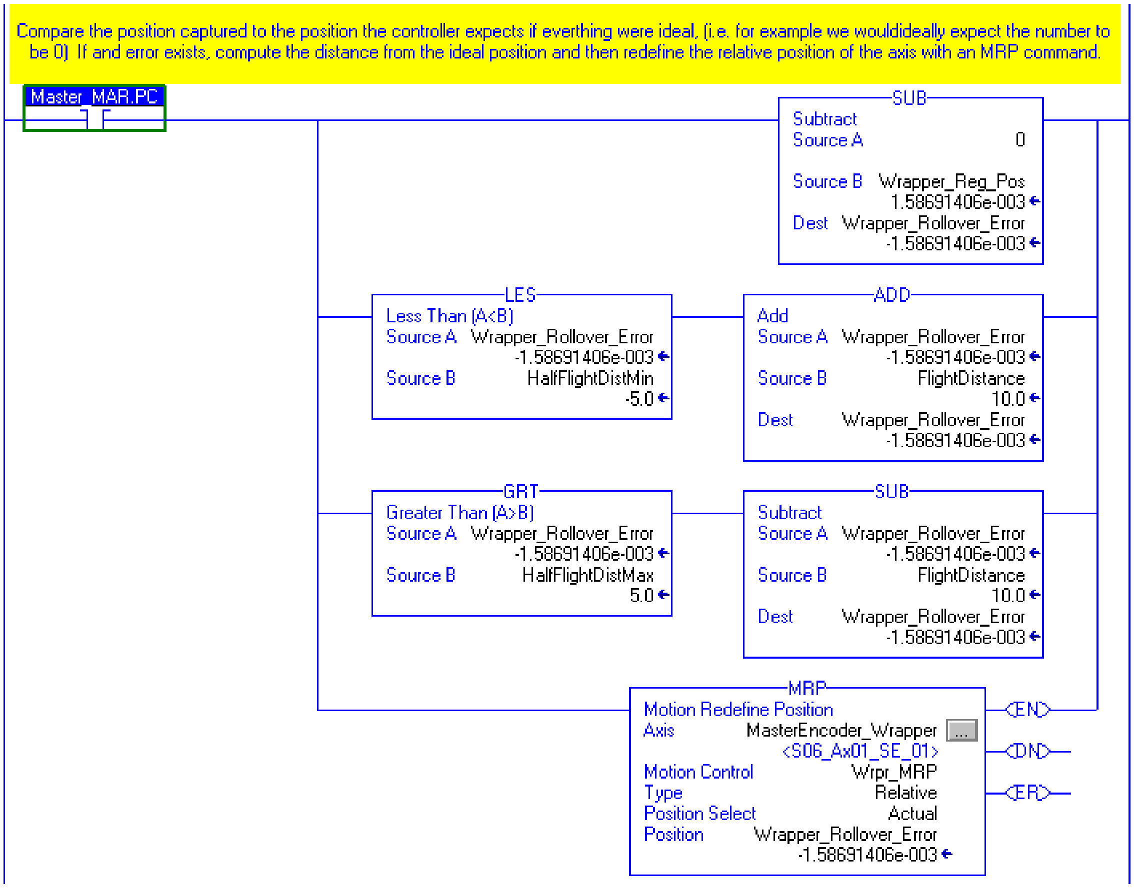
For the purposes of this application example, the sensors is connected to the registration inputs of their respective axes. Furthermore, the logic to follow demonstrates the ability to do a “soft registration event.” A soft registration event is the ability to capture the position of an axis other than the axis associated with the registration sensor.

The flight sensor serves the purpose of referencing a position to the movement of the flights. If the mechanics of the system were ideal, this referencing would only be necessary when the system is first powered up or immediately following an encoder fault. However, this paper assumes that the system is not ideal and therefore the unwind constant of the master encoder cannot be represented by an integer number of encoder counts. Using this assumption, the control needs to monitor the position of the flights and correct any accumulated position error due to the non-integer unwind constant. The monitoring procedure begins when a flight passes the flight sensor and triggers a registration event. The control then retrieves the registration position captured during the registration event. For purposes of this application, the registration position will be compared to zero. If the registration position does not equal zero, it can indicate that the controller position is accumulating error with respect to the actual position of the flight and a redefinition of the controller position will be done.

The following logic keeps the master encoder position referenced to the actual position of the flights:







By keeping the flights referenced to the position of the flight sensor, the control essentially knows where the flights are located at all times. Keep in mind that the objective of the application is to position the product with respect to the flights. Therefore, since the control knows where the flights are located, the missing data at this point is the position of the product *with respect to the flight*. The product sensor captures this information each time a product passes it. The logic is similar to that which references the flights, but will differ because:

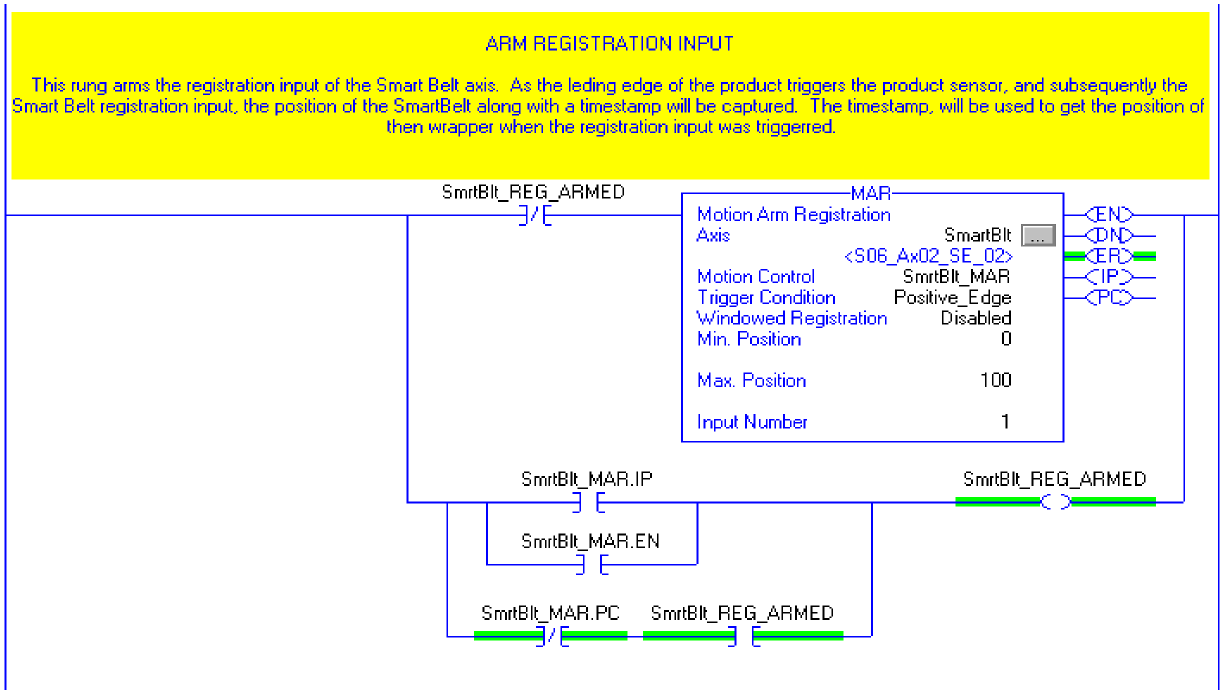
1. The position the control needs to capture is the flight position.
2. The product sensor is connected to the smart belt's registration input

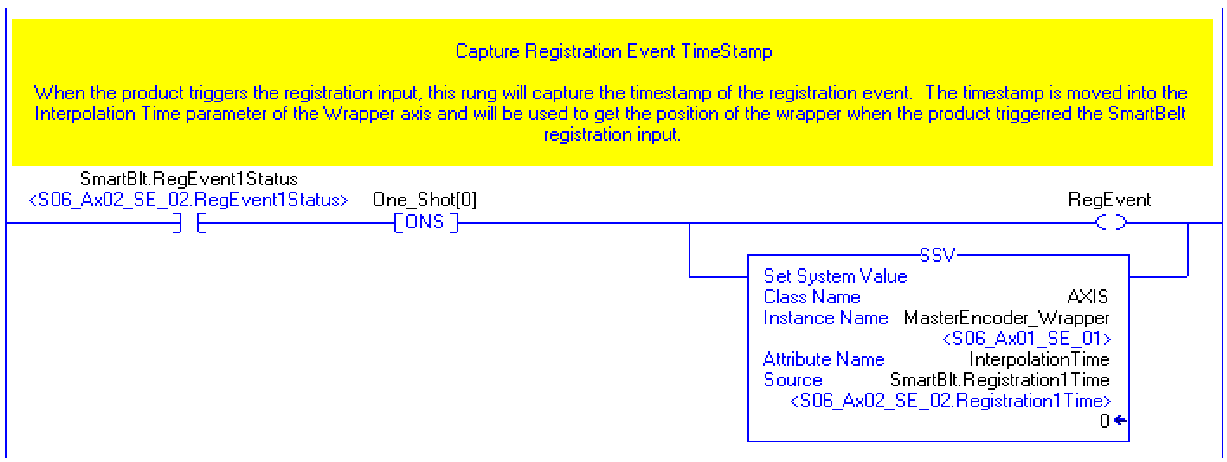
Why does the product trigger the capture of the Flight Position?

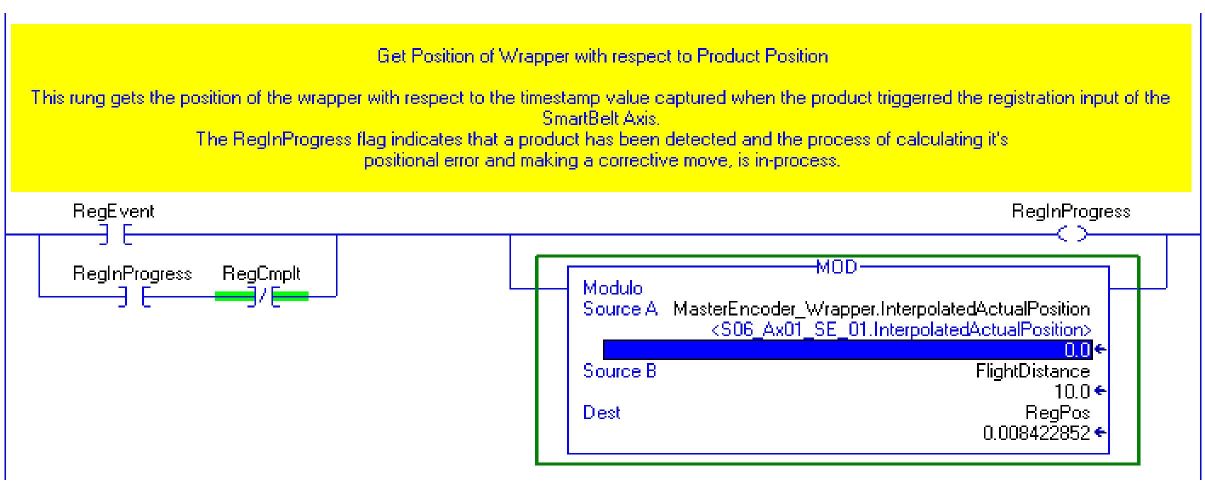
The control needs to capture the flight position because it needs the position of the product *with respect to the flight*.

This is the information where most programmers stumble. Intuition tells them that since there is a servo on the smart belt, the position of the smart belt must be important. To the contrary, the position of the smart belt axis is arbitrary and as random as the products being fed to it. However, the precision of the servo will allow the smart belt axis to make accurate incremental positional corrections to the product once the control can determine where the product is *with respect to the flight*, and where it needs to be.

The following logic is used to capture the "soft registration" position of the flights (ProductPositionPV) using the registration input of the smart belt axis.







Processing the Data

As stated above, the product error represents the corrective move distance required to position the product correctly. However, before the value is used for the corrective move, it must be tested to assure that the polarity and magnitude of the corrective distance are between +/- ½ the distance between flights.

Consider the following data:

Actual distance between Flights = 10 inches

Product Position = 9

Desired Product Position = 1

Using the previously mentioned formula:

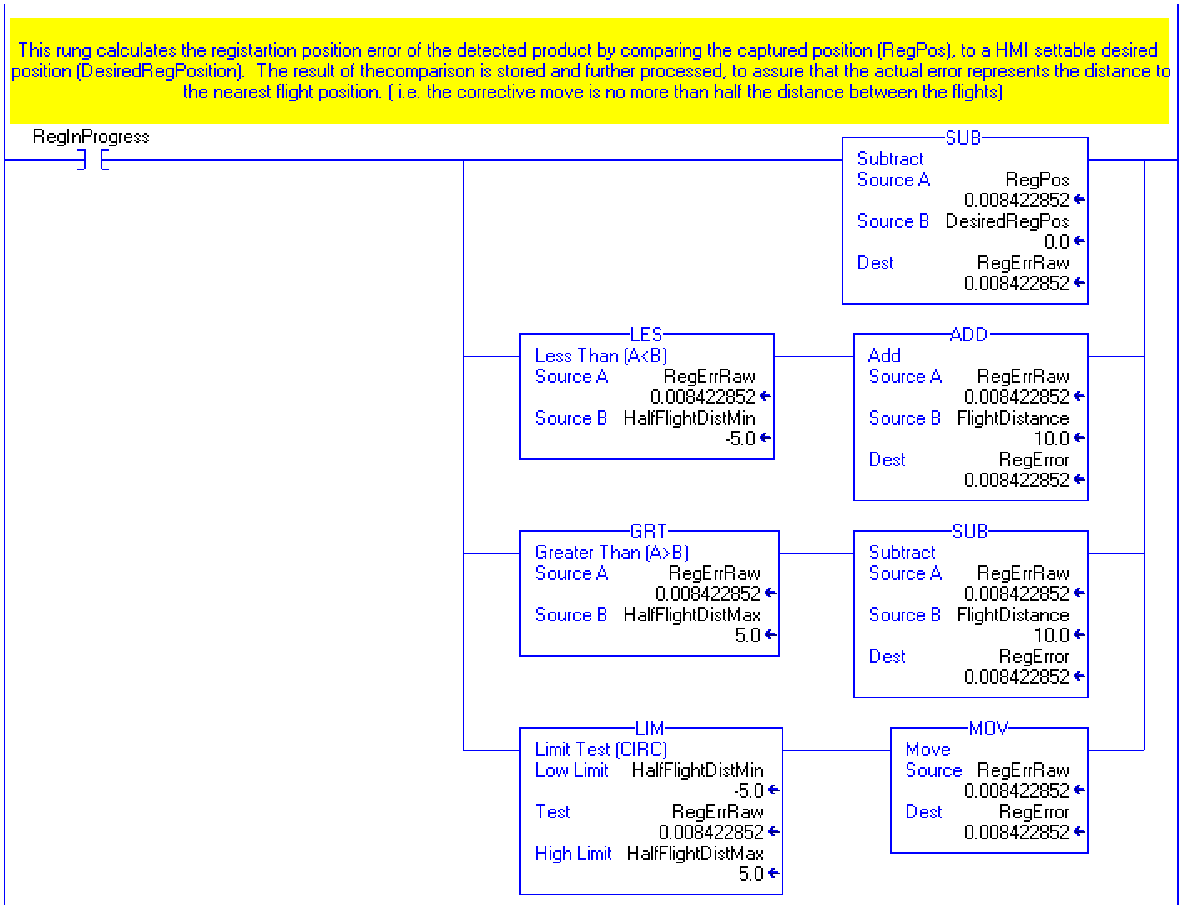
Product Error = Desired Product Position – Product Position

= 1 - 9

Product Error = **- 8 inches**

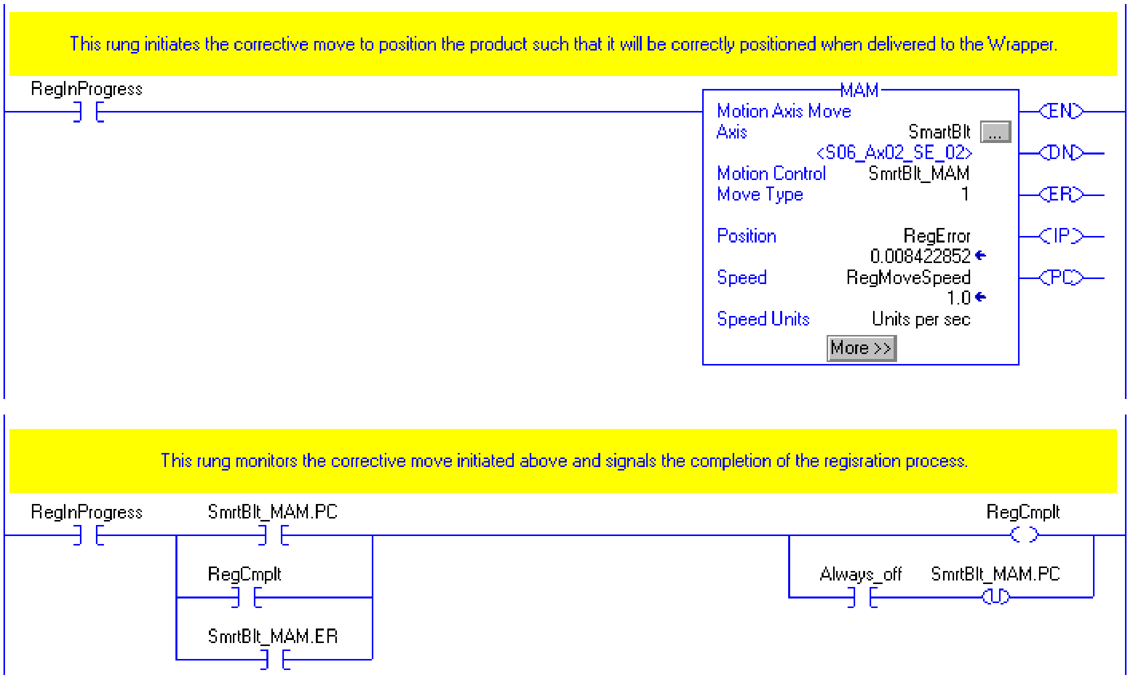
The above math implies that the system needs to correct the part a distance of 8 inches for the part to be positioned correctly. However, the math is misleading and not complete.

Keep in mind that a new flight approaches every 10 inches. Therefore, if we are sensing the front edge of the product, that edge could at most be out of position by 5 inches (i.e., ½ the flight distance). In the above example, the product error needs to be further processed because the magnitude of the correction is more than half the distance between the flights. Study the following rung to see the additional checks and math that may be needed to determine the actual correction distance.



Corrective Move

The following rungs execute the corrective move and monitor the move for completion.



The speed of the corrective move should be considered since it's value can cause undesirable affects. For instance, if the speed is set to operate at the maximum value the application can tolerate, and the speed of the machine, when a corrective move takes place, is lower than the corrective move speed, and the corrective move is negative, the corrective move can momentarily reverse the direction of the smart belt. You must consider if this condition can be tolerated.

An alternative method of setting the corrective move speed would be to set it at the average velocity of the wrapper or some fraction thereof. In this case, a negative correction will slow the product and at worse, stop it momentarily.

From the preceding examples, you must consider the effects of all negative corrections with respect to the successive products. If negative corrections can result in the collision of subsequent products, multiple smart belts might be required to stagger negative corrections and/or the gap conveyor might need to be controlled so as to compliment and more fully control the positioning of the products.

Summary

In conclusion, caution must be used in applying the logic and concepts discussed. Rockwell Automation is confident that the logic displayed can be applied to a wide range of smart belt applications; however, safety and application specific demands can require significant accessory logic and/or modification for any specific application. Please contact your local Rockwell Automation distributor for further assistance with your specific application needs

www.rockwellautomation.com

Corporate Headquarters
Rockwell Automation, 777 East Wisconsin Avenue, Suite 1400, Milwaukee, WI, 53202-5302 USA, Tel: (1) 414.212.5200, Fax: (1) 414.212.5201

Headquarters for Allen-Bradley Products, Rockwell Software Products and Global Manufacturing Solutions
Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation SA/NV, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Headquarters for Dodge and Reliance Electric Products
Americas: Rockwell Automation, 6040 Ponders Court, Greenville, SC 29615-4617 USA, Tel: (1) 864.297.4800, Fax: (1) 864.281.2433
Europe/Middle East/Africa: Rockwell Automation, Brühlstraße 22, D-74834 Elztal-Dallau, Germany, Tel: (49) 6261 9410, Fax: (49) 6261 17741
Asia Pacific: Rockwell Automation, 55 Newton Road, #11-01/02 Revenue House, Singapore 307987, Tel: (65) 6356-9077, Fax: (65) 6356-9011

Publication RA-AP006A-EN-P – August, 2003