LIGO

LIGO LASER BEAM TRACKING

Anirban Bairagi
LIGO SURF 2021
Mentors: Rana Adhikari, Yehonathan Drori, Tega Edo

20/09/2021

LIGO

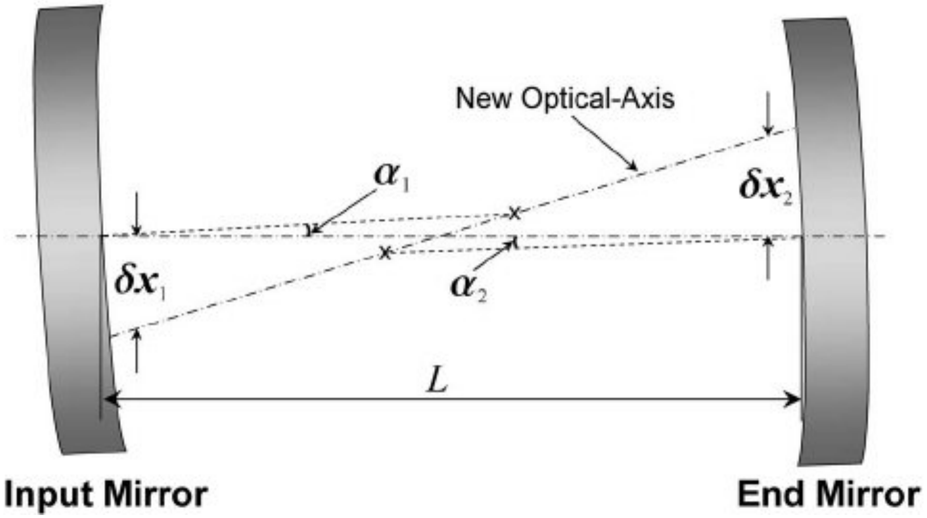
Overview

- Why does the laser beam spot move?
- Why do we need to track the beam spot?
- Image sensing and Data acquisition.
- Simulating realistic scattered beam spots.
- Application of CNN to get the position of the beam spots.

Motivation

Reason for Beam spot movement

- Due to ground motion, seismic vibration ITMs undergo angular movement causing a shift in beam position on ETMs and vice versa.



Need of Beam spot tracking

- Beam misalignment gives rise to angle to length coupling which couples angle noise into the GW readout. To understand angular movement of the mirrors so that a feedback control system may be employed in future to fix its position.
- RMS velocity of the beam spot helps in characterizing the detector.

Image Credits: Google

Image sensing and acquisition

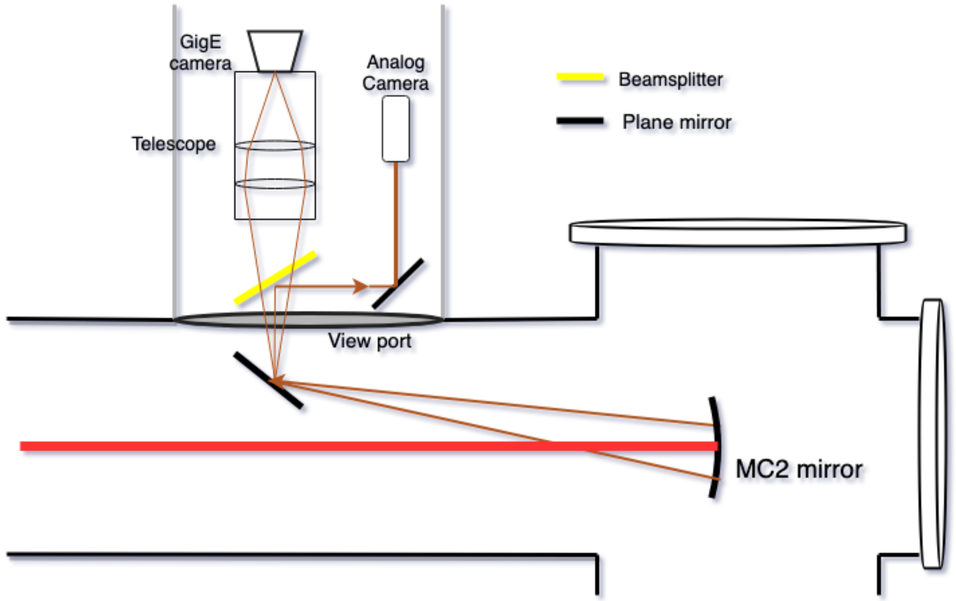


Image Credits: Kruthi

- CCD cameras are placed at an angle to the beam axis.
- The light incident on the mirror is scattered due to irregularities and point scatterers of the mirror, and it helps to capture the scattered beam spot from any angle.

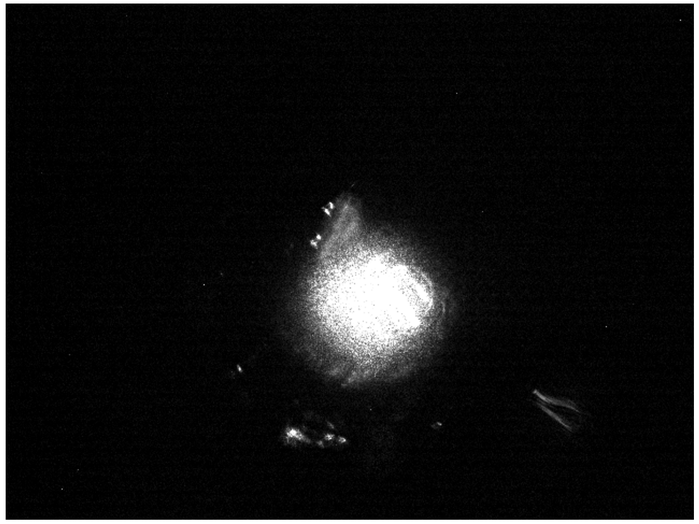


Image Credits: Dropbox / Surf_beam_motion_data

LIGO Simulation: Perfectly Gaussian Beam

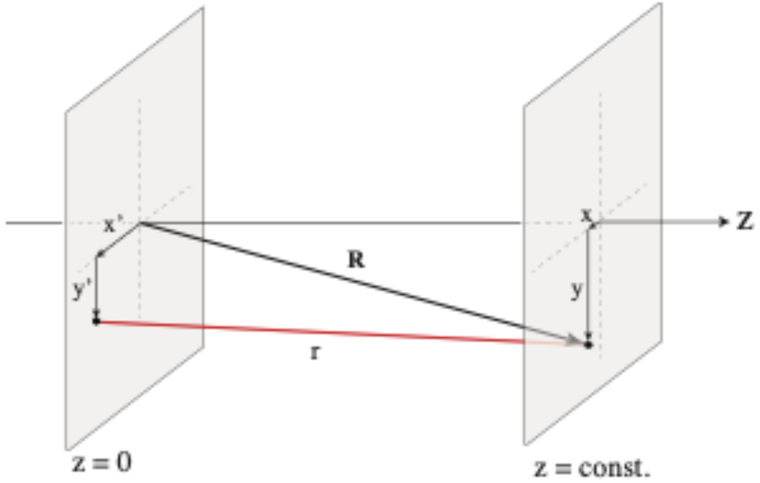
$$\vec{\mathrm{E}}(x',y',0) = \vec{\mathrm{E}}_0\, e^{-\frac{(x'-\mu_x)^2+(y'-\mu_y)^2}{w_0^2}}$$

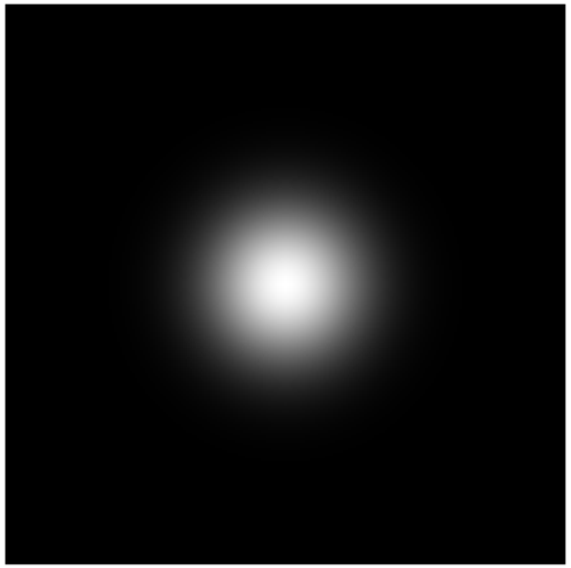
$$\tilde{\mathrm{E}}(k_x,k_y,0) = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\vec{\mathrm{E}}(x',y',0)e^{-i(k_x x'+k_y y')}dx'dy'$$

$$\tilde{\mathrm{E}}(k_x,k_y,z) = \tilde{\mathrm{E}}(k_x,k_y,0)e^{ik_z z}$$ → **Z=constant**

$$\vec{\mathrm{E}}(x,y,z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\tilde{\mathrm{E}}(k_x,k_y,z)dk_x dk_y$$

$$\mathrm{I}(x,y,z) = \vec{\mathrm{E}}^{\star}\vec{\mathrm{E}}$$



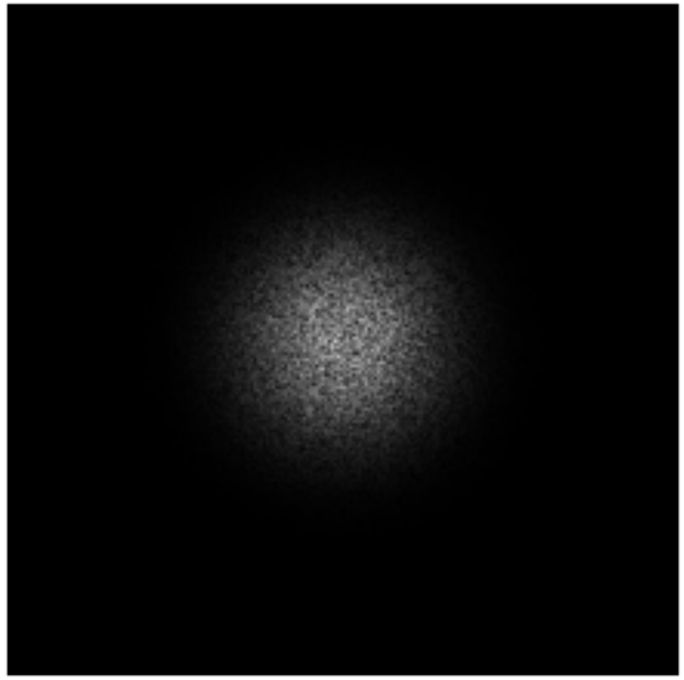


Simulation: Scattered Beam

$$\vec{E}_{\text{scatter}}(x, y, z) = \vec{E}(x, y, z)e^{ik_z \Delta z} - \vec{E}(x, y, z)$$

Δz defines the roughness at various points (x, y) of the mirror surface

$$I_{\text{scatter}}(x, y, z) = \vec{E}^{\star}_{\text{scatter}} \vec{E}_{\text{scatter}}$$

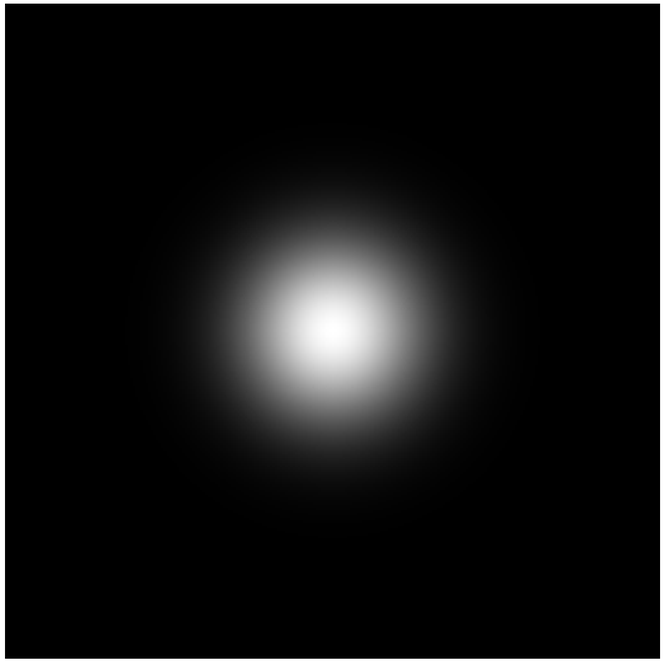


CCD Noises

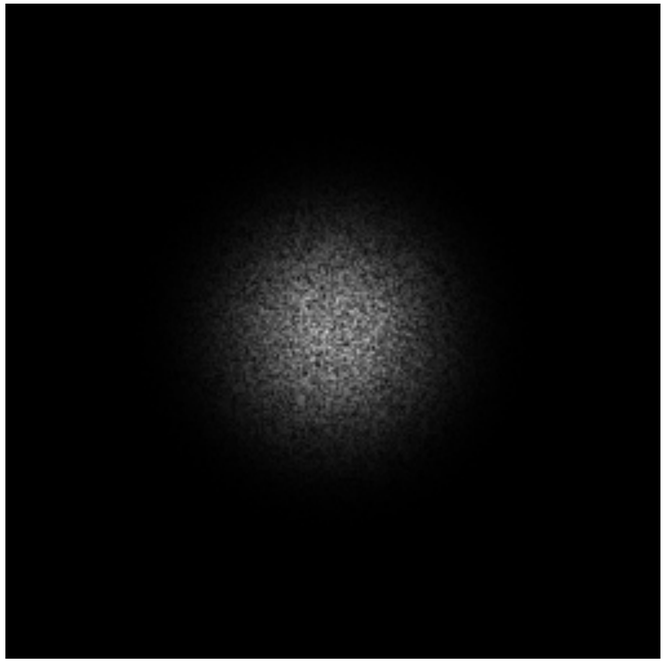
Shot Noise:

- Originates due to discrete nature of photons.
- Same number of photons can not generate same number of photo-electrons every time.
- Shows a Poisson distribution to the number of photo-electrons generated from the incident photons.

(a) Shot noise added to Gaussian Beam



(b) Shot noise added to Scattered Beam

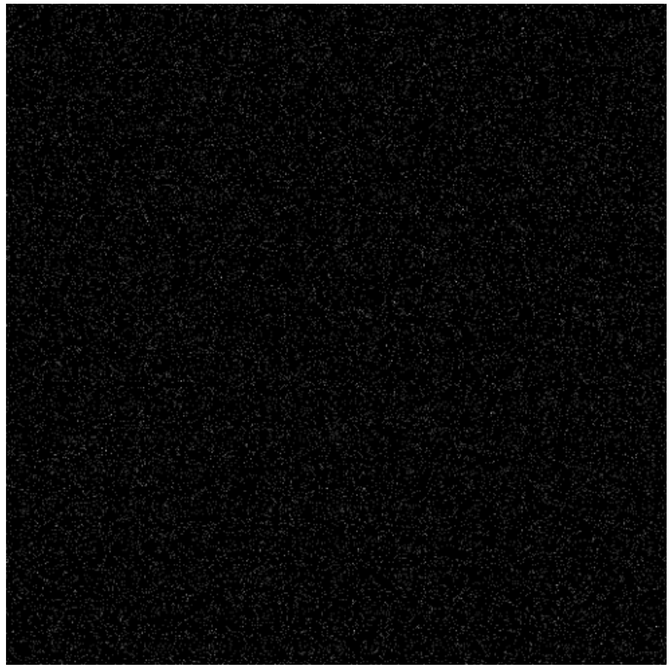


CCD Noises

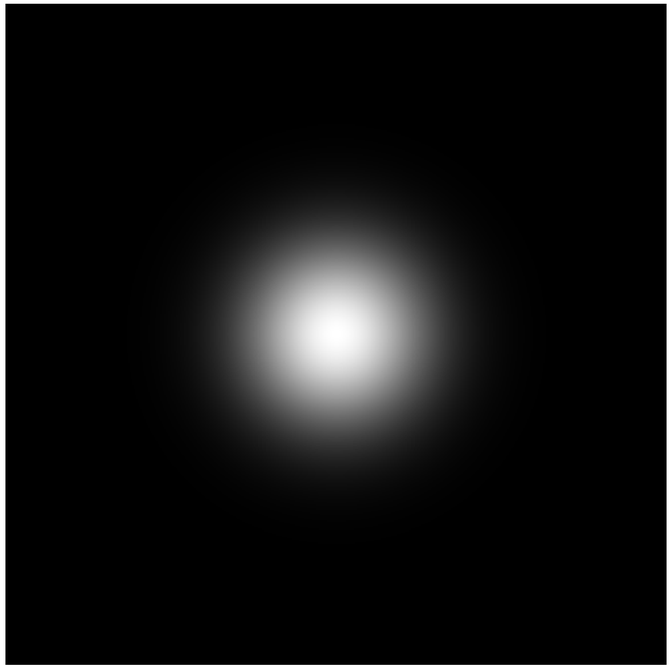
Dark Current Noise:

- Originates due to thermally produced electrons within the silicon structure of CCD.
- Depends on the temperature ; cryogenics may be used to reduce dark current.
- Follows Poisson statistics.

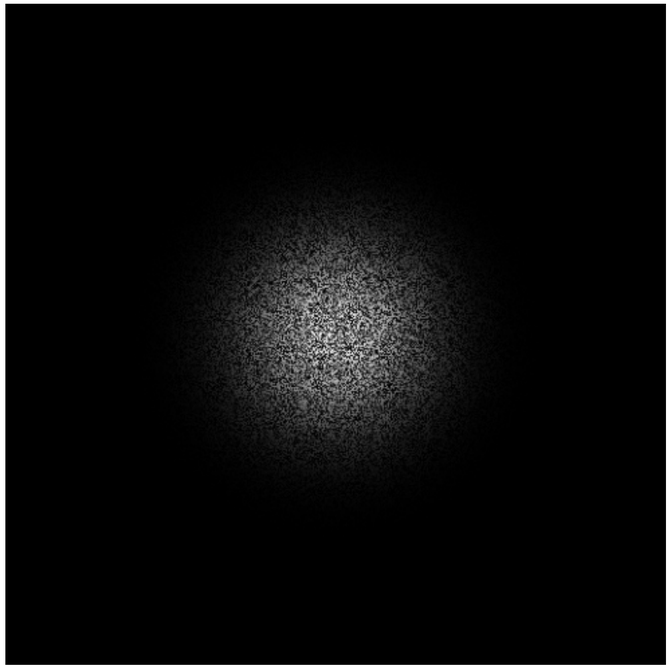
(a) Dark Current noise



(b) Dark Current noise added to Gaussian Beam



(c) Dark Current noise added to Scattered Beam

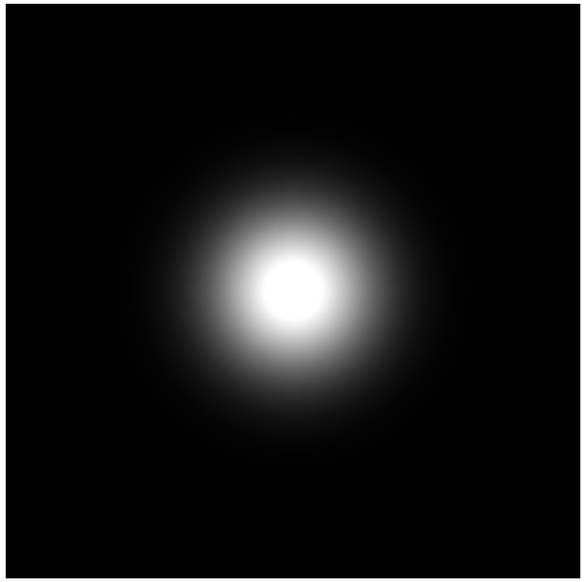


CCD Noises

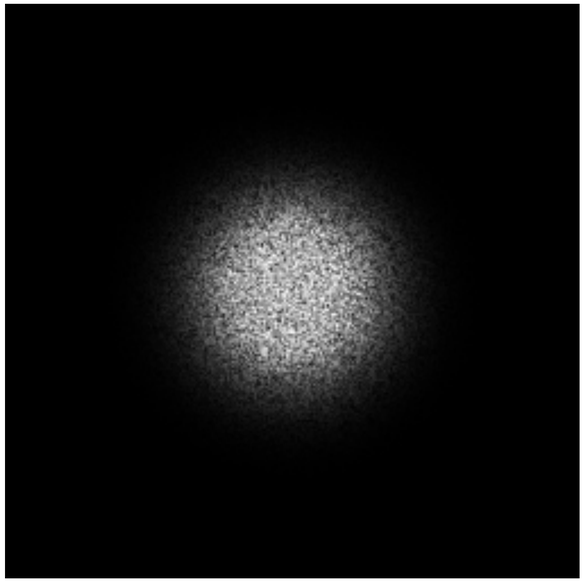
Saturation:

- After a certain threshold value of number of photons incident on the CCD camera, the generation of photo-electrons does not increase. Hence intensity reaches at a saturation value.

(a) Saturation effect on Gaussian Beam



(b) Saturation effect on Scattered Beam



Data Generation

Simulation Models:

- Gaussian Beam
- Scattered Beam
- Gaussian Beam with CCD noises
- Scattered Beam with CCD noises

Train , Validation and Test set

- Movement along X axis only $\mu_x \in [-1, +1]$
- Movement along Y axis only $\mu_y \in [-1, +1]$
- Movement along both direction $\mu_x, \mu_y \in [-1, +1]$

Screen: 34X34 cm2
Laser beam radius: 6.5 cm

Generated image resolution: 512X512

1 Pixel = 0.066 cm

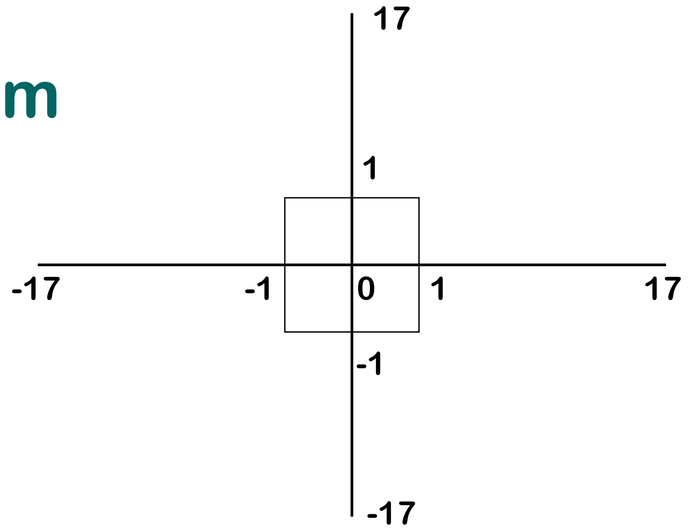


Fig: Coordinate system on CCD screen

Convolutional Neural Network

CNN is a class of Machine Learning technique which deals with images mostly to extract features from the images

Conv2D

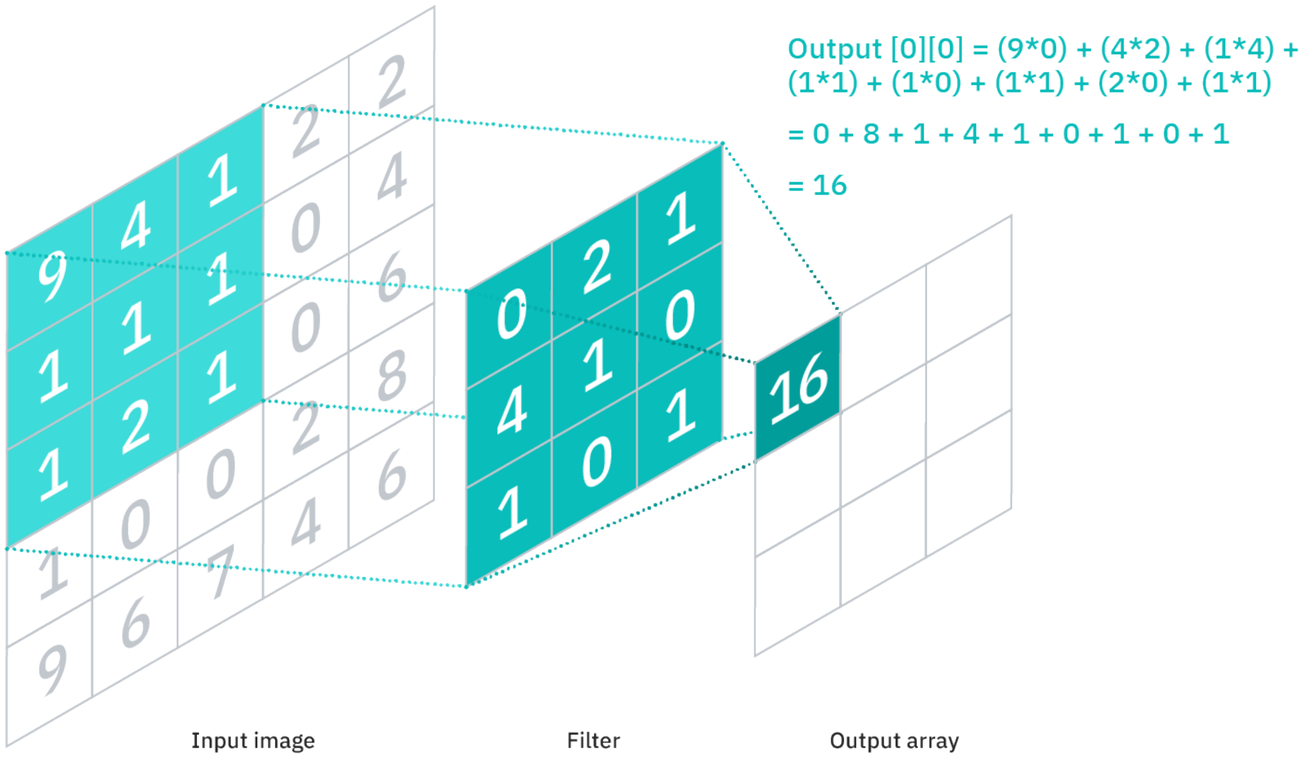


Image Credits: Google

Pooling Layer

- Max Pooling
- Average Pooling

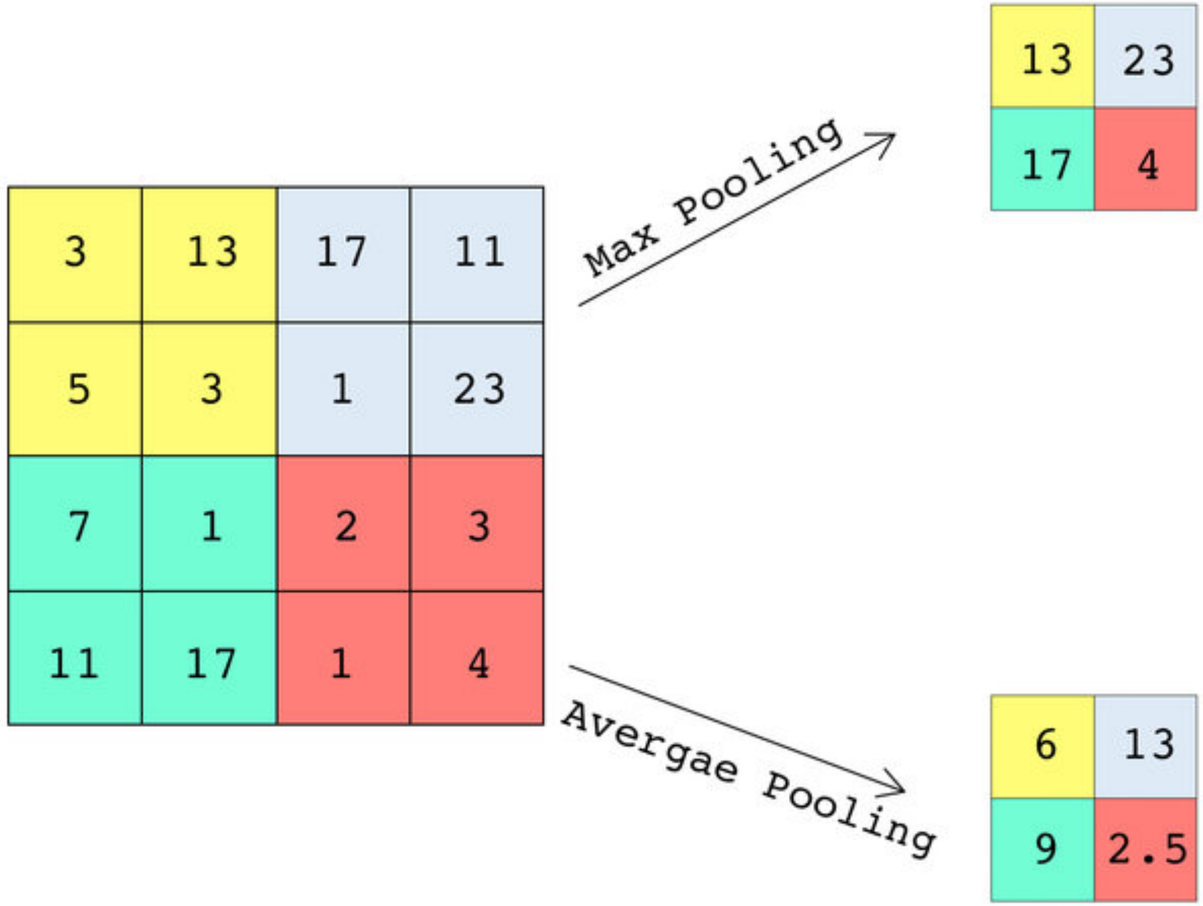
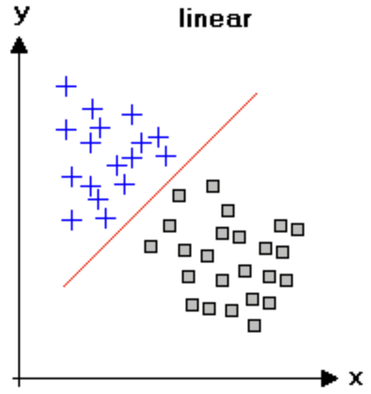
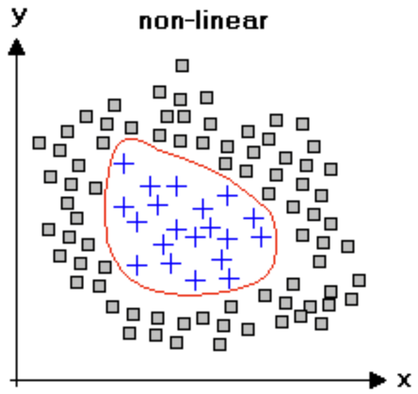


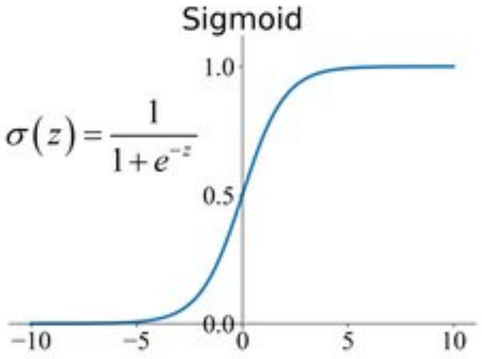
Image Credits: Google

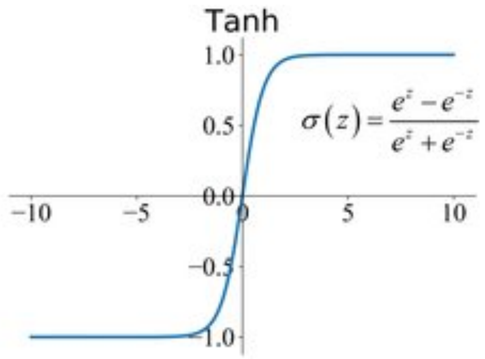
Activation Function

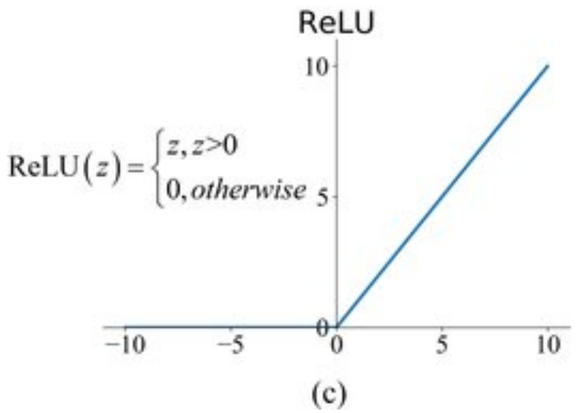
- Decides whether a neuron should be activated or not and how a input node should be transformed into an output.
- Sets a bound to the input layer before sending it to the next layer.
- Adds Non-linearity to the network which helps to detect more complex features.











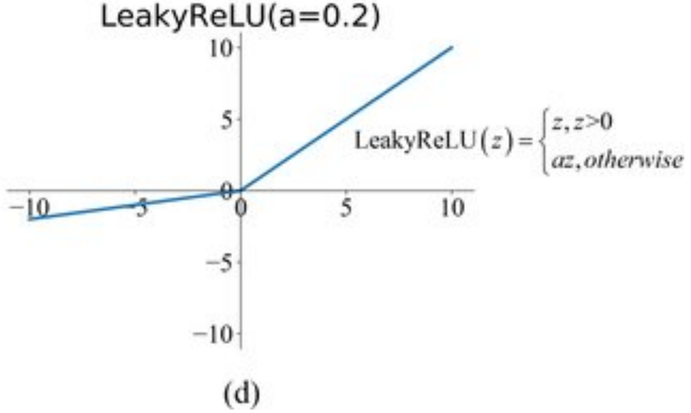
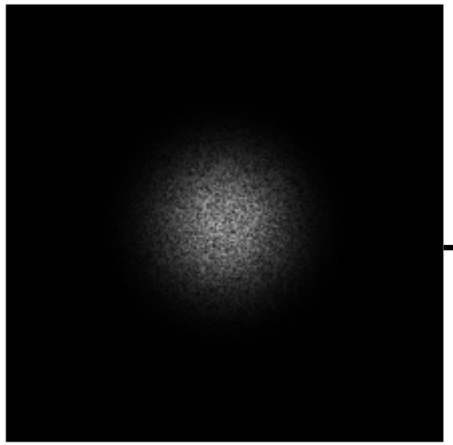


Image Credits: Google

CNN Model Architecture

input: InputLayer	input:	[(None, 512, 512, 3)]
	output:	[(None, 512, 512, 3)]
conv0: Conv2D	input:	(None, 512, 512, 3)
	output:	(None, 509, 509, 32)
activation: Activation	input:	(None, 509, 509, 32)
	output:	(None, 509, 509, 32)
max_pool0: MaxPooling2D	input:	(None, 509, 509, 32)
	output:	(None, 127, 127, 32)
conv1: Conv2D	input:	(None, 127, 127, 32)
	output:	(None, 124, 124, 64)
activation_1: Activation	input:	(None, 124, 124, 64)
	output:	(None, 124, 124, 64)
max_pool1: MaxPooling2D	input:	(None, 124, 124, 64)
	output:	(None, 31, 31, 64)
conv2: Conv2D	input:	(None, 31, 31, 64)
	output:	(None, 28, 28, 128)
activation_2: Activation	input:	(None, 28, 28, 128)
	output:	(None, 28, 28, 128)
max_pool2: MaxPooling2D	input:	(None, 28, 28, 128)
	output:	(None, 7, 7, 128)
flatten: Flatten	input:	(None, 7, 7, 128)
	output:	(None, 6272)
fc2: Dense	input:	(None, 6272)
	output:	(None, 2)

Beam Spot Center Position



CNN Model

Output: (0,0)

- How good is this model ?

Cost Function, Metric & Optimizer

Cost Function:

- **Mean Squared Error between true and predicted position of the training image set**

Metric:

- **Signal to Noise Ratio -> ratio of mean squared value of true position to mean squared value of difference between true and predicted position**

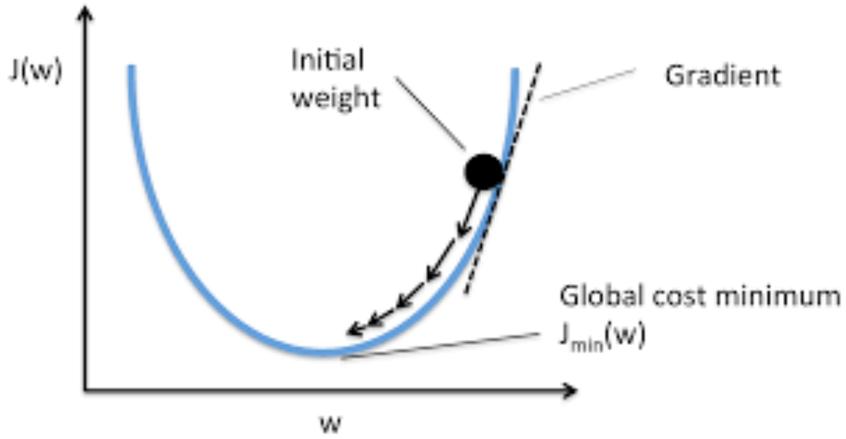
Optimizer:

- **Adaptive Moment (Adam) -> optimizes the value of the parameters using a modified gradient descent method**

Scheduler: StepLR

$$\mathrm{SGD}: w_{t+1} = w_t - \alpha \frac{\partial J}{\partial w}$$

Learning rate



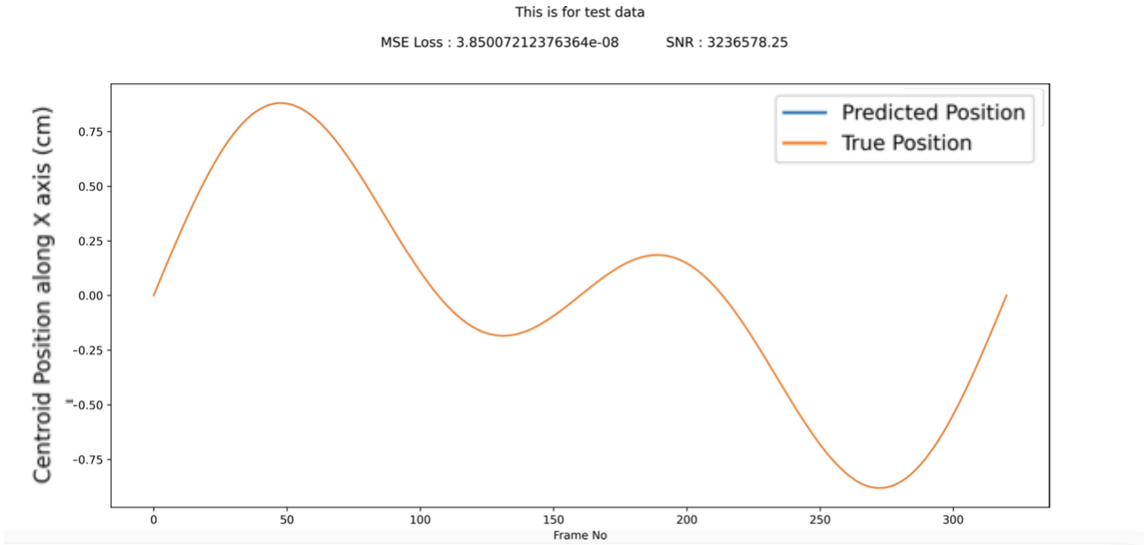
LIGO

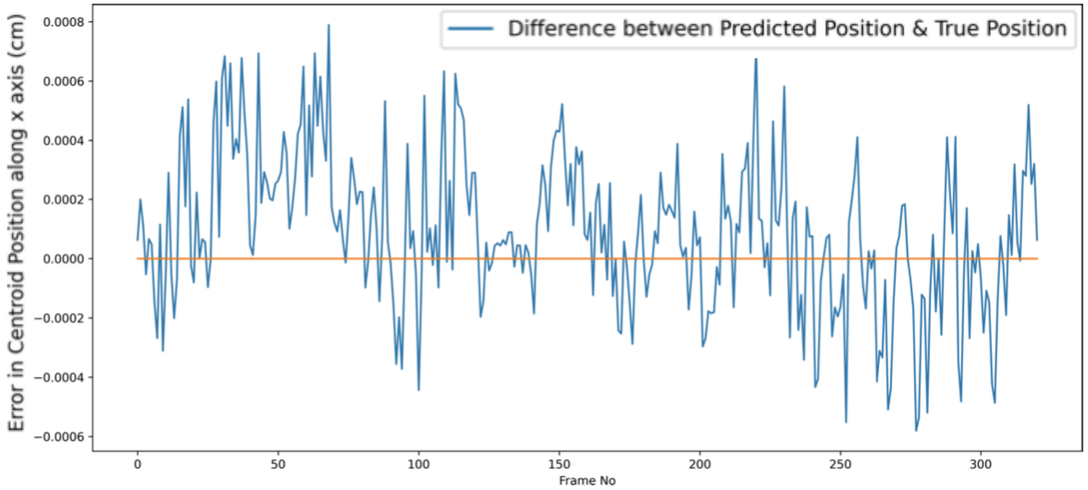
Hyperparameters

- Batchsize: 32
- Learning rate: 0.001
- LR scheduler step size: 10
- Gamma: 0.5
- Epochs: 50

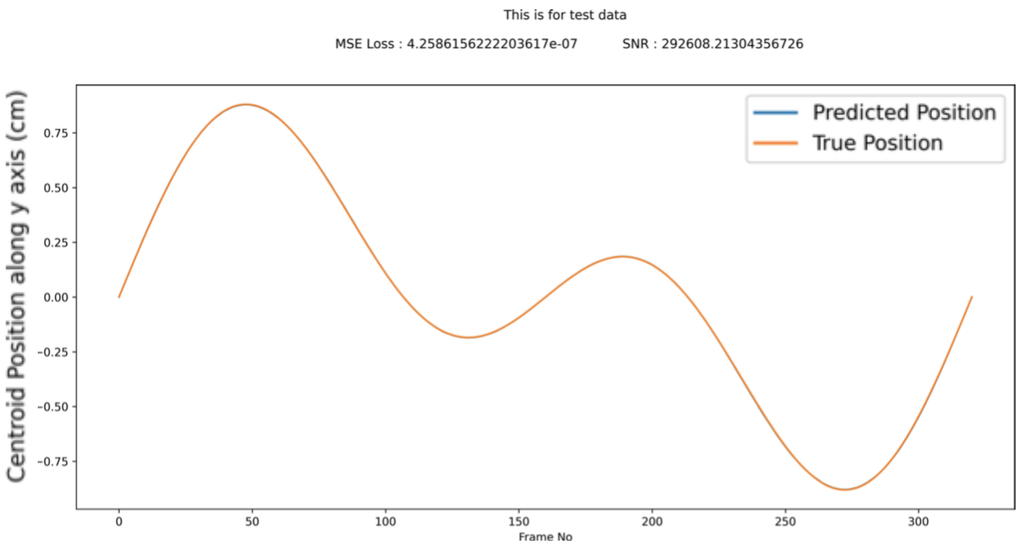
Results for Gaussian Beam

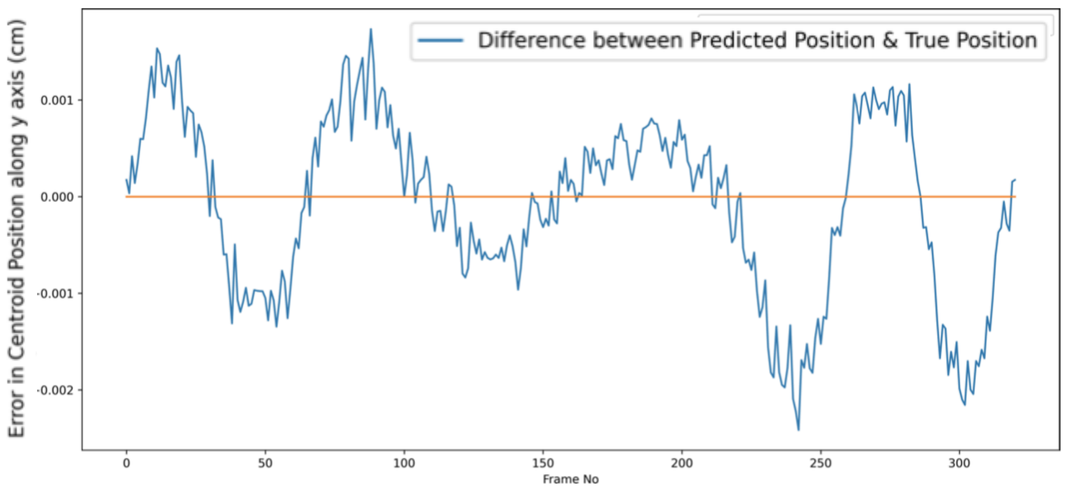
Movement along X axis only





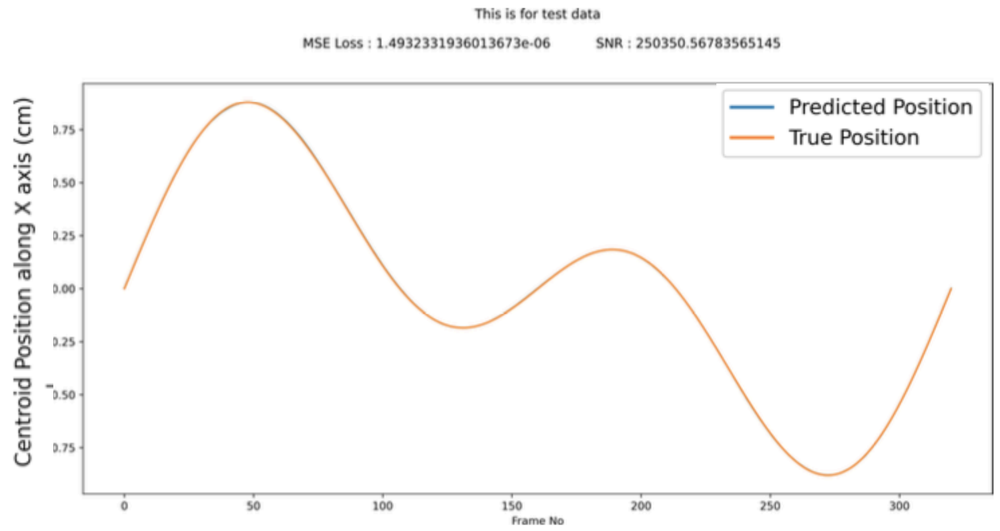
Movement along Y axis only

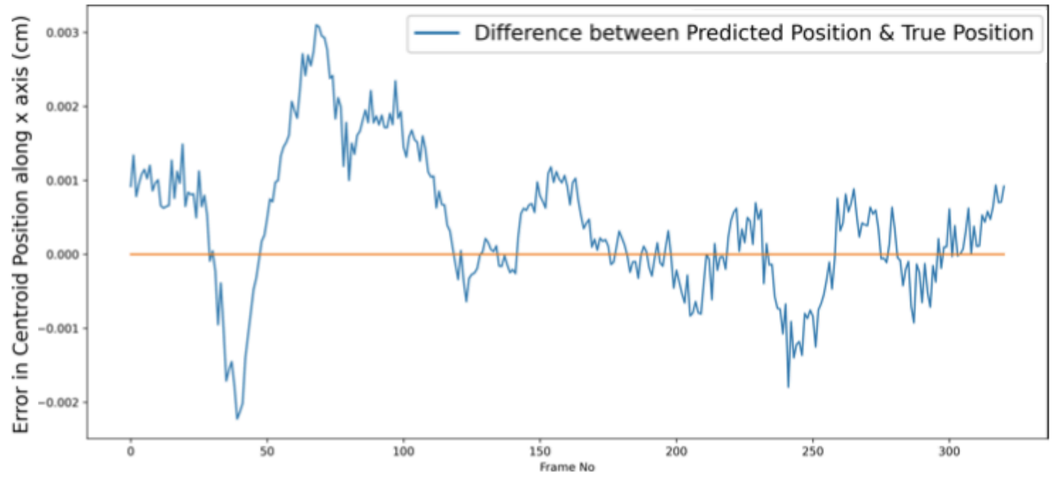


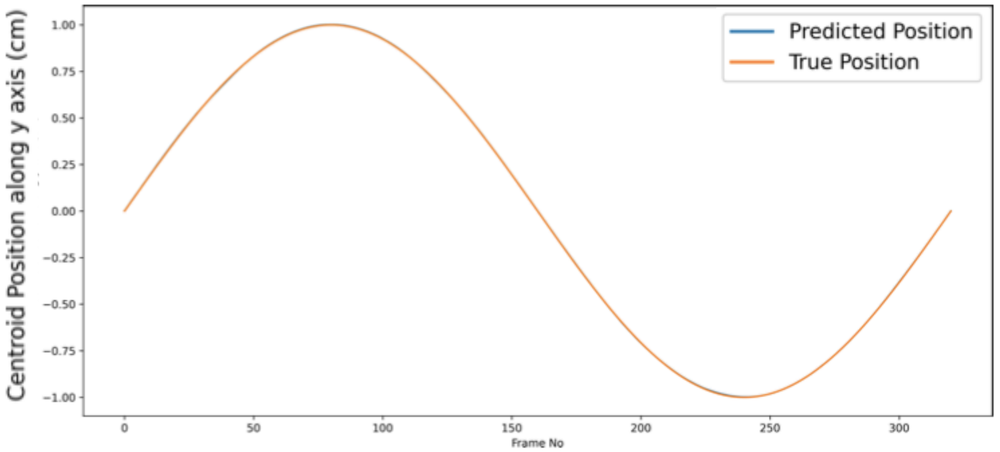


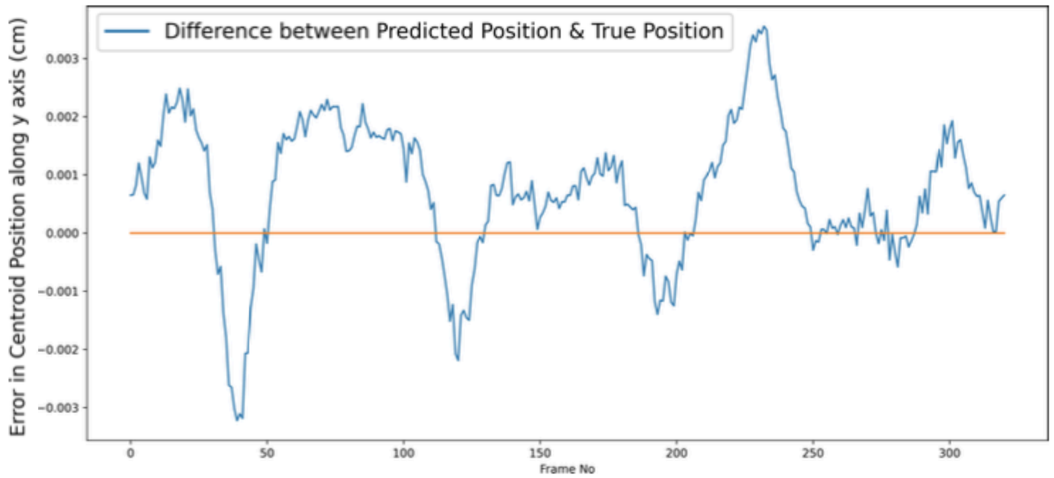
Results for Gaussian Beam

Movement along both X and Y axis



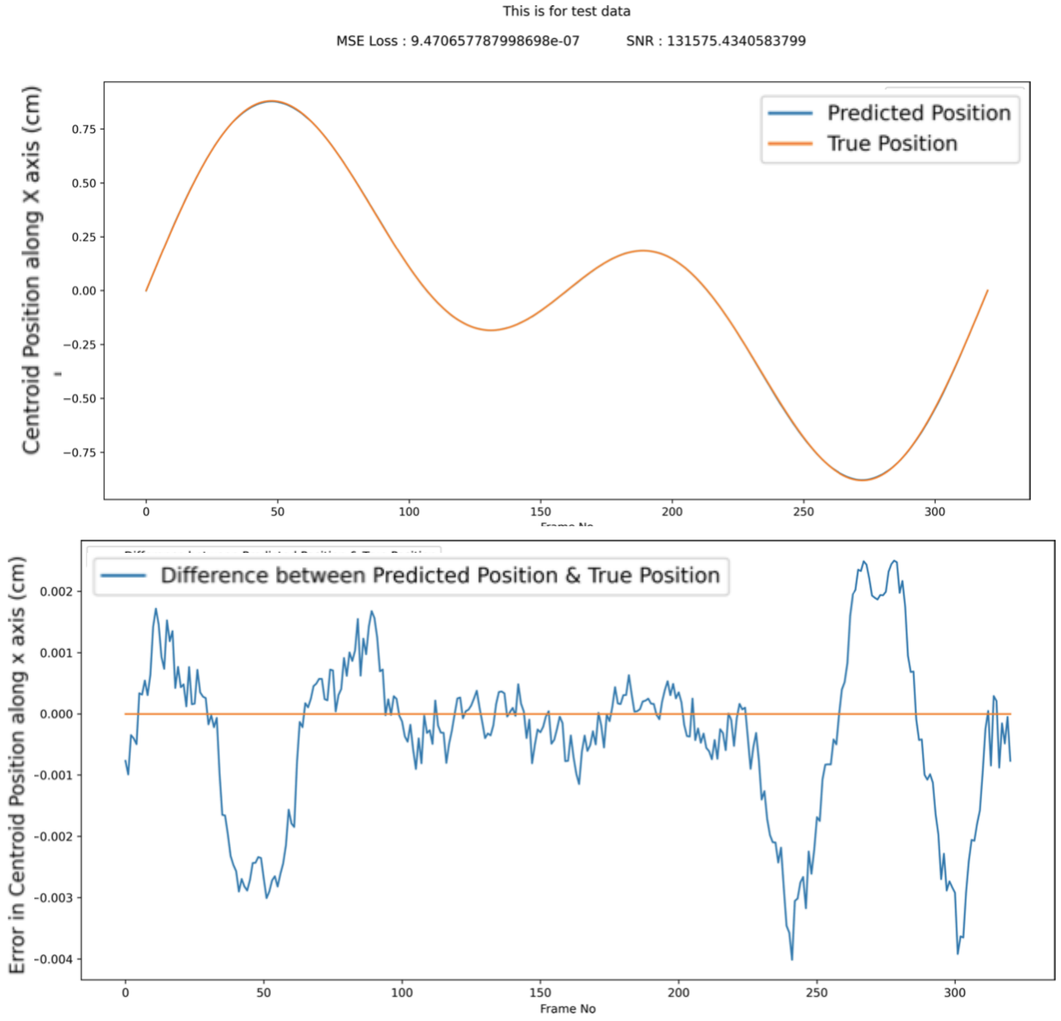




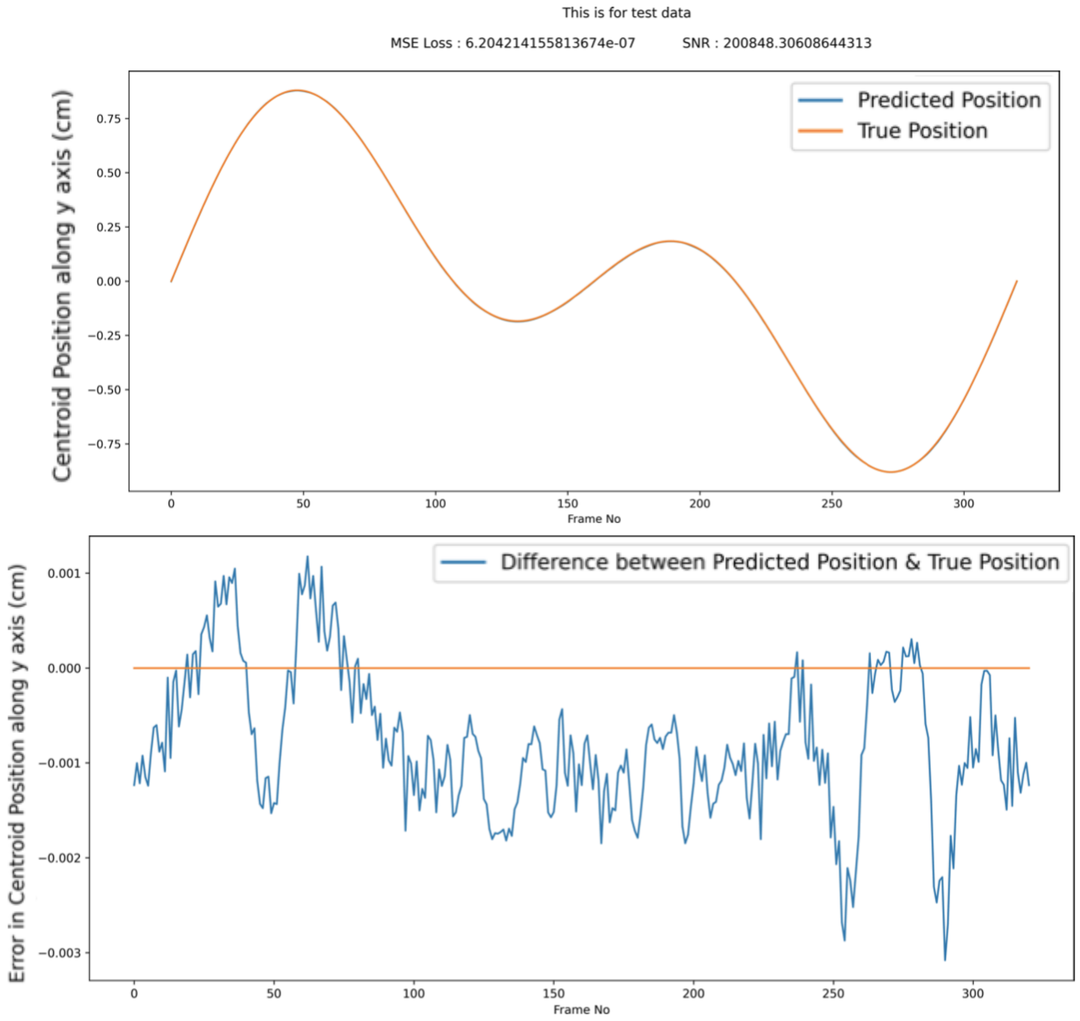


Results for Scattered Beam

Movement along X axis only

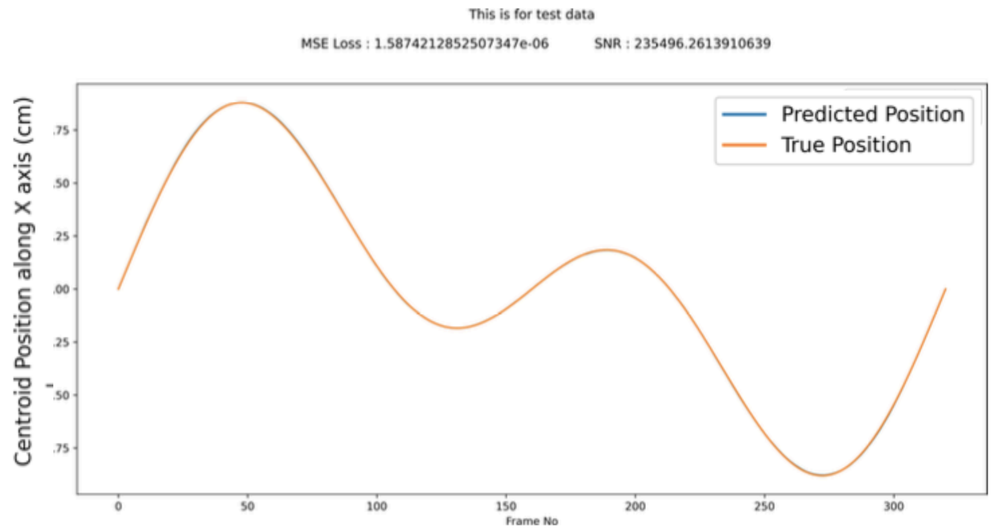


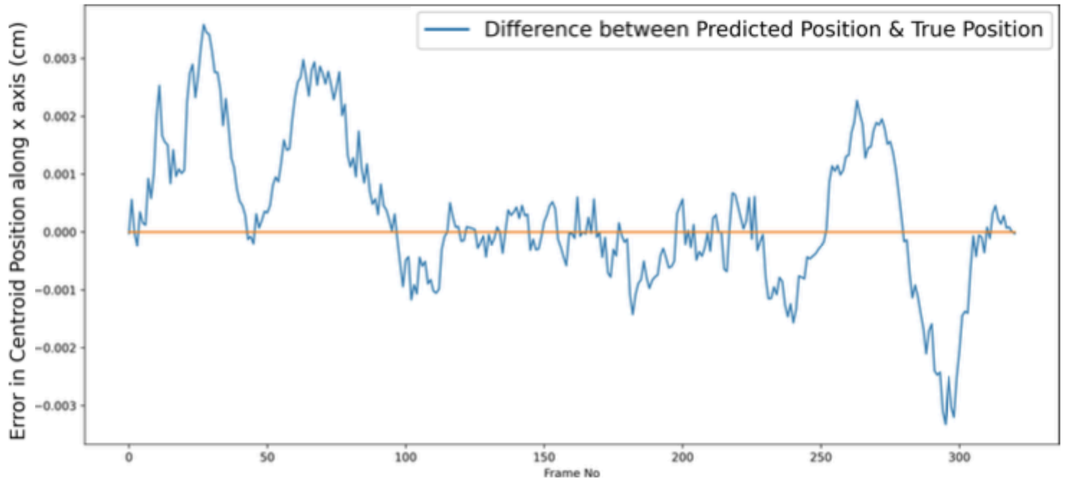
Movement along Y axis only

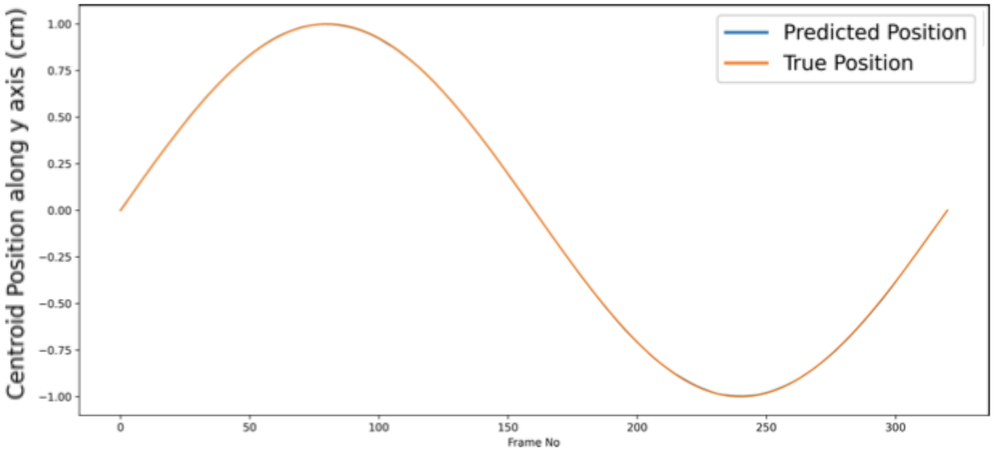


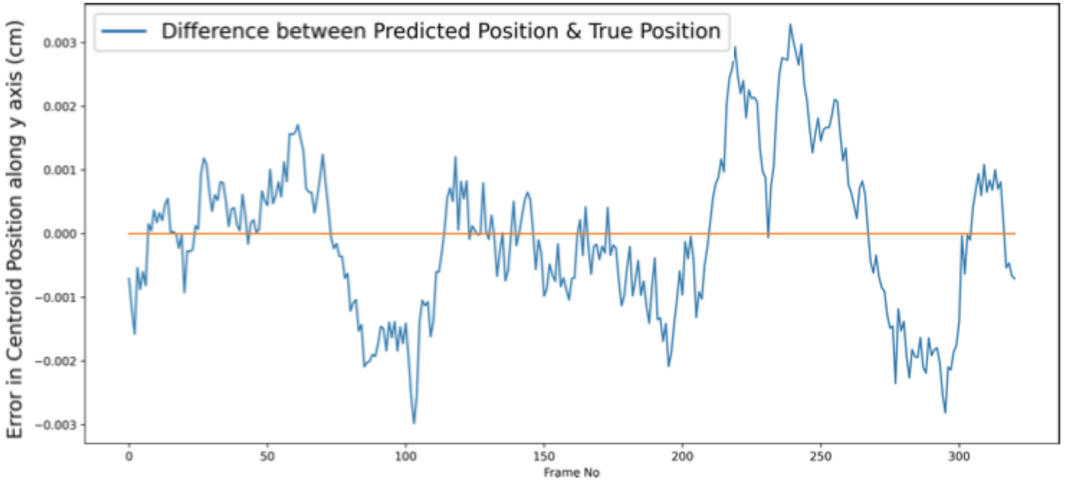
Results for Scattered Beam

Movement along both X and Y axis



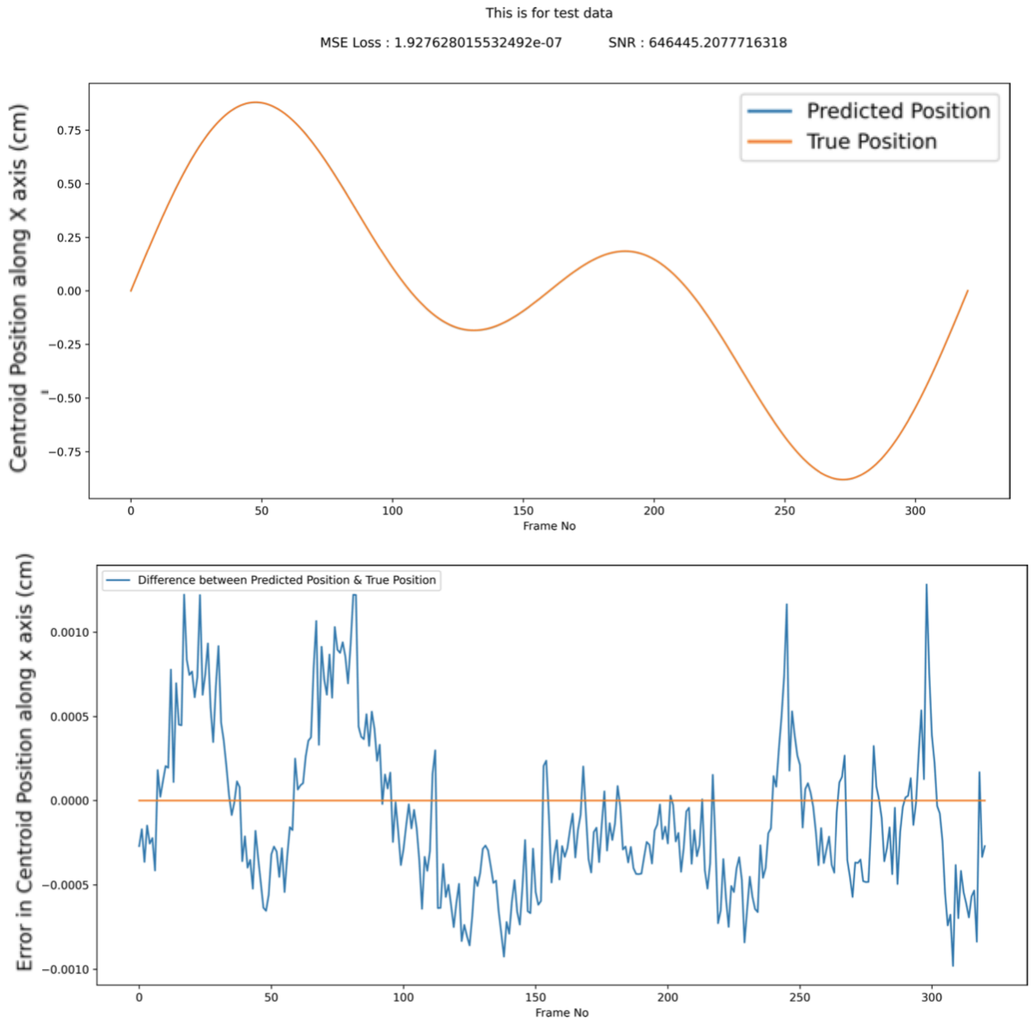




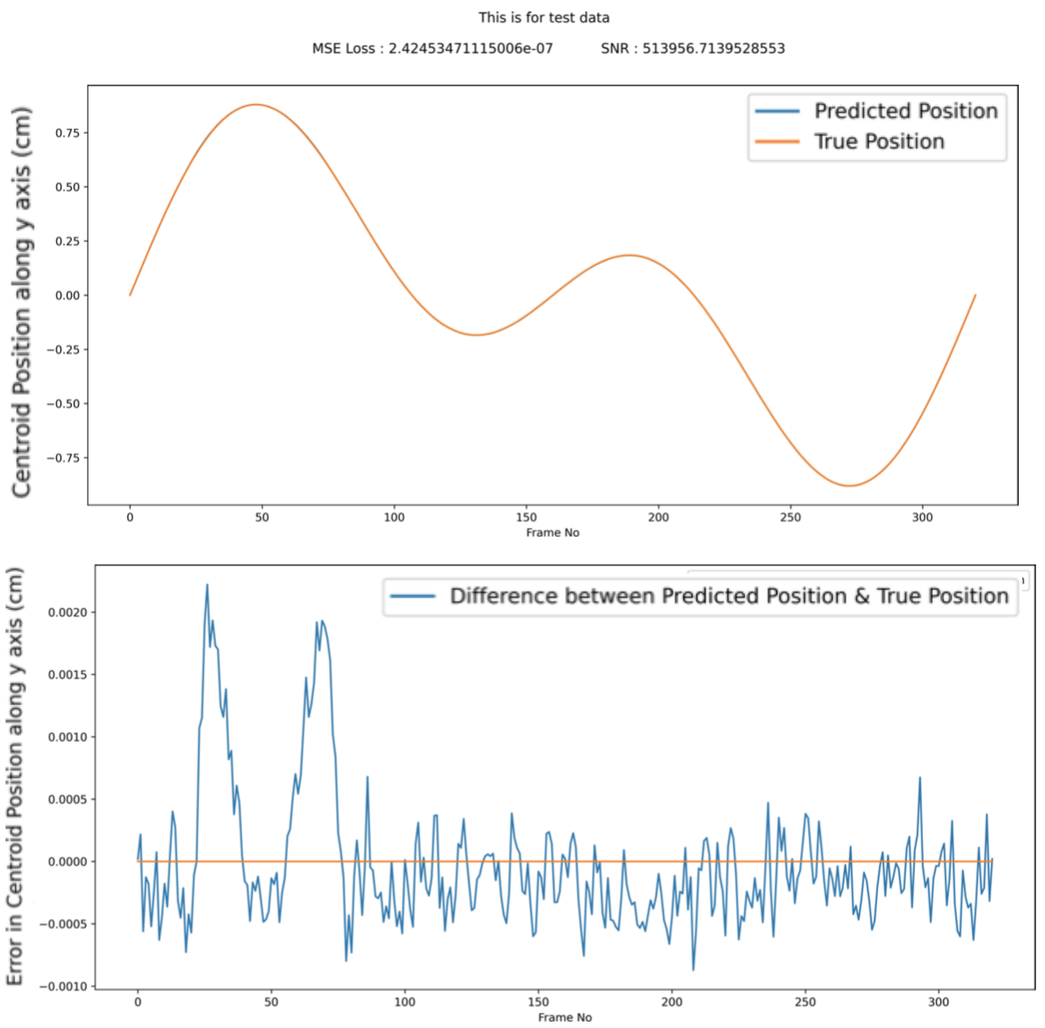


Results for Gaussian Beam with CCD noise

Movement along X axis only

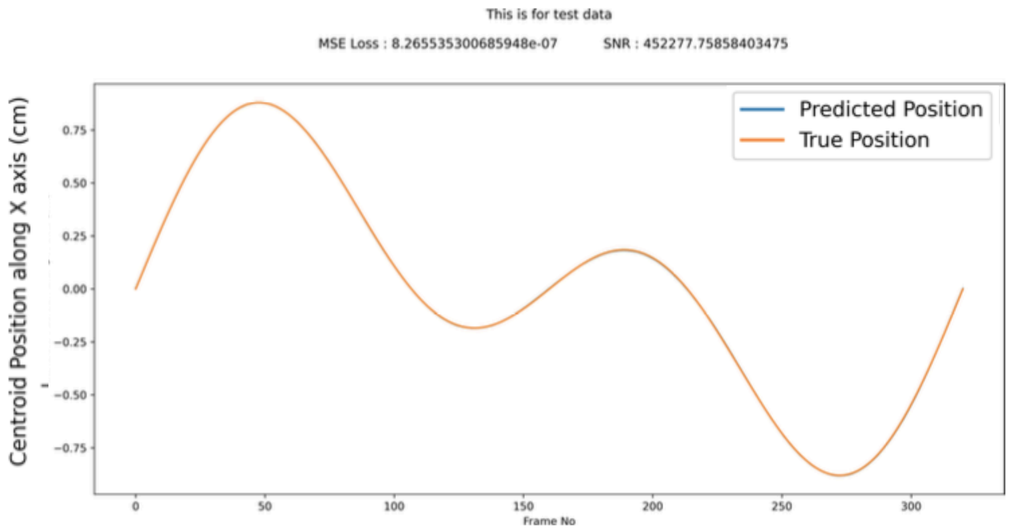


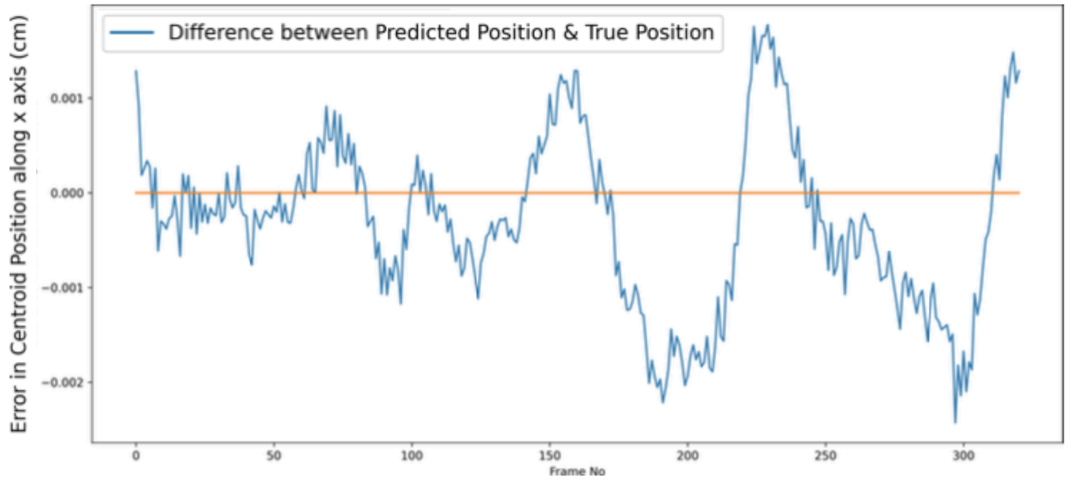
Movement along Y axis only

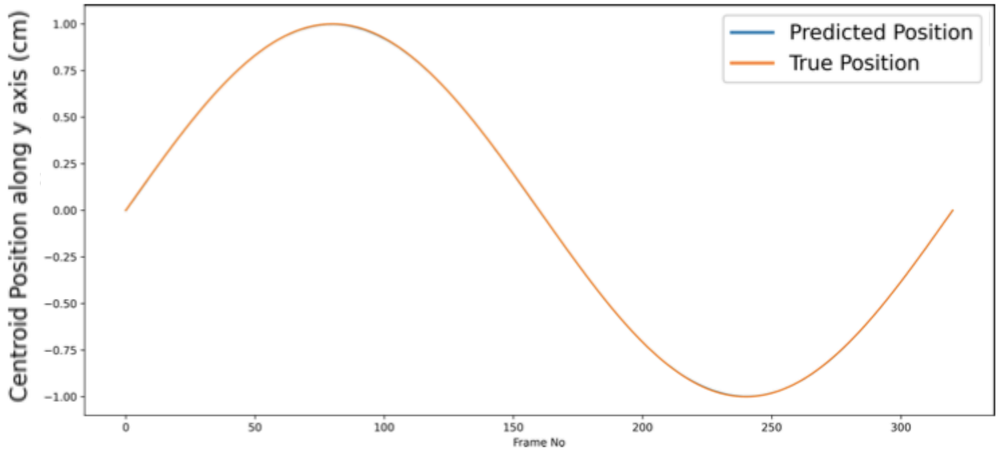


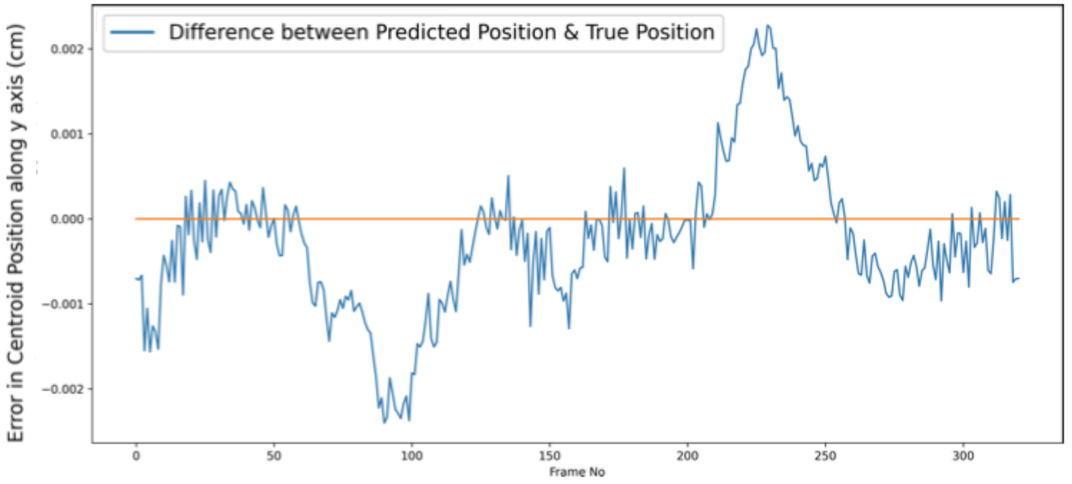
Results for Gaussian Beam with CCD noise

Movement along both X and Y axis



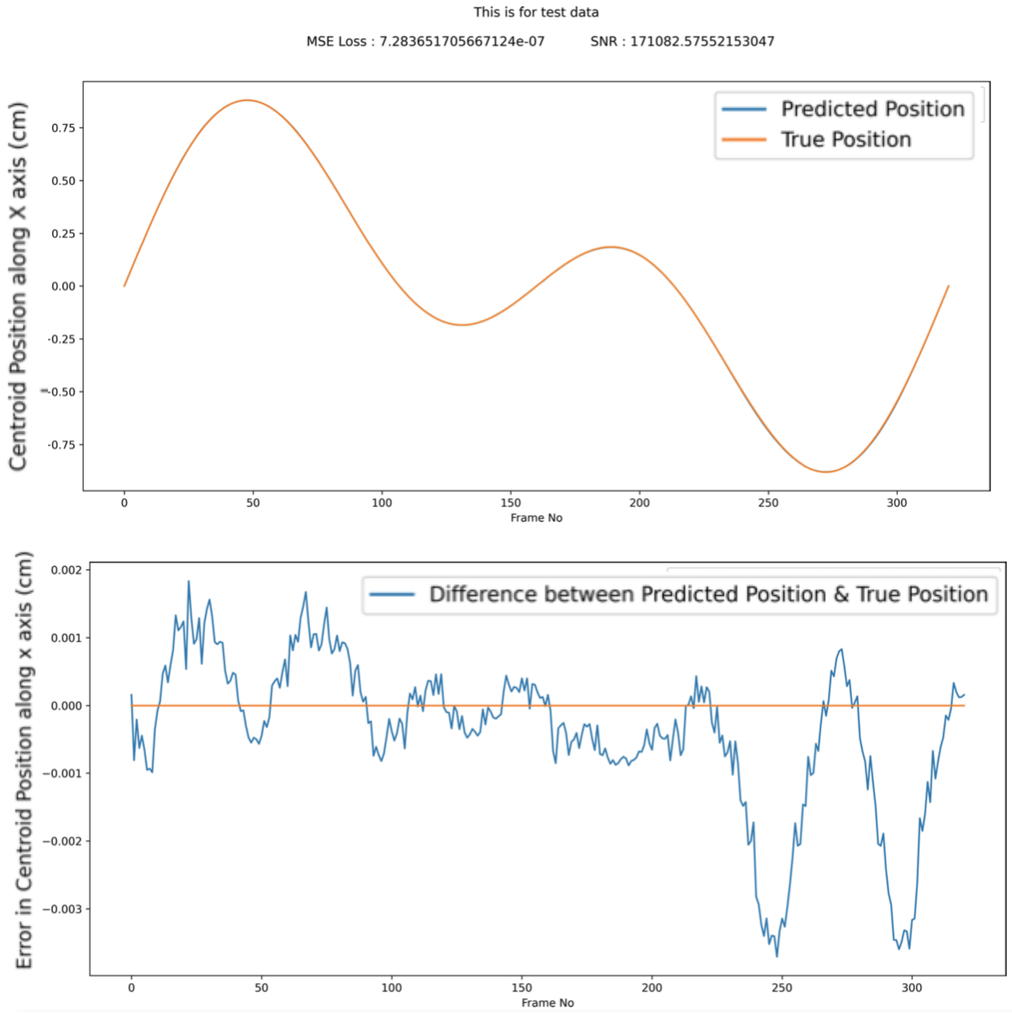




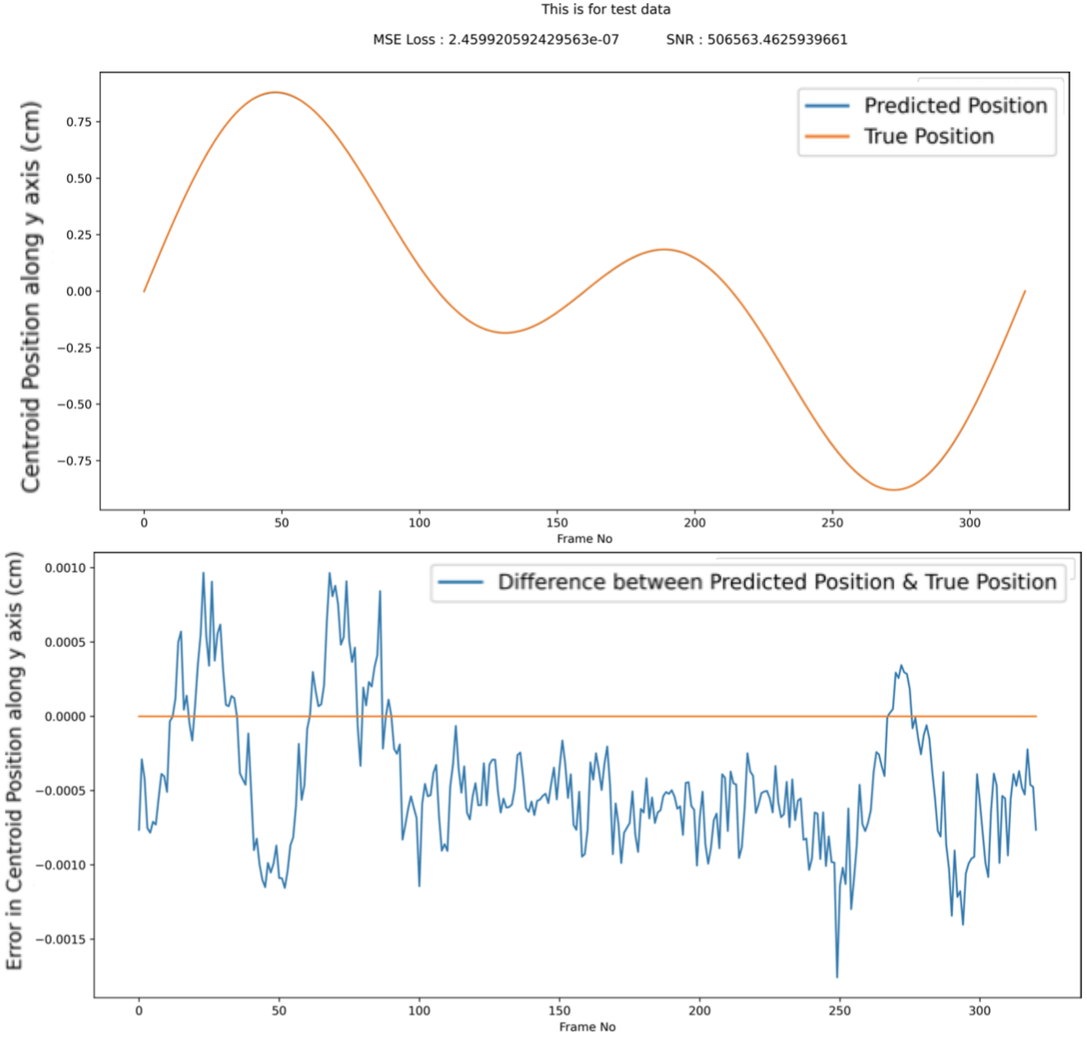


Results for Scattered Beam with CCD noise

Movement along X axis only

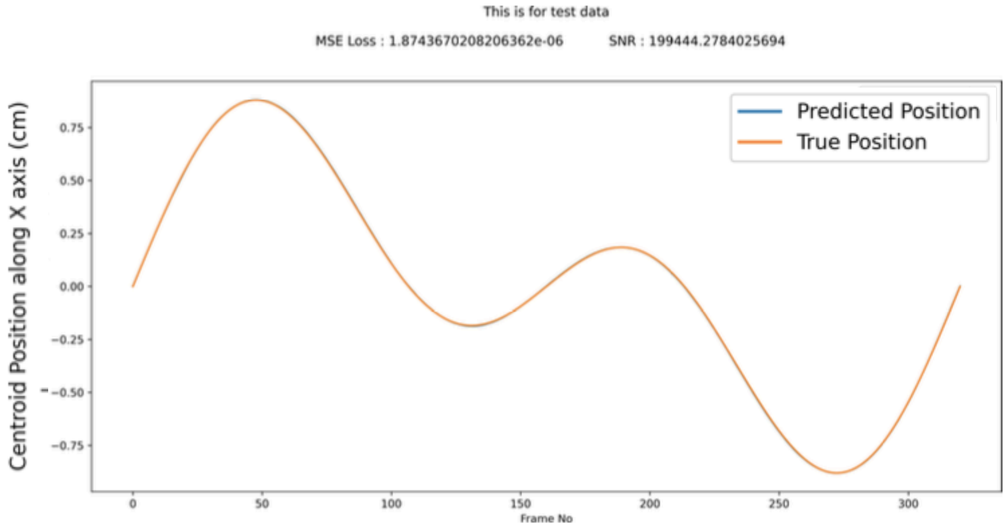


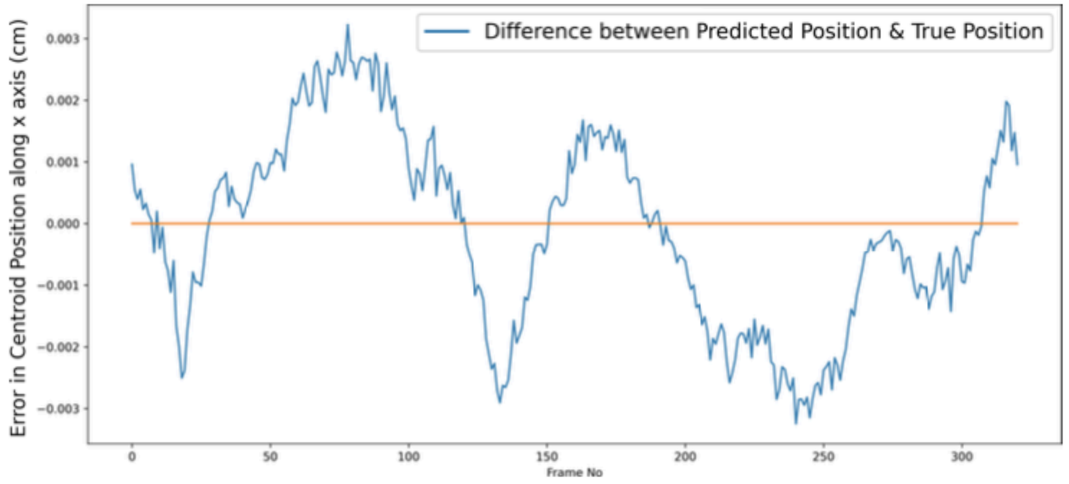
Movement along Y axis only

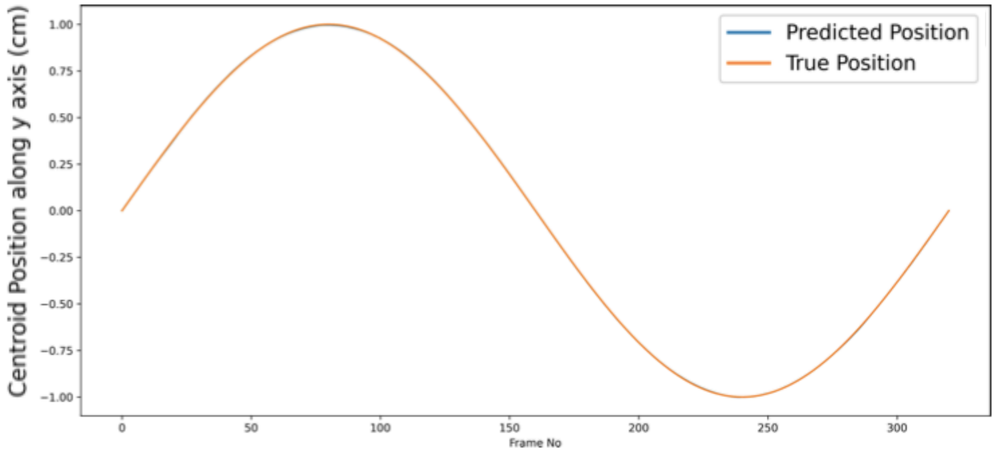


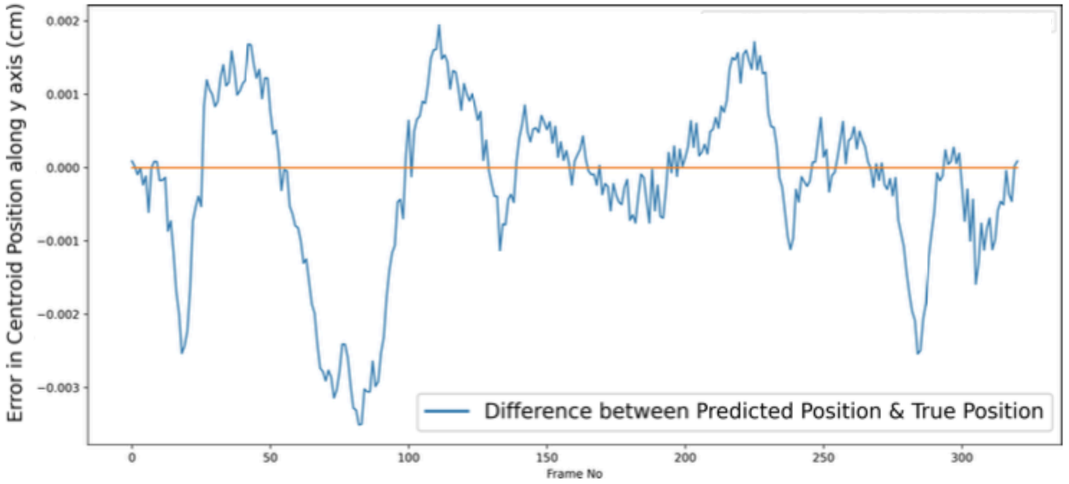
Results for Scattered Beam with CCD noise

Movement along both X and Y axis









LIGO

Summary

- We have been able to reach sub-pixel accuracy
- Maximum error in detecting centroid is 40 micron

Acknowledgement

- I am grateful to my mentors Prof. Rana Adhikari, Dr. Yehonathan Drori and Dr. Tega Edo for their guidance throughout the project.
- I would like to thank other fellow SURF students for making this a memorable summer.
- I am thankful to Prof. Alan Weinstein, all the members of Caltech LIGO Group, NSF, LIGO-India who have played a key role in enriching this internship experience

Thank You