Southern African Large Telescope

Prime Focus Imaging Spectrograph

PFIS Mechanism Summary

SALT-3140AE0028

Jeffrey W Percival

Modification Record

Version	Date	Comment
2.0	19-Oct-2009	
2.1	02-Nov-2009	
2.2	21-Feb-2010	
2.3	22-Mar-2010	KHN comments
2.4	21-Apr-2010	KHN comments

PCON LabVIEW notes

The PCON LabVIEW environment comes in several pieces.

- LabVIEW itself, a data flow programming language. PCON uses LV 6.1, although National Instruments is no longer supporting that version. The LV target for the NIR instrument is 8.6. NI has moved on to LV 2009.
- Measurement and Automation Explorer (MAX). This is a stand-alone program used to manage your "device environment". You can examine what software you have installed, and you can operate your NI devices (motors, digital ins and outs, analog ins) without needing to invoke LabVIEW itself.
- FlexMotion: this is an add-on module (Windows only, no Mac or Linux) that provides the VIs that operate your motors from within a LabVIEW program.

MAX Settings: MAX is also used to program the NI 7334 4-axis stepper motor controllers. Use MAX to set a stepper motor's steps per turn, or an encoder's counts per turn. A summary of the current settings can be on PCON is the following folder:

C:\Documents and settings\Administrator\My Documents\PFIS\PCS\pfis-max-settings.xls

PCON Initialization: PCON has many "constants" that are used to operate the instrument: gear ratios, encoder constants, station trim values. PCON initializes itself by reading a file of stored values:

C:\salt\svn\branches\sutherland\pcon\src\data\PCON Constants.txt

Some quantitative items (e.g. how many motor steps to move from one station to the next) are calculated from first principles from these constants. Instead of, say, wiring in a slitmask station width of 243 steps, we do a calculation in slitmask-algebra.vi that combines gear ratios, worm gear threading, steps per turn, etc. and calculate what we need to know. This is so that if a motor is replaced with a slightly different one, you just update the fundamental data and the VI will still operate as desired. Some constants may need to be adjusted as a result of reassembling or aligning a mechanism. These are listed in **boldface** font in the sections below.

Mechanism Numbers

05-Slitmask

06-Waveplate

07-Shutter

08-Focus

09-Etalons

10-Grating

11-Beamsplitter

12-Filter

13-Articulation

Control System Item Naming Scheme

Every actuator name starts with P or S

- P: Pneumatic
- S: Stepper

This is followed by the mechanism number (see above):

S.10	a stepper in the grating mechanism.

Actuators are numbered sequentially within a mechanism.

S.10.1	the 1st stepper in the grating
S.10.2	the 2nd stepper in the grating

Each actuator has inputs (in, out, end of travel) and outputs (insert, remove) associated with it.

Inputs are generally coded as "B" (binary or boolean). Inputs (including interlocks) are read from the SCB100 connector block by the PXI-6508 digital input card.

Outputs are generally coded as "R" (relays). Outputs are written to the SCB68 breakout boxes via the 32-bit output section of the PXI-7334 4-axis Stepper Motor Controller.

I/O items are numbered sequentially within an actuator:

S.10.1-B.1	magazine fwd hard
S.10.1-B.2	magazine rev hard
S.10.1-B.3	magazine home
S.10.1-B.4	magazine station 1
S.10.1-B.5	magazine station 2
S.10.1-B.6	magazine station (1 & not 2)

Interlocks are generated within the CPLDs and end with "-M" for "monitor".

Summary of Mechanism Transitions

The design concepts that underlie the PFIS mechanism control system are presented in detail in the document "Roadmap to the PFIS Control System Software". This document is available at the web link given above. The “flowcharts” that were used in coding the LabVIEW Vis are in the form of “state diagrams”. State diagrams appear frequently in SALT documentation, usually describing the states and flow of control for observatory systems (Init, Slew, Track, Maintenance, Shutdown). See SALT-221AW1026 for more details on this approach. The PFIS software design followed this approach, representing the flow of control in the form of state diagrams. This document will present them for each mechanism.

Each mechanism has a number of "states" defined, like "Opened", "Closed", "Inserted", "Removed". States are connected by "transitions", which are actions that take you from one state to the next. Transitions are actions, like closing a relay or stepping a motor.

Many transitions are interlocked: some set of conditions must be satisfied, or else the action is forbidden. For example, to fetch a slitmask into the elevator, you must be at a station. No station, no fetch. These interlocks are programmed into the CPLD logic devices and are monitored as digital inputs to the control system.

In software, each transition is implemented as a 3-step process:

1. Is the action OK to take?
2. If so, take the action
3. If taken, wait for it to complete (see timeout discussion in net section)

Step 1 is accomplished (usually) by checking the CPLD interlock. The CPLD interlock may depend on logic applied to individual sensors, like (removed and (fetched or stowed)), which the software could implement on its own, but then the logic is defined in two places (CPLD and LabVIEW), which is bad. So we just let the CPLD do its work, and peek at the result.

Step 2 generally involves asserting a digital output bit to control a relay, or moving a motor some number of steps.

Step 3 involves checking the state of sensors that indicate the desired state has been reached.

Some actions are always allowed (like Beamsplitter Insert); for those, an interlock is dutifully defined in the data base (P.11.1-R.1-M), but there is no corresponding CPLD bit.

For some mechanisms, just getting the actuator ready to go was more complicated than actually using the mechanism. Compare the waveplate-axis state diagram with the waveplate state diagram. The "Axis" state diagrams manage getting the axis ready to go and cleaning up after its done. It's just a way to modularize the transitions, so we don't end up with a 12-state machine.

Mechanism Timeouts

In the “ok?, do it, wait” triplet of Vis discussed above, the wait portion uses a timeout as it waits for the indication of a completed action. These timeouts are on the front panels of the “wait” Vis, are wired up to the connector pane, but are typically unwired from a higher level VI, so the front panel value is the one that’s used. The timeouts are given in each mechanism section below.

A Note on Station Creeping

The three magazine mechanisms (slitmask, grating, filter) use a “picket fence” of vane switches to achieve a very narrow active region at each magazine slot. We want to find the rising edge of the slot indicator to within 1 motor step. We do this by taking largish moves (the creep-home size, typically 5 steps) towards home, then creep away (creep-away size, 1 step) until we find the rising edge of the slot indicator.

For each of these mechanisms, there are a triplet of creeper Vis:

-creeper.vi (manages the creep process)

    -creep-home.vi (sub-VI)

    -creep-away.vi (sub-VI)

Creep sizes appear as front-panel controls in the two sub-Vis, but they are wired to the connector pane. Creep sizes appear again in the parent VI, and re wired to the proper terminals of the sub-Vis, thereby overriding them. Change them in the parent, not the sub-VIs.

This technique is used by the grating and filter state machines as well.

A Note on Motor Timeouts

During the large motorized motions (slitmask, focus, grating, filter, and articulation), there is a timeout in play that is different from the mechanism timeouts discussed above. The motor timeout is imposed on the duration of the motion; then, when it’s done, the “wait” VI uses the mechanism timeout to verify that it is done. The motor timeout is computed on the fly by using a “time per station” control on the front panel, multiplied by the number of stations we intend to move. The “time per station” control is not wired to the connector pane, and so cannot be programmatically configured. You have to go to the proper front panel to change it. This will be discussed in the mechanism sections below.

Slitmask (Mechanism 05)

Slitmask constants:

- Steps per Turn=200 // the main worm gear
- Turns per Inch=8 // the main worm gear
- Station Break=11 // the 1st of the narrower magazine slots
- Wide Station (mm)=9.000000 // the size (height) of the 10 wide slots
- Narrow Station (mm)=5.800000 // the size (height) of the remaining narrow slots
- Station 1 (steps)=-2180 // the motor position of the 1st slot
- Elevator Home (volts)=8.680000 // the voltage on the yo-yo encoder when at home
- Encoder Constant (volts/mm)=0.02504 // for the yo-yo encoder
- **Home Trim** (steps)=-20 // how home varies from the home switch
- **Station Trim** (steps)=0 // how much stations vary from the station indicator

05-Slitmask: selects, fetches, inserts, removes, and stows slitmasks

slitmask-t1: select (2 s per station, timeout 5000 ms)
action: move slitmask elevator to desired magazine slot
interlock: slitmask removed AND (fetched OR stowed)
indicator: slitmask station on (= station1 AND NOT station2)

The stations (“station1” and “station2”) are indicated by a pair of "picket fence" of steel vanes read by 2 Hall-effect sensors. They are slightly offset from each other, so that the actively sensed region is quite narrow. The logic used is "A and NOT B", where A and B are the active regions of each sensor by itself. The breakdown of actions is:

1. Given the station, calculate the desired step count using slitmask-algebra.vi.
2. Move to that step count.
3. Now do a high-resolution search for the leading edge of the combined vane switch signal:
   - Creep toward home one step at a time, until A and B are both off
   - Creep away from home until (A and NOT B) appears
   - The creep-home and creep-away step sizes are on the front panel of slitmask-creeper.vi
   - The creep-step sizes can differ, to speed things up: do a rough creep for the first one, then a 1-step creep next.
4. Apply a fixed trim value of steps (**Station Trim**)

slitmask-t2: fetch (pull slitmask out of the magazine into the elevator) (timeout 5000 ms)
action: set fetch then clear stow
interlock: slitmask station (= station1 AND NOT station2) OR fetched

indicator: slitmask fetched

slitmask-t3: seek home (send the elevator down to the focal plane) (timeout 5000 ms)
  action: perform a seek home operation
  interlock: removed AND (fetched OR stowed)
  indicator: slitmask homed

The home sequence is:

1. Do a FlexMotion seek home sequence on the home sensor.
2. Apply a fixed trim value of steps (**Home Trim**)

slitmask-t4: insert (puch the slitmask into the beam for observing) (timeout 10000 ms)
  action: set slitmask insert
  interlock: slitmask home
  indicator: slitmask inserted

slitmask-t5: remove (pull the slitmask out of the beam) (timeout 10000 ms)
  action: clear slitmask insert
  interlock: slitmask home
  indicator: slitmask removed

slitmask-t6: stow (return the slitmask to its home in the magazine) (timeout 5000 ms)
  action: clear fetch then set stow
  interlock: slitmask station (= station1 AND NOT station2) OR stowed
  indicator: slitmask stowed

A note on how slitmasks are managed:

There is a little difference in the way the slitmask state table works, compared to the non-magazine mechanisms. Suppose you are in the state “inserted” with mask A. Now you want to be in the state “inserted” with mask B. A naïve look at the state table would show that if your current state in “inserted” and your desired state is “inserted”, then the table would tell you to do nothing.

The magazine mechanisms differ from the others in that they have *data*: the optic that happens to be in use. So “inserted+A” differs from “inserted+B”. The way the software manages this is in the “what should I do next” module. Normally, the “what should I do next” modules (one for each state machine) take two inputs (the current state and the desired state) and produce one output: what action to take next.

The magazine modules take four inputs: current state + current optic, and desired state + desired optic. The module uses the data to override the usual state table output. It overrides the state table until you have stowed A, and then lets go and allows the state table to get you to optic B.

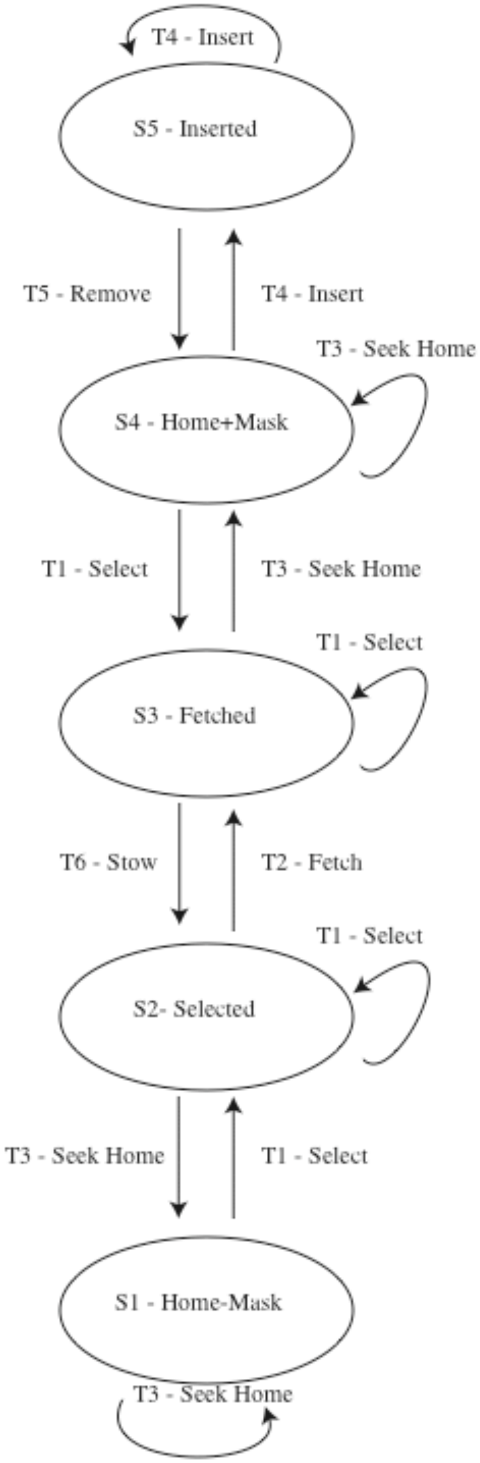
This technique is used by the grating and filter state machines as well.

	Desired State					
Current State	S0	S1	S2	S3	S4	S5
S0	T0/S0	T0/S0	T0/S0	T0/S0	T0/S0	T0/S0
S1	T0/S0	T3/S1	T1/S2	T1/S2	T1/S2	T1/S2
S2	T0/S0	T3/S1	T1/S2	T2/S3	T2/S3	T2/S3
S3	T0/S0	T6/S2	T6/S2	T1/S3	T3/S4	T3/S4
S4	T0/S0	T1/S3	T1/S3	T1/S3	T3/S4	T4/S5
S5	T0/S0	T5/S4	T5/S4	T5/S4	T5/S4	T4/S5

Inserted	Fetched	
0	0	S2
0	1	S3
1	0	S0
1	1	S5

State Detection

PFIS Slitmask State Diagram



05-Slitmask-Axis: manages motor power

This sub-sub-system powers up the elevator motor before making a move, and powers it down after the move is complete.

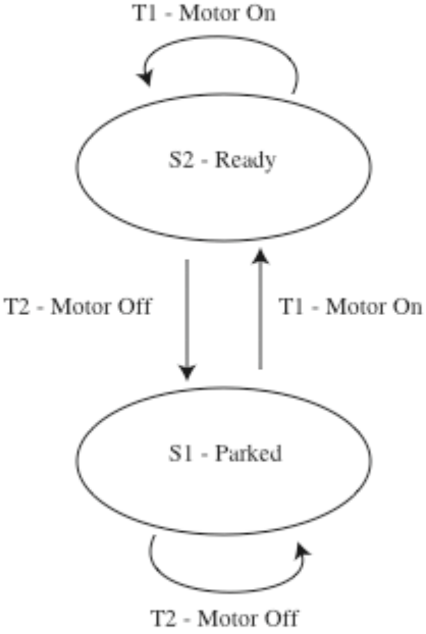
sm-axis-t1: motor on (timeout 5000 ms)
- action: clear slitmask "All Windings Off" (VI/ FlexMotion/ PXI/ 7334/UMI)
- interlock: none
- indicator: slitmask motor power on (UMI/7334/PXI/FlexMotion/VI)

sm-axis-t2: motor off (timeout 5000 ms)
- action: set slitmask "All Windings Off" (VI/ FlexMotion/ PXI/ 7334/UMI)
- interlock: none
- indicator: slitmask motor power off (UMI/7334/PXI/FlexMotion/VI)

		Desired State		
		S0	S1	S2
Current State	S0	T0/S0	T0/S0	T0/S0
	S1	T0/S0	T2/S1	T1/S2
	S2	T0/S0	T2/S1	T1/S2

Motor Killed	
0	S2
1	S1

State Detection



PFIS Slitmask Axis State Diagram

Waveplate

Waveplate constants:

- Bearing Teeth=144 // the big circular bearing
- Pinion Teeth=30 // the little driving gear
- Steps per Turn=500 // for the pinion
- Encoder Counts per Rev=81000 // for the Mercury 3000 encoder
- Stations per Rev=32 // number of notched observing stations
- **HWP Station 0** (steps)=758 // location of HWP station 0
- **QWP Station 0** (steps)=1314 // location of QWP station 0
- Encoder delay=500 // time to wait for the encoder to be power-up happy (ms)

06-Waveplate: manages the waveplate configuration (linear, circular)

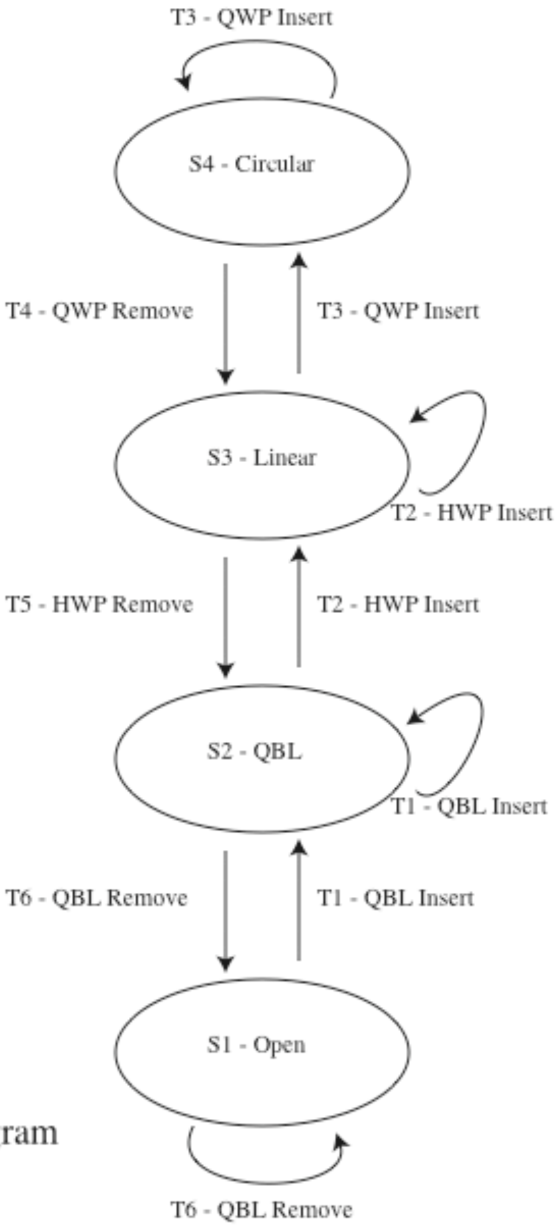
waveplate-t1: qbl insert (timeout 15000 ms)
action: set qbl insert (the compensator blank for the quarter wave plate)
interlock: none
indicator: qbl inserted
waveplate-t2: hwp insert (the half wave plate) (timeout 10000 ms)
action: set hwp insert
interlock: qbl inserted
indicator: hwp inserted
waveplate-t3: qwp insert (the quarter wave plate) (timeout 10000 ms)
action: set qwp insert
interlock: hwp inserted
indicator: qwp inserted
waveplate-t4: qwp remove (timeout 10000 ms)
action: clear qwp insert
interlock: none
indicator: qwp removed
waveplate-t5: hwp remove (timeout 10000 ms)
action: clear hwp insert
interlock: none
indicator: hwp removed
waveplate-t6: qbl remove (timeout 10000 ms)
action: clear qbl insert
interlock: none
indicator: qbl removed

Desired State

Current State	S0	S1	S2	S3	S4
S0	T0/S0	T0/S0	T0/S0	T0/S0	T0/S0
S1	T0/S0	T6/S1	T1/S2	T1/S2	T1/S2
S2	T0/S0	T6/S1	T1/S2	T2/S3	T2/S3
S3	T0/S0	T5/S2	T5/S2	T2/S3	T3/S4
S4	T0/S0	T4/S3	T4/S3	T4/S3	T3/S4

QWP Inserted	HWP Inserted	QBL Inserted	
0	0	0	S1
0	0	1	S2
0	1	0	S0
0	1	1	S3
1	0	0	S0
1	0	1	S0
1	1	0	S0
1	1	1	S4

State Detection



PFIS Waveplate State Diagram

06-Waveplate-Axis: manages the waveplate encoder, motor, and detente

wp-axis-t1: encoder on (timeout 5000 ms)
    action: set waveplate encoder power bit
    interlock: none
    indicator: none

This is a complicated transition. The waveplate encoder uses an LED that science detector can “see” during an observation. To prevent this, we turn off the encoder after the waveplate is latched into the correct position. But turning off the encoder loses the position, so it is cached prior to turning it off. When turning it on, first we apply power to it, then we have to load the saved step-value into FlexMotion. The FlexMotion initialization is:

1. Turn on the encoder (apply power)
2. Wait a fixed delay (the **Encoder delay** constant)
3. Disable the axis
4. Open the loop
5. Unmap the encoder
6. Reset the encoder (load it with the saved value)
7. Map the encoder
8. Close the loop
9. Enable the Axis
10. Read the encoder to verify the correct value

Note that there is an “Encoder delay” control on the wp-axis-t1 front panel; this is not used, it is unwired on the block diagram. The value from the Encoder Constants is used instead.

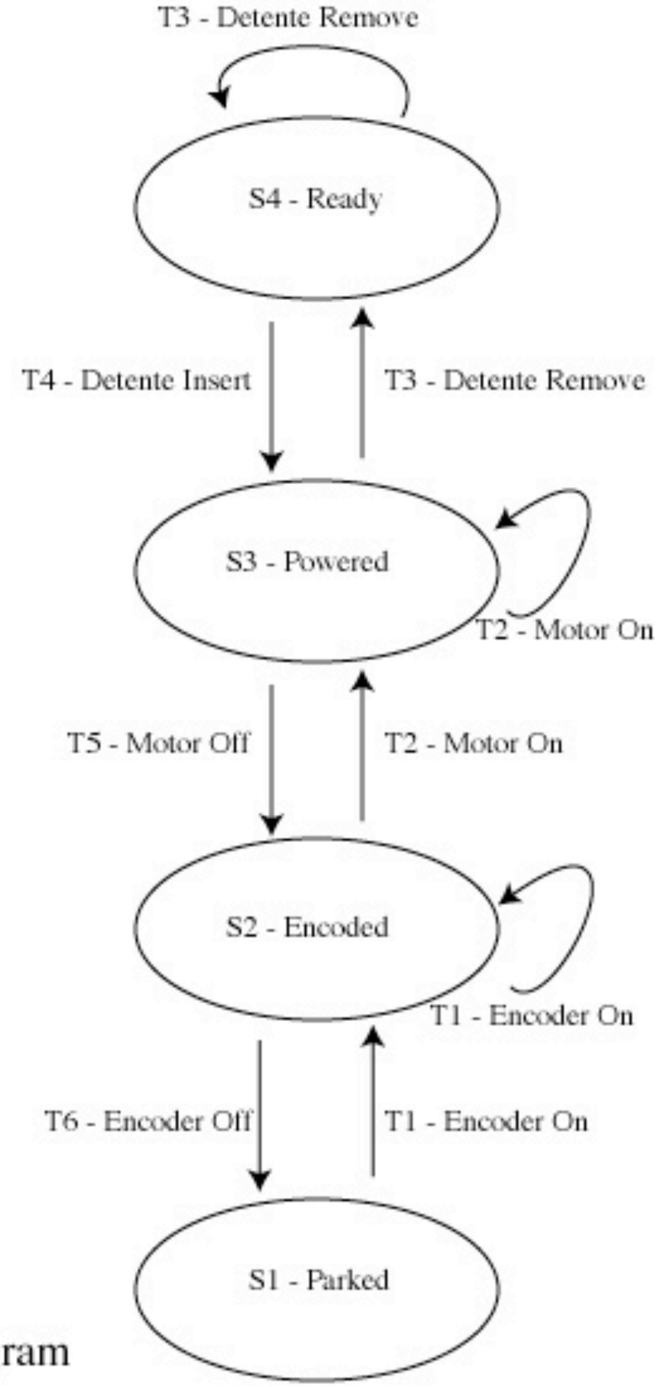
wp-axis-t2: motor on (timeout 5000 ms)
    action: clear waveplate "All Windings Off"
    interlock: none
    indicator: waveplate motor power on
wp-axis-t3: detente remove (timeout 5000 ms)
    action: set the waveplate detente remove bit
    interlock: none
    indicator: waveplate detente removed
wp-axis-t4: detente insert (timeout 5000 ms)
    action: clear the waveplate detente remove bit
    interlock: none
    indicator: waveplate detente inserted
wp-axis-t5: motor off (timeout 5000 ms)
    action: set waveplate "All Windings Off"
    interlock: none
    indicator: motor power off
wp-axis-t6: encoder off (timeout 5000 ms)

action: cache encoder value then clear waveplate encoder power bit
interlock: none
indicator: none

		Desired State				
		S0	S1	S2	S3	S4
Current State	S0	T0/S0	T0/S0	T0/S0	T0/S0	T0/S0
	S1	T0/S0	T0/S1	T1/S2	T1/S2	T1/S2
	S2	T0/S0	T6/S1	T1/S2	T2/S3	T2/S3
	S3	T0/S0	T5/S2	T5/S2	T2/S3	T3/S4
	S4	T0/S0	T4/S3	T4/S3	T4/S3	T3/S4

Detente Removed	Motor On	Encoder On	
0	0	0	S1
0	0	1	S2
0	1	0	S1
0	1	1	S3
1	0	0	S1
1	0	1	S2
1	1	0	S1
1	1	1	S4

State Detection



PFIS Waveplate Axis State Diagram

06-Waveplate-Move: moves to the specified observing angle

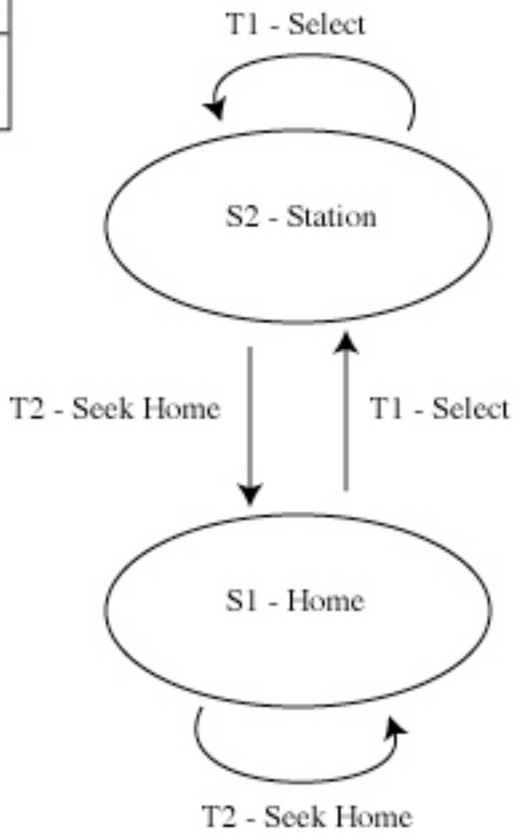
wp-move-t1: select station (timeout 5000 ms)
    action: move waveplate to selected station
    interlock: waveplate detente removed
    indicator: none

In this action, we compute the difference in steps between the current station and the desired station. We then decide which direction to go (the short way around), and do a relative move (a step adjustment rather than an absolute step value) to get to the station.

wp-move-t2: seek home (uses index mark as home indicator) (timeout 5000 ms)
    action: perform a seek index operation
    interlock: waveplate detente removed
    indicator: waveplate indexed

Here we do a FlexMotion seek operation, but use the encoder's index pulse instead of a home sensor. After finding the index, we apply a station 0 trim (the waveplate constants **HWP Station 0** or **QWP Station 0**). The trim is simply the step location of station 0.

		Desired State		
		S0	S1	S2
Current State	S0	T0/S0	T0/S1	T0/S2
	S1	T0/S0	T2/S1	T1/S2
	S2	T0/S0	T2/S1	T1/S2



Assume S2
State Detection

PFIS Waveplate Movement State Diagram

Shutter

Shutter constants: none

07-Shutter: opens and closes the shutter

“shutter-assist-enable” is enabled by a jumper in PSC1.

If "shutter-assist-enable" is off:

shutter-t1: open (timeout 5000 ms)
    action: set shutter open
    interlock: none
    indicator: shutter opened
shutter-t2: assist on (timeout 5000 ms)
    action: set shutter assist
    interlock: shutter opened
    indicator: shutter assist on
shutter-t3: relax (turn off solenoid) (timeout 5000 ms)
    action: clear shutter open
    interlock: none
    indicator: none
shutter-t4: assist off (timeout 5000 ms)
    action: clear shutter assist
    interlock: none
    indicator: shutter assist off
shutter-t5: close (timeout 5000 ms)
    action: clear shutter open
    interlock: none
    indicator: shutter closed

If "shutter-assist-enable" is on:

shutter-t1: open
    action: set shutter open (CPLD sets pneumatic and turns off solenoid)
    interlock: none
    indicator: shutter opened, shutter assist on
shutter-t2: n/a
shutter-t3: same as shutter-t5
shutter-t4: n/a
shutter-t5: close
    action: clear shutter open
    interlock: none
    indicator: shutter closed, shutter assist off

Desired State

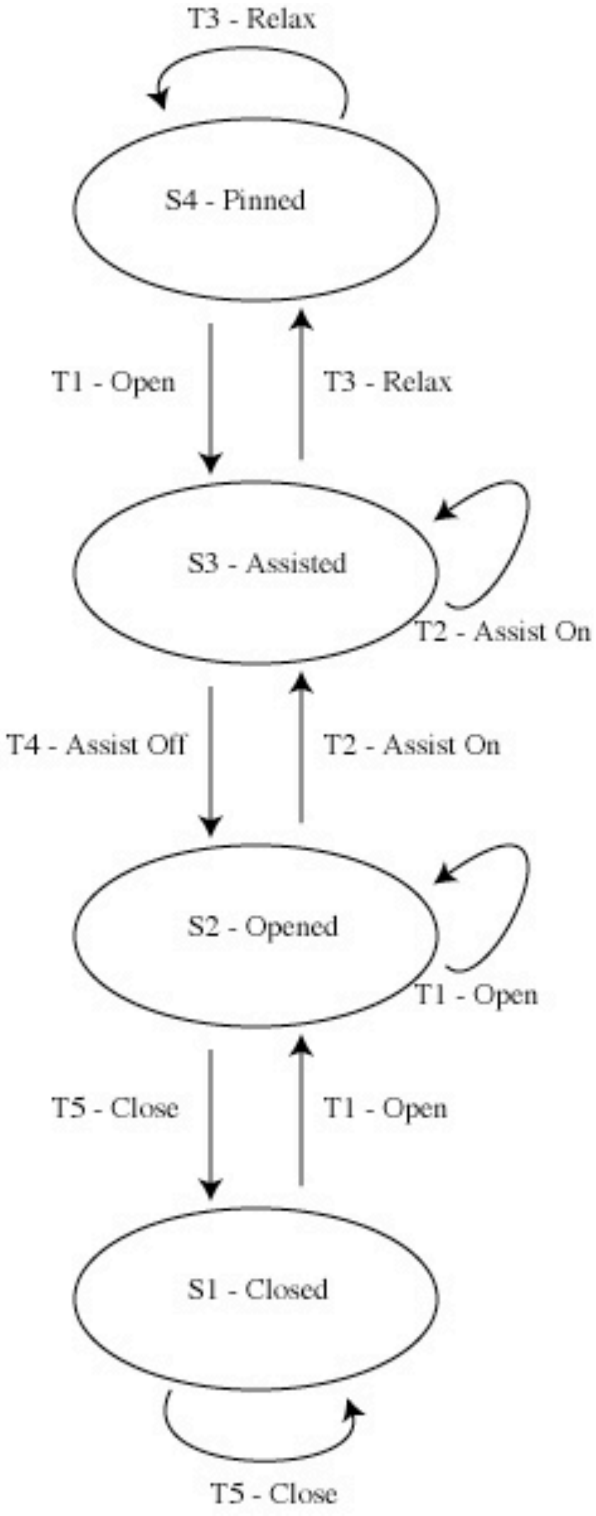
Current State	S0	S1	S2	S3	S4
S0	T0/S0	T0/S0	T0/S0	T0/S0	T0/S0
S1	T0/S0	T5/S1	T1/S2	T1/S2	T1/S2
S2	T0/S0	T5/S1	T1/S2	T2/S3	T2/S3
S3	T0/S0	T4/S2	T4/S2	T2/S3	T3/S4
S4	T0/S0	T1/S3	T1/S3	T1/S3	T3/S4

Assisted	Opened	
0	0	S1
0	1	S2
1	0	S0
1	1	S3

State Detection

Note: T3 (Relax) is like T5 (Close) except that no sensors change state.

PFIS Shutter State Diagram



Focus

Focus constants:

- Steps per Turn=48 // of the stepper mike
- Gear Ratio=28.444444 // internal to the stepper mike
- Screw Pitch (mm/turn)=0.400000 // internal to the stepper mike
- **LVDT @ Home (volts)**=8.755000// measured
- Encoder Constant (V/mm)=-4.000000 // from the data sheet?
- **LVDT @ Fwd Limit (volts)**=2.5850 // measured
- Steps @ Fwd Limit=51942 // measured
- **LVDT @ Rev Limit (volts)**=8.870000 // measured

08-Focus

focus-t1: select focus position (0.1 s per micron, timeout 5000 ms)
action: set the focus to the desired setting
interlock: none
indicator: none

This is a simple FlexMotion move. We convert the desired focus in microns into steps, and move. There is a timeout imposed on the move that is proportional to the relative motion.

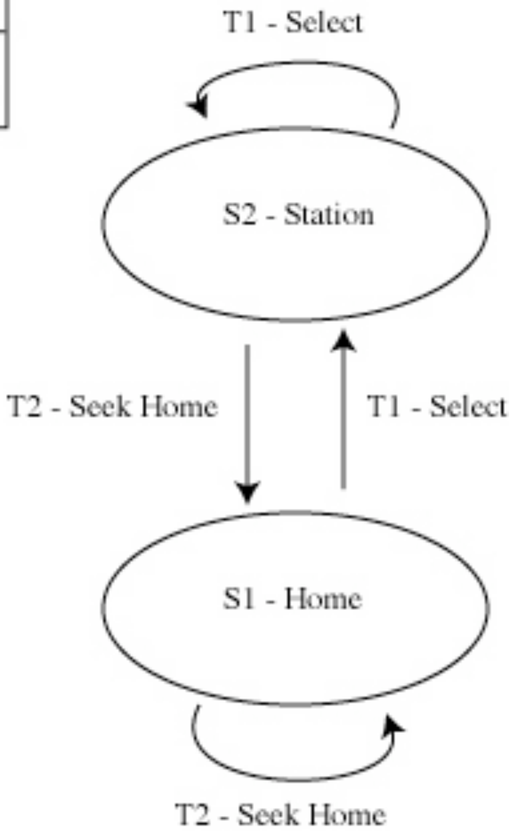
There is an LVDT providing absolute position sensing, but this is not used in the controlling the move.

focus-t2: seek home (timeout 5000 ms)
action: perform a focus seek home procedure
interlock: none
indicator: focus home

This action does a FlexMotion home seek operation on the home sensor.

The LVDT analog sensor is not used as part of the control loop. The focus stepper has a quadrature encoder on it that is used by the NI 7334 stepper motor controller to close the loop. The LVDT sensor indicates the physical movement of the optic.

		Desired State		
		S0	S1	S2
Current State	S0	T0/S0	T0/S0	T0/S0
	S1	T0/S0	T2/S1	T1/S2
	S2	T0/S0	T2/S1	T1/S2



Assume S2

State Detection

PFIS Focus State Diagram

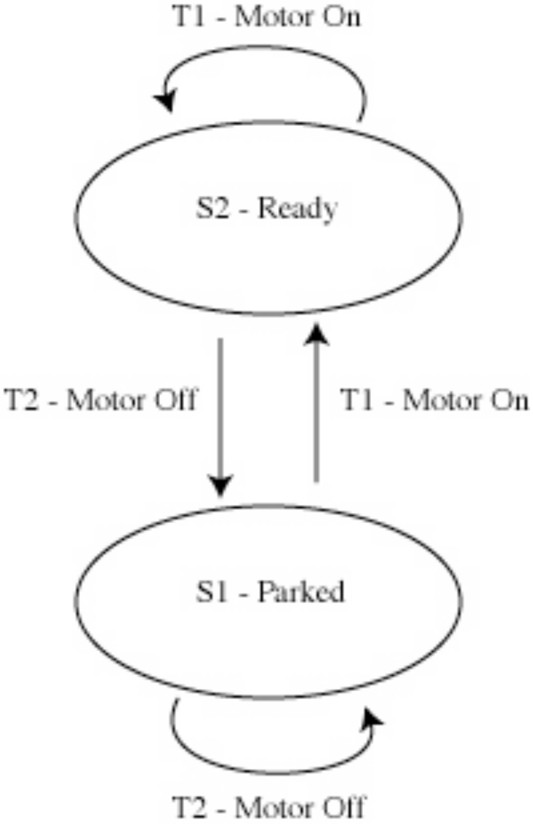
08-Focus-Axis: manages focus motor power

fo-axis-t1: motor on (timeout 5000 ms)
    action: clear focus "All Windings Off"
    interlock: none
    indicator: motor power on
fo-axis-t2: motor off (timeout 5000 ms)
    action: set focus "All Windings Off"
    interlock: none
    indicator: motor power off

		Desired State		
		S0	S1	S2
Current State	S0	T0/S0	T0/S0	T0/S0
	S1	T0/S0	T2/S1	T1/S2
	S2	T0/S0	T2/S1	T1/S2

Motor Killed	
0	S2
1	S1

State Detection



PFIS Focus Axis State Diagram

Etalons

Etalon constants: none

09-Etalons: manages the two etalons

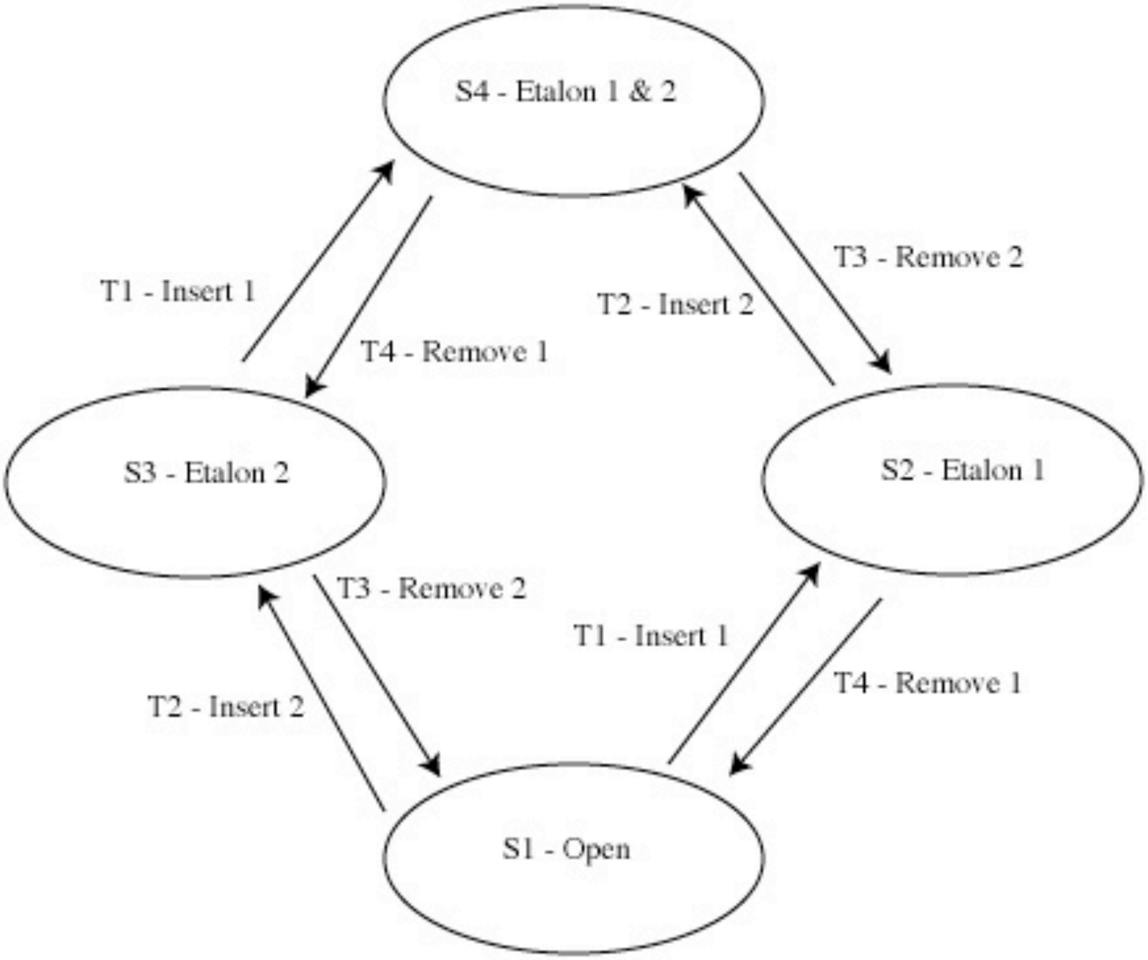
This code moves the two etalons into and out of the beam.

elaton-t1: insert 1 (timeout 5000 ms)
  action: select etalon 1 state S4
  interlock: see above
  indicator: see above
etalon-t2: insert 2 (timeout 5000 ms)
  action: select etalon 2 state S4
  interlock: see above
  indicator: see above
etalon-t3: remove 2 (timeout 5000 ms)
  action: select etalon 2 state S1
  interlock: see above
  indicator: see above
etalon-t4: remove 1 (timeout 5000 ms)
  action: select etalon 1 state S1
  interlock: see above
  indicator: see above

		Desired State				
		S0	S1	S2	S3	S4
Current State	S0	T0/S0	T0/S0	T0/S0	T0/S0	T0/S0
	S1	T0/S0	T0/S1	T1/S2	T2/S3	T1/S2
	S2	T0/S0	T4/S1	T0/S2	T4/S1	T2/S4
	S3	T0/S0	T3/S1	T3/S1	T0/S3	T1/S3
	S4	T0/S0	T3/S2	T3/S2	T4/S3	T0/S4

ET-Axis 2 State S1	ET-Axis 1 State S1	
0	0	S4
0	1	S3
1	0	S2
1	1	S1

State Detection



PFIS Etalon State Diagram

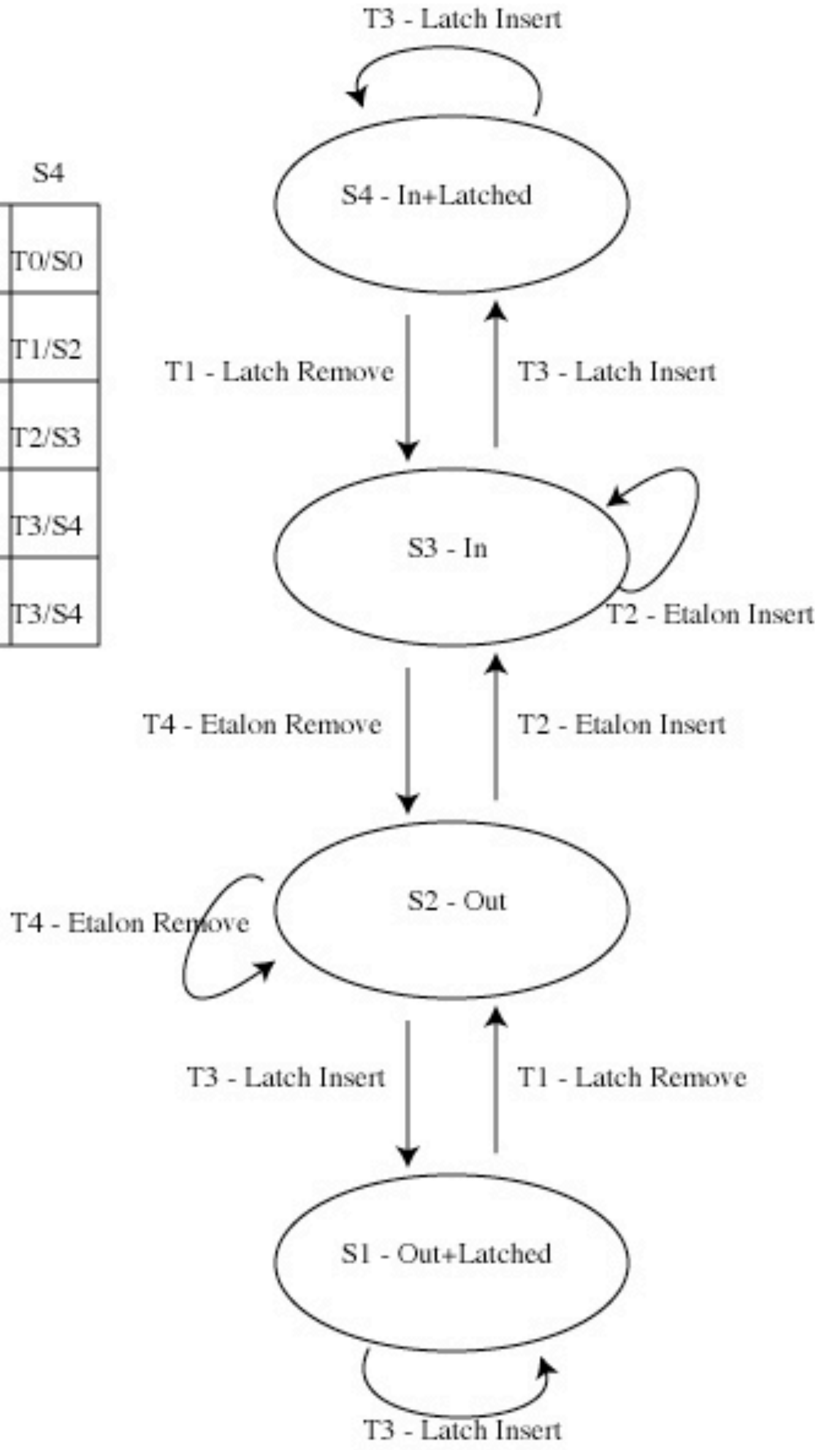
09-Etalon-Axis: moves a single etalon

et-axis-t1: remove latch (timeout 5000 ms)
  action: set latch remove
  interlock: articulation detente caught
    AND articulation home
    AND grating removed
    AND grating rotate home
  indicator: latche removed
et-axis-t2: insert etalon (timeout 30000 ms)
  action: clear remove then set insert
  interlock: latch removed OR etalon inserted
  indicator: etalon inserted
et-axis-t3: insert latch (timeout 5000 ms)
  action: clear latch remove
  interlock: none
  indicator: latch inserted
et-axis-t4: remove etalon (timeout 30000 ms)
  action: clear insert then set remove
  interlock: latch removed or etalon removed
  indicator: etalon removed

		Desired State				
		S0	S1	S2	S3	S4
Current State	S0	T0/S0	T0/S0	T0/S0	T0/S0	T0/S0
	S1	T0/S0	T3/S1	T1/S2	T1/S2	T1/S2
	S2	T0/S0	T3/S1	T4/S2	T2/S3	T2/S3
	S3	T0/S0	T4/S2	T4/S2	T2/S3	T3/S4
	S4	T0/S0	T1/S3	T1/S3	T1/S3	T3/S4

Etalon Removed	Latch Inserted	
0	0	S3
0	1	S4
1	0	S2
1	1	S1

State Detection



PFIS Etalon Axis State Diagram

Grating

Grating constants:

- Magazine Steps per Turn=500 // stepper motor
- Rotate Steps per Turn=800 // stepper motor
- Magazine Pitch (mm)=4.000000 // of the worm gear
- Rotate Degrees per Turn=2.000000 // per turn of the stepper motor
- Rotate Encoder Counts per Turn=2048 // data sheet
- Encoder Constant (Volts/mm)=0.026247 // of the yo-yo encoder
- Magazine Station 1 (steps)=-630 // depends on mechanical alignment
- Magazine Station Width (mm)=30.000000 // as designed
- Magazine Home (volts)=4.960000 // measured
- Rotate Home Trim (steps)=50 // apply after a home sequence
- Magazine Station Trim (steps)=-40 // apply after a home sequence
- Station Hysteresis (steps)=-300 // station depends on direction of travel

10-Grating: selects, inserts, rotates, unrotates, and removes gratings

grating-t1: select (3 s per station, timeout 5000 ms)
    action: move motor to desired station
    interlock: etalon 1 removed
        AND etalon 1 latched
        AND etalon 2 removed
        AND etalon 2 latched
        AND grating insert relay off
        AND grating magazine break release relay on
        AND grating remove relay on
        AND grating removed on
    indicator: grating station on

The interesting feature about the move to station is the management of hysteresis. When the move is started, the target position is equal to the desired grating station plus the "station hysteresis", which is stored annoyingly enough in grating-t1, not grating-constants. After the move (station + hysteresis) is done, a 2nd move, just to station, is done.

grating-t2: insert (timeout 10000 ms)
    action: clear remove then set insert
    interlock: articulation home
        AND articulation detente caught
        AND etalon 1 removed
        AND etalon 1 latched
        AND etalon 2 removed
        AND etalon 2 latched
        AND (grating inserted OR (grating magazine station AND grating rotate

home))
    indicator: grating inserted
grating-t3: rotate to observing angle (timeout 5000 ms, not used)
    action: step motor to desired step count
    interlock: etalon 1 removed
        AND etalon 1 latched
        AND etalon 2 removed
        AND etalon 2 latched
    indicator: none

This move is similar to grating-t1. We first move a little more positive than desired, then back to the desired angle. The fudge factor is not in grating constants, nor is it a control on grating-t3. It appears as a constant on the diagram itself, with a comment "Mike says to go 1 deg more positive, then go to the desired angle"!

grating-t4: rotate home (timeout 5000 ms)
    action: perform seek home operation
    interlock: etalon 1 removed
        AND etalon 1 latched
        AND etalon 2 removed
        AND etalon 2 latched
    indicator: grating rotate home

There are two actions here:

1. A FlexMotion home seek on the home indicator.
2. A small absolute move to the "Rotate Home Trim" value, this one in grating-constants.vi.

grating-t5: remove (timeout 10000 ms)
    action: clear insert then set remove
    interlock: articulation home
        AND articulation detente caught
        AND grating magazine station
        AND (grating removed OR grating rotate home)
    indicator: grating removed
grating-t6: magazine home (timeout 5000 ms)
    action: perform a seek home operation
    interlock: same as grating-t1 above
    indicator: grating magazine home

There are two actions here:

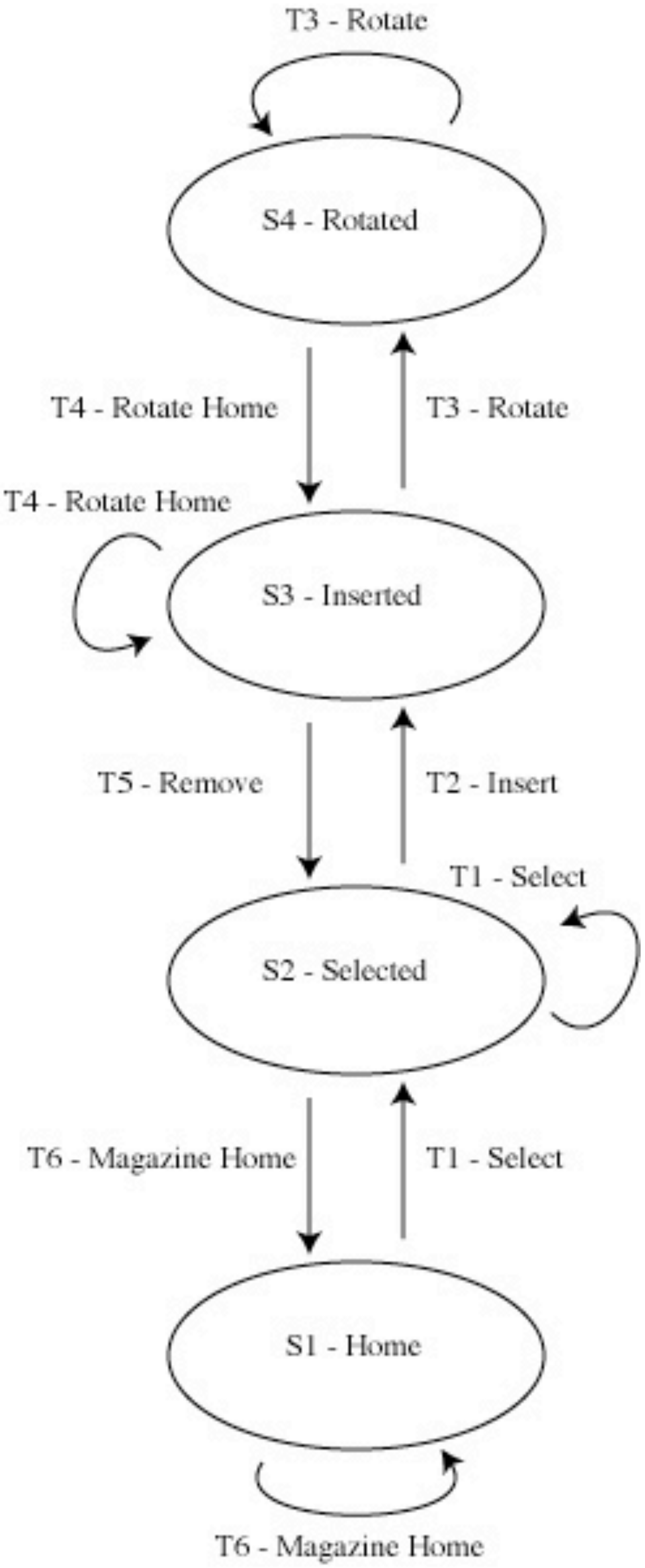
1. A FlexMotion home seek on the home indicator.
2. An abandoned move to the "Magazine Station Trim" value, fossilized in grating-constants.vi. A note on the diagram reads "Ignore the trim; the grating magazine axis is not detented and we have to creep to station anyway

See the special note on handling magazines at the end of the slitmask section above.

		Desired State				
		S0	S1	S2	S3	S4
Current State	S0	T0/S0	T0/S0	T0/S0	T0/S0	T0/S0
	S1	T0/S0	T6/S1	T1/S2	T1/S2	T1/S2
	S2	T0/S0	T6/S1	T1/S2	T2/S3	T2/S3
	S3	T0/S0	T5/S2	T5/S2	T4/S3	T3/S4
	S4	T0/S0	T4/S3	T4/S3	T4/S3	T3/S4

Inserted	
0	S2
1	S4

State Detection



PFIS Grating State Diagram

10-Grating-Magaxine-Axis: manages the magazine motor and brake

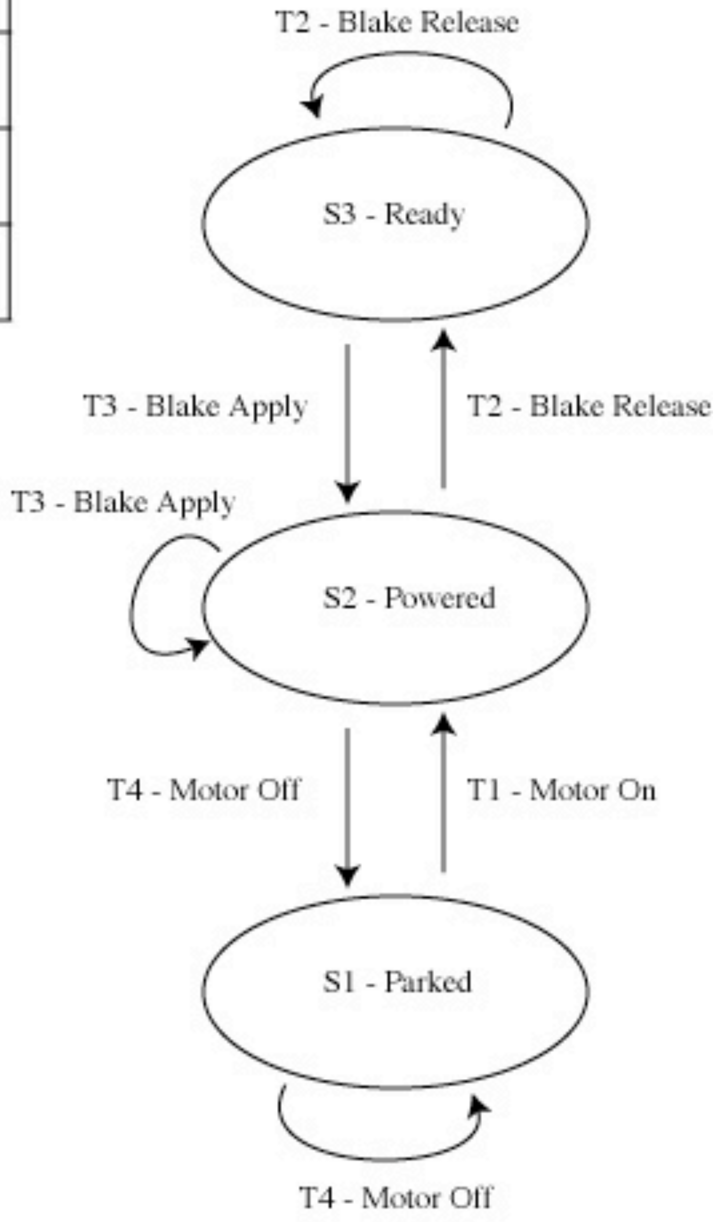
gm-axis-t1: motor on (timeout 5000 ms)
    action: clear grating magazine "All Windings Off"
    interlock: none
    indicator: motor power on
gm-axis-t2: release brake (timeout 5000 ms, not used)
    action: set brake release
    interlock: motor power on
    indicator: none
gm-axis-t3: apply brake (timeout 5000 ms, not used)
    action: clear brake release
    interlock: none
    indicator: none
gm-axis-t4: motor off (timeout 5000 ms)
    action: set grating magazine "All Windings Off"
    interlock: brake release relay off
    indicator: motor power off

Desired State

Current State	S0	S1	S2	S3
S0	T0/S0	T0/S0	T0/S0	T0/S0
S1	T0/S0	T4/S1	T1/S2	T1/S2
S2	T0/S0	T4/S1	T3/S2	T2/S3
S3	T0/S0	T3/S2	T3/S2	T2/S3

Blake Released	Motor Killed	
0	0	S2
0	1	S1
1	0	S3
1	1	S1?

State Detection



PFIS Grating Magazine Axis State Diagram

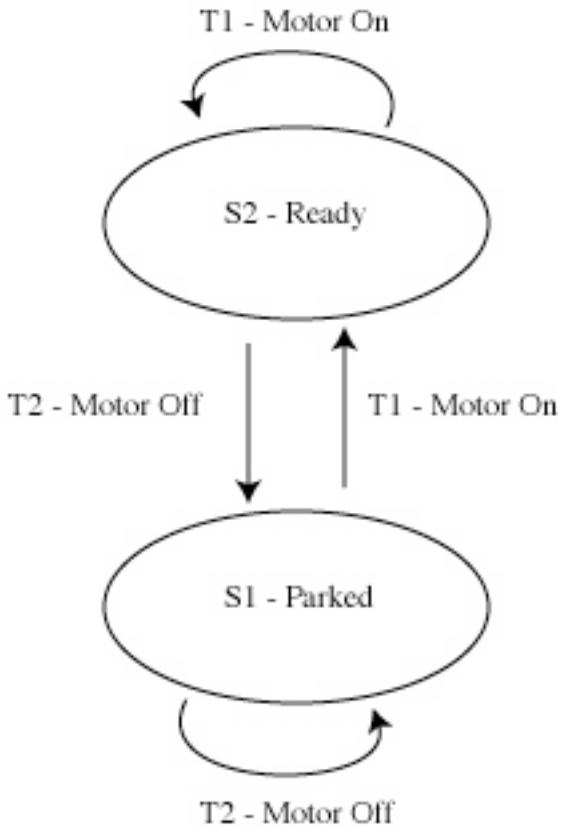
10-Grating-Rotation-Axis: manages the rotator motor

gr-axis-t1: motor on (timeout 5000 ms)
    action: clear grating rotator "All Windings Off"
    interlock: none
    indicator: motor power on
gr-axis-t2: motor off (timeout 5000 ms)
    action: set grating rotator "All Windings Off"
    interlock: none
    indicator: motor power off

		Desired State		
		S0	S1	S2
Current state	S0	T0/S0	T0/S0	T0/S0
	S1	T0/S0	T2/S1	T1/S2
	S2	T0/S0	T2/S1	T1/S2

Motor Killed	
0	S2
1	S1

State Detection



PFIS Grating Rotation Axis State Diagram

Beamsplitter:

Beamsplitter constants: none

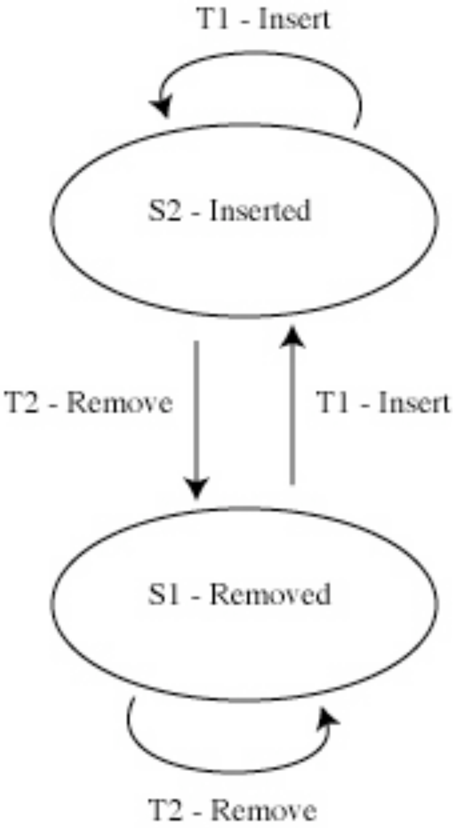
11-Beamsplitter: inserts and removes the beamsplitter

beamsplitter-t1: insert (timeout 15000 ms)
    action: set beamsplitter insert
    interlock: none
    indicator: beamsplitter inserted
beamsplitter-t2: remove (timeout 15000 ms)
    action: clear beamsplitter insert
    interlock: none
    indicator: beamsplitter removed

		Desired State		
		S0	S1	S2
Current State	S0	T0/S0	T0/S0	T0/S0
	S1	T0/S0	T2/S1	T1/S2
	S2	T0/S0	T2/S1	T1/S2

Inserted	
0	S1
1	S2

State Detection



PFIS Beamsplitter State Diagram

Filter

Filter constants:

- Steps per Turn=200 // the main drive motor
- Turns per Inch=20 // the worm gear
- Station Width (mm)=14.000000 as designed
- Encoder Constant (V/mm)=0.026247 // for the yo-yo encoder
- Station 1 (steps)=-818 // measured, depends on alignment
- Magazine Home (volts)=1.680000 // measured
- **Station Trim (steps)**=20 // measured, to get the alignment right

12-Filter: selects, inserts, and removes filters

filter-t1: select (5 s per station, timeout 5000 ms)
action: selects the desired filter station
interlock: filter removed
indicator: filter magazine station

There are 3 actions, similar to the slitmask:

1. A blind FlexMotion move to the calculated step value
2. A creep toward home
3. A creep away from home

The creep step sizes are on the panel of filter-creeper.vi.

filter-t2: insert (timeout 25000 ms)
action: set filter insert
interlock: filter magazine station
indicator: filter inserted

filter-t3: remove (timeout 25000 ms)
action: clear filter insert
interlock: none
indicated: filter removed

filter-t4: seek home (timeout 5000 ms)
action: perform a seek home operation
interlock: filter removed
indicator: filter magazine home

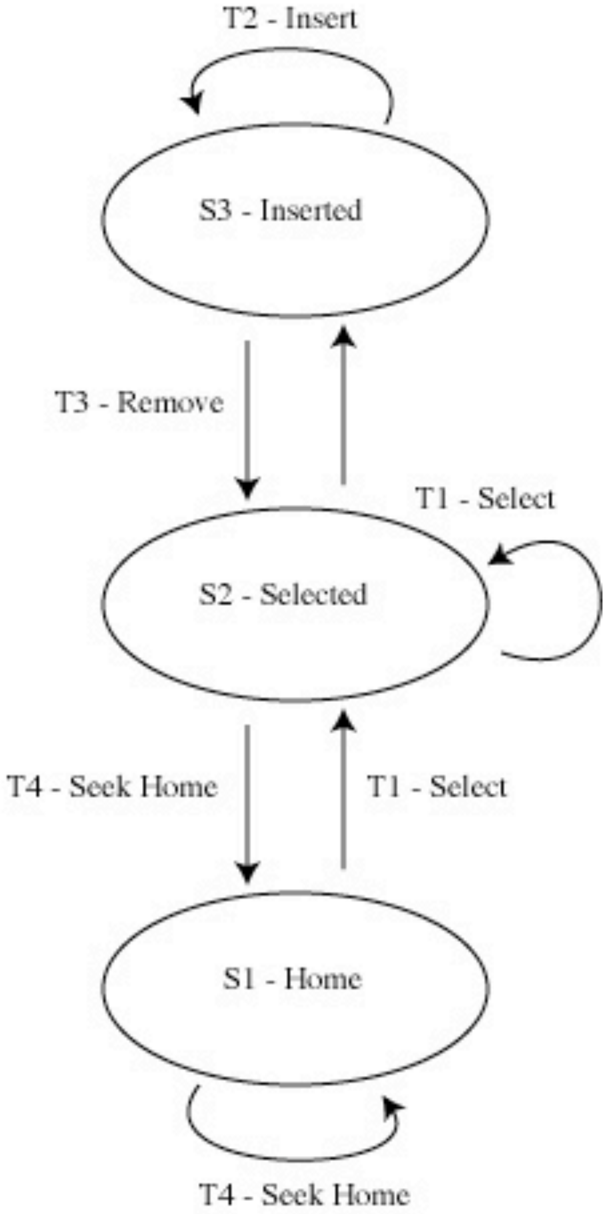
This is a simple FlexMotion home seek operation on the home indicator.

See the special note on handling magazines at the end of the slitmask section above.

		Desired State			
		S0	S1	S2	S3
Current State	S0	T0/S0	T0/S0	T0/S0	T0/S0
	S1	T0/S0	T4/S1	T1/S2	T1/S2
	S2	T0/S0	T4/S1	T1/S2	T2/S3
	S3	T0/S0	T3/S2	T3/S2	T2/S3

Inserted	
0	S2
1	S3

State Detection

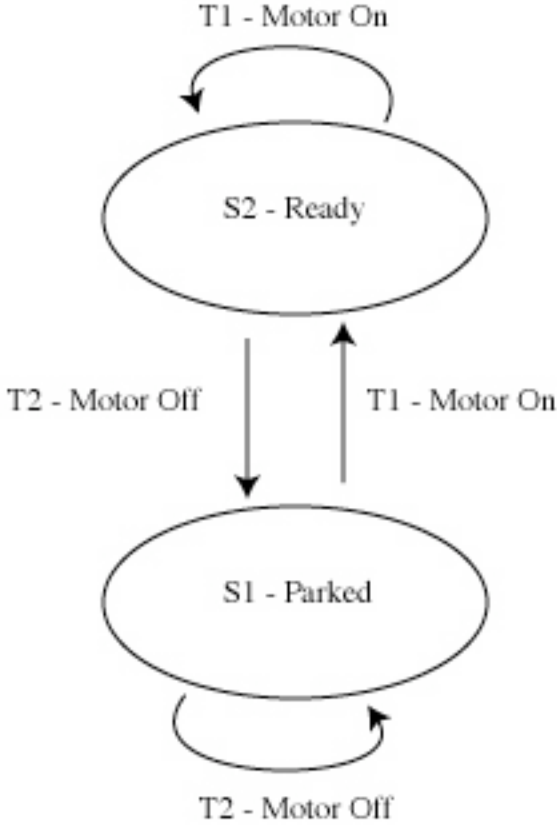


PFIS Filter State Diagram

12-Filter-Axis: manages the filter motor power

fi-axis-t1: motor on (timeout 5000 ms)
  action: clear filter "All Windings Off"
  interlock: none
  indicator: motor power on
fi-axis-t2: motor off (timeout 5000 ms)
  action: set filter "All Windings Off"
  interlock: none
  indicator: motor power off

		Desired State		
		S0	S1	S2
Current State	S0	T0/S0	T0/S0	T0/S0
	S1	T0/S0	T2/S1	T1/S2
	S2	T0/S0	T2/S1	T1/S2



Motor Killed	
0	S2
1	S1

State Detection

PFIS Filter Axis State Diagram

Articulation

Articulation constants:

- Steps per Turn=5000 // the stepper motor
- Gear Box Ratio=40 // data sheet
- Pinion Teeth=15 // the small drive gear
- Rack Teeth=200 // on the curved rail
- Station Spacing (deg)=0.750000 // as designed
- Station 1 Offset (deg)=1.750000 // as designed
- **Home Trim (steps)**=9 // measured to help alignment
- Encoder Counts per Turn=7684 // data sheet

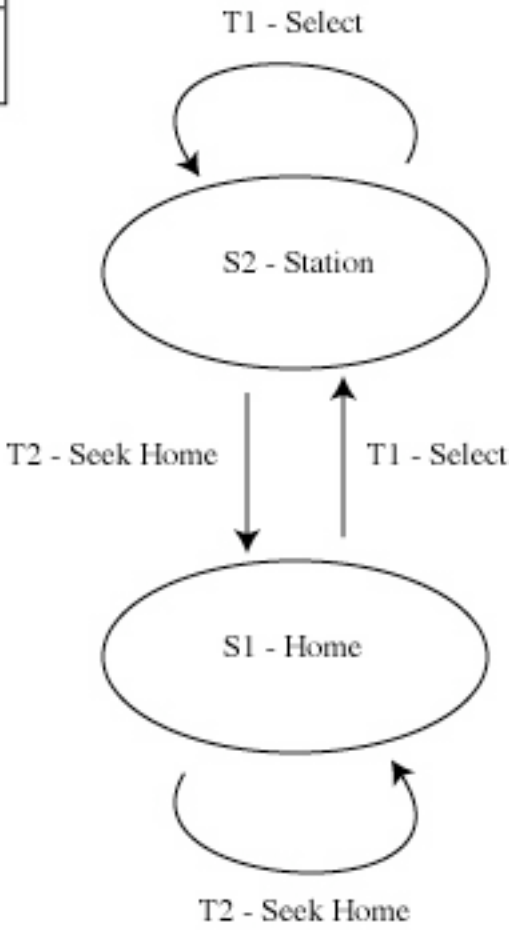
13-Articulation: selects the desired articulation station

articulation-t1: select observing station (2 s per station, timeout 5000 ms, not used)
    action: step to the desired station
    interlock: brake released
        AND detente removed
        AND (grating inserted OR grating removed)
    indicator: none
This just does a FlexMotion move to the calculated step count for the desired station.

articulation-t2: seek home (timeout 5000 ms)
    action: perform a home seek operation
    interlock: brake released
        AND detente removed
        AND (grating inserted OR grating removed)
    indicator: articulation home
This just does a FlexMotion home seek operation on the home indicator, and applies the articulation constant **Home Trim**.

		Desired State		
		S0	S1	S2
Current State	S0	T0/S0	T0/S0	T0/S0
	S1	T0/S0	T2/S1	T1/S2
	S2	T0/S0	T2/S1	T1/S2

Assume S2
State Detection



PFIS Articulation State Diagram

13-Articulation-Axis: manages the articulation brake, motor, and détente

A few words on the articulation detente operation. The articulation has evenly-spaced notches, and a detente that is driven into one of these notches to establish an observing position. The detente is not necessarily *exactly* lined up with the notch. The problem with trying to drive it all the way into the notch is that the motor, still powered, will fight the detente as it tries to seat itself. We can’t turn off the motor prior to the detente insertion, because the loss of torque would allow the articulated camera cradle to free-fall into a hard stop. The solution was to establish an intermediate position, “caught”, at which the detente is in far enough to prevent free-fall, but not so deep as to fight the motor. So we insert the detente just far enough to be “caught”, then we turn off the motor, and finally we push the detente in all the way. This final push will take out any slight misalignment with the notch, moving the cradle ever so slightly to establish the final observing position.

ar-axis-t1: release brake (timeout 5000 ms, not used)
  - action: set brake release
  - interlock: detente caught OR motor power on
  - indicator: none

ar-axis-t2: motor on (timeout 5000 ms)
  - action: clear articulation "All Windings Off"
  - interlock: none
  - indicator: motor power on

ar-axis-t3: remove détente (timeout 5000 ms)
  - action: set detente remove
  - interlock: motor power on
    - AND etalon 1 removed
    - AND etalon 1 latched
    - AND etalon 2 removed
    - AND etalon 2 latched
  - indicator: articulation detente removed

ar-axis-t4: insert détente (timeout 5000 ms)
  - action: clear detente remove
  - interlock: none
  - indicator: articulation detente caught

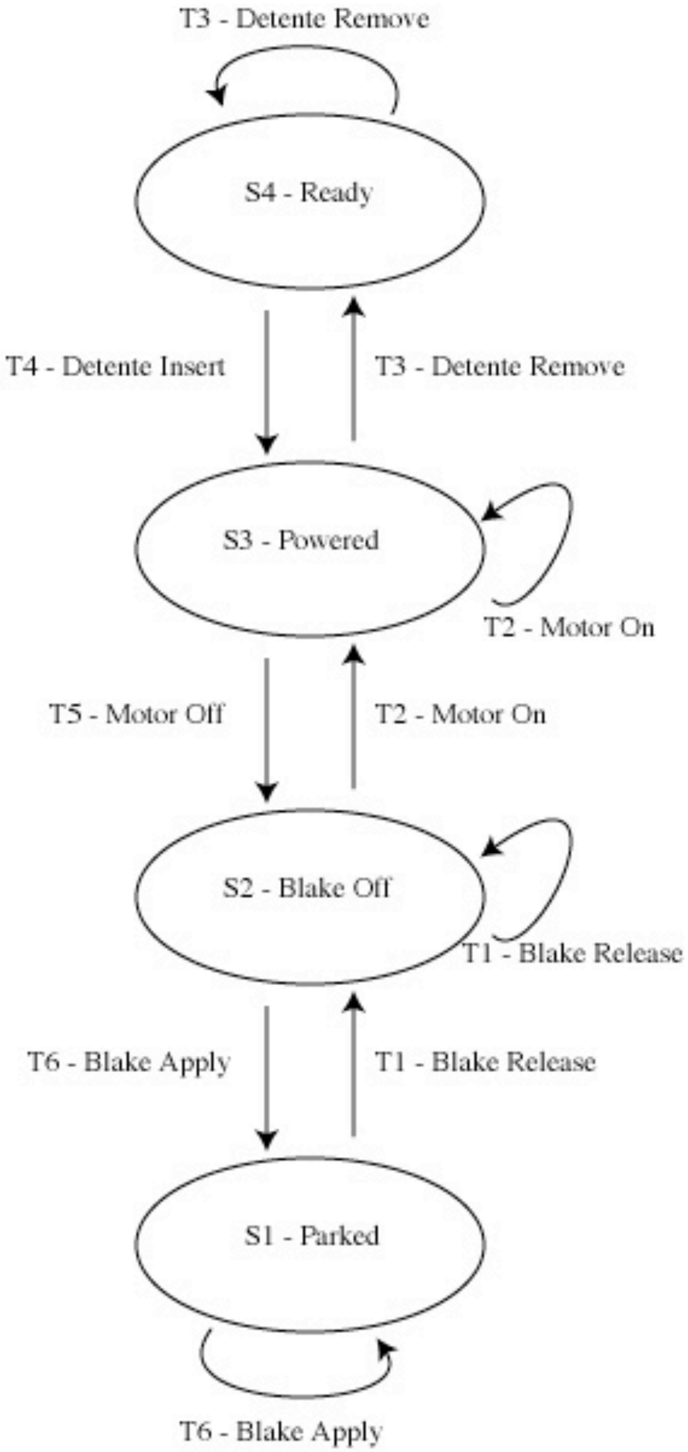
ar-axis-t5: motor off (timeout 5000 ms)
  - action: kill motor power
  - interlock: articulation detente caught
  - indicator: articulation all windings off AND articulation détente inserted

ar-axis-t6: apply brake (timeout 5000 ms, not used)
  - action: clear brake release
  - interlock: none
  - indicator: none

		Desired State				
		S0	S1	S2	S3	S4
Current State	S0	T0/S0	T0/S1	T0/S2	T0/S3	T0/S4
	S1	T0/S0	T6/S1	T1/S2	T1/S2	T1/S2
	S2	T0/S0	T6/S1	T1/S2	T2/S3	T2/S3
	S3	T0/S0	T5/S2	T5/S2	T2/S3	T3/S4
	S4	T0/S0	T4/S3	T4/S3	T4/S3	T3/S4

Detente Removed	Motor Killed	Blake Released	
0	0	0	S1?
0	0	1	S3
0	1	0	S1
0	1	1	S2
1	0	0	S1?
1	0	1	S4
1	1	0	S1?
1	1	1	S1?

State Detection



PFIS Articulation Axis State Diagram