Project: Multi-motor imaging device

PI: Geunyoung Yoon
Prepared by: Chloe Degre Kendrick

Introduction

This device consists of four motors and two cameras. All four motors must be controlled synchronously within approximately 10 ms precision, some of them moving at different speeds or different ranges, in preset configurations. The accompanying software should provide live camera feeds for both cameras, controls for setting the scanning directions and speed for each scan, timing control for capturing images, and camera control (gain, exposure). Live camera views can be shown at lower resolution than saved images, if necessary.

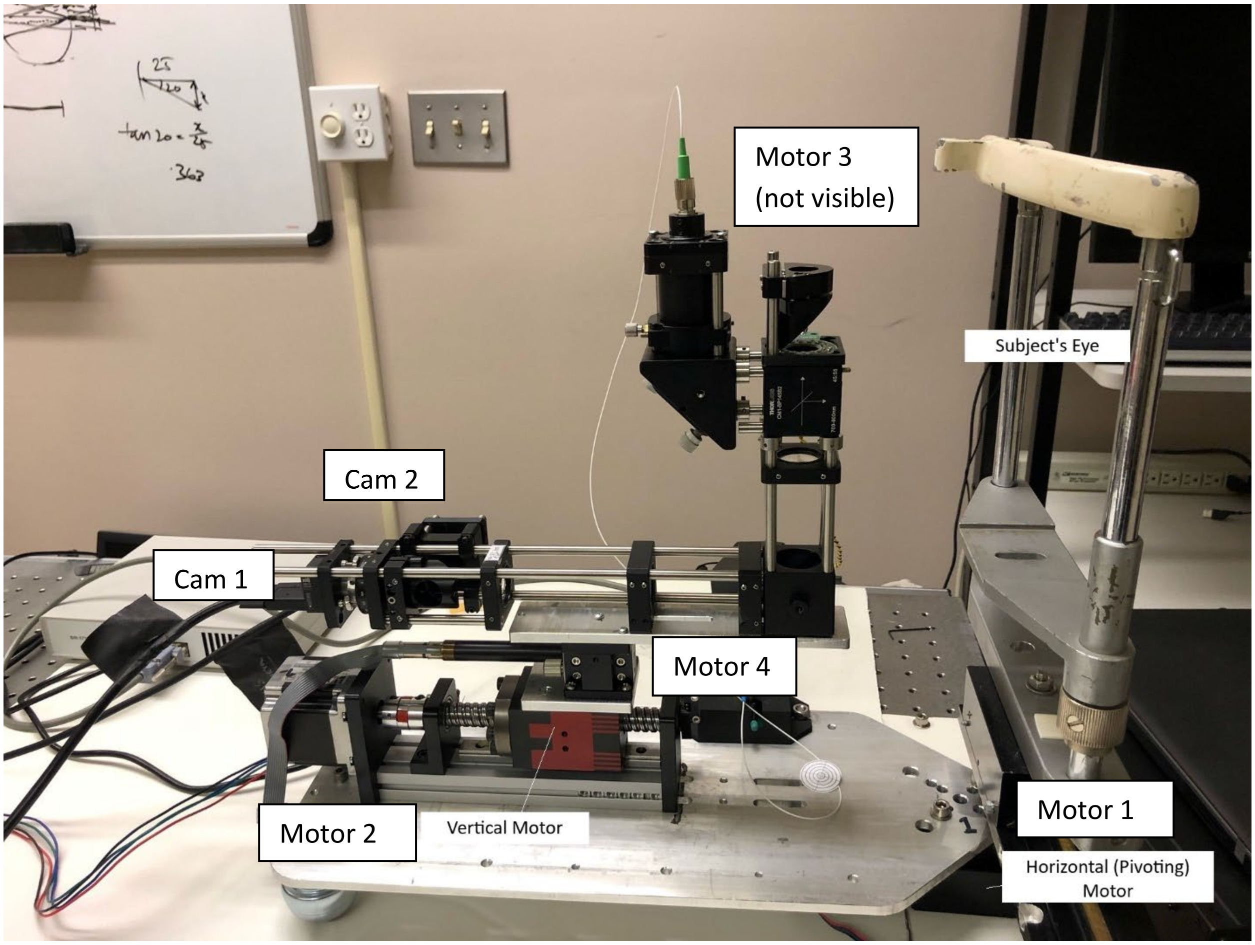
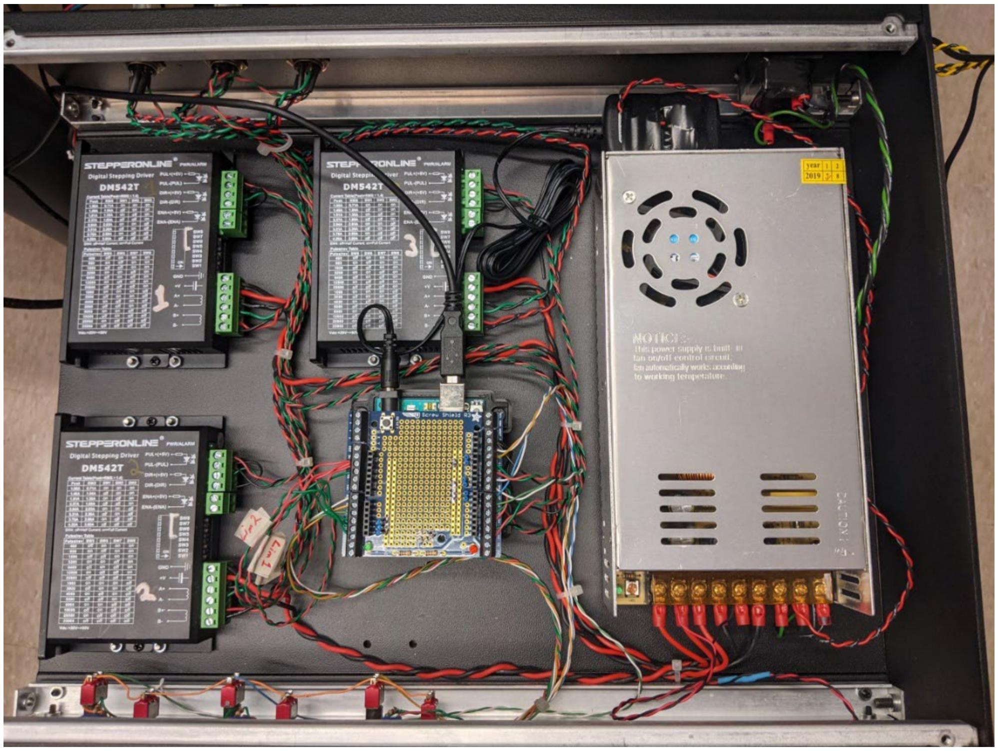
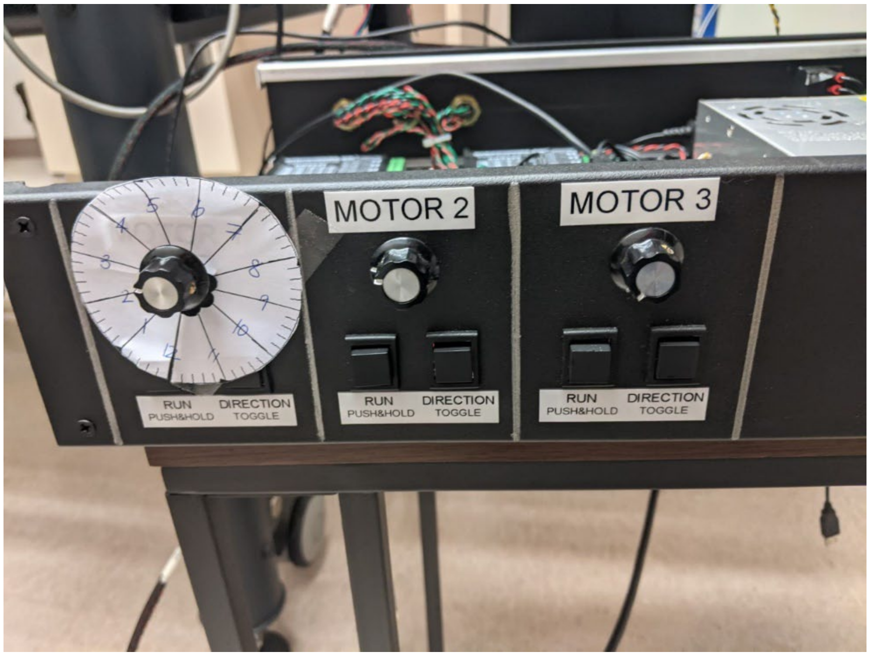


Figure 1. The device



Drivers for Motors 1-3, and current Arduino set up. Currently operated by an Arduino application that can control the Stepperonline motor drivers used for this system based on analog velocity controllers (knobs that control potentiometers) and digital direction and start/stop signals.



Currently, each motor is manually operated using a speed adjustment knob and two toggle buttons: run (click and hold) and direction change (click).

Scanning and image capture: One or more than one motors need to be operated simultaneously, but each motor will initially be set to have a different speed and range. The two cameras need to be synchronized with the motors to capture images at a preset time interval. The precision of the synchronization can be approximately 10ms. The images captured during the scan need to be saved as an individual image file or a video file. Each frame needs to be timestamped.

Components

Camera 1

Ximea model MQ013RG-E2 is a near infrared monochrome camera used to capture the Shack-Hartmann spot image used to measure the wavefront.
https://www.ximea.com/en/products/cameras-filtered-by-sensor-types/mq013rg-e2

Camera 2

Ximea model MQ013MG-E2 is a monochrome camera used to view the pupil. When the image of the pupil is well-centered, the WFS optical system is also well-aligned.
https://www.ximea.com/en/products/cameras-filtered-by-sensor-types/mq013mg-e2

Stepper Motors (Motors 1, 2, 3)

Three of the stepper motor drivers are STEPPERONLINE model DM652T. These drive the horizontal, vertical and optical axis motors.
https://www.omc-stepperonline.com/digital-stepper-driver-10-42a-20-50vdc-for-nema-17-23-24-stepper-motor-dm542t.html

Motor 4

A fourth stepper motor controls the tilt of the beam-splitter. Its driver is the Thorlabs Kinesis® Brushed Motor Controller model KDC101. The software control options for this driver include the Kinesis® software package (.NET or low-level DLLs) or a legacy, ActiveX APT package. Since ActiveX technology is legacy technology, the Kinesis® package will be implemented.

Motor: https://www.thorlabs.com/thorproduct.cfm?partnumber=ZST213
Controller: https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=2419

Current operation

The Windows 10 application provides for camera control and motor (#4) control. The three motor drivers are controlled by the Arduino processor, but by manually changing the knob and the toggle buttons, and dip switches.

Motor Calibration Panel

The UI provides a Calibration module which moves each motor independently in coarse or fine steps in the requested position. This process is used to define a state where the motor is at the centered (zero) position for all motors. At the end of every sweep operation, the software returns the motors to the zero position.

The UI parameters are described in the table below. Note that the translation of one UI step is defined by the UI and is distinct from the translation of the stepper motor steps.

Parameter	**Units**	**Type**	**Operation/Status**
Minimum Position	Degrees	edited text	Negative value that defines the most negative value permitted by the software. This limit is defined relative to the user-defined zero position.
Maximum Position	Degrees	edited text	Positive value that defines the most positive value permitted by the software. This limit is defined relative to the user-defined zero position.
Enable Motor	True/False	checkbox	Motor moves only when checked
Coarse step size	Degrees	List-box	Defines the translation of one UI step in visual angle degrees in “Coarse” mode
Fine step size	Degrees	Label or list-box*	Defines the translation of one UI step in visual angle degrees in “Fine” mode
Use coarse steps	True/False	Checkbox	Selects to use coarse steps when checked. Use fine steps when unchecked.
Step Backward	n/a	Pushbutton	Take one UI step in minus degree direction. Button is disabled until the step is completed to prevent excessive repetitions by the user.
Step Forward	n/a	Pushbutton	Take one UI step in plus degree direction. Button is disabled until the step is completed to prevent excessive repetitions by the user.
Move Backward	n/a	Pushbutton	Take continuous steps in the minus degree direction until the user presses the “Stop” or “Stop All” button on the UI, or the motor reaches the limits defined by the Minimum Position parameter.
Move Forward	n/a	Pushbutton	Take continuous steps in the plus degree direction until the user presses the “Stop” or “Stop All” button on the UI, or the motor reaches the limits defined by the Maximum Position parameter.

Parameter	Units	Type	Operation/Status
Stop	n/a	Pushbutton	Stops the respective motor.
Stop All	n/a	Pushbutton	Stops all motors.
Zero	n/a	Pushbutton	Return to zero position

*The Fine Step Size may simply be a label that indicates the degrees per pulse step provided by the stepper motor controller, for the user's reference. If these steps are too small, the fine step could be a list box that provides for a limited number of options for 1, 2 or 3 pulse steps and the equivalent degrees.

The host computer informs the Arduino when the motors are in the "zero" position of their respective axes, and then the Arduino internally maintains each motor's position as the number of steps from zero at all times, so that it can return to zero.

Sweep Parameters

	Scan type: horizontal	Scan type: vertical	Scan type: diagonal	Starting Position	Ending Position	Velocity
Motor 1	Yes	No	Yes	-x1 degrees	+x1 degrees	y1 deg/sec
Motor 2	No	Yes	Yes	-x2 mm	+x2 mm	y2 mm/sec
Motor 3	No	Yes	Yes	-x3 mm	+x3 mm	y3 mm/sec
Motor 4	No	Yes	Yes	-x4 mm	+x4 mm	y4 mm/sec

Parameter	Operation/Status
Initialize	Moves the motors from the zero position to the most negative position of the sweep. (The actual position may be adjusted for a calibrated "start-up" distance if the motor requires a short start-up interval for acceleration.)
Capture	Starts the sweep and image capture operation. If the motors are not in the initial position, the "Initialize" operation is automatically performed first.
Stop	Stops all motors.
Zero	Return to zero position

Typical Sequence

The software returns all motors to the zero position when it closes, so that the system starts with the motors close to the zero position for the next session. At startup, the operator makes any manual adjustments with the WideScanner Calibration panel using the last zero position as a reference and sets the current motor position as the zero position for the alignment step.

Camera 1 and 2 are put in “live” mode. Adjusting camera parameters (exposure time, gain) by up/down pushbuttons or typing in numbers may be required to optimize image quality. Aligning may require additional motor adjustments. The motor Calibration is re-set to zero to provide a suitable reference position for the sweep.

The operator next inputs the sweep parameters. The user enters the positions of the synchronized motors and sets the “Initialize” command. When the initialize command is issued, the motors move to the starting position. Next, the user provides a “Capture” command. This command initiates the scanning and image capture sequence. Once the sweep is completed, the images are saved.

File Naming Conventions

Images from both cameras should be saved as a sequence of images in .avi or .mpg format with each frame being timestamped or individual images (.tif) with timestamp filenames.