Stage Database & Motor "Tuning" (Future procedure, In Progress)

Current user stages should be entered here. This page lists the part number, has a link to manufacturer documentation and gives each stage a "name". This name corresponds to the configuration in the parameter manager which contains all the settings of the controller that make the stage work well. Entering this name in the questionnaire for a beam time will enable a single command to push all the right settings to the motor controller after the stage has been connected. These configurations are shared among hutches as stages are moving between hutches. This list is also shared with the beam line engineer/designer so that they know what stages are available when planning an upcoming experiment. As more stages become available in LCLS, the page should also be updated. Also seeing as though some stages under the same name can be outfitted with a gear reducer, these configurations should be saved within the parameter manager and the confluence page updated

The procedure for setting up IMS/dumb motion stages:

1. Open the confluence page and look for the stage you will be using and note its corresponding parameter manager name. If a new stage is being used, discuss with your local ECS staff member on the support available.
2. Open the questionnaire for the specific experiment and navigate to the "CDS" tab
    a. You will notice that there is a space to enter information about the various motor stages needed. The information entered in these columns allows the stages to be automatically loaded into the local hutch python sessions and the archiver without the need for manual additions to separate experiment files. Scientists and ECS staff can work together to populate these columns.
        i. Purpose -
        ii. python/archive name -
        iii. Stage Identifier -
        iv. Location -
        v. Base PV -
3. Fill in the necessary information to the best of your ability. Keep in mind that the stage identifier should match exactly what can be found in the parameter manager (i.e. confluence page)
4. Begin the experimental set up of your motors and stages.(i.e, affix to the beamline, connect cables to appropriate motor controllers, etc...). Contact your local ECS staff member if assistance is needed during this step.
    a. If you haven't input the base PV in the questionnaire before this step, now is the time since all motors should be connected to a smart/dumb motor controller channel. The controller should be labeled with some information regarding the base PV of each channel or found in hutch documentation.
5. Open a terminal and start a new hutch python session (i.e. <hutch>3) on the specific hutch machine (i.e. <hutch>-control, <hutch>-daq, etc...).
    a. You will notice that among the devices loaded on the screen, there should be a section that loads in the experiment questionnaire. There you will see your stage names. After the session loads, you should run x.<stage python name>.diff_configuration() to list all the parameters that have changed between the configuration (stage identifier) listed in the questionnaire and the parameters restored from previous setups (limits are not usually included here). If you accept these changes, it is best to run x.<stage python name>.configure() to apply those changes. If you do not accept the changes, it is best to discuss with an ECS staff member about updating the stage parameters for future applications, if the parameters have been successfully tested. To see what the different "FLD_"s are check out the table below.
6. Once your parameters have been confirmed, now is the time to test the functionality of your motion stage. This step also applies to testing out parameters that weren't accepted during the initial configuration.
    a. First, set velocity, acceleration, user and dial limits (adjust as necessary, basic hutch python commands can also be found below)
    b. Confirm that the motor moves the correct distance when commanded. This parameter usually corresponds to the UREV parameter which, if configured properly, should not need to be amended. If the motor does not move at all, and it is confirmed that the cables are all connected properly there could be a few reasons to take note of here:
        i. The PV associated with the stage does not correspond to the motor controller port used.
        ii. Limits have been set to unreasonable values (i.e. 0)
        iii. Velocities and accelerations have not been set
        iv. The run current is too low.
    c. Manually check limit switch functions if the stage is equipped with them. It may be easiest to confirm this through the motor-expert-screen.

Notable hutch python IMS motion stage field descriptions

Field	Description
FLD_LM	Limit Stop Mode
FLD_SM	Stall Mode
FLD_SF	Stall factor
FLD_STSV	Stall severity level for reporting
FIELD_ERSV	Error Severity Level for reporting
FLD_EE	Encoder Enabled
FLD_EL	Encoder Lines
FLD_MT	Motor Settling Time (ms)

Field	Description
FLD_EGU	Engineering units name
FLD_UREV	Units per Revolutions (EGU/rev)
FLD_FREV	Full steps per rev
FLD_SREV	micro-steps per revolution
FLD_ERES	Encoder Step Size (EGU)
FLD_MRES	Motor Resolution (EGU/micro-step)

Field	Description
FLD_SMAX	max speed (rev/s)
FLD_S	speed (rev/s)
FLD_BS	backlash speed
FLD_HS	Home speed
FLD_ACCL	Acceleration
FLD_BACC	Backlash acceleration
FLD_HACC	Homing acceleration
FLD_TWV	Tweak Value
FLD_HOMD	Dial Value at home

FLD_HT	Holding Current Delay Time (ms)
FLD_RCMX	Run current max (0..100 %)
FLD_RC	Run current (0..100 %)
FLD_HCMX	Holding current max (0..100 %)
FLD_HC	Holding current (0..100 %)
FLD_MODE	Run Mode
FLD_PDBD	Position Tolerance for monitoring (EGU)

FLD_DIR	Direction
FLD_OFF	User Offset
FLD_FOFF	Adjust Offset/Controller
FLD_HTYP	Homing Type
FLD_HEGE	Homing edge of index or limit
FLD_BDST	Backlash distance (EGU)
FLD_HDST	Back-off distance for limit-switch-homing (EGU)
FLD_RTRY	Max # of Retries
FLD_RDBD	Retry deadband (egu)

FLD_EGAG	Use external gauge
FLD_ESKL	External Gauge Scale
FLD_DLVL	Debugging Level
FLD_S1	Limit switch setting 1
FLD_S2	Limit switch setting 2
FLD_S3	Limit switch setting 3
FLD_S4	Limit switch setting 4
PV_FW__MEANS	Name of forward direction
PV_REV__MEANS	Name of reverse direction
FLD_SBAS	base speed (rev/s)

Basic hutch-python 3 commands

Function	Command	Description
Absolute move	.mv(value)	Absolute move to a position
Relative move	.mvr(value)	Relative move from this position
Set velocity	.velocity.put(value)	
Set user limits	.set_high_limit(value)/.set_low_limit(value)	
set dial limits		