Automatic alignment of laser beams

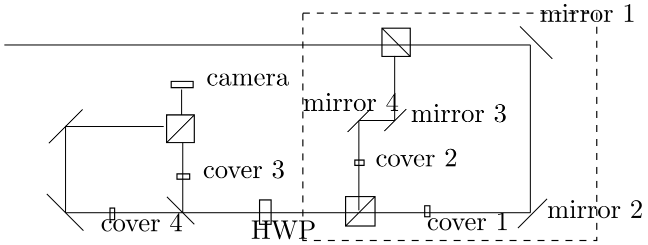
Jakub Wornbard

27 september 2019

Introduction

The goal is to align two laser beams by making thir positions overlap on the matrix of a camer. The setup is shown in the figure. In practice the region marked with a dashed line could be any other setup with TWO (there was one for testing purposes) beams and mirrors that allow for their control. The covers allow us to switch betweeen parts of the beam that reach the camera, so that their position can be determined separately but with the use of a single camera. Suppose that we want the two images corresponding to a particular beam to appear in a particular part of the camera matrix. Then the procedure for moving it is as follows

testing setup



1. Take photos of both parts of that beam reflected to the camera separately. Fit a gaussian funciton to the brightness distribution and take the center to be the peak of the function. This is done by the gaussian_fit.py module.

Memorize these positions

2. Move the knob on each mirror by a set amount (around 3° seems to work) and see how each of these movements affects the position of both images. Use these to compute the gradient of a chosen distance function in a given position
3. Move the knobs according to their input to the calculated gradient i.e. if the distance function is $d(\theta_1 \dots \theta_4)$ and the total rotation per step is α then the angle by which the mirror i is turned is $\frac{-(\nabla d)_i}{\|\nabla d\|}\alpha$
4. Repeat these steps until a satisfying distance marigin is reached.

Technical details

The images are taken by an IDS camera and the module responsible for taking the photos is ccd_time_snap.py. Parameters such as exposure time need to be set manually in align_lasers.py
Motors responsible for mirror knob rotation are controlled by an Arduino MEGA with the Motorizedi_Kinematic_Mount script. In order to communicate with the Arduiono, motor_control.py is used.
Similarly the servos responsible for covering the beams are controlled with an arduino UNO with the servo_control sketch and servo_control.py responsible for the communication. More details about pin setup and usage can be found in the code.

Issues

The diameter of the laser beam seemed to be too large for the system to work. Reducing its size with the use of lenses did not work because it was not collimated. An aperture could be used to cover the majority of the beam but it is possible that in this case the gaussian distribution of intensity could be lost and a different method for locating the center might be needed