

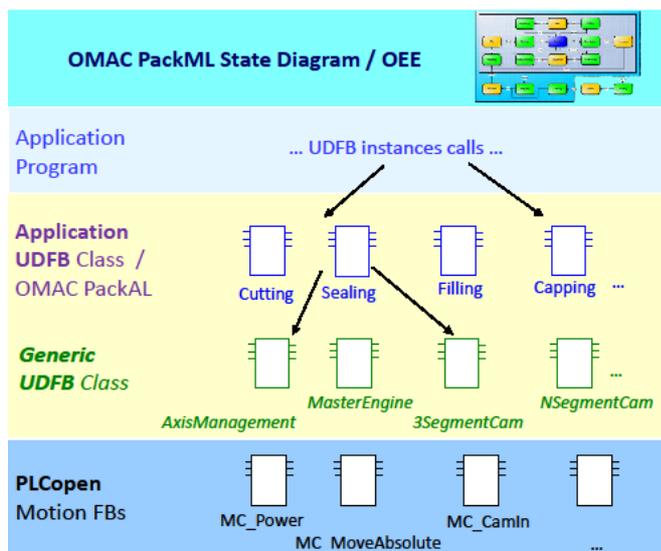
PLCopen Motion Control and OMAC PackAL – a good combination

The OMAC Packaging Workgroup has defined a set of specifications, incl. PackML and PackAL. PackML defines a state diagram and PackAL defines a number of function blocks at a higher level, like technology functions, specifically aimed at packaging machines. This specification now falls under the ISA.

PLCopen has added several specifications to the IEC 61131-3 standard for the purpose of motion control. OMAC is referring to both the IEC 61131-3 standard and the PLCopen Motion Control specifications.

This paper shows how some of the OMAC / PackAL specified functions can be created from the set of function blocks as defined by PLCopen for motion control as well as by the IEC 61131-3 standard.

One can represent the different levels in a diagram. The top layer is here represented by the OMAC PackML state diagram. However at this level other state diagrams can be implemented also.



These functionalities can be created from the set of Function Blocks as defined by PLCopen for Motion Control as well as by the IEC 61131-3 standard concerning the basic functionalities, here represented at the lowest level. With these basic functionalities one can create generic User Derived Function Blocks, UDFBs, like MasterEngine, for which one encapsulates the lower level functionalities. The next higher level contains functionalities closer to the application level, like cutting and sealing. Based on these functionalities it becomes quite straightforward to implement the application program itself. As such, the coupling to a state diagram like PackML helps to structure the software development process, while providing a harmonized look-and-feel of the machines application.

PackAL

PackAL has defined 3 sets of functionalities: Package Process Functions, Machine Communications, and

Machine Behavior Organization. In this specification we only deal with the ‘Packaging Process Functions’. They consist of:

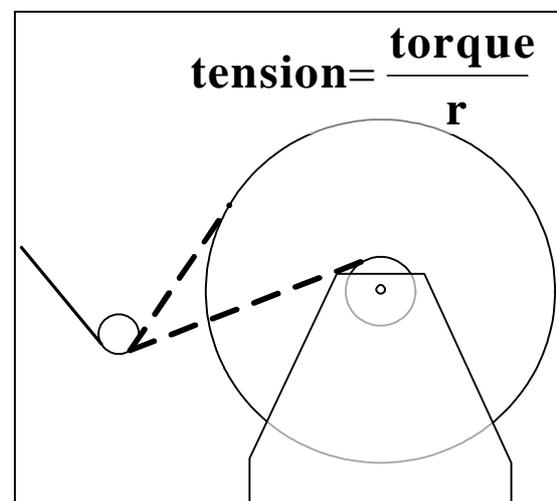
- Wind / Unwind Axis (constant surface velocity)
- Wind / Unwind Axis (constant torque mode)
- Dancer Control
- Registration
- Registration Correction
- Indexing
- Batch Counter
- Digital PLS (Digital CAM Switch)
- Set Override
- Jog Axis
- Flying Sync
- Gear_In with Dynamic Gear Factor
- Motion Command Stop with oriented halt

Some of these functionalities can be directly coupled to PLCopen motion control function blocks. However, some other functions need additional programming, or alternatives are possible especially concerning different inputs or feedback. For this reason we elaborated here on the Wind / Unwind functionality with constant surface velocity.

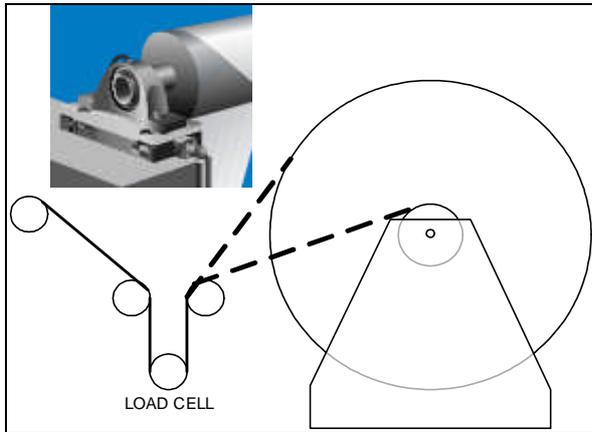
Wind / Unwind – General introduction

The OMAC Packaging Workgroup PackAL specification has defined two “Wind / Unwind” functionalities. However there are more solutions. This section deals with understanding of the winding / unwinding function. Following chapters will go more in detail of the mapping to the PackAL functions.

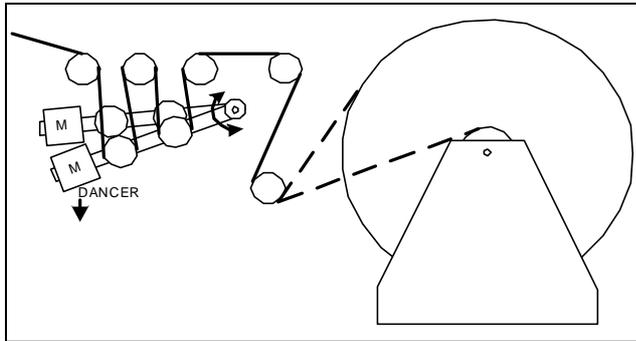
Winding with torque feedback can be done in several ways. A simple solution does not have any additional measuring inputs. Without measure, the motor torque is used. This limits the torque range and accuracy due to the diameter transformation and variable losses. See the first picture on the next page for this solution.



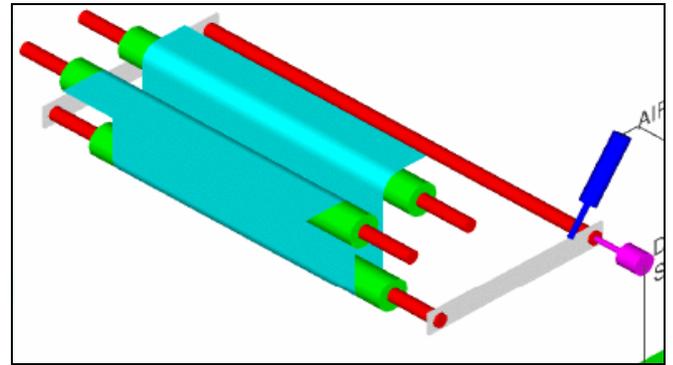
Another solution can be based on the feedback from a load cell. A load cell is fast and accurate but expensive.



Yet another solution could be with a so called dancer control, taking care of a constant tension.



This dance control system could be coupled to a sensor for feedback, like here:



With so many different mechanical solutions it is clear that there is not one unique control solution to it. For this reason PLCopen did not standardize this functionality at a high level, but wants to show how one can standardize its own mechanical and control solution as basis for a library of function blocks for wide spread use.

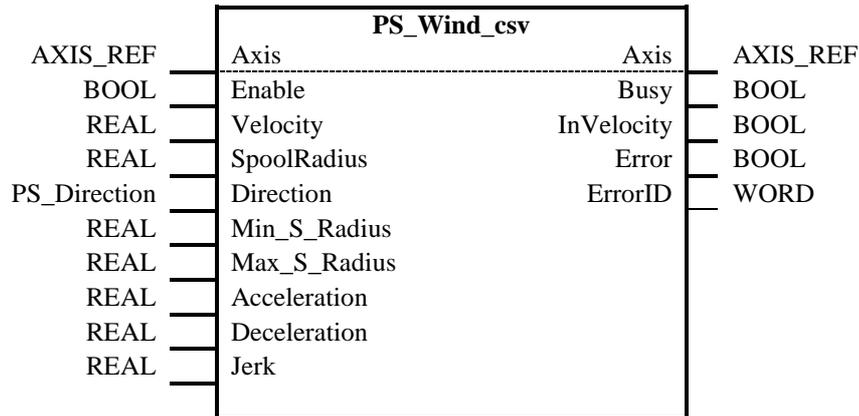
PackAL Wind / Unwind Axis (constant surface velocity, csv mode)

FB-Name		PS_Wind_csv	
This function block commands a controlled motion at a specified circumflex velocity for a wind / unwind axis. The circumflex is calculated from the radius of a wind/unwind spool, measured with a sensor.			
VAR_IN_OUT			
B	Axis	AXIS_REF	
VAR_INPUT			
B	Enable	BOOL	Start the motion at high level, stop at low level
B	Velocity	REAL	Value of the maximum Surface velocity [u/s]
B	SpoolRadius	REAL	Value of the radius of the spool [u]
B	Min_S_Radius	REAL	Value of the Minimal Spool Radius, initial value [u]
B	Max_S_Radius	REAL	Value of the Maximal Spool Radius, initial value [u]
E	Direction	PS_Direction	Enum type (pos, neg)
E	Acceleration	REAL	Value of the acceleration (increasing energy of the motor) [u/s ²]
E	Deceleration	REAL	Value of the deceleration (decreasing energy of the motor) [u/s ²]
E	Jerk	REAL	Value of the Jerk [u/s ³]
VAR_OUTPUT			
B	Busy	BOOL	Executing status
B	InVelocity	BOOL	command circumflex velocity phase active
B	Error	BOOL	Signals that error has occurred within Function block
B	ErrorID	WORD	Error Number

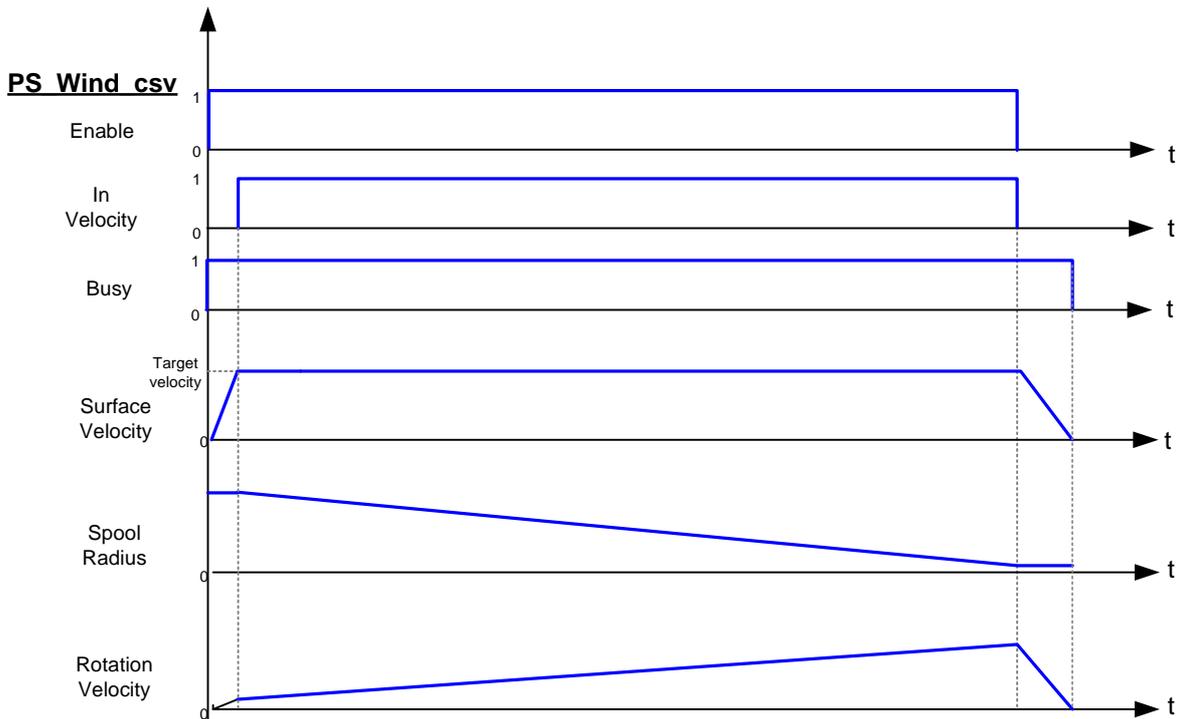
Note: This FB contains the mathematical calculation for coupling between a rotary slave axis and a translatory master axis. Its purpose is to generate and re-adjust a constant surface velocity (peripheral speed), relative to the master axis, for the rotary calibrated and controlled slave axis, depending on its spool diameter. Via a signal proportional to the spool radius, the spool radius of this slave axis is automatically evaluated and used for the calculation. This radius must never have the value 0.0 mm, since otherwise a calculation is no longer possible.

- The FB decelerates the axis to stop while winding above the Max_S_Radius
- The FB decelerates the axis to stop while unwinding below the Min_S_Radius
- SpoolRadius needs to satisfy $\text{Min_S_Radius} < \text{SpoolRadius} < \text{Max_S_Radius}$ to execute the action, otherwise Error = True, ErrorID set
- Error ID is implementation specific

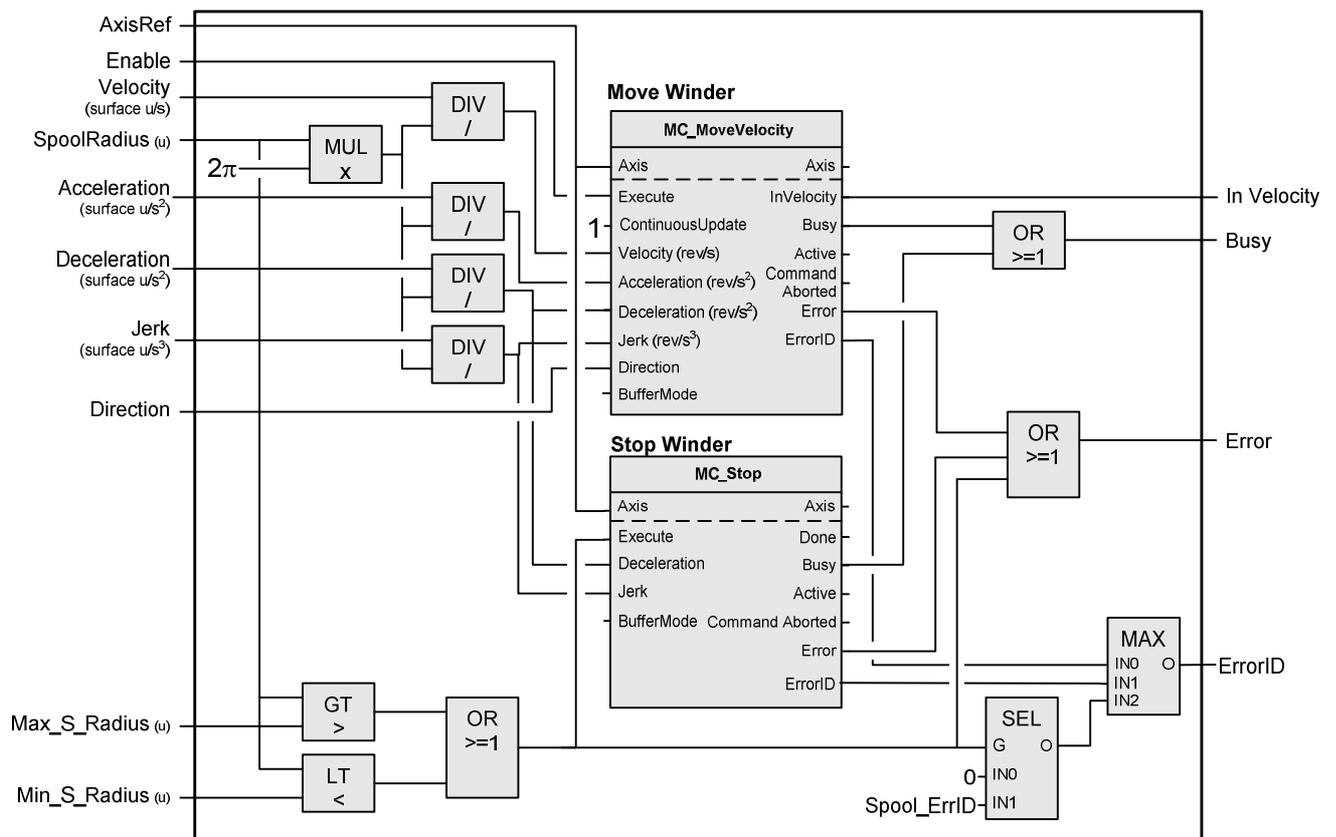
A graphical representation of this as an IEC 61131-3 Function Block:



And the corresponding timing diagram:



PS_wind_csv timing diagram



Explanation:

The “SpoolRadius” times 2 pi defines the relationship towards the velocity, the acceleration, deceleration, and jerk (all surface related). These are issued as inputs for the MC_MoveVelocity function block.

There is a check included if the “SpoolRadius” is between the “Max_S_Radius” and “Min_S_Radius”. If outside this range, it stops the winder while setting the error output. The combination of the two ErrorIDs only works if there is only one Error at a time. If necessary, this part of the program can be changed.

This functionality is easy to implement, and by adding this function block to the company library of the development system one can easily re-use this within the whole organization. This can be done for different systems, if applicable. In this way the same functionality is provided over and over again, while still providing the same interface to the programmer as well as the operator and maintenance people. This helps in understanding, training and maintenance.

PLCopen has defined motion control functionality during several years, and many suppliers have implemented these specifications on multiple platforms. The current suite of PLCopen Motion Control Specifications consists of the following parts:

<p>The suite of PLCopen Motion Control Specification:</p> <ul style="list-style-type: none"> • Part 1 - PLCopen Function Blocks for Motion Control • Part 2 - Extensions • Part 3 - User Guidelines • Part 4 - Coordinated Motion • Part 5 - Homing Procedures and Extensions • Part 6 - Fluid Power 	
--	--

For more specific information on PLCopen Motion Control, an overview of the certified products, and other activities, check www.PLCopen.org.