Python 3.6 Update to Lens Alignment

Matthew Hoover[1], Zachary Lentz[2+]

[1]University of California at Santa Cruz

[2]Linac Coherent Light Source, SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA.

[+]Contact:zlentz@slac.stanford.edu

Introduction

When focusing a laser, having to modify all three axes of the Beryllium lens precisely can be difficult and time consuming. When there is a very small amount of acceptable variance with the end target, human error can cripple an important project. The align function from the original 2.7 Python has been updated and improved upon for 3.6 to preserve its ability to save time and increase precision when moving motors. It will also provide a basis for future programs using the pseudo motor to automate alignment or using the LensStack object.
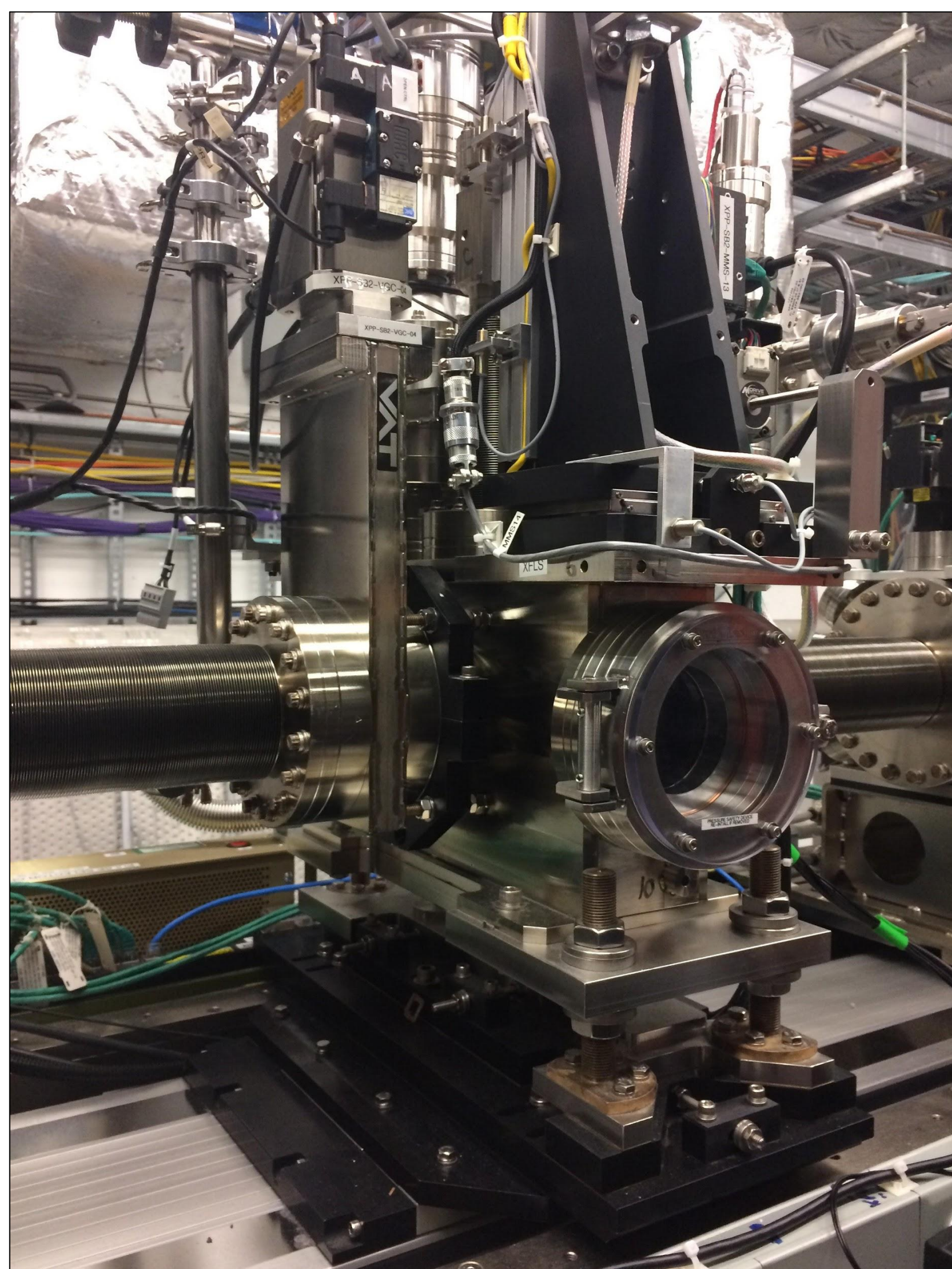
Figure 1: The actual lens alignment machine. The three motors move with z moving in the same direction as the beam line, y moving up and down, and x moving perpendicular to the beam. This is why x and y are for focusing the beam, while z is for adjusting the focal length.

Method

The main feature of this function is a pseudo motor that directly controls the z motor, but also moves the x and y motors to a focused position. The positions are based on data that must be made before using the pseudo motor, called in the align function of the LensStack class. The data is saved between runs in a user set folder, with each individual motor getting a permanent log file that only uses the most recent data.

Before using the pseudo motor for aligning the laser automatically, the LensStack class must be calibrated manually to the beamline path at two positions in order to make a formula for the laser's position. This can be done with the align function, which uses user input to get two positions for the three motors where the beam is focused on the sample. The align function uses the z-motor's upper and lower limit for the two positions, and uses the tweak function from mv_interface to give the user an interfaceable way to adjust the focus of the x- and y-motors without repeatedly calling a move function. These positions are saved in a folder that must be set by the user before calling the align function in a motor specific file. It keeps all saved positions by each motor in order to easily reference it later.

That data is what is called upon when using the pseudo motor. The pseudo motor moves the z-motor directly to the position it is given, but also moves the x- and y-motors. From the data stored, a slope intercept model of the beam is made with x and y dependant on z. The motors are then moved to their respective positions, putting the beam into alignment at a different focal length.

Figure 2: After the LensStack object has been aligned, the pseudo motor can be called as any other motor without more alignment. Beyond changing the move function to move all the motors, it responds to functions like the z motor. Realignment is only necessary when the beam angle changes, letting the motor be used continuously.

$$x = \frac{x_1 - x_2}{z_1 - z_2}(z - z_1) + x_1$$

Equation 1: Point slope form as used in program. Also used with x replaced by y.
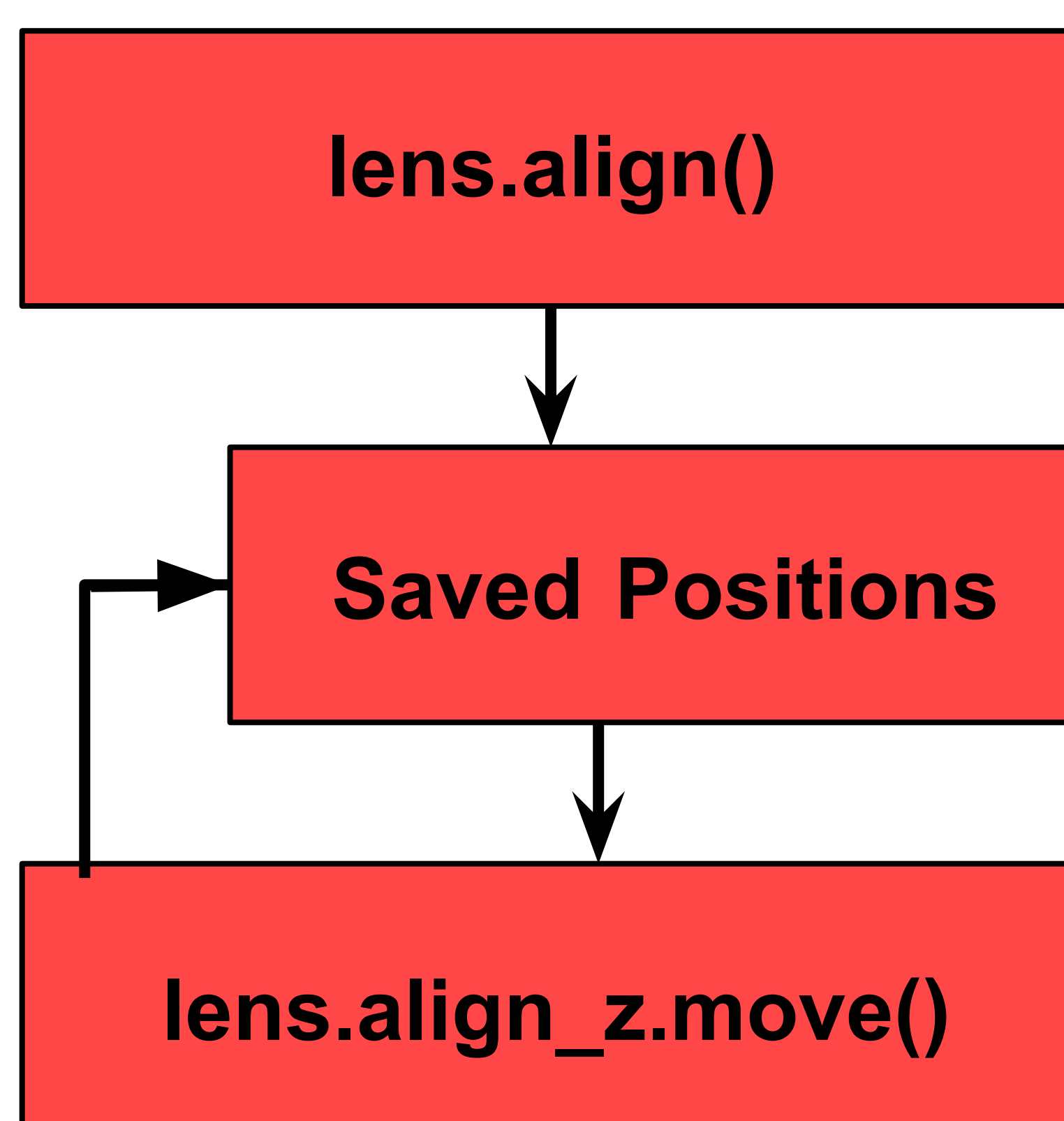
Future Applications

While this base motor is useful, it can be improved upon by automating more of the functions is provides. After the first alignment it is very useful, but that still means that the user has to align it on each end. Making a program to focus the x and y motors of the lens would further improve the precision of the pseudo motor and decrease the time needed to use the function. However, this would take a large amount of sample data to ensure it works properly.

Another similar use for the pseudo motors would be for aligning the crystals in the X-ray Correlation Spectroscopy (XCS) instrument. The distance the seperated x-rays need to go is determined by the user, but aligning the crystals is inconsistent across their range, and any small error is noticeable in the output. Making a pseudo motor control the movement after taking data on the crystal's movements would allow the user to only set the distance between pulses. The pseudo motor would then align the crystals into the necessary positions with its in depth knowledge of all the mirrors and lenses needed to reflect the x-ray the necessary distance.

A different function that relies on this pseudo motor is one which automates the control of the focal length of the project. The pseudo motor alone makes it easy to be focused at the position, but finding it automatically would improve the efficiacy of the projects.

Acknowledgments

Date: 08/03/2018