

# MOTION CONTROL OF THE MIRROR CHAMBER

Z. H. Zhang, W. H. Jia, P. Liu, L. F. Zheng,  
Shanghai Institute of Applied Physics (SINAP), CAS, Shanghai 201800, P. R. China

## Abstract

The initial beamlines of SSRF have several mirror chambers. The motion mechanism of the mirror chamber consists of six step motors. Three vertical motors are used to achieve the height translation, the roll adjustment and the pitch adjustment. Two horizontal motors are used to achieve the horizon translation and the yaw adjustment. The mirror bending is achieved by a single motor. Except the bending, every motion needs two or three motors to work simultaneously. The VME systems are used in the control system. The EPICS (Experimental Physics and Industrial Control System) toolkits [1] are adopted to develop the control software. This paper describes how to use motor records, transform records and the EPICS database linking mechanism to form soft loops, which solve the difficulties of the complex motions perfectly. The system has been successfully applied in the mirror chambers at SSRF.

## INTRODUCTION

SSRF has developed mirror chambers [2], which are used to manipulate the collimating mirror and the focusing mirror at beamlines. The structure of the chamber and the coordinate sketch are illustrated in Fig. 1. The motion mechanism of the mirror chamber consists of six step motors. Three physical variables like the height translation, the roll and the pitch are achieved by three vertical motors ( $y_1, y_2, y_3$ ). Other two physical variables like the horizon translation and the yaw are achieved by two horizontal motors ( $x_1, x_2$ ). The mirror bending is achieved by a single motor ( $b$ ).

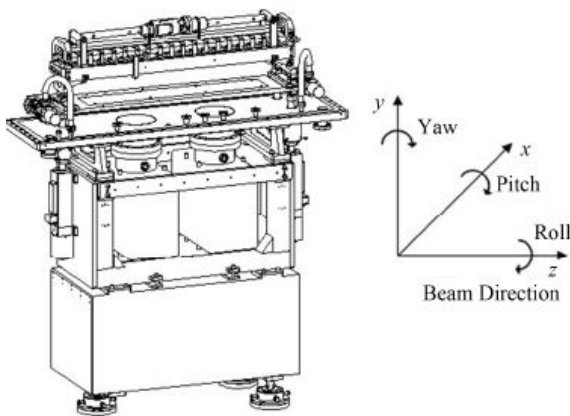


Figure 1: Structure of the chamber and coordinate sketch.

All motors are 5-phase step motors. To achieve the adjustment of the mirror properly, two or three motors must work coordinately. Physical variables, corresponding motors and their formulas are listed in Table 1. Standard hardware and software are selected to

set up the control system. The hardware is based on VME bus, the software is developed by using EPICS toolkits.

Table 1: Physical variables, corresponding motors and their formulas,  $W$  is the width of the mirror,  $L$  is the length of the mirror

Physical Variables	Motors	Formulas
X	$x_1, x_2$	$X=(x_1+x_2)/2$
Y	$y_1, y_2, y_3$	$Y=(2y_1+y_2+y_3)/4$
Roll	$y_2, y_3$	$\text{Roll}=\tan^{-1}((y_3 - y_2)/W)$
Pitch	$y_1, y_2, y_3$	$\text{Pitch}=\tan^{-1}(((y_2 + y_3) - y_1)/2L)$
Yaw	$x_1, x_2$	$\text{Yaw}=\tan^{-1}((x_2 - x_1)/L)$
Bend	$b$	$\text{Bend} = b$

## HARDWARE

The hardware includes an operator interface(OPI), an input/output controller(IOC) and motor drivers as shown in Fig. 2. The OPI is a HP-PC with linux operating system and EPICS environment, the operator interfaces run on this PC. The IOC is a VME crate, which houses a Motorola MVME5500 board and a Pro-dex MAXv-8000 motion controller. MVME5500 is with powerpc604 CPU running VxWorks5.5 as a real time operating system. MAXv-8000 is a VME-bus-based motion controller, which can manage up to 8 axes of open-loop stepper, closed-loop stepper or servo systems and synchronize all independent or coordinated motion of up to 8 axes. SLS2015 is a 5-phase stepper motor driver developed by Swiss Light Source. The OPI communicates with the IOC through Ethernet. The IOC communicates with motor drivers through SCSI cables [3].

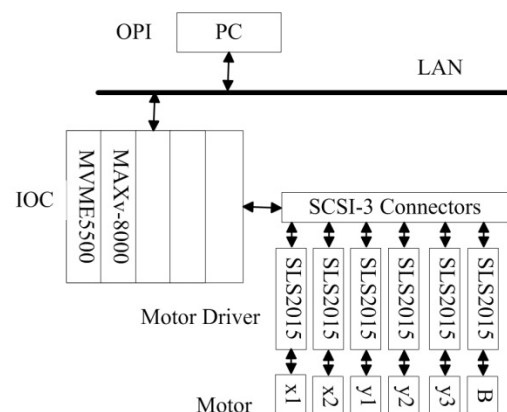


Figure 2: Hardware architecture.

## SOFTWARE

### Software Structure

EPICS is used as the control software. We use the latest version of EPICS (Release R.3.14.8.2) on linux. The IOC is a VME controller with VxWorks operating system. An EPICS run-time database on the IOC and Channel Access (CA) constitute the core of EPICS software. EPICS database is edited using a simple text editor and compiled to be loaded into the IOC by a VxWorks startup script file. Records defined in database are accessed by EPICS client software on the OPI using the field names of the records.

Three EPICS support modules are used to implement the control function. They are motor-6-2, calc-2-8 and autosave-4-7, which can be found in the APS synApps package [4]. The motor module contains motor record [5], soft motor record and the MAXv-8000 device support. The calc module contains transform record [6], which is used to perform the mathematical calculation. The autosave module is used to save values of parameters in run time and restore them in boot time. The software structure is shown in Fig. 3.

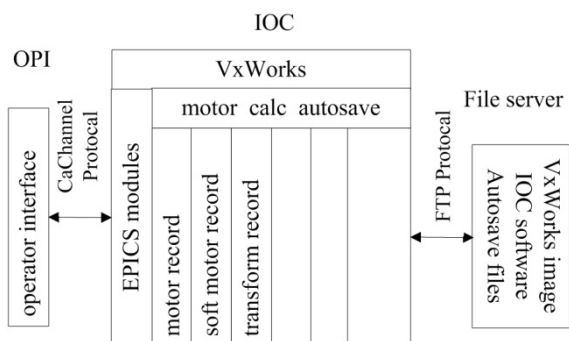


Figure 3: Software structure.

### Motion of Single Stepper Motor

After the mirror chamber arrives, we need to test the fundamental function of each stepper motor. The main parameters of a stepper motor include resolution, run direction, velocity, etc. The motor record has more than 100 fields corresponding to the parameters of a stepper motor. To drive a stepper motor, we need to configure these fields correctly.

### Soft Loop in the Database

After testing each motor, we implement to control the physical variables by combining the soft motor records, transform records and motor records. In our EPICS database, the physical variables are represented by soft motor records, stepper motors are represented by motor records, transform records are used to calculate the formulas listed in Table 1. A transform record has 16 float variables, each variable has independent input, calculation and output functions. The INP, CLC and OUT fields of transform record perform these functions. A transform record gets values of physical variables through the OUT fields of the soft motor records, then executes calculation

automatically according to the specified formulas in CLC fields and gets values of motors' positions. Finally the position values are output to the motor records through the OUT fields of the transform record and the motors run immediately. This is the forward half loop.

When motors are running, the read-back value of each motor is continuously changing. The read-back state of the soft motor records should be updated simultaneously. We also use a transform record. After receiving the read-back values from motor records, the transform record converts them to read-back values of the physical variables according to the formulas and outputs these values to the corresponding soft motor records. This is the backward half loop. The soft loop is shown in Fig. 4.

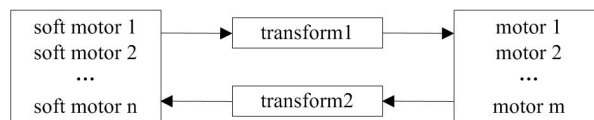


Figure 4: Soft loop in the database.

In the soft loop described above, records are linked by the INP or OUT fields. There are three main linking modes between EPICS records [7]. They are PP (process passive) mode, NPP(not process passive) mode and CP (continuous process) mode. PP mode means doing the INP or OUT operation and triggering the process of the target record. NPP mode means doing the INP or OUT operation and not triggering the process of the target record. CP mode means continuously doing the INP or OUT operation and triggering the process of the target record. In the soft loop, we use PP linking mode in the forward half loop. In the backward half loop, the linking between motor record and transform record is CP mode, others are NPP modes.

Because in the backward half loop the transform record is linked to the soft motor records by NPP, the backward half loop can't trigger the forward half loop when we directly drive the motors. In this case, the read-back values of the soft records are updated in time, while the corresponding intermediate variables in the transform record remain old values. These not-updated variables will cause the loop to work wrong when we change the physical variables next time in the soft motor records. It is necessary to synchronize the intermediate variables in the forward half loop with the read-back values. The intermediate variables in the forward transform record get their values through the OUT fields of the backward transform record using NPP linking mode. After this synchronization, the soft loop can work smoothly without losing beats. The final soft loop is showed in Fig. 5.

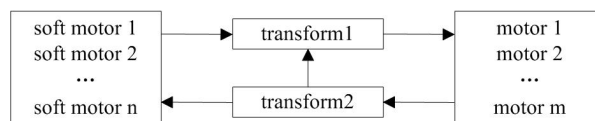


Figure 5: Final soft loop in the database.

### Save and Restore of Parameters

The MAXv-8000 controller can't save and restore values of controlled variables and motors. Some important values will disappear after cold rebooting the controller. The autosave support module can help save and restore field values of EPICS records. The basic settings are as following [8]:

- Write two text files named "positions.req" and "settings.req". The "positions.req" file lists the DVAL field names of motor records. The "setting.req" file lists other field names of EPICS records, such as velocity, resolution, etc.
- According to the "positions.req" and "settings.req", EPICS IOC will generate "position.sav" and "settings.sav" automatically in run time. When the controller cold reboots, the values saved in these two files will be restored automatically.
- In the VxWorks startup script file, specify "positions.sav" to be restored before record initialization and "settings.sav" to be restored after record initialization.

### OPI Interface

There are several tools to create EPICS operator interfaces, such as edm, medm and css. We use edm to build the interface. Fig. 6 is a screenshot of the control interface.

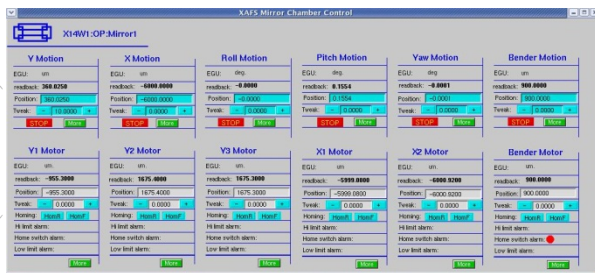


Figure 6: Edm-based control interface.

### CONCLUSION

The motion control system of the mirror chamber has been developed and tested during the commissioning operation of SSRF. By now, the system has worked stably for 3 years. The EPICS database soft loop described in this paper has become a standard method of developing applications of beamline motion control at SSRF.

### REFERENCE

- <http://www.aps.anl.gov/epics/about.php>
- Y. FU, W. Q. ZHU, and S. XUE, "The development of SSRF beamline mirror manipulator systems", Nucl Tech. 2010, 33(10).
- Y. HU, "EPICS-Based Motion Control System", Shanghai Institute of Applied Physics, 2007.
- <http://www.aps.anl.gov/bcda/synApps/>
- <http://www.aps.anl.gov/bcda/synApps/motor/R6-2/motorRecord.html>
- <http://www.aps.anl.gov/bcda/synApps/calc/R2-4/transformRecord.html>
- Martin R. Kraimer EPICS Input/Output Controller Application Developer's Guide, Argonne National Laboratory Advanced Photon Source, June 1998
- <http://www.aps.anl.gov/bcda/synApps/autosave/autoSave>