SPT NC Basic Slave

Last updated by | Nick Higgins | Oct 11, 2022 at 12:41 PM EDT

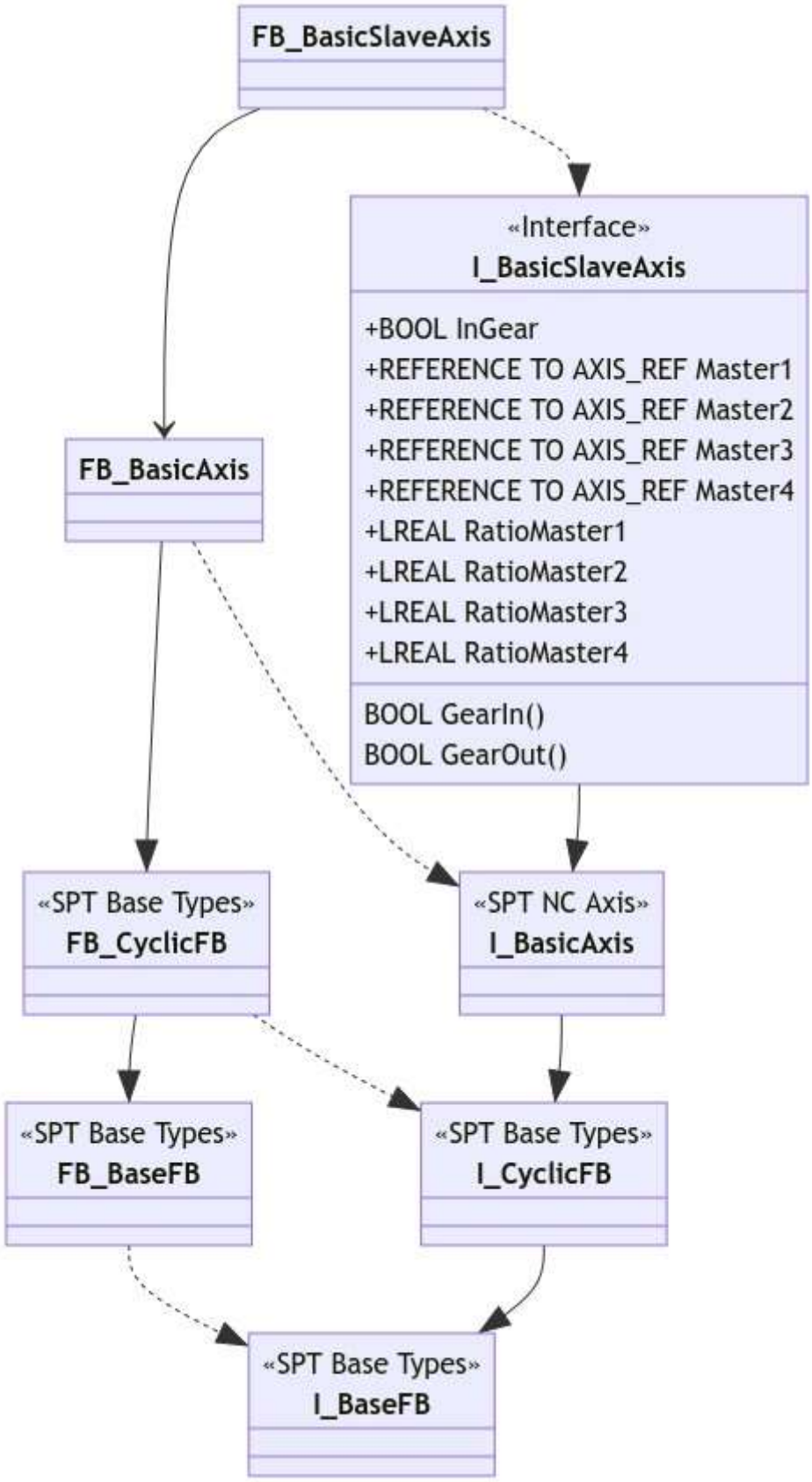
Contents

Overview

NC axis wrapper function block including basic coupling features (Gearing). This function block can be used without any PackML-related functions and does not by itself implement any of the component model interfaces. Use `FB_Component_BasicSlaveAxis` for PackML-based projects.

Extends `FB_BasicAxis` and thus can be used in exactly the same way for non-gearing functions.

Class Diagram



Interfaces

I_BasicSlaveAxis

(extends `I_BasicAxis`)

Defines basic required functionality for a basic NC axis plus gearing functions.

Properties

Property	Type	Access	Description
Master1	REFERENCE TO AXIS_REF	RW	Get/Set the first master axis to couple with
Master2	REFERENCE TO AXIS_REF	RW	Get/Set the second master axis to couple with
Master3	REFERENCE TO AXIS_REF	RW	Get/Set the third master axis to couple with
Master4	REFERENCE TO AXIS_REF	RW	Get/Set the fourth master axis to couple with
RatioMaster1	LREAL	RW	Get/Set the coupling ratio of the first master axis
RatioMaster2	LREAL	RW	Get/Set the coupling ratio of the second master axis
RatioMaster3	LREAL	RW	Get/Set the coupling ratio of the third master axis
RatioMaster4	LREAL	RW	Get/Set the coupling ratio of the fourth master axis
InGear	LREAL	RO	Get coupling status of this axis

Methods

Method	Return Type	Access	Description
GearIn	BOOL	PUBLIC	Establish coupling with master axes
GearOut	BOOL	PUBLIC	Release the coupling with master axes

Function Blocks

FB_BasicSlaveAxis

(extends `FB_BasiciAxis`, implements `I_BasicSlaveAxis`)

Complete implementation of I_BasicSlaveAxis. For use as a PackML component, use `FB_Component_BasicSlaveAxis`.

Notes

- See `FB_BasicAxis` for documentation of base motion funcntions

Examples

Couple two axes at 1:1 ratio

```
CASE State OF
  0:
    MySlaveAxis.Master1 REF= MyMasterAxis.Axis;
    MySlaveAxis.RatioMaster1 := 1.0;
    IF MySlaveAxis.GearIn() THEN
      State := State + 10;
    END_IF

  10:
    IF MySlaveAxis.InGear THEN
      State := State + 10;
    END_IF
END_CASE
```

Adjusting ratio on the fly

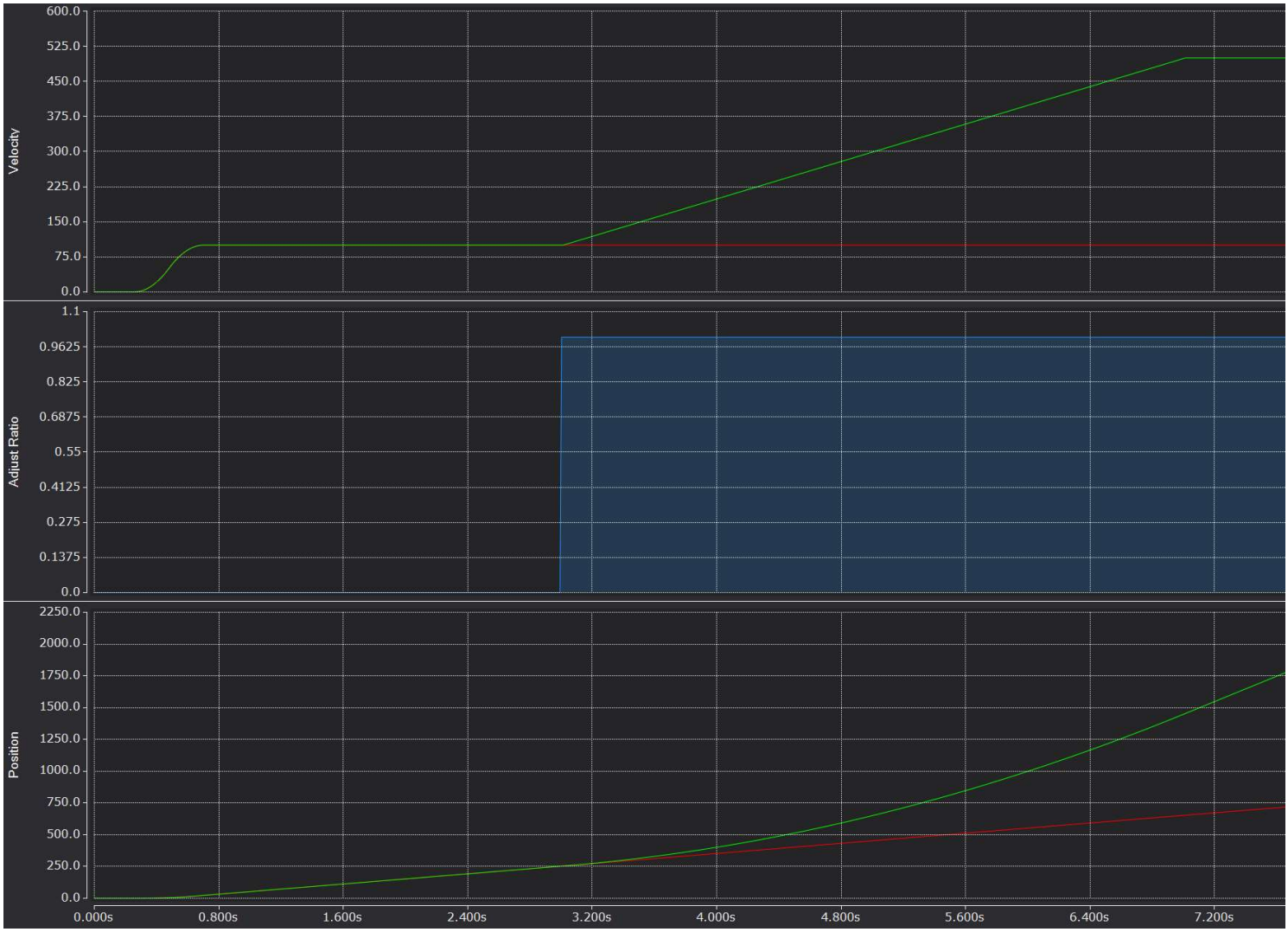
```
CASE State OF
  0:
    MySlaveAxis.Master1 REF= MyMasterAxis.Axis;
    MySlaveAxis.RatioMaster1 := 1.0;
    IF MySlaveAxis.GearIn() THEN
      State := State + 10;
    END_IF

  10:
    IF MySlaveAxis.InGear THEN
      State := State + 10;
    END_IF

  20:
    IF MyMasterAxis.MoveVelocity(100, FALSE) THEN
      State := State + 10;
    END_IF

  30:
    IF AdjustRatio THEN
      MySlaveAxis.RatioMaster1 := 5.0;
      State := State + 10;
    END_IF
END_CASE
```

Result



Phase adjustments using multimaster gearing

You can gear a second (and third/fourth) master to the same slave axis. This can be useful when you need to make phase adjustments to a master/slave coupling (conveyor gapping, etc.) Moves made to the additional masters are superimposed against the other masters according to the dynamics of the phase adjustment moves.

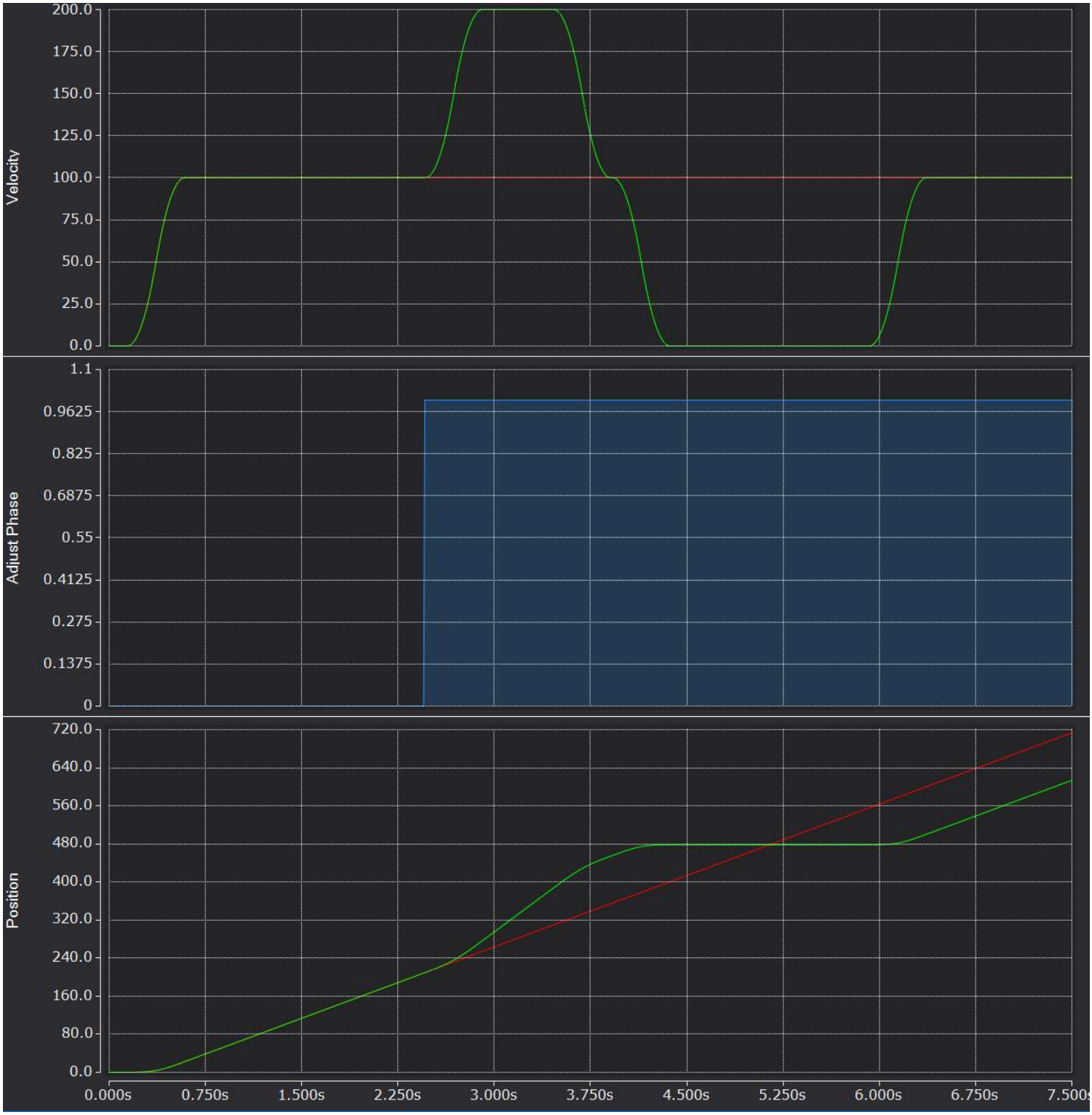
```
CASE State OF
  0:
    MySlaveAxis.Master1 REF= MyMasterAxis.Axis;
    MySlaveAxis.Master2 REF= MyPhaseAdjustAxis.Axis;
    MySlaveAxis.RatioMaster1 := 1.0;
    MySlaveAxis.RatioMaster2 := 1.0;

    IF MySlaveAxis.GearIn() THEN
      State := State + 10;
    END_IF

  10:
    IF MySlaveAxis.InGear THEN
      State := State + 10;
    END_IF

  20:
    //Start master axis
    IF MyMasterAxis.MoveVelocity(100, FALSE) THEN
      State := State + 10;
    END_IF
  30:
    IF AdjustPhase THEN
      //Advance phase of slave by +100 units
      IF MyPhaseAdjustAxis.MoveRelative(100, FALSE) THEN
        State := State + 10;
      END_IF
    END_IF
  40:
    IF NOT MyPhaseAdjustAxis.Busy THEN
      //Advance phase of slave by -200 units
      IF MyPhaseAdjustAxis.MoveRelative(-200, FALSE) THEN
        State := State + 10;
      END_IF
    END_IF
END_CASE
```

Result


Velocity
200.0
175.0
150.0
125.0
100.0
75.0
50.0
25.0
0.0
Adjust Phase
1.1
0.9625
0.825
0.6875
0.55
0.4125
0.275
0.1375
0
Position
720.0
640.0
560.0
480.0
400.0
320.0
240.0
160.0
80.0
0.0
0.000s
0.750s
1.500s
2.250s
3.000s
3.750s
4.500s
5.250s
6.000s
6.750s
7.500s