IR Spectrometer

Member: Brent Leinonen

Advisor: Dr. Malinowski

The scope of this project is an implementation of a 2D Infrared Spectrometer with a time recorder. The Spectrometer will be used to observer the changes in bond length of molecules during experiments. This is done by measuring the amount of light from the laser that is not absorbed, since different bonds will absorb different frequencies of light. The procedure is done by utilizing mirrors to allow the laser to meet with the camera at the sample to measure the light absorbed. Therefore, this system contains an infrared camera, several mirrors and an infrared laser.

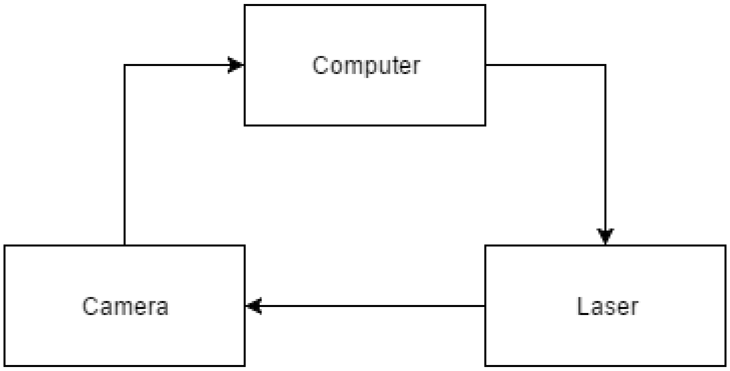
The majority of the work, however, will be done on a computer that controls the IR laser and the camera. This computer uses a Debian OS platform (presently Ubuntu 14.04 or 16.04), to host a python-based program that controls several console-based programs written in C. One of the already partially developed programs acquires the raw data from the camera and stores it in a binary file. Another program is planned to read the data from the stored binary file, process areas of the 2D spectrometer image selected by the user and store the results in another field, for example excel type, for further visualization. The processing involves averaging, noise reduction and subtracting the current image from the calibrated data in the form of the background image recorded previously without the tested sample to get the desired image.

System level:

1. Inputs:
    a. Request to calibrate, it is necessary to calibrate the system before use when external environment has changed activated by user.
    b. Request to immediately power up or power down the laser.
    c. Power to the camera, laser and computer.
    d. Request to start the experiment according to the specification stored in a configuration file, this has many inputs such as range of laser frequencies, time to be averaged, and the overall time of recording.
    e. The configuration file for the experiment.
    f. Request to process the experiment results for the selected area or areas of the sample.
2. Outputs:
    a. Archived background image/calibration data.
    b. Archived partially-processed 2D images from the spectrometer.
    c. Final processed images.

Subsystem level:

Block Diagram:



Laser:

The laser receives data from the computer telling the laser to go to a specified frequency and how long to hold that frequency.

Inputs:

1. Power
2. Frequency and time data from computer

Outputs:

1. Infrared laser

Camera:

The camera gathers the data from the laser going through the sample and reports that to the computer via USB

Inputs:

1. Image from laser + sample
2. Power

Outputs:

1. Data sent to computer to be analyzed

Computer:

The computer will be controlling the laser via Python scripts. Utilizing hand-made controller files, the program will tell the laser at what frequencies and how long each frequency should be set for. This provides the ability to average frames captured to get more reliable data.

Once the camera receives the image it reports that to the computer to allow for image processing. The computer then subtracts the new data from the camera with the previously gathered data from calibration (to cut out noise from the image we want) and displays both the subtracted version and the raw input from the camera.

The computer is running on an Ubuntu operating system and uses Python, C++ and Octave (GNU Matlab equivalent) to perform these operations.

Inputs:

1. Power
2. User specified controls, such as frequencies and exposure time
3. Signal from camera

Outputs:

1. Raw image from camera
2. Final image (subtraction from calibration and raw)