

3 Pole Wiggler Control System

Dean Andrew Hidas

July 21, 2015

Abstract

This document is a description of the control system for the 3 Pole Wiggler. The device was originally designed by ADC and the first prototype built by them. The design was modified by BNL ID staff and the remainder produced in-house at BNL. The control system was “built” also in-house from scratch with involvement from the ID and Controls groups. This document describes the control system at the level of the delta tau, EPICS, and the CSS interfaces. It also describes actions to be taken in the event of a power cycle, kill switch, and emergency stop.

1 Limit, Kill, and Sensing Switches

Location and M-variable of all switches are given in table 1 and shown in figure 1.

1.1 Kill Switches

Kill switches just before the in and out hard stops physically cuts power to the motors, but does not relay information back to the controller. An amplifier fault will occur on an attempt to move again. As long as the kill switch signal is activated no motor motion is possible. You must turn by hand the drive until off of the switch. Once this is done you must power cycle the delta tau. You will see an amplifier fault if trying to move when the emergency stop has been reset but delta tau not reset.

1.2 Emergency Stop

If the emergency stop is pressed power to the motors is physically cut. You must reset the emergency stop switch and power cycle the delta tau. No signal is sent back to the delta tau for an emergency stop and it is not reported in any GUI. You will see an amplifier fault if trying to move when the emergency stop has been reset but delta tau not reset.

1.3 Limit Switches

There are 2 limit switches, one for positive (in) and negative (out) limits. The delta automatically stops motion. Motion in the direction opposite of the limit switch is allowed. Both switches have indicator lights on the GUI.

Location	M-Variable	Action
Beampipe Upper Upstream	M4	Motion Disable
Beampipe Upper Upstream	M5	Motion Disable
Beampipe Upper Upstream	M6	Motion Disable
Beampipe Upper Upstream	M7	Motion Disable
Negative Limit	MX	Motion Disable in direction of Limit
Positive Limit	MX	Motion Disable in direction of Limit
Negative Kill	N	Motor Power Cut
Positive Kill	N	Motor Power Cut
IN 1	MX	Notify
IN 2	MX	Notify
OUT 1	MX	Notify
OUT 2	MX	Notify

Table 1: Location and M-variable for all switches. Switches which do not report back to the delta tau are indicated with an N in the M-variable column.

Figure 1: Location of all switches. Limit switches are circled in blue, IN and OUT in white, kill in red, and beampipe in purple.

1.4 IN and OUT Switches

There are 2 IN indicator switches and 2 OUT indicator switches. The gui will indicate IN or OUT based on an “or” of the two for each.

1.5 Beampipe Switches

There are 4 beampipe sensing switches. If any one of these switches is depressed the device is dangerously close to the beampipe and may have already caused damage. All motion is stopped. This cannot be overridden. You will need to crank it off of the beampipe yourself. An indicator will light on the operator GUI indicating “Beampipe Hit”. On the engineering GUI you will see each individual sensor. The position and M-Variables for each is given in table 1.

2 CSS Interfaces

The 3PW CSS interfaces live in mercurial under boy/ID/3PW at <https://code.nsls2.bnl.gov/hg/>

2.1 Operator Interface

The operator interface allows for only basic operation. It consists of 2 buttons: IN and OUT. The IN and OUT buttons will move the device in and out of the beam to predefined

locations which are also reported on the gui. The values of these locations can be changed only at the level of the delta tau and should only be done by an expert. The IN and OUT functionality is disabled at the EPICS record level if homing is not complete.

On this display there is also an indicator panel of lights as shown in figure ??.

2.2 Engineering Interface

In addition to what the operator interface allows the Engineering interface also allows one to do the following

- Input a desired position directly to the motor record. This functionality is disabled at the EPICS record level if homing is not complete.
- Stop and Kill motor motion, the difference between the two being that Stop is a slow deceleration and Kill is fast. Both are safe for this system.
- Enable/Disable the encoder loss override flag. Disabling this will allow movement (via another expert gui: asynRecord/pmacStatusAxis) without encoder feedback. This should only be done by a PMAC expert as it requires changing the position and velocity feedback addresses in the controller.
- Open the nsls2 standard CSS pages for the records: motor, motorstatus, pmacStatus, pmacStatusAxis, and asynRecord.

3 EPICS

3.1 Standard Records

The following are taken from the standard NSLS2 repository:

- motor.db
- motorstatus.db
- pmacStatus.db
- pmacStatusAxis.db
- asynRecord.db
- pmac_asyn_motor.db

3.2 Specific 3PW Records

ThreePW.db

4 Controller Software - PMAC

4.1 PLCs

4.1.1 PLC 2 - Stepper Motor

This PLC enables the motor as a stepper motor safely according to the manual specification. Sets axis to stepper mode, clears errors, and write protects against strobe word.

4.1.2 PLC 3 - Encoder Loss

The encoder loss PLC monitors the presence of the encoder feedback. If this feedback is lost all motion is halted via the Kill command. Further amplifier enables are blocked by this PLC while the encoder feedback is not live. An additional flag accessible from EPICS allows an expert to override this blocking mechanism. The Encoder loss status as well as the override status are reported in the operator and engineering CSS pages. One can toggle the override bit only from the engineering page. Encoder loss is reported in P301 and the override bit in P300.

4.1.3 PLC 5 - CPU Load

This PLC is taken directly from the NSLS2 repository tpmac and used as is. It is used to monitor CPU and PLC activity for the pmacStatus EPICS records.

4.1.4 PLC 6 & 7 - User Commanded IN and OUT

These two PLCs serve as the basic IN and OUT commanded position settings from the GUI. The positions are hard coded and should only be changed by an expert. These PLCs check for the encoder loss bit and check the homing status bit, then set the desired position to M172, and finally execute a J=* command.

4.1.5 PLC 10 - Homing

This PLC is taken directly from the NSLS2 repository tpmac (written by Wayne Lewis) and used as is. It performs the following routine:

- Move to the limit switch being used as a reference.
- Perform a homing move away from the limit switch until it deactivates.
- Return to the starting position.

4.1.6 PLC 11 - Beampipe Sensors

This PLC monitors the beampipe sensor switches and issues a kill command if any switch is depressed and motor is enabled. This prevents any motion as long as any of these switches is depressed. There is no override functionality for this.