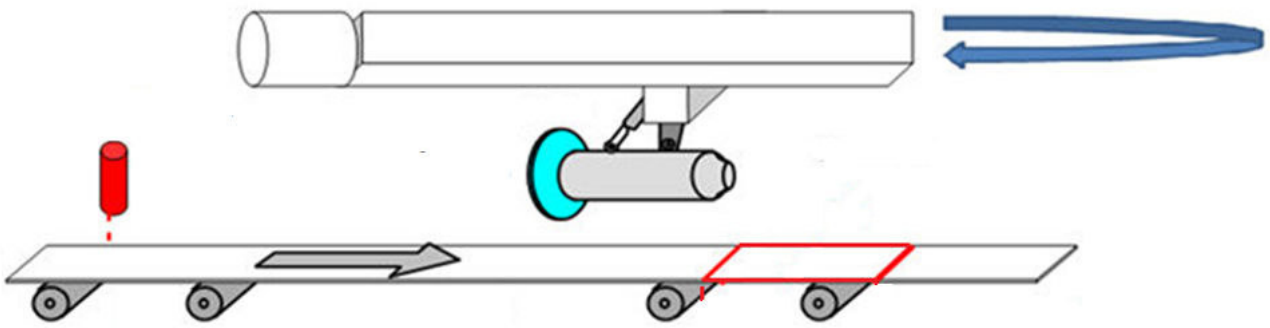
Technical Manual

Linear Flying Shear Application Solution Package for MPiec Controllers



Subject: Technical Manual	Product: MPiec Controllers	Doc#: TM.MPIEC.02
Title: Linear Flying Shear Application Solution Package for MPiec Controllers		

Contents

Subject: Technical Manual	Product: MPiec Controllers	Doc#: TM.MPIEC.02
Title: Linear Flying Shear Application Solution Package for MPiec Controllers		

Application Overview:

The Linear Flying Shear Application Solution Package (ASP) offers an easy way to implement linear flying shear applications. This ASP applies to other applications such as bottle filling because the motion required is identical. This solution package was designed for machines that process products spaced at fixed intervals and machines that must make dynamic corrections based on product registration marks. The Linear Flying Shear ASP project can be used as a template on which the user can complete the application, or the ASP can be added as a user library into an existing project. The Linear Flying Shear ASP focuses only on the core motion features required to operate a flying shear and does not include other functionality such as servo enable, homing, manual modes, alarm handling, etc. The user is responsible for incorporating other basic logic functionality required for the complete application.

The main steps involved in completing a linear flying shear application are:

1) Configure the axes in Hardware Configuration.
2) Import all required user libraries.
3) Create cam profiles. (Two choices)

   a) Use the 'LinearShearCamGen' function block in the ASP (for the speed matching axis). Create a custom cam profile for a secondary axis if present.

   OR

   b) Create custom cam profiles for the speed matching axis and secondary axis.
4) Home the speed matching axis and secondary axis. Set the CamMasterCycle for all slave axes.
5) Use the 'LinearFlyingShear' function block with or without the Registration input.

Application Highlights:

1. Designed for Linear Flying Shear applications with or without product registration.
2. Designed to accommodate a secondary slave axis in addition to the speed matching axis.
3. Uses smooth cam motion to reduce jerk.

Implementation:

1. Configuring the Master and Slave Axes

Configure the master and slave axes in the Hardware Configuration. Configure the master and slave 'load type' as linear.

2. Adding User Libraries

If the Linear Flying Shear ASP is being used as a template, no additional user libraries need to be added.

If the Linear Flying Shear ASP is applied as a user library:

1. Delete all Yaskawa Data Type files from the main project.
2. Add the Y_Motion firmware library to the main project.

Add the following toolboxes to the main project in the order in which they appear. The minimum versions are given in Figure 1 below:

1. DataTypes_Toolbox_v200
2. Math_Toolbox_v202
3. Yaskawa_Toolbox_v204
4. PLCopen_Toolbox_v206
5. Cam_Toolbox_v205
6. Add the Linear Flying Shear ASP project as a User library in the main project.

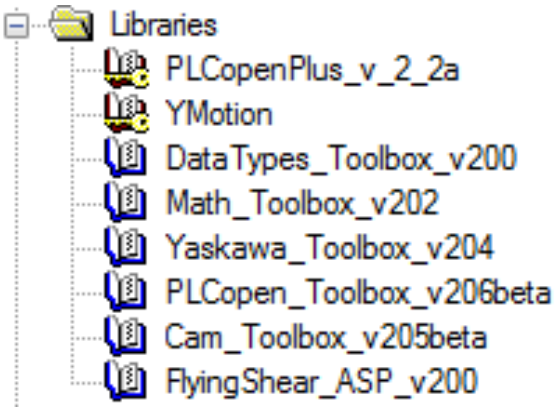


Figure 1: User libraries added to Linear Flying Shear project

3. Generating Cam Profiles

The user has two options for creating cam tables:

1) Use the LinearShearCamGen function block for the speed matching axis. The function block creates cam profiles with Tangent Matching segments. If a secondary slave axis exists, create a custom profile using Y_CamFileSelect, CamGenerator, and Y_CamStructSelect.
2) Create custom cam profiles for the speed matching axis and the secondary axis using Y_CamFileSelect, CamGenerator, and Y_CamStructSelect. Custom profiles may be necessary for applications with unique mechanisms where the user desires to create unique cam profiles for the application.

3 a) Generating Cam Profiles using LinearShearCamGen

Add LinearShearCamGen to a POU running in a slow / low priority task (1000 ms task update).

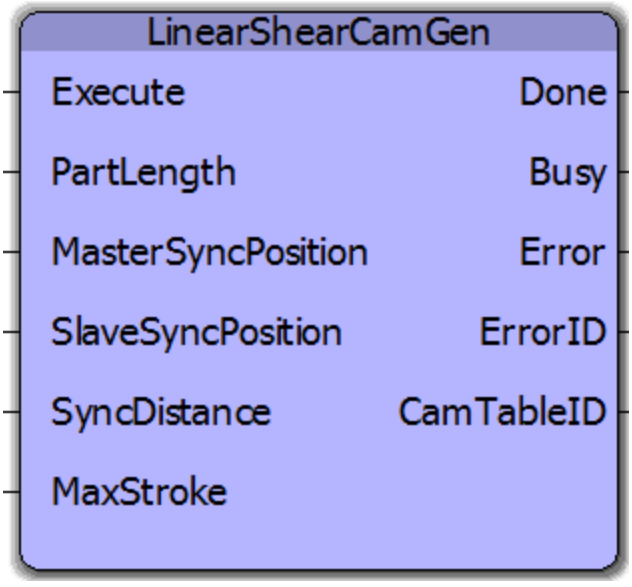


Figure 2: LinearShearCamGen function block

Parameter		Data type	Description	Default
VAR_INPUT				
U*	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the Execute input.	FALSE
U	PartLength	LREAL	Length of the part to be processed. If randomly spaced parts are to be processed, enter a nominal distance between registration marks.	LREAL#0.0
U	MasterSyncPosition	LREAL	The position of the master in the cam table where both axes must be moving at the same speed.	LREAL#0.0
U	SlaveSyncPosition	LREAL	The position of the slave axis in the cam table where both axes must be moving at the same speed.	LREAL#0.0
U	SyncDistance	LREAL	The distance travelled by the slave axis while in sync with the master. Note that the total travel distance will be greater due to the acceleration and deceleration phase.	LREAL#0.0
U	MaxStroke	LREAL	The maximum allowable stroke length for the linear slave axis. If the calculated stroke required is greater than MaxStroke, an Error will be generated.	LREAL#0.0
VAR_OUTPUT				
F*	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
F	Busy	BOOL	Set high upon the rising edge of the 'Execute' input, and reset if Done, or Error is true.	
F	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' goes low	
F	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' goes low	
F	CamTableID	UINT	CamTable ID for the speed matching slave axis	

* U: Input to be provided by user

* F: Parameter generated by the function block

A cam profile that illustrates various inputs for the LinearShearCamGen function block is shown in figure 3.

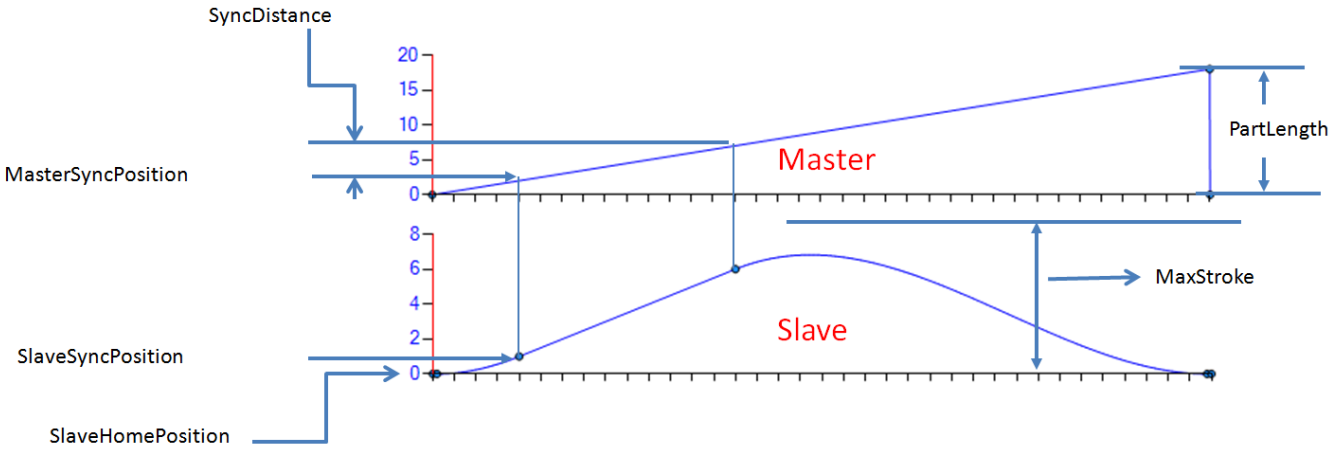


Figure 3: Sample cam profile for a linear shear application

If the application uses a secondary axis for punching/slitting/cutting while the linear axis is synchronized with the master, and a cam is required for the secondary operation, use the following function blocks to create a cam profile yourself.

1) Y_CamFileSelect (If the cam will be loaded in the controller memory as a .csv file)
2) CamGenerator (If the cam profile will be created in the IEC application program.)
3) Y_CamStructSelect (If the cam profile was created in the IEC application program.)

ErrorID	Meaning
10190	PartLength must be greater than zero.
10191	The SyncDistance input must be greater than zero.
10192	The MasterSyncPosition must be greater than zero.
10193	The PartLength must be greater than the SyncDistance.
10194	The calculated stroke length is greater than the allowable 'MaxStroke'. Increase MaxStroke or reduce the SyncDistance.
	All other ErrorIDs are listed under help for CamGenerator (Cam_Toolbox), CamTableManager (Cam_Toolbox), Y_CamStructSelect (PLCopenPlus_v_2_2a). (Refer to the Toolbox manual at: **www.yaskawa.com/iectb**)

3 b) Generating custom cam profiles

If the application requires custom profiles and the LinearShearCamGen function block is not used because of mechanism constraints, use the following function blocks:

1) Y_CamFileSelect
2) CamGenerator
3) Y_CamStructSelect

If the application uses a secondary slave axis for punching/slitting/cutting while the primary axis is synchronized with the master, and a cam profile is required for the secondary servo as well, use the function blocks mentioned above. Cam table IDs for the primary speed matching axis and the secondary axis (if servo is being used as a cam slave) must be obtained for using the LinearFlyingShear function block discussed in section 5. A sample profile for the speed matching shear axis is shown in figure 4a. A sample profile for the secondary axis is shown in figure 4b.

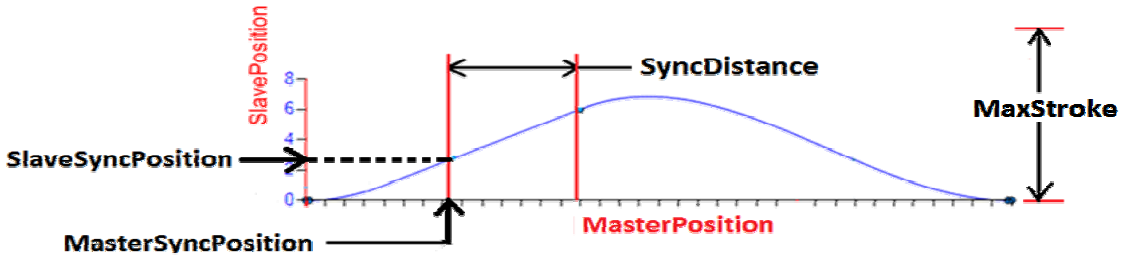


Figure 4a: Sample cam profile for the speed matching axis in linear flying shear applications

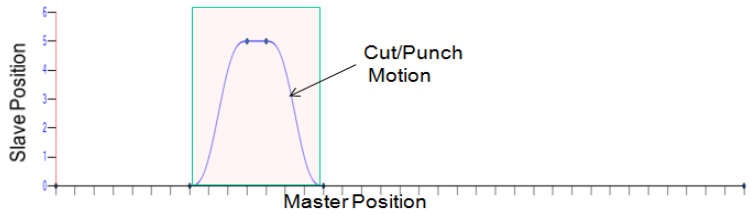


Figure 4b: Sample cam profile for secondary axis in linear flying shear applications

4. Setting the CamMasterCycle

Linear flying shear applications that require product registration must make adjustments to the cam cycle to synchronize with the first part before motion starts. To initialize the CamMasterCycle, execute the SetCamMasterCycle function block after the desired running CamTableIDs have been generated and the servos are enabled. If the machine includes a secondary slave axis, set the cam master cycle for that axis using another instance of the SetCamMasterCycle function block.

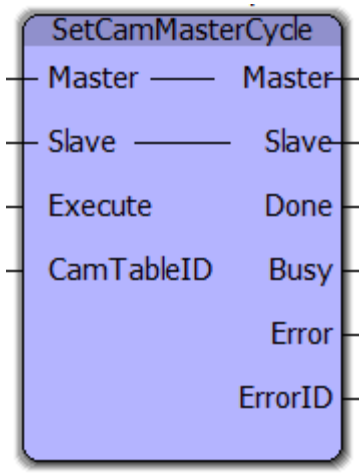


Figure 5: Setting the CamMasterCycle

Parameter		Data type	Description	Default
VAR_IN_OUT				
B	Master	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number)	
B	Slave	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number)	
VAR_INPUT				
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the Execute input	FALSE
B	CamTableID	UINT	A reference to the cam memory in the motion engine	UINT#0
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when Execute goes low.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' input, and reset if Done, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' goes low	
E	ErrorID	UINT	If Error is true, this output provides the Error ID. This output is reset when 'Execute' goes low	

5. Linear Flying Shear Motion

Two types of linear shear applications are supported:

a) Constant product size without registration.

b) Variable product size with registration.

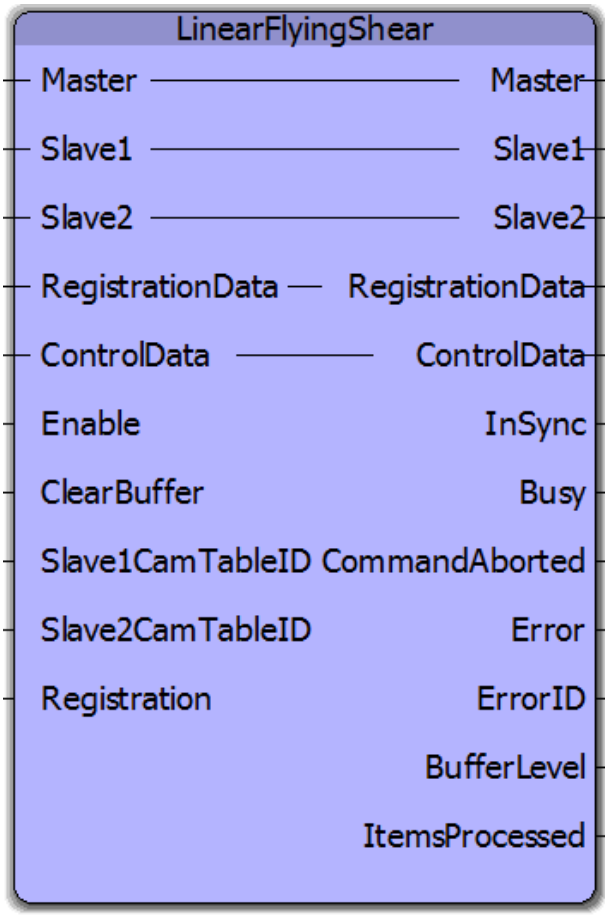


Figure 6: LinearFlyingShear function block

Parameter		Data type	Description	Default
VAR_IN_OUT				
U*	Master	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number)	
U	Slave1	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number). Refers to the speed matching linear axis in the linear shear system.	
U	Slave2	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number). Refers to the secondary axis in the linear shear system. If a secondary servo is not used, use UINT#0 for this AXIS_REF.	
U	RegistrationData	ProductBufferStruct	Structure containing all information for the circular buffer to operate. The user inputs in this structure will be used in managing the product buffer that keeps track of registration marks.	
U	ControlData	CamSyncStruct	Structure containing all information that enables the LinearFlyingShear function to correct the slave axis motion for varying product sizes and to determine when the slave axis should disengage from the master if no products are being registered	
VAR_INPUT				
U	Enable	BOOL	The function will continue to execute while Enable is held high.	FALSE
U	ClearBuffer	BOOL	While high, new products are not added to the product buffer. The products that are already in the buffer will get processed by the function block.	FALSE
U	Slave1CamTableID	UINT	Cam Table ID corresponding to the speed matching linear axis.	UINT#0
U	Slave2CamTableID	UINT	If being used in the system, Cam Table ID corresponding to the secondary servo.	UINT#0
U	Registration	BOOL	Selects if registration based corrections are required. If TRUE, adjustments will be performed to synchronize with the registration marks on the master axis.	FALSE
VAR_OUTPUT				
F*	InSync	BOOL	Indicates that the outputs of the function block are valid and the salves are following the master. It does not indicate if the slave are matching speed with the master.	
F	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
F	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	
F	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' goes low.	
F	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' goes low.	
F	BufferLevel	INT	Number of products detected by the sensor which have not been processed by the machine.	
F	ItemsProcessed	UDINT	Provides a count of the number of products processed since this function was enabled.	

* U: Input to be provided by user. * F: Parameter generated by the function block

ErrorID	Meaning
	All ErrorIDs are listed under help for CamShiftControl (Cam_Toolbox), CamControl (Cam_Toolbox), Y_CamIn (PLCopenPlus_v_2_2a), Y_CamOut (PLCopenPlus_v_2_2a), ProductBuffer (PLCopen_Toolbox) (Refer to the Toolbox manual at: www.yaskawa.com/iectb)

5 a) Constant cut length without registration

Use the LinearFlyingShear function block with the 'Registration' input set to FALSE. If only one slave is used, connect UINT#0 to Slave2 and Slave2CamTableID. Connect variables for RegistrationData and ControlData. These variables must be declared but need not have valid data. (These structures are not used for applications that do not use registration).

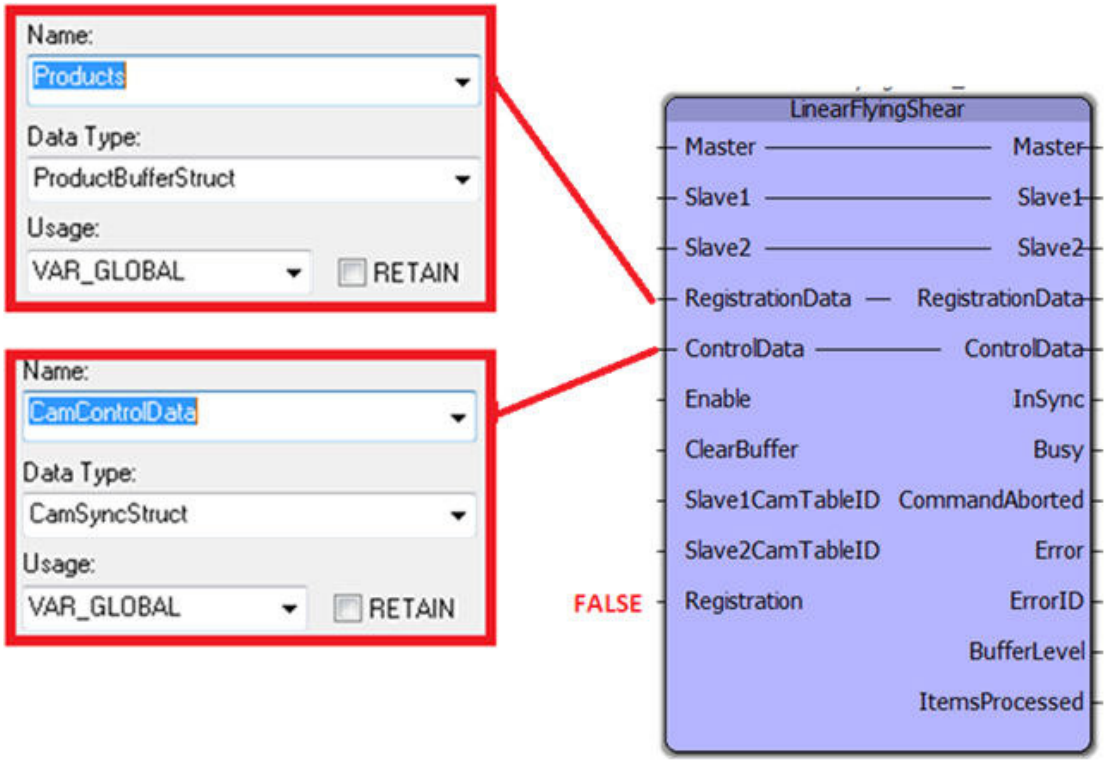


Figure 7: Linear Flying Shear function block

5 b) Applications requiring product registration

Use the LinearFlyingShear function block with the 'Registration' input set to TRUE.

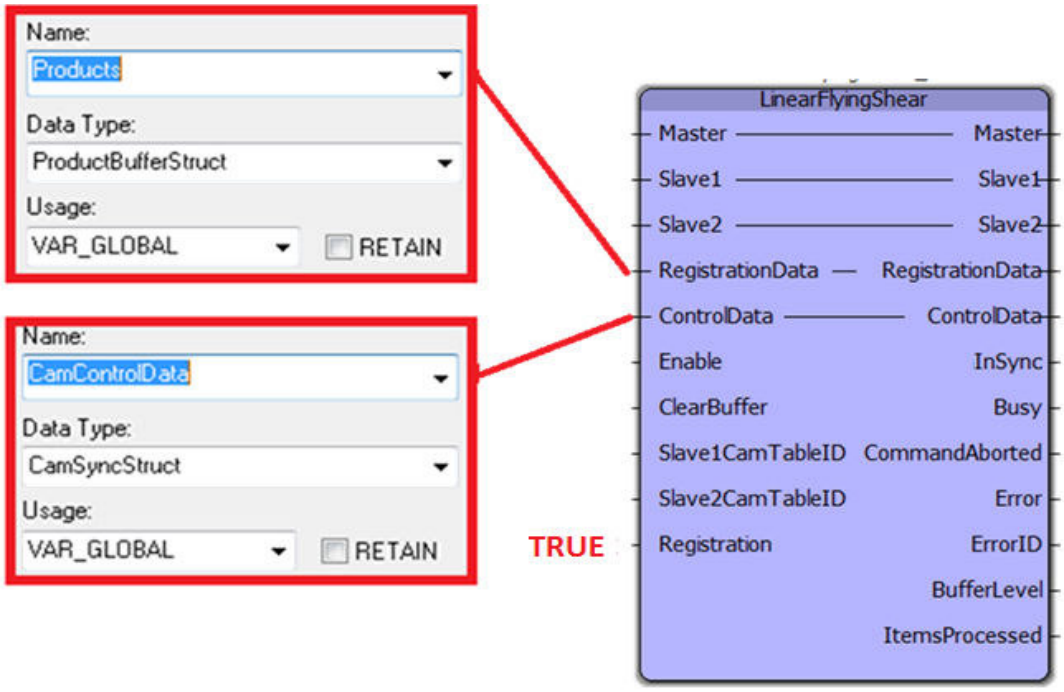


Figure 8: LinearFlyingShear with registration

1. Define a variable for RegistrationData (datatype: ProductBufferStruct) and initialize the following elements of the structure:

- BufferSize
- LockoutDistance
- SensorDistance
- ProductAwayDistance
- Sensor.Bit

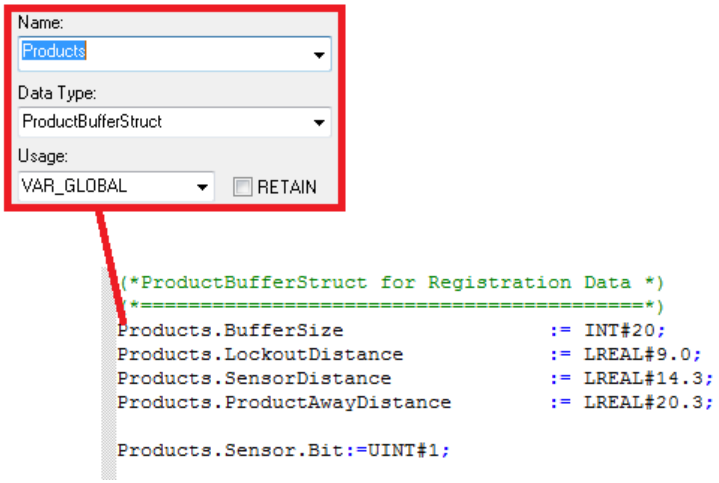


Figure 9: Definition of RegistrationData input

A graphical representation of the various elements of the RegistrationData input are shown in figure 10 below.

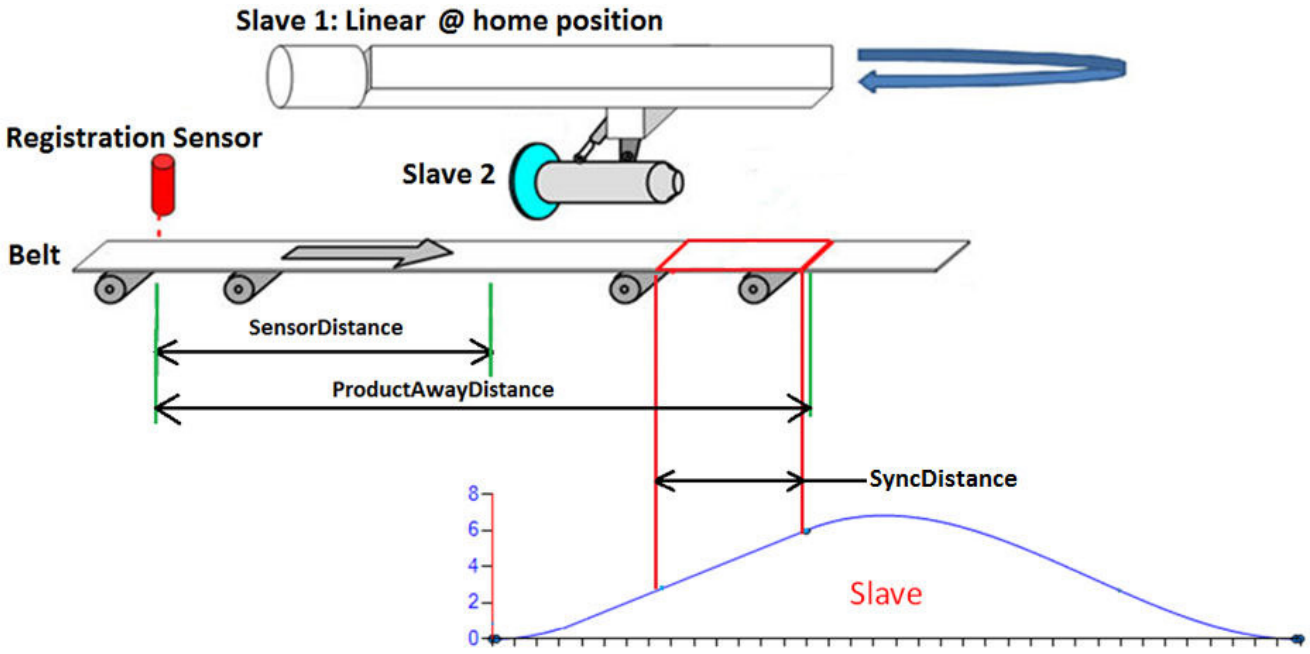


Figure 10: RegistrationData elements

2. Define a variable for ControlData. Populate the structure elements with:

- Mode
- DecisionPosition
- EndSyncPosition
- StartSyncPosition

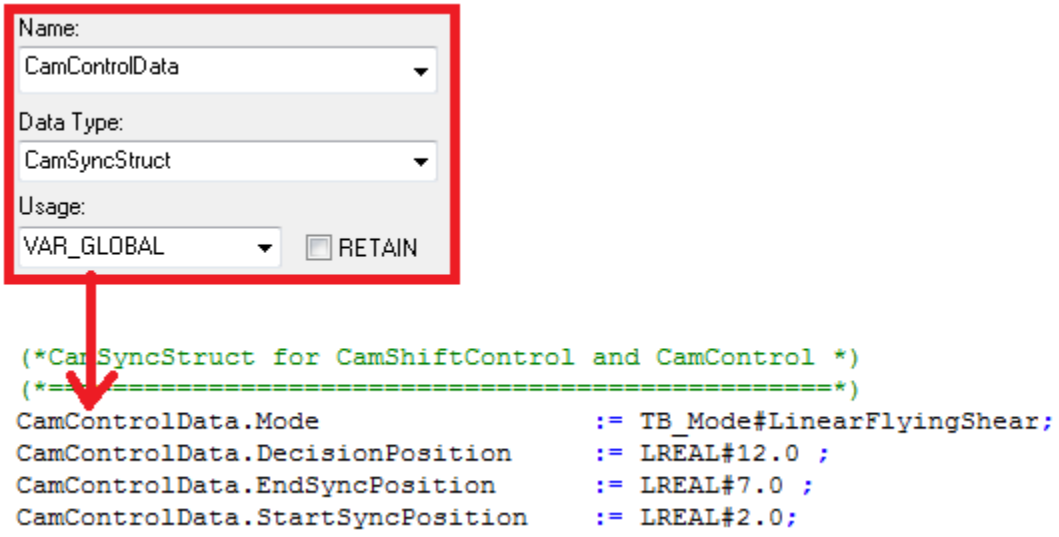


Figure 11: ControlData input structure

Figure 12 illustrates the various elements that form the ControlData structure.

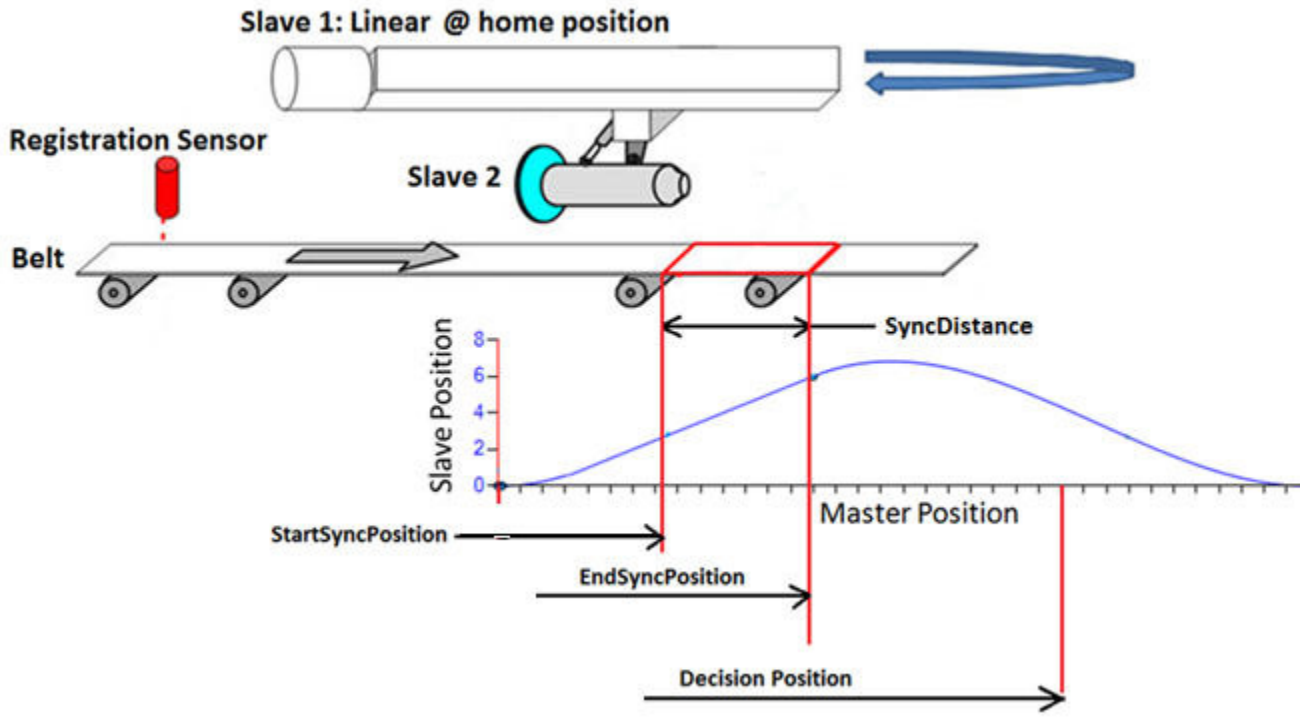


Figure 12: ControlData elements

3. Ensure that all slave axes are homed. The LinearFlyingShear function block assumes that the slave axes are at their home positions when the function block is enabled. If the slave axes are not at their home positions, the slave axes may run into mechanical limits depending on the stroke length of the designed cam.
4. Enable the LinearFlyingShear block.