Physical Mount for Hyperspectral Camera in LDFZ

Naman J. Parikh

July 26, 2024

Motivation

In designing the mount to place the hyperspectral camera in the LDFZ, the primary concern was that the camera needs to move. Rather than a traditional two-dimensional image, the camera capures a one-dimensional row of pixels, oriented horizontally across the furnace. To create a two-dimensional image, the camera needs to move vertically and scan across the desired area. The camera manufacturer's solution is a moving stage with a mount for the camera above, so the sample on the stage moves while the camera remains still. In the furnace, we adapt this to have the camera move instead.

Design

To move the camera on command, we mount the camera on a motor such that the camera moves vertically when the motor rotates. The motor is controlled by a Raspberry Pi (R-Pi), which is connected to the camera's dedicated computer. The Arduino sketch found in my GitHub repository at `"MotorControl/smsweep/smsweep.ino"` is the program running on the R-Pi. When the camera's Hyperspec III software sends commands for motion to the stage, the R-Pi translates them into appropriate motion for the motor. The R-Pi also responds to Hyperspec III, mimicking the signals the stage would send to allow for easy usage of the software.

The motor is programmed to move down and then up repeatedly when signalled to move, and to return to it's start position when signalled to stop. 1 rotation of the motor translates to 10 mm of motion in Hyperspec III. So, if the motor is programmed to move 1 rotation in each direction, the end position in Hyperspec III should be set greater than 10 mm, so the motor will not receive a stop signal and continue to oscillate vertically until manually stopped or the desired number of frames are reached.

Adjusting the Motion

The distance the motor travels vertically can be easily adjusted in the Arduino sketch by changing the following code segment. `numSteps` is the number of motor steps to move, and 1 full rotation is 200 steps. After changing the number of steps as desired, the sketch needs to be uploaded to the R-Pi from the computer with the sketch. The R-Pi can then be reconnected to the camera's computer and used.

```
78     /* Adjust number of motor rotations or steps here */
79    int numRotations = 2;
80    int numSteps = numRotations * 200;
81    motor->setTargetPosition(dir ? numSteps : 0);
```

References