EPICS Training 2015
Using Motors

Ronald L. Sluiter

2015-02-16

The Advanced Photon Source is an Office of Science User Facility operated for the U.S. Department of Energy Office of Science by Argonne National Laboratory

Overview

- Acknowledgements.
- What's in the EPICS motor module and what's it for?
- Supported motor controller Manufacturers and models.
- Motor record features.
- Configuration example.
- Feedback.
- Retries.
- Backlash Correction.
- Distribution.

Note: All of following refers to R6-9 of the motor distribution.

Acknowledgments

- Jim Kowalkowski - original author, until…
- Tim Mooney and Joe Sullivan, until…
- Myself, since 1998/9, until…
- Kevin Peterson and myself are the current “module owners”.
- Many contributors to the motor module.
  - Mark, BCDA, Diamond Light Source, vendors

Terminology

- MM ⇔ Motor Module distribution
- MR ⇔ Motor Record
- EGU ⇔ Engineering Units
- RPS ⇔Revolutions Per Second
- Stepper motors *step* *and* encoders *tick.*
- Readback ⇔ Feedback
- *target position* - a user requested, static, absolute position (VAL).
- *commanded position* – the motor controller's current, dynamic, absolute reference position (RMP).

What's in the Motor module?

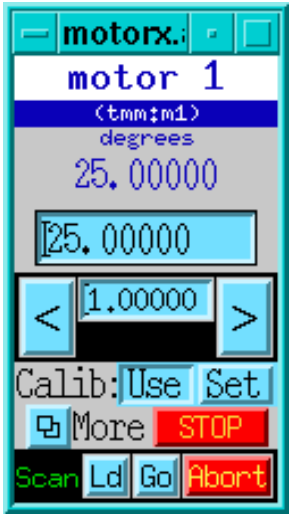
- User displays - medm, CSS/Boy, caQtDm (mostly commissioning, some application specific; e.g., fly scanning) (Location: <motor>/motorApp/op/*)
- Databases – (templates referenced by *.substitution files) (Location: <motor>/motorApp/Db)
- The MR (*.dbd and 4,000+ *.cc) and common device/driver code shared by all device/drivers (Location: <motor>/motorApp/MotorSrc)
  - 3 different motor module device/driver architectures.
  - Model 1 (unique device, used asyn for comm.), Model 2 & 3 (common device, asyn integral)
  - https://subversion.xray.aps.anl.gov/synApps/motor/trunk/documentation/motorDeviceDriver.html
- Vendor specific directories, device/driver code and *.dbd (README docs) (Location: <motor>/motorApp/<vendor-name>Src).
- Save/restore *.req files (Location: <motor>/motorApp/Db).

What's in the Motor module?

- Two IOC build examples (w/ and w/o Asyn) with motor device/drivers (build option) (Location: <motor>/motorExApp)
- iocBoot examples (controller specific st.cmd and supporting *.cmd files) (build option) (Location: <motor>/iocBoot/*)
- Documentation (release notes, motor record and device/driver info) (Location: <motor>/documentation)

What's the Motor module for?

- Device independence – motor hardware is transparent to users.
  - Same user displays and motor motion behavior, for all devices.



Available operations from this display;

1. Make absolute or incremental moves.
2. Define the current position.
3. Stop the current move.

without any controller specific information.

- Common device and driver level software for all supported controllers.
- Unsupported MR features are handled at the device/driver level; typically, by ignoring the request (the price - LCD motor record features).

Supported Manufacturers

1. Pro-Dex (OMS)
2. Newport
3. Schneider Electric
4. Advanced Control Systems
5. Mclennan
6. Physik Instrumente (PI)
7. MicroMo
8. Micos
9. Faulhaber
10. Parker Hannifin
11. New Focus
12. ACS Motion Control
13. Spectra-Physics
14. Thorlabs
15. Animatics
16. piezosystem
17. Kohzu
18. attocube
19. Aerotech
20. Hytec
21. nPoint
22. Micronix
23. SmarAct
24. Phytron

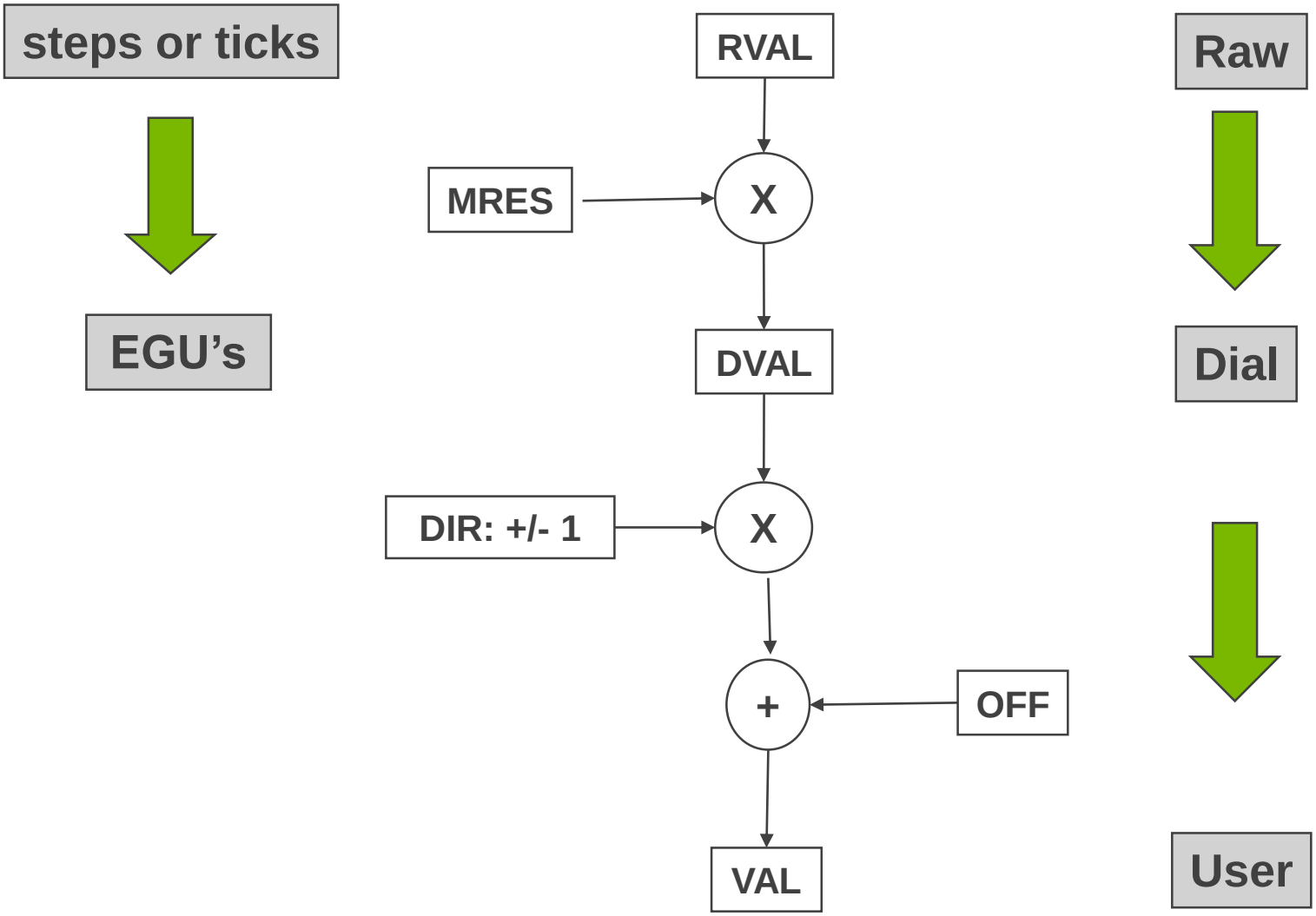
Does motor controller XYZ have EPICS support?

- Motor record document.
  - http://www.aps.anl.gov/bcda/synApps/motor/R6-9/motorRecord.html
  - MM location: <motor>/documentation/motorRecord.html
- EPICS Hardware Support: by Manufacturer
  - http://www.aps.anl.gov/epics/modules/manufacturer.php
  - Not all controller support is distributed in the MM; e.g., Galil
- EPICS tech-talk (search - ask)
  - http://www.aps.anl.gov/epics/tech-talk/index.php
  - Search for “Delta Tau Turbo PMAC2 VME Ultralight”
    - http://www.aps.anl.gov/epics/tech-talk/2014/msg01793.php
- Support for stepper, servo, piezo (deflection and slip-stick) motors.

Features - scope

- The scope of the MR is limited to single axis, non-coordinated, point to point moves.
- For multi-axis coordinated motion and trajectory scans see…
  - https://subversion.xray.aps.anl.gov/synApps/motor/trunk/documentation/trajectoryScan.html

Features - coordinate systems.



Features - coordinate systems.

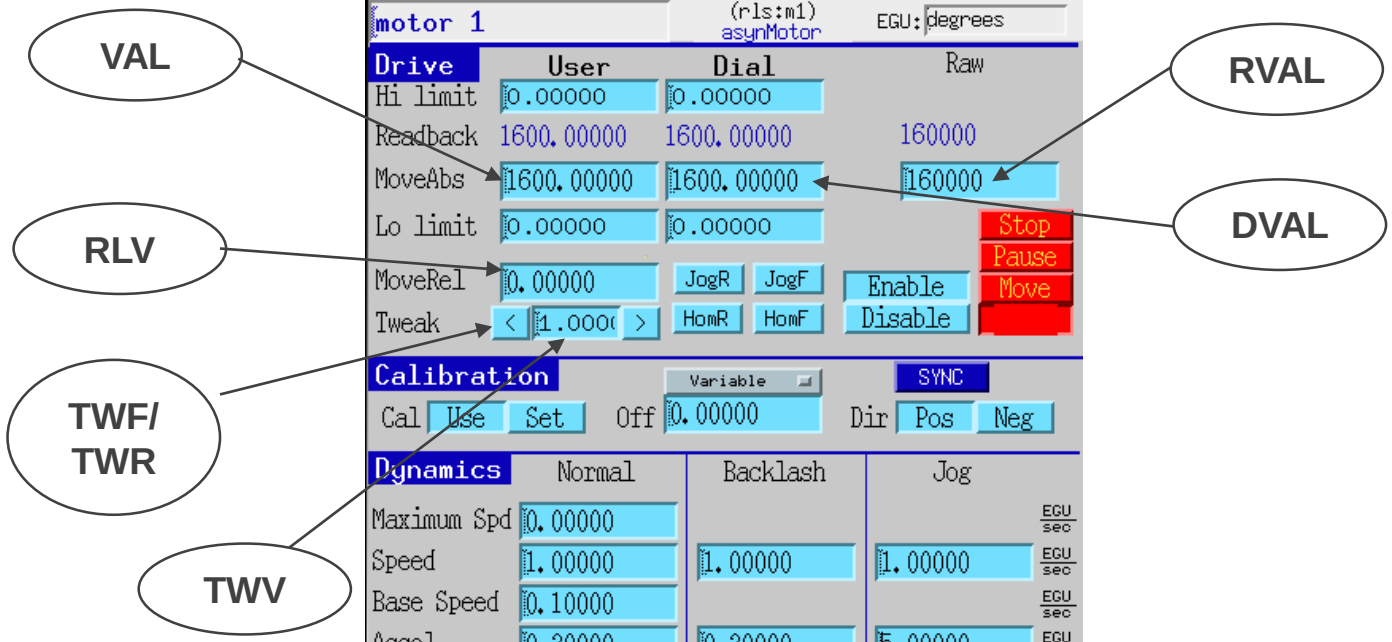
- Some controllers use raw units to communicate.
  - The MR communicates to device support in raw units.
  - Position, velocity and acceleration MR commands to device support are in steps, steps/sec and steps/sec^2.
- Other controllers communicate in engineering units (EGU's); e.g., inch, mm, degrees. Device support handles the conversion.
- Motor resolution field (MRES) converts raw values to dial values.
  - **Convert raw target position (RVAL) to dial target (DVAL):**

    DVAL [EGU's] = RVAL [steps] * MRES [EGU's/step]
- User units are based on dial units, the User Direction field (DIR) and the User Offset field (OFF)
  - **Convert dial target position (DVAL) to user target (VAL):**

    VAL = (DVAL * DIR) + OFF; where DIR = +/- 1.

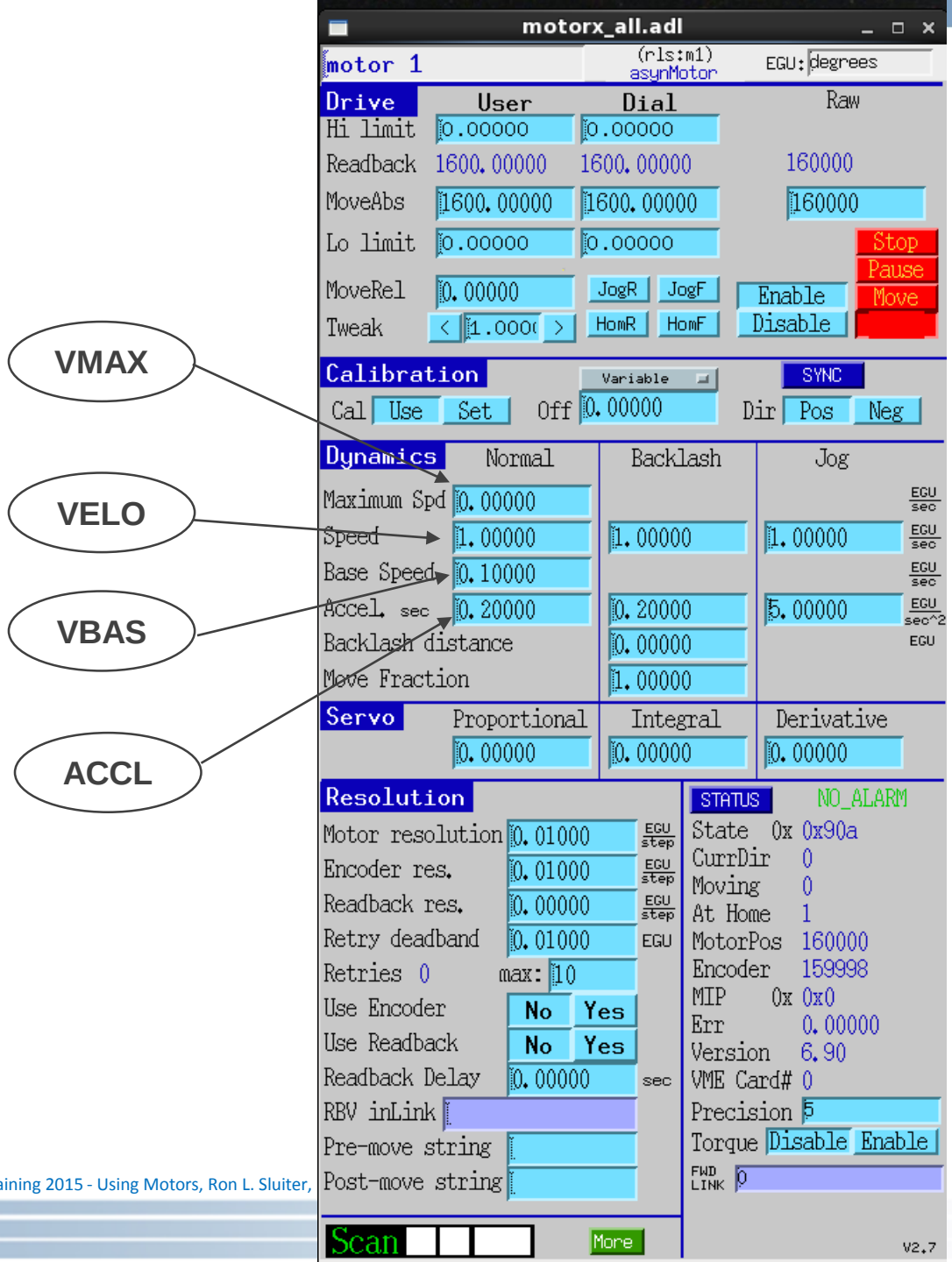
Features – Move types

- Absolute (VAL, DVAL, RVAL)
- Relative (RLV)
- Incremental (TWF, TWR, TWV)


VAL
RLV
TWF/
TWR
TWV
RVAL
DVAL
motorx_all.adl
motor 1
(rls:m1)
asynMotor
EGU: degrees
Drive
User
Dial
Raw
Hi limit
0.00000
0.00000
Readback
1600.00000
1600.00000
160000
MoveAbs
1600.00000
1600.00000
160000
Lo limit
0.00000
0.00000
Stop
Pause
Move
MoveRel
0.00000
JogR
JogF
Enable
Tweak
1.000
HomR
HomF
Disable
Calibration
Variable
SYNC
Cal
Use
Set
Off
0.00000
Dir
Pos
Neg
Dynamics
Normal
Backlash
Jog
Maximum Spd
0.00000
Speed
1.00000
1.00000
1.00000
Base Speed
0.10000
Accel. sec
0.20000
0.20000
5.00000
Backlash distance
0.00000
Move Fraction
1.00000
Servo
Proportional
Integral
Derivative
0.00000
0.00000
0.00000
Resolution
Motor resolution
0.01000
Encoder res.
0.01000
Readback res.
0.00000
Retry deadband
0.01000
Retries 0
max: 10
Use Encoder
No
Yes
Use Readback
No
Yes
Readback Delay
0.00000
sec
RBV inLink
Pre-move string
Post-move string
STATUS
NO_ALARM
State 0x 0x90a
CurrDir 0
Moving 0
At Home 1
MotorPos 160000
Encoder 159998
MIP 0x 0x0
Err 0.00000
Version 6.90
VME Card# 0
Precision 5
Torque
Disable
Enable
FWD LINK
0
Scan
More
V2.7

Features - Velocity and Acceleration

- Maximum Velocity (VMAX).
- Slew Velocity (VELO)
- Base Velocity (VBAS) – for stepper motors only to minimize resonance.
- Slew Acceleration time (ACCL [sec]) – acc. rate [step/s^2] = (VELO - VBAS) / ACCL



VMAX

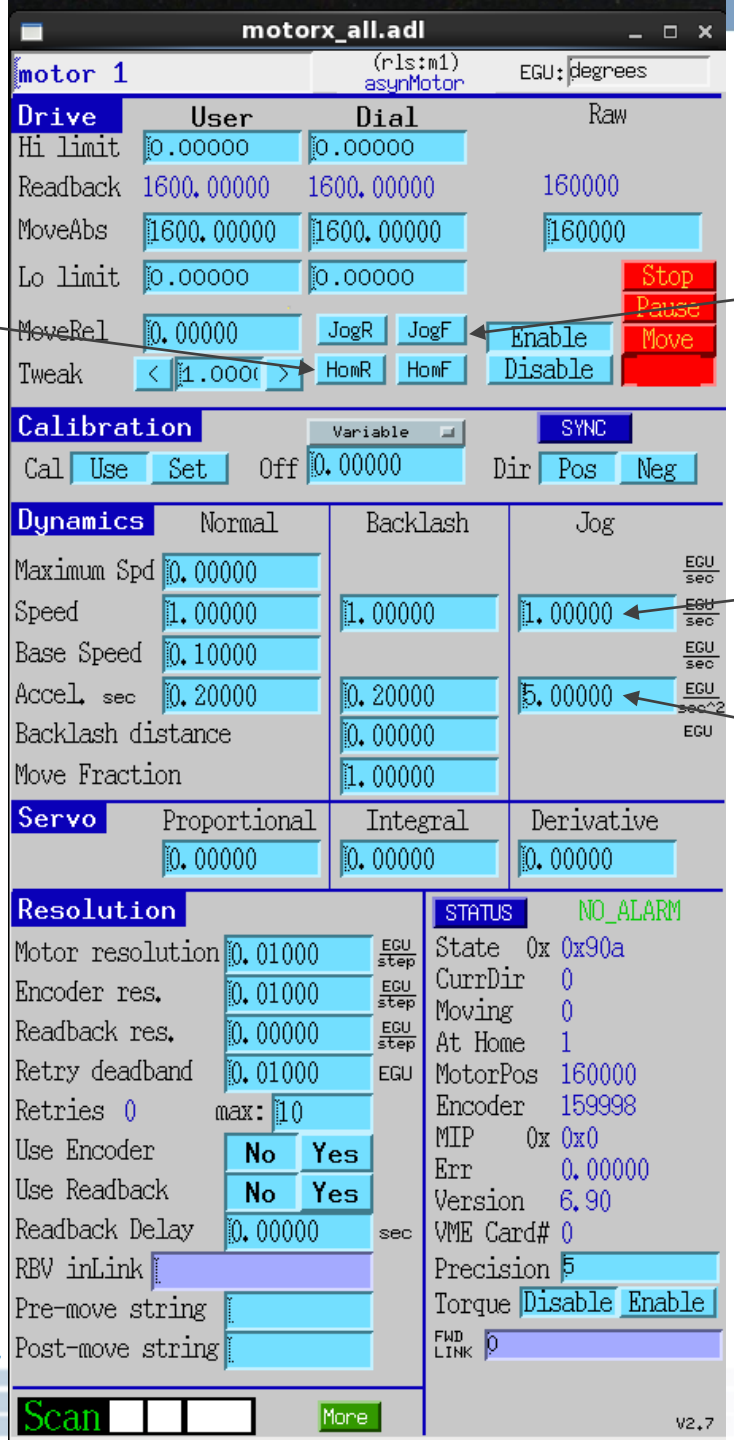
VELO

VBAS

ACCL

Features - Homing, Jogging

- Home search (HOMF, HOMR, HVEL).
- Jogging (JOGF, JOGR, JVEL, JAR).
- Safety issues
  - Commissioning a stage? Check hardware limit switches 1st.
  - pinch hazards, remote operation, servo systems



HOMF/HOMR

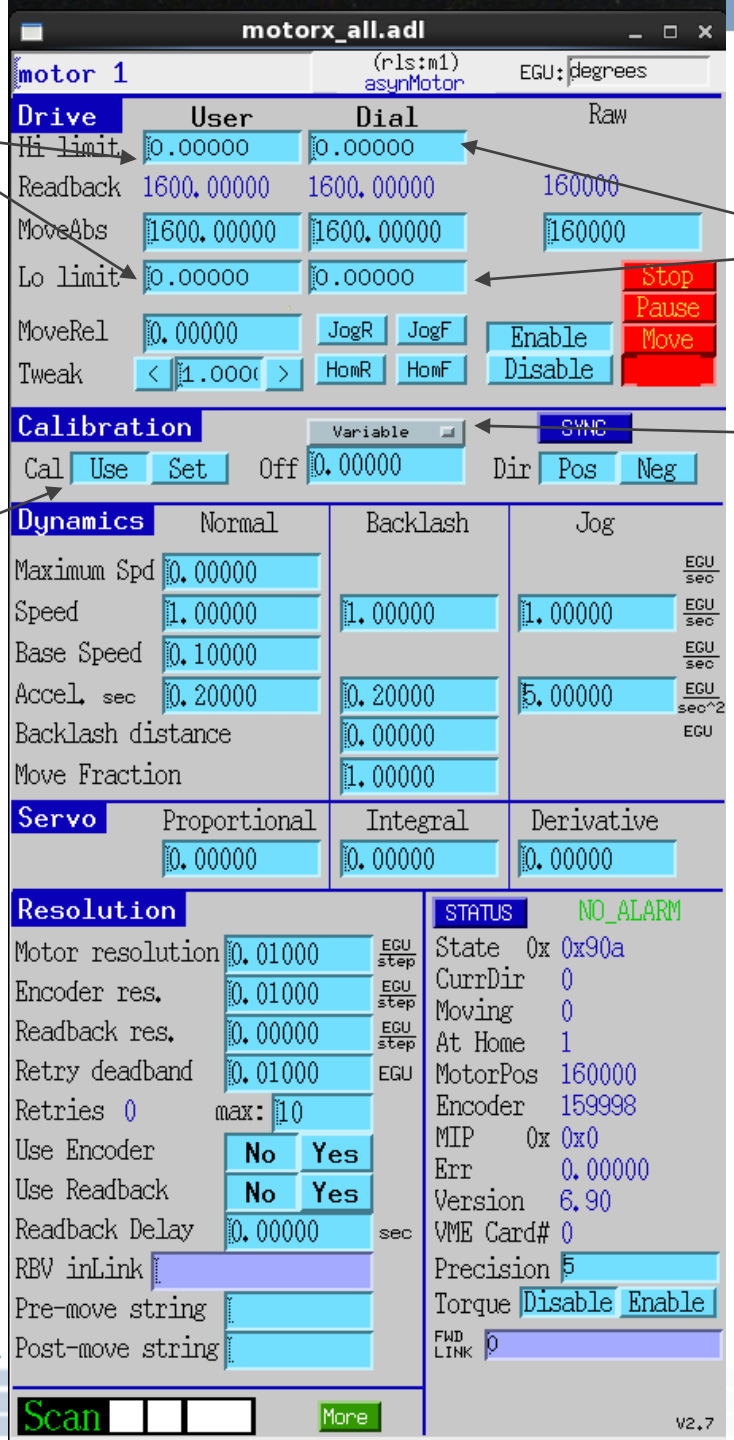
JOGF/JOGR

JVEL

JAR

Features – Set position, soft travel limits

- Define current position (SET, FOFF).
- Software travel limits (HLM, LLM, DHLM, DLLM) (disabled if DHLM = DLLM = 0)



HLM/LLM

DHLM/DLLM

FOFF

Set

Features

- Send motor controller command primitives at initialization, pre-move and post-move; (INIT, PREM, POST).
- Drive Power Monitoring and Motor Synchronized DB Puts via Device Directives.
- User initiated updates via the Status Update field (STUP).
- A few fields are motor type specific;
  - PID parameters (PCOF/ICOF/DCOF) for servo motors (normalized to 0 -> 1).
  - Velocity base (VBAS) for stepper motors.
  - Done Moving Input Link (DINP) for Soft Channel.
- Position feedback from either a EPICS PV link or a motor controller.

Configuration example

- **Set the engineering units field (EGU); inch, mm, degrees .**
- **Set the motor resolution (MRES) field which is in units of …**
  - (EGU's / motor step) for stepper motors.
  - (EGU's / encoder tick) for servo motors.

Cont'd Configuration example

- What should MRES be set to?
- For a simple linear stage with a leadscrew directly driven by a stepper motor.

$$\mathrm{MRES} = \left(\frac{\mathrm{EGU's}}{1\,\mathrm{leadscrew\ rev.}}\right) \times \left(\frac{1\,\mathrm{motor\ rev.}}{\mathrm{motor\ steps}}\right)$$

Cont'd Configuration example

- UREV = EGU's / 1 leadscrew rev.
- SREV = motor steps / 1 motor rev. > 0
- MRES = UREV / SREV
- MRES and UREV allow negative values so that the record's coordinate system can be configured to the opposite polarity of the motor controller's.
- Never change MRES while the motor is moving.

Cont'd Configuration example

- Gear box example
  - **Gear ratio definition; "The ratio of the powered gear of a gear train to that of the final or driven gear." Hence, a 3:1 gear ratio means that the motor makes 3 revolutions per 1 revolution of the lead screw**

$$\mathrm{MRES} = \left(\frac{\mathrm{EGU's}}{1\,\mathrm{leadscrew\ rev.}}\right) \times \left(\frac{1\,\mathrm{leadscrew\ rev.}}{3\,\mathrm{motor\ revs.}}\right) \times \left(\frac{1\,\mathrm{motor\ rev}}{\mathrm{motor\ steps}}\right)$$

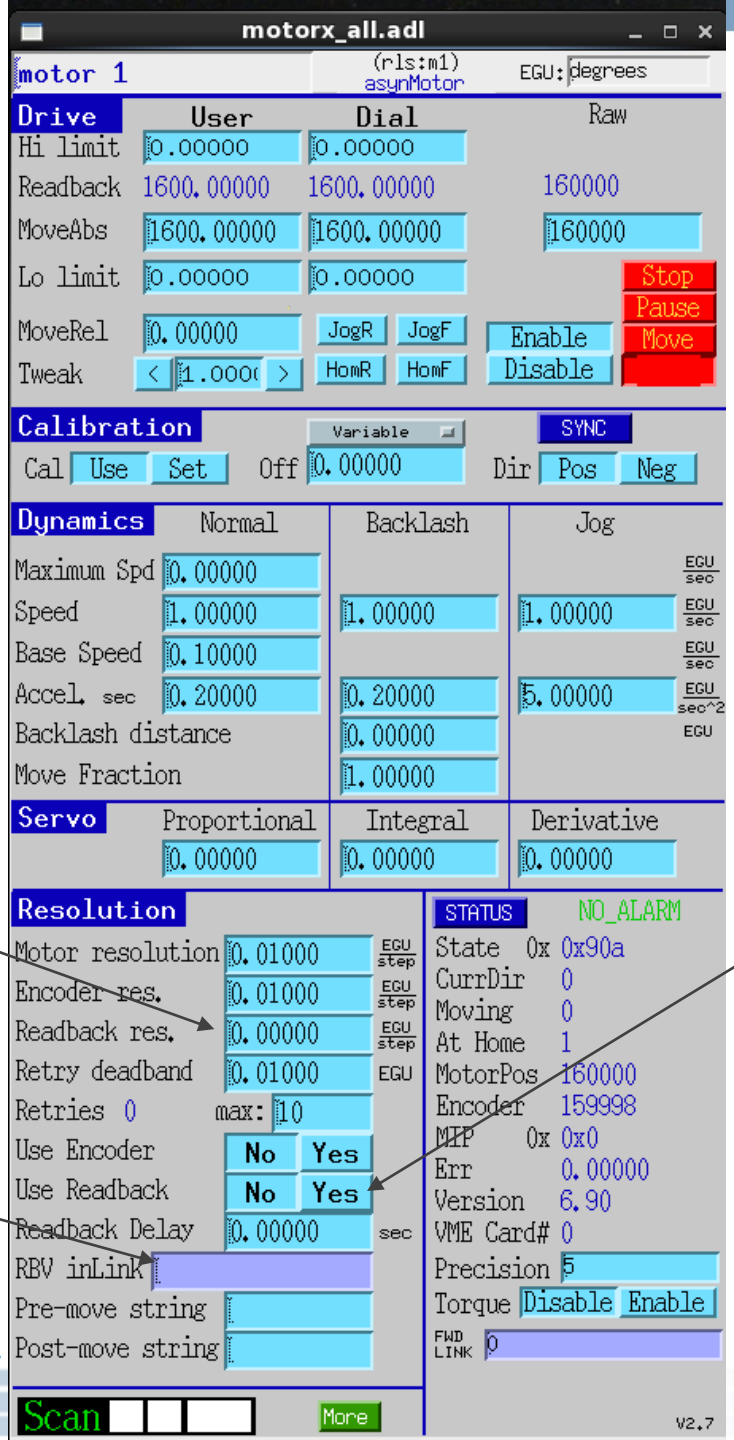
- **For a servo motor with a motor mounted encoder, change *motor steps* to *encoder ticks*.**

Feedback

- Two ways to input position feedback to the motor record:
  1. From driver support, via the motor controller.
  2. From an EPICS PV, via a stand-alone, feedback device,
- For feedback from driver support:
  - Driver level code sets the *Encoder is Present* bit (EA_PRESENT) in the Motor Status field (MSTA) to True at initialization.
  - The Raw Encoder Position field (REP) has the motor controllers' raw encoder value [ticks].
- Configuring feedback from driver support:
  - Set the Encoder Resolution field (ERES) to convert encoder ticks to EGU's.
  - The Use Encoder If Present field (UEIP) determines if the REP is used [Yes/No].

Cont'd Feedback

- Configuring feedback from an EPICS PV:
  - Set Readback PV link (RDBL).
  - Set the Readback Resolution (RRES) to convert the RDBL PV units to EGU's.
  - The Use Readback If Present field (URIP) determines if the RDBL PV link is used [Yes/No].
- Raw Motor Position (RMP) is the current commanded position read from the motor controller. If motor record closed-loop control is off (UEIP & URIP both set to *No*), then RVAL = RMP after every move.
- Raw Readback Value field (RRBV) can be in units of either steps or ticks.

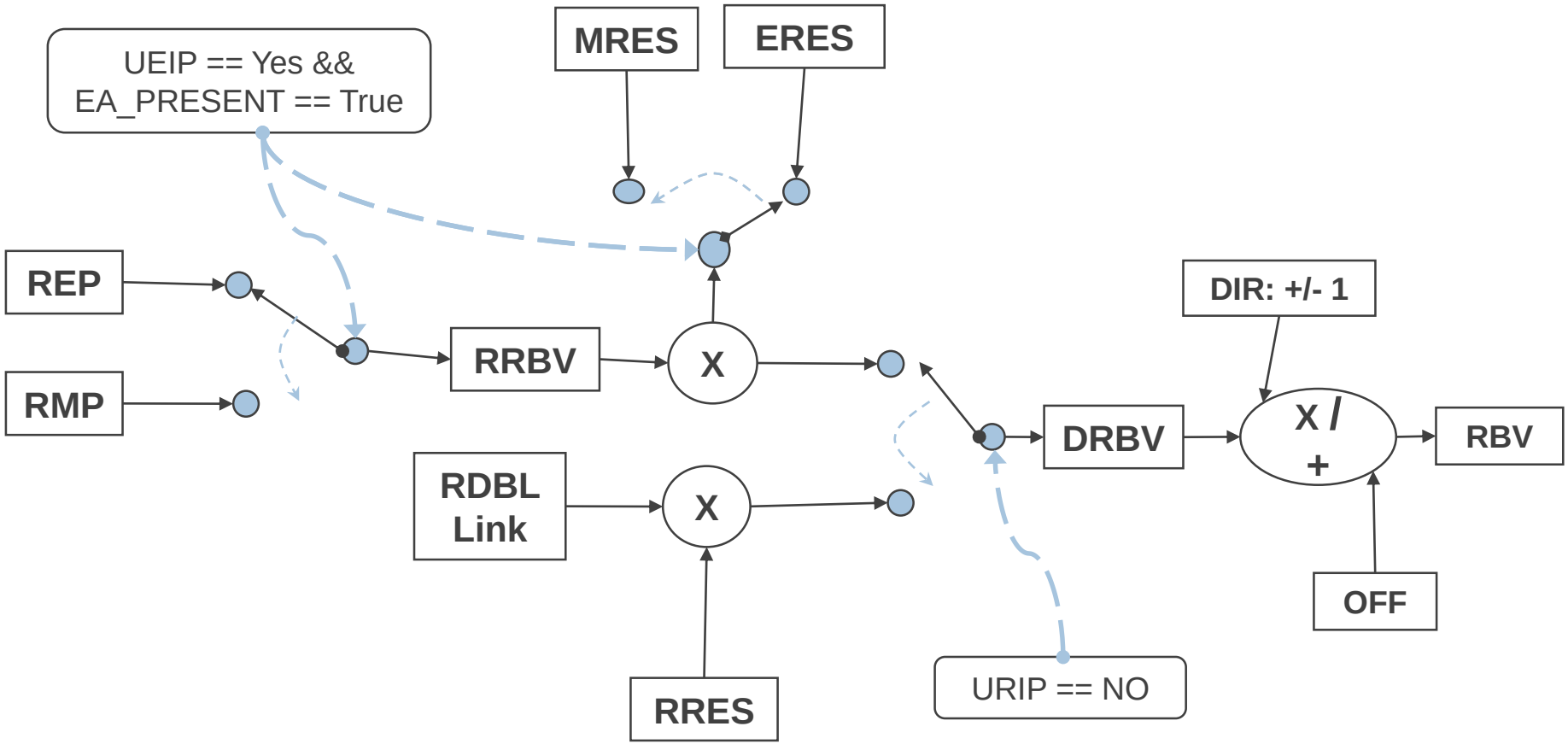


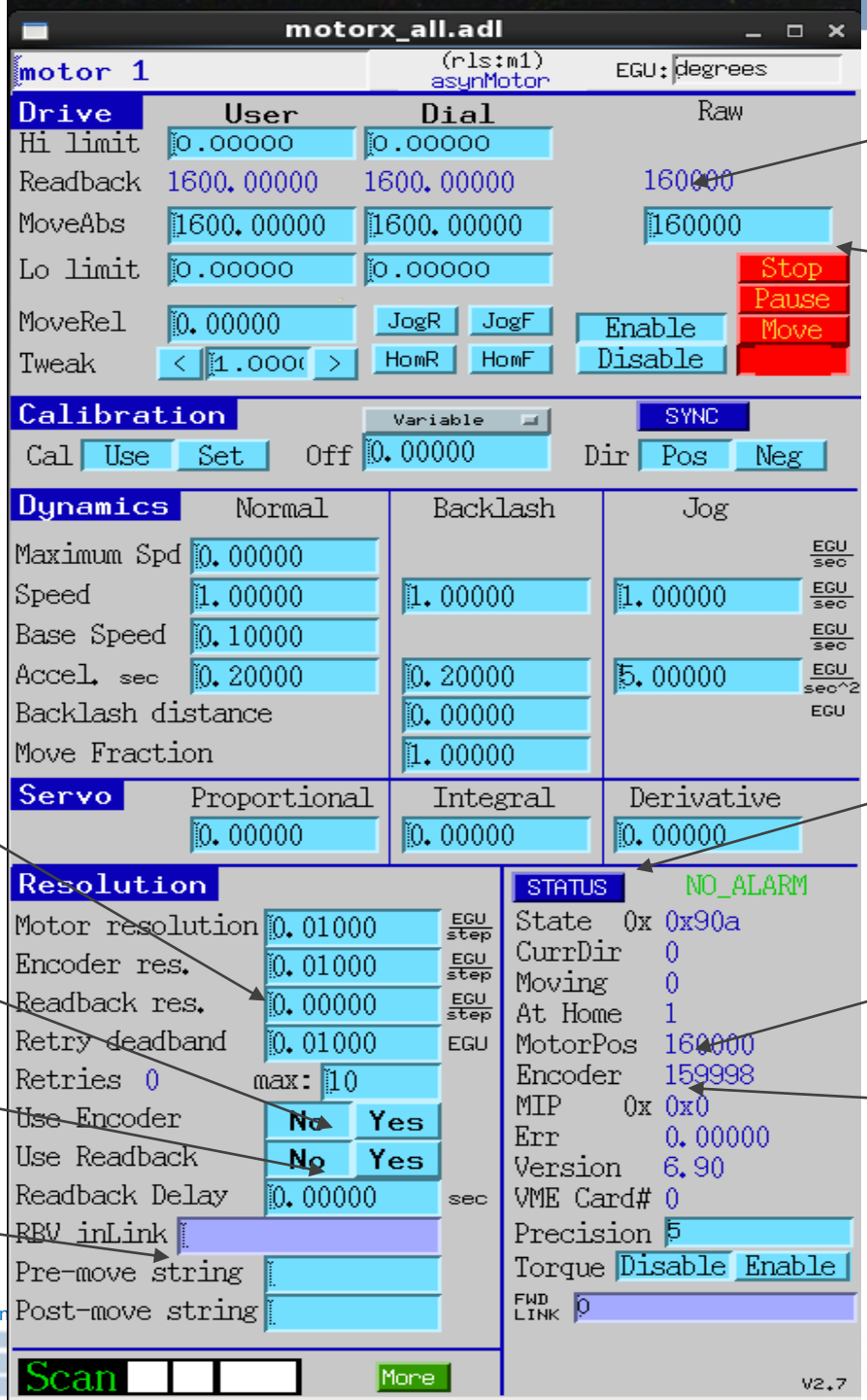
RRES

RDBL

URIP

Feedback data flow





RRBV

RVAL

RRES

STUP

UEIP

RMP

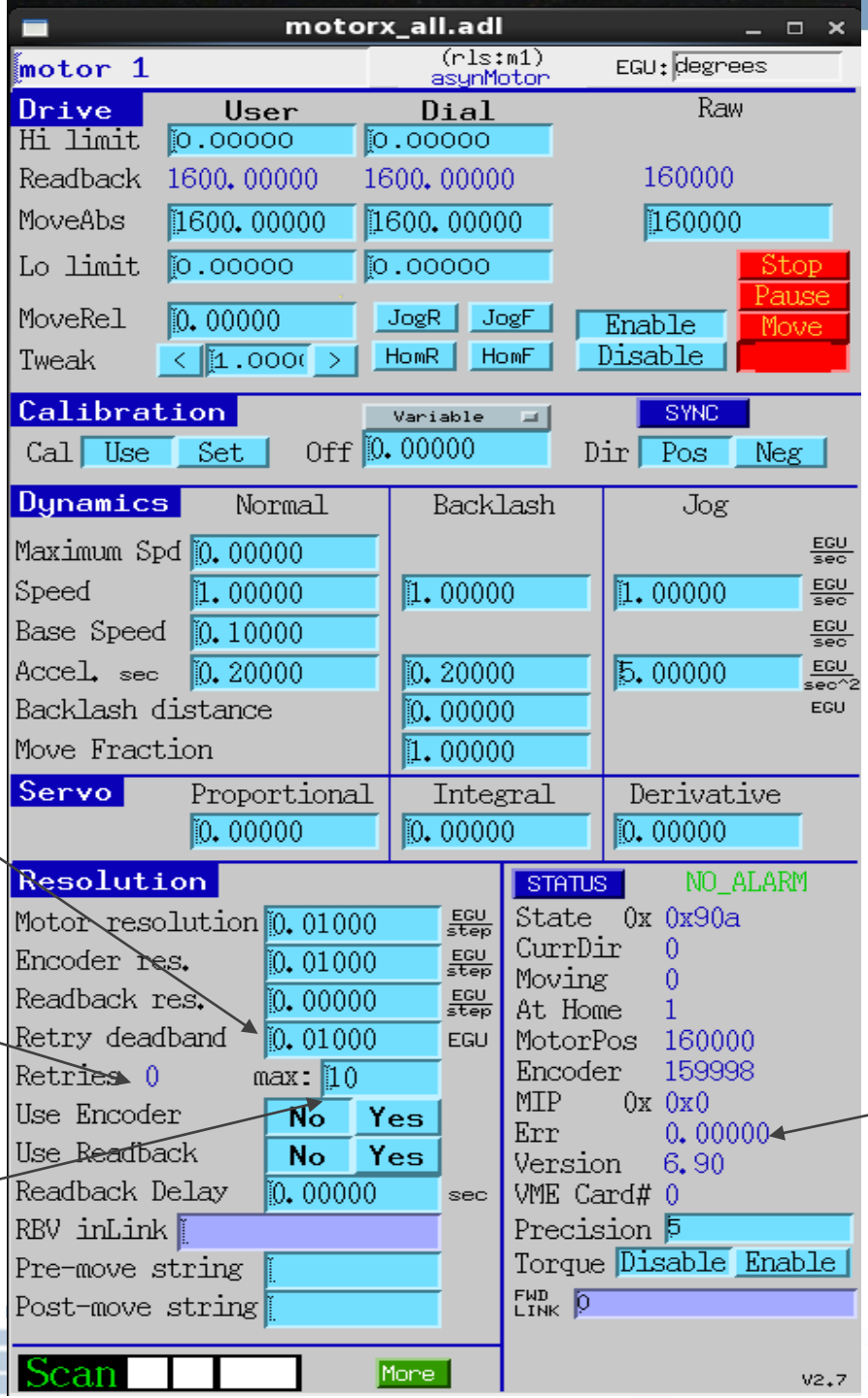
URIP

REP

RDBL

Retries

- What it isn't. Retries are not continuous, dynamic loop closure.
- What it is. Retries try to eliminate dial position error (DIFF) by making, consecutive, relative moves based on the DIFF field.
- DIFF = DVAL – DRBV
- If [RTRY !=0, AND, ((EA_PRESENT = True, AND, UEIP == Yes), OR, URIP == Yes)] is True, then all motor record moves are relative moves.
- Configuring retries:
  - Usually set UEIP == No for a servo motor.
  - Set Retry Deadband (RDBD); retry if (DIFF > RDBD).
  - Set Max Retry Count (RTRY) - maximum number of retries.
  - Current Retry Count (RCNT) – cleared at the start of every move.



RDBD

RCNT

RTRY

DIFF

Backlash Correction

- Backlash - lost motion due to mechanical imperfections.
- Backlash configuration:
  - Backlash distance (BDST) determines the magnitude and direction of the backlash move.
  - The backlash has its' own velocity (BVEL) and acceleration (BACC) parameters.
- Backlash correction algorithm:
  - Preferred direction - the sign of BDST.
  - Slew parameters - slew velocity (VELO) and acceleration (ACCL).
  - Backlash parameters - BVEL and BACC.

Backlash Correction Logic

- Given a move to a position called Target:

IF backlash is disabled (0 = |BDST| < |MRES|), OR, (move is in the preferred direction, AND, backlash parameters == slew parameters).

THEN, skip backlash, move to Target using slew parameters.

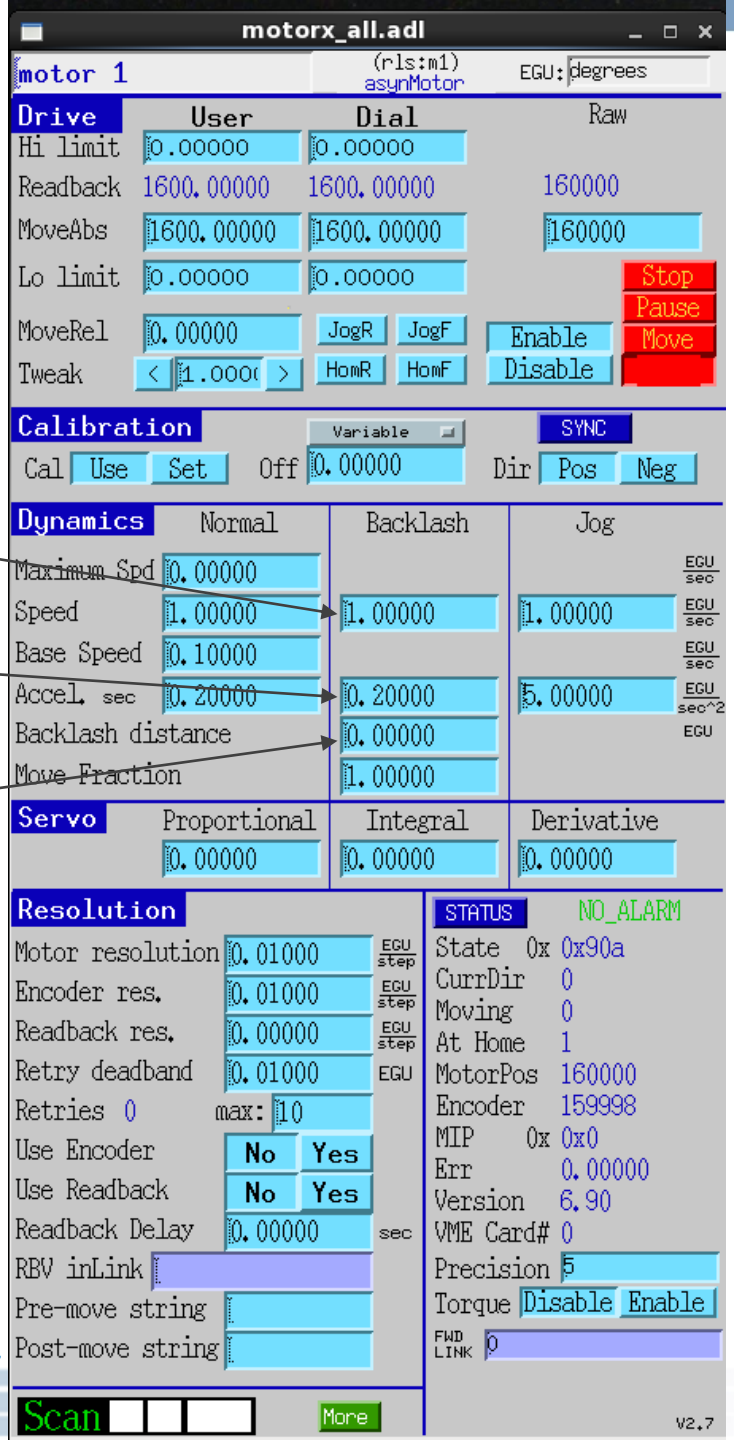
ELSE IF the incremental distance > |BDST|, OR, the move is Not in the preferred direction.

THEN, move to position (Target - BDST) using slew parameters, then move to Target using backlash parameters.

ELSE IF the incremental distance <= |BDST|, AND, the move is in the preferred direction.

THEN, backlash is assumed to have been taken out; move to Target using backlash parameters.

ENDIF



BVEL

BACC

BDST

Motor module distribution and build customization

- Motor module web page.
  - http://www.aps.anl.gov/bcda/synApps/motor/index.html
- Motor module build customization.
  - To build examples; uncomment three lines in <motor>/Makefile that begin with “#!”.
  - <motor>/configure/RELEASE:
    - If only EPICS_BASE is defined, only OMS and Soft Channel model #1 device/drivers are built.
    - ASYN is required for everything else.
  - <motor>/motorApp/Makefile: Define which device/driver modules to build.