Machine Learning Techniques for Fabry-Pérot Cavity Alignment

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Haraprasad Nandi

IISER PUNE

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

May, 2025

Supervisor: Dr. Sanjit Mitra

Certificate

This is to certify that this dissertation entitled Machine Learning Techniques for Fabry-Pérot Cavity Alignment towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Haraprasad Nandi at Inter-University Centre for Astronomy and Astrophysics, Pune under the supervision of Dr. Sanjit Mitra, Professor, Department of Physics , during the academic year 2024-2025.

Dr. Sanjit Mitra

Committee:

Dr. Sanjit Mitra

Dr. Umakant Rapol

This thesis is dedicated to my parents.

Declaration

I hereby declare that the matter embodied in the report entitled Machine Learning Techniques for Fabry-Pérot Cavity Alignment are the results of the work carried out by me at the Department of Physics, Inter-University Centre for Astronomy and Astrophysics, Pune, under the supervision of Dr. Sanjit Mitra and the same has not been submitted elsewhere for any other degree.

HNandi

Haraprasad Nandi

Acknowledgments

I would like to express my sincere gratitude to everyone who has supported me throughout the journey of this thesis.

First and foremost, I am deeply grateful to my thesis supervisor Dr. Sanjit Mitra for the invaluable guidance, encouragement, and insightful feedback that have been crucial in shaping this work.

I would also like to extend my heartfelt thanks to the scientific officers, Ashish Mhsake, who helped me shape this whole work to what it is today and T. R. Saravanan, whose expertise and support greatly contributed to the success of this project.

I am forever indebted to my parents for their unwavering support, love, and encouragement throughout this challenging yet rewarding journey.

I would also like to thank Aniket, Kwanit, Bisweswar, Anamitra, Shinjini, Manjima, Shalini, Ashmi for being with me through the thick and the thin and making the BSMS life less boring. I would like to thank Sneha for putting up with me through the years and making the memories that I cherish most.

To those who I forgot to mention because of the short timeline, I want to extend my thanks and love.

Thank you all for your continued support and belief in me.

List of Figures

List of Tables

Contents

Abstract

This thesis presents a novel approach to the alignment of Fabry-Pérot cavities using advanced machine learning techniques. Precise cavity alignment is critical for applications ranging from high-resolution spectroscopy and laser stabilization to gravitational wave detection. Traditional manual alignment methods are often labor-intensive and prone to error, motivating the need for automated solutions. Also, we want to understand how effectively we can implement modern techniques in GW interferometer alignments. In this work, theoretical framework and simulations of a cavity setup was done to model the beam profiles present in the cavity. A convolutional neural network (CNN) model was developed to recognize the order of the mode images based on the simulation. Finally, this model was integrated with a Reinforcement Learning (RL) model to predict adjustments of the redirection mirrors to perfect the alignment. In the validation phase of the model, the model showed high precision in aligning the beam by adjusting the redirection mirrors and bringing the cavity mode to zero order TEM mode. This demonstrates that machine-learning based alignment systems can improve on the alignment efficiency and continuous realignment procedures. It will also reduce calibration time maintaining operational precision. This paves a way towards fully automated cavity alignment systems even in operational systems.

Chapter 1

Introduction

Optical interferometers are central in the application of detection of waves that cannot be detected in traditional detectors based on other mechanisms. Laser interferometry has advanced to the point where detection of waves in the displacement sensitivity of $10^{-18}m$ is now possible over a four kilometers interferometer length, therefore necessitating the need of precise alignment and operational efficiency. My thesis provides an outline on using the latest advancements of automation using machine learning and computer vision techniques to precisely align Fabry-Pérot cavity setups. First we explore the underlying theory associated with Gaussian beams and optical cavities, followed by the alignment techniques.

1.1 Background and Motivation

Alignment of optical system to a very high precision plays a important role in spectroscopy, interferometry, laser stabilization, and gravitational wave detection. Normally this alignment is done using iterative adjustments and observing results thereof with other instruments. However, manual alignment comes with a lot of fallacies:

- Desired precision is not always achieved and repetitive adjustments are necessary.
- Manual adjustments are subject to a variety of errors like parallax and so on.

- Environmental disturbances are subject to happen during alignment before the application of isolation.

Automated alignment systems have turned out to be promising in removing these limitations. Machine Learning methods have provided to be particularly useful in finding inaccuracies and resolving them and thereby increase precision. The machine learning models are capable of modeling non-linear relationships and does precise alignment at the same time provides premises to changing circumstances and realignment when some external disturbances creep in.

1.2 Research Problem and Objectives

1.2.1 Research Problem

Optical systems involving resonant cavities such as the Fabry-Pérot are extremely sensitive to misalignments. Even slight angular or positional deviations can lead to failure of basic function, including power losses, mode distortions, and reduced finesse. These challenges become more mounted when the systems are complex and involves multiple optical components and cavities.

Problem Statement: How can the alignment of a Fabry-Pérot cavity setup be automated to achieve high precision and reproducibility, so that the highly complex setups can achieve higher precision without accounting for all the factors by using machine learnings modeling?

1.2.2 Objectives

The primary objectives of this thesis are:

1. Explore the theoretical perspectives of Gaussian beam propagation in a particular optical setup and derive the beam parameters and phase factors during various places of propagation while maintaining stability.
2. Design and perfect alignment of a Fabry-Pérot cavity and derive relations for spot size, mode volume, finesse, and quality factor.

3. Using feedback loop to identify alignment problems and explore higher order modes in the cavity due to slight misalignments.
4. To integrate machine learning algorithms into the alignment process, enabling automated adjustments based on real-time feedback.
5. To validate the proposed approach through simulation and experimental case studies, thereby demonstrating improvements in alignment precision and system performance.

1.3 Methodology Overview

Here, I list the methodologies undertaken during the work, mainly focusing on the theoretical work and then comparison with the physical setup that was constructed.

- **Theoretical Modeling:** Derivation of key equations and relationships starting from Maxwell's equations, leading to the paraxial wave equation and Gaussian beam solutions, beam propagation until the cavity and mode matching to remove the Laguerre Gaussian (LG) Modes.
- **Optical Cavity Analysis:** Detailed investigation into the stability, spot size, mode volume, and phase characteristics of Fabry-Pérot cavities, beam spot size matching using lenses to that of the cavity.
- **Interferometric Techniques:** Application of interferometric methods to diagnose alignment errors, usage of computer vision models to auto detect mode orders and providing relevant feedback to the neural network.
- **Machine Learning Integration:** Development and training of machine learning models to predict and correct alignment errors based on interferometric data.
- **Experimental Validation:** Implementation of the proposed alignment strategies in a laboratory setup, followed by performance evaluation through analysis of goal completion.

This approach with theoretical modeling and then implementation in the physical setup ensures the robustness and validation of the machine learning algorithm.

1.4 Summary

To summarise, this work tackles the challenges of manual alignment in Fabry-Pérot cavities by developing an automated machine learning based alignment system that combines optical physics, interferometry, and machine learning. In the next chapters we will go through theoretical derivations, experimental validations, and detailed methodologies used in my work and the results that we obtained.

Next, we move on to the theory of Gaussian beams which will help us in understanding beam propagation and alignment within optical cavities.

Chapter 2

Gaussian Beams

Laser physics and optics revolves around the usage of optical beams with a Gaussian profile. These Gaussian beams serve as the fundamental solutions to the paraxial wave equation. We begin our approach to Gaussian beams by working through Maxwell's equations, then deriving the paraxial approximation and Gaussian beam solution, and then determine beam parameters, mode structure, phase factors, higher order modes, and the ABCD matrix formalism. Finally, we discuss the practical considerations to emphasize the importance and wide use of Gaussian beams in various optical systems.

2.1 Introduction and Motivation

Laser beams are modeled generally by Gaussian Beams, it being the most common solution of the paraxial wave equation [2, 16]. Gaussian beams have well defined beam waist, Rayleigh range, and divergence which is essential in the mapping of the beam propagation, focusing and resonator design.Here we derive the Gaussian Beam equation from the first principle and then move on to higher order modes.

2.2 Theoretical Foundation

2.2.1 Maxwell's Equations and the Wave Equation

We start from the Maxwell's equations in free space. In the absence of charges and currents, we get the homogeneous wave equation for the electric field **E** from the Maxwell's equations:

$$\nabla^2\mathbf{E} - \mu_0\varepsilon_0\frac{\partial^2\mathbf{E}}{\partial t^2} = 0. \tag{2.1}$$

Now, we take the time dependence as $e^{-i\omega t}$, and we reduce the above to the time-independent Helmholtz equation:

$$\nabla^2\mathbf{E} + k^2\mathbf{E} = 0, \tag{2.2}$$

where $k = \omega\sqrt{\mu_0\varepsilon_0} = \frac{2\pi}{\lambda}$ [18].

2.2.2 Separation into Transverse and Longitudinal Components

Here, we assume, for practicality, that the field variation in the direction of propagation (z) is much slower than in the transverse directions (x and y). Therefore, we write the electric field as:

$$E(x,y,z) = E_0\,\psi(x,y,z)\,e^{-ikz}, \tag{2.3}$$

where $\psi(x,y,z)$ is varying slowly. Substituting (2.3) into (2.2) and assuming ψ is slowly varying along z gives the paraxial wave equation.

2.3 Derivation of the Paraxial Wave Equation

Starting with the ansatz in (2.3), the Laplacian is:

$$\nabla^2 E = \nabla_t^2 E + \frac{\partial^2 E}{\partial z^2}, \tag{2.4}$$

where ∇_t^2 represents the Laplacian in the transverse coordinates x and y.

2.3.1 Computing the Derivatives

We compute the derivatives of $E(x,y,z)$:

$$\frac{\partial E}{\partial z} = E_0 \left[\frac{\partial \psi}{\partial z} - ik\psi\right] e^{-ikz}, \tag{2.5}$$

$$\frac{\partial^2 E}{\partial z^2} = E_0 \left[\frac{\partial^2 \psi}{\partial z^2} - 2ik\frac{\partial \psi}{\partial z} - k^2\psi\right] e^{-ikz}. \tag{2.6}$$

Substituting into the Helmholtz equation (2.2), we get:

$$\nabla_t^2 \psi - 2ik\frac{\partial \psi}{\partial z} + \frac{\partial^2 \psi}{\partial z^2} = 0. \tag{2.7}$$

A full derivation of this approximation can be found in Siegman [2] and Kogelnik & Li [17].

In deriving the paraxial wave equation, we assume that the envelope function $\psi(x,y,z)$ varies slowly along the z-direction compared to the rapidly oscillating phase factor $\exp(-ikz)$. Mathematically, if L is the characteristic length scale over which ψ varies, then we have

$$\frac{\partial \psi}{\partial z} \sim \frac{\psi}{L} \quad \text{and} \quad \frac{\partial^2 \psi}{\partial z^2} \sim \frac{\psi}{L^2}.$$

Since the wavelength λ is much smaller than L (i.e. $\lambda \ll L$), the ratio λ/L is very small, and hence the term $\frac{\partial^2 \psi}{\partial z^2}$ is negligible compared to $2ik\frac{\partial \psi}{\partial z}$. This leads to the condition

$$\frac{\lambda}{L} \ll 1,$$

which is the fundamental requirement for the paraxial approximation to hold. Physically, this means that the beam remains well-collimated over distances much larger than its wavelength, ensuring that its profile does not change rapidly along the propagation direction. Therefore, we get:

$$\nabla_t^2 \psi - 2ik\frac{\partial \psi}{\partial z} = 0. \tag{2.8}$$

This is the paraxial wave equation.

2.4 Derivation of the Gaussian Beam Solution

A solution of the paraxial equation that satisfies proper boundary conditions is given by the Gaussian beam. We assume a solution of the form:

$$\psi(x,y,z) = \frac{w_0}{w(z)} \exp\left[-\frac{x^2+y^2}{w^2(z)}\right] \exp\left[-ik\frac{x^2+y^2}{2R(z)} + i\zeta(z)\right], \tag{2.9}$$

where:

- w_0 is the minimum beam waist (at $z = z_0$).
- $w(z)$ is the beam radius at position z.
- $R(z)$ is the radius of curvature of the wavefront.
- $\zeta(z)$ is the Gouy phase.

2.4.1 Substitution into the Paraxial Equation

Now, substituting (2.9) into (2.8) and matching the coefficients of same powers of x and y, we get a set of differential equations. The derivation gives:

$$w(z) = w_0\sqrt{1+\left(\frac{z-z_0}{z_R}\right)^2}, \tag{2.10}$$

$$R(z) = (z-z_0)\left[1+\left(\frac{z_R}{z-z_0}\right)^2\right], \tag{2.11}$$

$$\zeta(z) = \arctan\left(\frac{z-z_0}{z_R}\right), \tag{2.12}$$

and the Rayleigh range:

$$z_R = \frac{\pi w_0^2}{\lambda}. \tag{2.13}$$

2.4.2 Discussion of the Gaussian Solution

The solution in (2.9) shows the amplitude and phase distribution of the Gaussian beam. The following factors should be noted:

- The amplitude factor $\frac{w_0}{w(z)}$ ensures energy conservation.
- The exponential term with $x^2 + y^2$ in the numerator determines the spatial decay.
- The quadratic phase term involving $R(z)$ represents the curvature of the beam.
- The Gouy phase $\zeta(z)$ adds an extra phase shift that is present in focused beams.

2.5 Beam Parameters and Their Physical Significance

2.5.1 Beam Waist and Divergence

The beam waist w_0 is the location of the beam where the beam is focused most tightly. From (2.10), we can see that $w(z)$ increases with $|z - z_0|$. The divergence angle θ of the beam for $z \gg z_R$ is:

$$\theta \approx \frac{\lambda}{\pi w_0}. \tag{2.14}$$

2.5.2 Rayleigh Range

The Rayleigh range z_R (see (2.13)) is the distance over which the beam radius increases by a factor of $\sqrt{2}$. It quantifies how quickly the beam diverges.

2.5.3 Wavefront Curvature

The radius of curvature $R(z)$ (see (2.11)) is the curvature of the beam's phase fronts. At the beam waist $z = z_0$, $R(z) \to \infty$ (a planar wavefront), while far from the waist, $R(z) \approx z - z_0$.

2.5.4 Gouy Phase Shift

The Gouy phase $\zeta(z)$ (see (2.12)) represents an additional phase shift that adds to the beam upon propagation. Over the full propagation from $-\infty$ to $+\infty$, the total Gouy phase shift is π for the fundamental mode.

2.6 Higher Order Modes

Practical laser beams contains higher order modes in addition to the fundamental TEM_{00} Gaussian mode.

2.6.1 Hermite-Gaussian Modes

The Hermite-Gaussian (HG) modes are solutions to the paraxial wave equation in Cartesian coordinates. They are expressed as:

$$\begin{aligned}\psi_{mn}(x,y,z) = \psi_{00}(x,y,z)\, H_m\left(\sqrt{2}\frac{x}{w(z)}\right) H_n\left(\sqrt{2}\frac{y}{w(z)}\right)\\ \times \exp\left[-i(m+n)\zeta(z)\right],\end{aligned} \tag{2.15}$$

where H_m and H_n are the Hermite polynomials of orders m and n. HG modes appear due to x and y misalignments of the input beam to the cavity.

2.6.2 Laguerre-Gaussian Modes

The Laguerre-Gaussian (LG) modes are solutions in cylindrical coordinates with an azimuthal index l and a radial index p:

$$\psi_{pl}(r,\phi,z) = \frac{w_0}{w(z)}\left(\frac{\sqrt{2}r}{w(z)}\right)^{|l|} L_p^{|l|}\left(\frac{2r^2}{w^2(z)}\right)\exp\left(-\frac{r^2}{w^2(z)}\right)$$
$$\times\exp\left[-ik\frac{r^2}{2R(z)}\right]\exp\left[-i(2p+|l|+1)\zeta(z)\right]\exp(il\phi), \tag{2.16}$$

where $L_p^{|l|}$ are the generalized Laguerre polynomials. We will be able to see LG modes if there is a mismatch of the radius of the beam waist of the cavity and the input beam.

2.7 Phase Factors and Their Significance

2.7.1 Longitudinal Phase

The term $\exp(-ikz)$ in (2.3) shows the rapid phase oscillation due to propagation. This longitudinal phase is the dominates the field.

2.7.2 Transverse Phase

The quadratic phase factor $\exp\left[-ik\frac{x^2+y^2}{2R(z)}\right]$ shows the curvature of the wave fronts. This phase factor is responsible for the focusing properties and beam divergence.

2.7.3 Gouy Phase

As mentioned before in Section 2.5, the Gouy phase $\zeta(z)$ leads to an additional phase shift which affects interference and mode matching. The Gouy phase represents an extra phase shift due to diffraction, emerging as a beam converges and diverges, and it reflects the gradual evolution of the beam's wavefront curvature near the focal region.

2.8 ABCD Matrix Formalism and the q-Parameter

2.8.1 Definition of the q-Parameter

The complex beam parameter $q(z)$ is defined as:

$$\frac{1}{q(z)} = \frac{1}{R(z)} - i\frac{\lambda}{\pi w^2(z)}. \tag{2.17}$$

This parameter captures the curvature and width of the beam and helps in the analysis of beam propagation.

2.8.2 Propagation Through Optical Systems

When a beam propagates through an optical system represented by an ABCD matrix, the q-parameter changes as:

$$q_2 = \frac{Aq_1 + B}{Cq_1 + D}. \tag{2.18}$$

This relation helps us in calculating how beam parameters change after passing through lenses, mirrors, and free space.

2.8.3 ABCD Matrix: Focusing by a Thin Lens

Let us take a thin lens with focal length f. The ABCD matrix for the lens is:

$$\begin{pmatrix} 1 & 0 \\ -1/f & 1 \end{pmatrix}. \tag{2.19}$$

Taking the input beam to have a parameter q_1, then after the lens, the parameter becomes:

$$q_2 = \frac{q_1}{-q_1/f + 1}. \tag{2.20}$$

From here, we can calculate the new beam waist and position after focusing.

2.9 Numerical Analysis & Simulation Profile

2.9.1 Calculation of Beam Parameters

For a laser with wavelength $\lambda = 632.8\,\mathrm{nm}$ and a beam waist (radius) $w_0 = 0.405\,\mathrm{mm}$, the Rayleigh range is given by:

$$z_R = \frac{\pi w_0^2}{\lambda} = \frac{\pi \left(0.405 \times 10^{-3}\,\mathrm{m}\right)^2}{632.8 \times 10^{-9}\,\mathrm{m}} \approx 0.815\,\mathrm{m}.$$

At a distance $z = 3\,\mathrm{m}$ from the waist, the beam radius becomes:

$$w(z) = w_0\sqrt{1+\left(\frac{z}{z_R}\right)^2} = 0.405\,\mathrm{mm}\sqrt{1+\left(\frac{3}{0.815}\right)^2} \approx 0.405\,\mathrm{mm}\sqrt{14.54} \approx 1.55\,\mathrm{mm}.$$

2.9.2 Knife's Edge Method for Beam Parameter Measurement

The Knife's Edge method is a classical technique for determining key spatial characteristics of a laser beam—such as the beam diameter, waist, and divergence. In this method, a sharp edge is translated across the beam profile while recording the transmitted power as a function of the knife edge position.

Theory

Assuming a Gaussian intensity profile, the beam intensity is given by:

$$I(x,y) = I_0 \exp\left(-2\,\frac{x^2+y^2}{w^2}\right),$$

where I_0 is the peak intensity, w is the beam radius (defined at the $1/e^2$ intensity level), and x and y are the spatial coordinates.

As the knife edge moves, it progressively blocks the beam. The transmitted power $P(x)$ can be expressed by integrating the unblocked portion of the intensity profile:

$$P(x) = \frac{P_0}{2}\left[1 - \operatorname{erf}\left(\frac{\sqrt{2}x}{w}\right)\right],$$

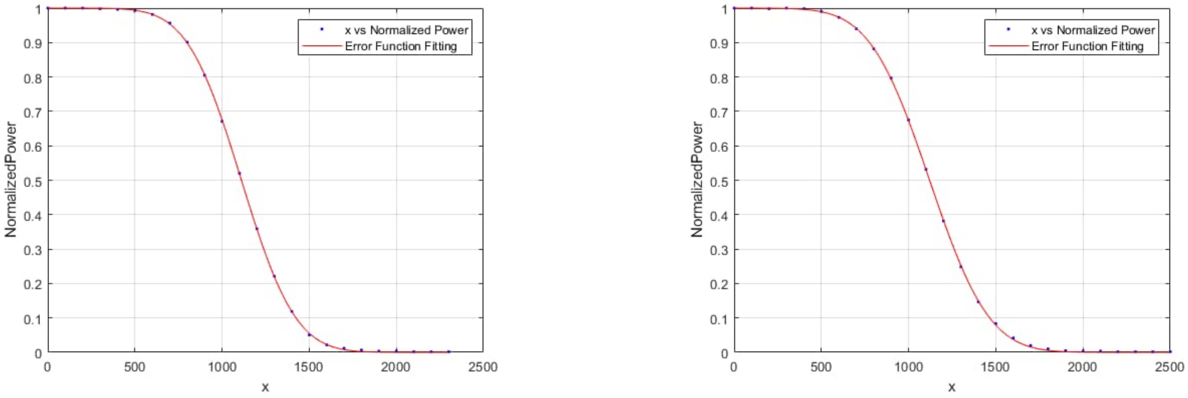
where P_0 is the total beam power and the error function is defined as:

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2}\, dt.$$

Plotting $P(x)$ versus x enables the extraction of the beam radius w from the slope of the transition region.

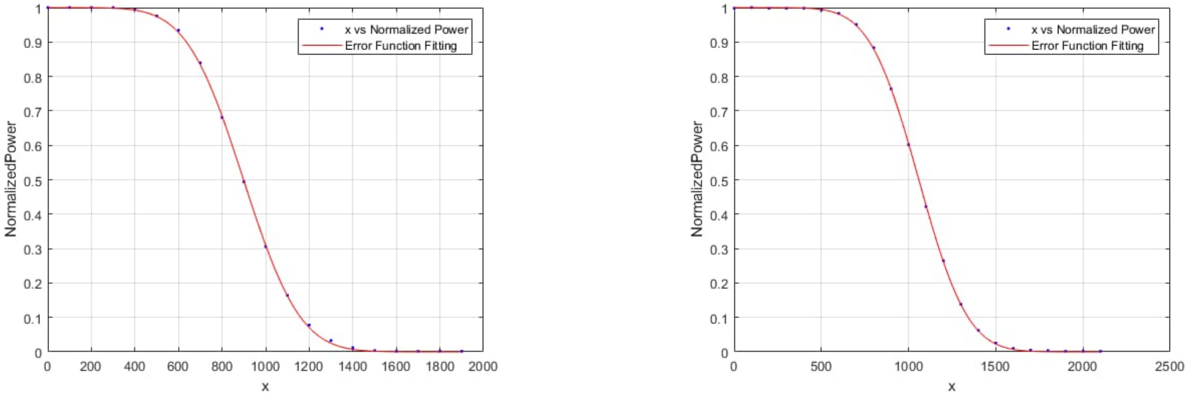
Experimental Results

Figure 2.1 displays the experimental data obtained using the Knife's Edge method under various conditions. The subfigures illustrate the measurements at different positions and the evolution of the beam waist.



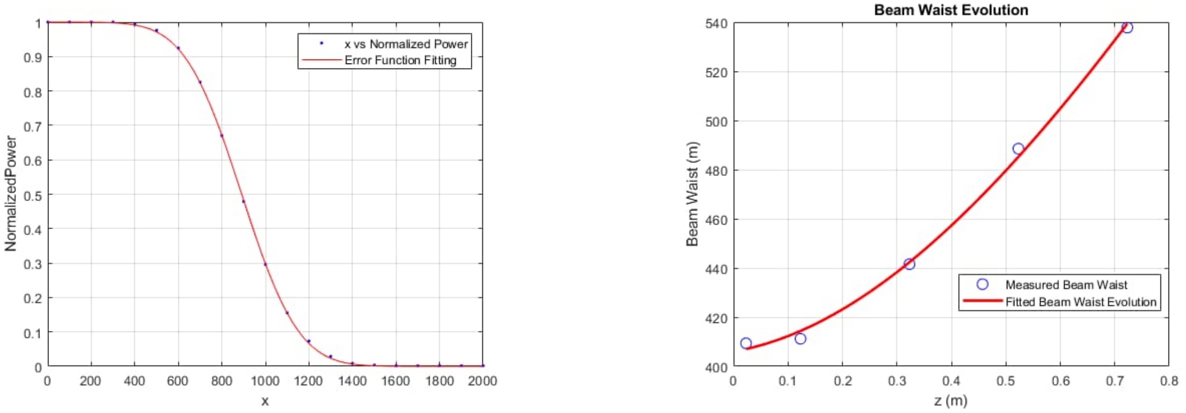
(a) Position 1

(b) Position 2



(c) Position 3

(d) Position 4



(e) Position 5

(f) Beam Waist Evolution

Figure 2.1: Experimental graphs from the Knife's Edge method showing the beam profile and waist evolution at different positions.

The extracted beam parameters from these measurements are summarized in Table 2.1.

Position (m)	Beam Waist w (μm)	Beam Center x_0
0.023	409.37	898.533
0.123	411.197	891.08
0.323	441.538	1058.8
0.523	488.508	1111.18
0.723	537.753	1120.58

Table 2.1: Measured beam waist and beam center at various positions.

Beam Waist Evolution Fit

A fit to the beam waist evolution data yields the following parameters (with 95% confidence bounds, units omitted):

- $w_0 = 404.4$ (378.1, 430.8)
- $z_0 = -0.08032$ (-0.396, 0.2354)
- $z_r = 0.9104$ (0.6339, 1.187)

The beam waist position is determined to be $z_0 = -0.08032$ m.

2.9.3 Simulation of Beam Propagation

Using python, we plot the evolution of the beam's intensity profile. Figure 2.2 shows the beam profile at various z-positions, with the increasing beam diameter and the change of the phase front.

2.9.4 Impact of Higher Order Modes

Small additions of higher order modes in the simulation shows distortions in the interference fringes. This emphasizes the need of pure modes and how perfect alignment can help in achieving pure modes and restrict higher order modes in the optical setup.

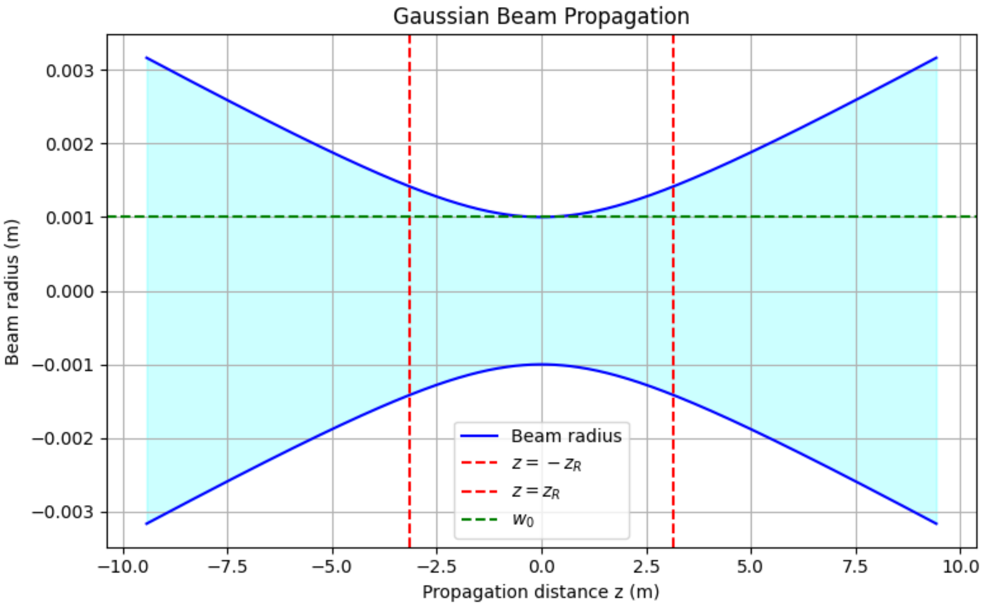


Figure 2.2: Gaussian Beam Propagation Evolution.

2.10 Summary

Here, in this chapter we outlined the basic concepts and theories associated with Gaussian Beams and their propagation methods and applications, starting from the Maxwell's equations combined with the parameter of paraxial approximation. We also derived the various parameters like the beam waist, Rayleigh range, wavefront curvature, Guoy phase. We extended the discussion to higher order mode solution to the equation and the relevant Hermite-Gaussian and Laguerre-Gaussian modes along with the simulations, which reflect the necessity of alignment to remove the higher order modes from an optical setup to prevent distortions among others.

Also, we discussed the ABCD matrix formalism and the complex q-parameter approach, which simplify the analysis of beam propagation through various optical elements.

Next, we move on to discuss optical cavities and their properties.

Chapter 3

Optical Cavities

Here, we will be discussing optical cavities which are very essential in precision instruments and detectors. We start from the propagation of Gaussian Beams and then move on to derive the stability conditions. Following this, we find the evolution of the beam spot size, effective mode volume and then we find the fundamental properties related to Fabry-Pérot cavity.

3.1 Gaussian Beams in Resonators and Stability Conditions

While propagation in an optical resonator, the Gaussian Beam must retrace the same path after every round trip. This is described by the complex beam parameter $q(z)$:

$$\frac{1}{q(z)} = \frac{1}{R(z)} - i\frac{\lambda}{\pi w^2(z)}, \tag{3.1}$$

[2, 18] where $R(z)$ is the radius of curvature of the beam's wavefronts, $w(z)$ is the beam radius, and λ is the wavelength.

3.1.1 Self-Consistency and the ABCD Matrix Formalism

For a resonator composed of two mirrors with radii of curvature R_1 and R_2 separated by a distance L, the beam parameter transforms according to the ABCD matrix:

$$q_{\text{out}} = \frac{A\,q_{\text{in}} + B}{C\,q_{\text{in}} + D}. \tag{3.2}$$

For self-consistency in every round trip, we require:

$$q_{\text{out}} = q_{\text{in}}. \tag{3.3}$$

From the above equation, we can derive the beam parameters. Therefore, only certain values of q will reproduce themselves after one complete round-trip through the cavity.

3.1.2 Stability Condition

For stability, the beam must remain confined and should not diverge to infinity. For a two-mirror cavity, the stability condition is expressed as:

$$0 \le g_1 g_2 \le 1, \tag{3.4}$$

[2, 16] where

$$g_i = 1 - \frac{L}{R_i}, \quad (i = 1, 2). \tag{3.5}$$

Detailed ABCD Matrix Formulation

Consider the following matrices:

1. **Propagation over a distance L:**

$$M_{\text{prop}} = \begin{pmatrix} 1 & L \\ 0 & 1 \end{pmatrix}. \tag{3.6}$$

2. **Reflection off a spherical mirror of radius R:**

$$M_{\text{mirror}} = \begin{pmatrix} 1 & 0 \\ -\frac{2}{R} & 1 \end{pmatrix}. \tag{3.7}$$

For a full round-trip in a symmetric cavity, the total ABCD matrix therefore becomes:

$$M = M_{\text{prop}} \cdot M_{\text{mirror}} \cdot M_{\text{prop}} \cdot M_{\text{mirror}}. \tag{3.8}$$

After doing the matrix multiplications, we get a matrix with elements A, B, C, and D. The eigenvalues of M are:

$$\lambda = \frac{(A+D) \pm \sqrt{(A+D)^2 - 4}}{2}. \tag{3.9}$$

For the cavity to be stable, these eigenvalues must have a magnitude less than or equal to 1. This means:

$$|A+D| \leq 2. \tag{3.10}$$

Substituting $A+D$ in terms of g_1 and g_2 gives us:

$$0 \leq g_1 g_2 \leq 1. \tag{3.11}$$

Therefore, if the cavity parameters satisfy this inequality, the beam will remain confined and the resonator will operate in a stable configuration.

3.2 Spot Size and Mode Volume Relations

3.2.1 Evolution of the Beam Spot Size

The Gaussian beam's spot size $w(z)$ changes along the direction of propagation. The minimum beam waist w_0 is located at $z = z_0$, and the spot size at any point z is given by:

$$w(z) = w_0 \sqrt{1 + \left(\frac{z - z_0}{z_R}\right)^2}, \tag{3.12}$$

where the Rayleigh range z_R is defined as:

$$z_R = \frac{\pi w_0^2}{\lambda}. \tag{3.13}$$

The Rayleigh range is the distance over which the beam cross-section expands highly (nearly doubles in area).

3.2.2 Effective Mode Volume

The effective mode volume V measures how confined the electromagnetic field is within the cavity. A detailed definition is the integral over the electric field energy density:

$$V = \frac{\int \varepsilon(\mathbf{r}) \, |\mathbf{E}(\mathbf{r})|^2 \, dV}{\max\{\varepsilon(\mathbf{r}) \, |\mathbf{E}(\mathbf{r})|^2\}}, \tag{3.14}$$

but for a Fabry-Pérot cavity, we can write it as:

$$V \approx \pi w_0^2 L_{\text{eff}}, \tag{3.15}$$

[19]

where L_{eff} is the effective cavity length. For higher order modes, the effective mode volume increases due to the broader spatial distribution and the presence of additional nodes, resulting in a less confined electromagnetic field within the cavity

3.3 Resonant Optical Cavities: The Fabry-Pérot Cavity

3.3.1 Basic Configuration and Resonance Condition

A Fabry-Pérot cavity is formed of two mirrors facing each other. The resonance condition for an ideal cavity is that the optical path length is an integral multiple of the wavelength:

$$2L = m\lambda, \quad m \in \mathbb{Z}. \tag{3.16}$$

[18]

In real cavities, several phase shifts must be taken into account to modify this condition.

3.3.2 Detailed Analysis of Phase Shifts

The sources of phase shifts in a Fabry-Pérot cavity are as follows:

- **Propagation Phase:** The phase added due to propagation over the distance L is given by:

$$\phi_{\text{prop}} = \frac{2\pi}{\lambda} \cdot 2L. \tag{3.17}$$

- **Mirror Phase Shift:** Each reflection at a mirror can add a phase shift of ϕ_{mirror}.
- **Gouy Phase Shift:** A Gaussian beam gets an extra phase known as the Gouy phase shift, $\zeta(z)$, which, for a round-trip, contributes a net phase shift of 2ζ. For a beam focused at z_0, the Gouy phase is:

$$\zeta(z) = \arctan\left(\frac{z - z_0}{z_R}\right). \tag{3.18}$$

Therefore, the total round-trip phase shift is:

$$\Phi_{\text{total}} = \phi_{\text{prop}} + \phi_{\text{mirror}} + 2\zeta. \tag{3.19}$$

For resonance, we need:

$$\Phi_{\text{total}} = 2\pi m, \quad m \in \mathbb{Z}, \tag{3.20}$$

or,

$$\frac{4\pi L}{\lambda} + \Delta\phi_{\text{mirror}} + 2\zeta = 2\pi m. \tag{3.21}$$

This expression predicts accurately the resonance frequencies in real optical cavities.

3.3.3 Finesse, Quality Factor, and Photon Lifetime

The performance of a Fabry-Pérot cavity is quantified by several parameters:

- **Finesse ($\mathscr{F}$):** It is defined as the ratio of the free spectral range (FSR) to the resonance linewidth $\Delta\nu$:

  $$\mathscr{F} = \frac{\text{FSR}}{\Delta\nu}. \tag{3.22}$$

  For an ideal cavity with mirror reflectivity R, the finesse can be written as :

  $$\mathscr{F} = \frac{\pi\sqrt{R}}{1-R}. \tag{3.23}$$

  [2]

- **Quality Factor (Q):** It measures the energy stored relative to the energy lost per cycle:

  $$Q = \frac{\omega_0}{\Delta\omega}, \tag{3.24}$$

  where ω_0 is the resonance angular frequency and $\Delta\omega$ is the resonance linewidth.

- **Photon Lifetime (τ):** The average time a photon remains in the cavity is:

  $$\tau = \frac{Q}{\omega_0}. \tag{3.25}$$

  [21]

3.4 Higher Order Modes and Transverse Mode Spacing

3.4.1 Fundamental and Higher Order Modes

In an ideal cavity, the lowest order mode is the TEM_{00} mode, which has a Gaussian profile. Higher order modes, such as the Hermite-Gaussian (HG) and Laguerre-Gaussian (LG) modes, show more complicated spatial profiles. They can be expressed as:

$$\psi_{mn}(x,y,z) = \psi_{00}(x,y,z)\, H_m\left(\sqrt{2}\frac{x}{w(z)}\right) H_n\left(\sqrt{2}\frac{y}{w(z)}\right) \exp\left[-j(m+n)\zeta(z)\right], \tag{3.26}$$

where H_m and H_n are Hermite polynomials of orders m and n respectively.

3.4.2 Transverse Mode Frequency Spacing

Due to the added Gouy phase shift, the resonant frequencies of the higher order modes are slightly shifted compared to the fundamental mode. The resonance frequency for the (m,n) mode is:

$$\omega_{qmn} = \omega_q + (m+n+1)\Delta\omega_{\mathrm{Gouy}}, \tag{3.27}$$

with ω_q being the fundamental resonance frequency and $\Delta\omega_{\mathrm{Gouy}}$ is the spacing induced by the Gouy phase.

3.5 Analysis of Cavity Setup

3.5.1 Stability of a Symmetric Fabry-Pérot Cavity

In our setup, the cavity consists of one plane mirror ($R_1 = \infty$) and one curved mirror with a radius of curvature $R_2 = 1\,\mathrm{m}$. The stability of an optical cavity is determined by the g-parameters:

$$g_1 = 1 - \frac{L}{R_1}, \quad g_2 = 1 - \frac{L}{R_2},$$

where L is the mirror separation. The cavity is stable if

$$0 < g_1 g_2 < 1.$$

Since the plane mirror has $R_1 = \infty$, we have

$$g_1 = 1 - \frac{L}{\infty} = 1.$$

Thus, the stability condition reduces to

$$0 < g_2 = 1 - \frac{L}{1\,\mathrm{m}} < 1.$$

This implies that the cavity will be stable if

$$0 < L < 1\,\mathrm{m}.$$

For a numerical example, with a laser wavelength $\lambda = 632.8\,\mathrm{nm}$, the cavity remains stable provided that the mirror separation L satisfies the above condition. A detailed numerical investigation by varying L and computing the corresponding g-parameters is shown in Figure 3.1.

3.5.2 Finesse and Quality Factor

For our setup, we have initial mirror reflectivity of $R_1 = 0.90$ and an effective final mirror reflectivity of

$$R_2 = 0.99 \times 0.99 = 0.9801,$$

the finesse for a two-mirror cavity with unequal reflectivities is approximated by:

$$\mathscr{F} \approx \frac{\pi (R_1 R_2)^{1/4}}{1 - \sqrt{R_1 R_2}}.$$

We first compute:

$$R_1 R_2 = 0.90 \times 0.9801 = 0.88209,$$

$$\sqrt{R_1 R_2} \approx \sqrt{0.88209} \approx 0.9382, \quad \text{and} \quad (R_1 R_2)^{1/4} \approx \sqrt{0.9382} \approx 0.9687.$$

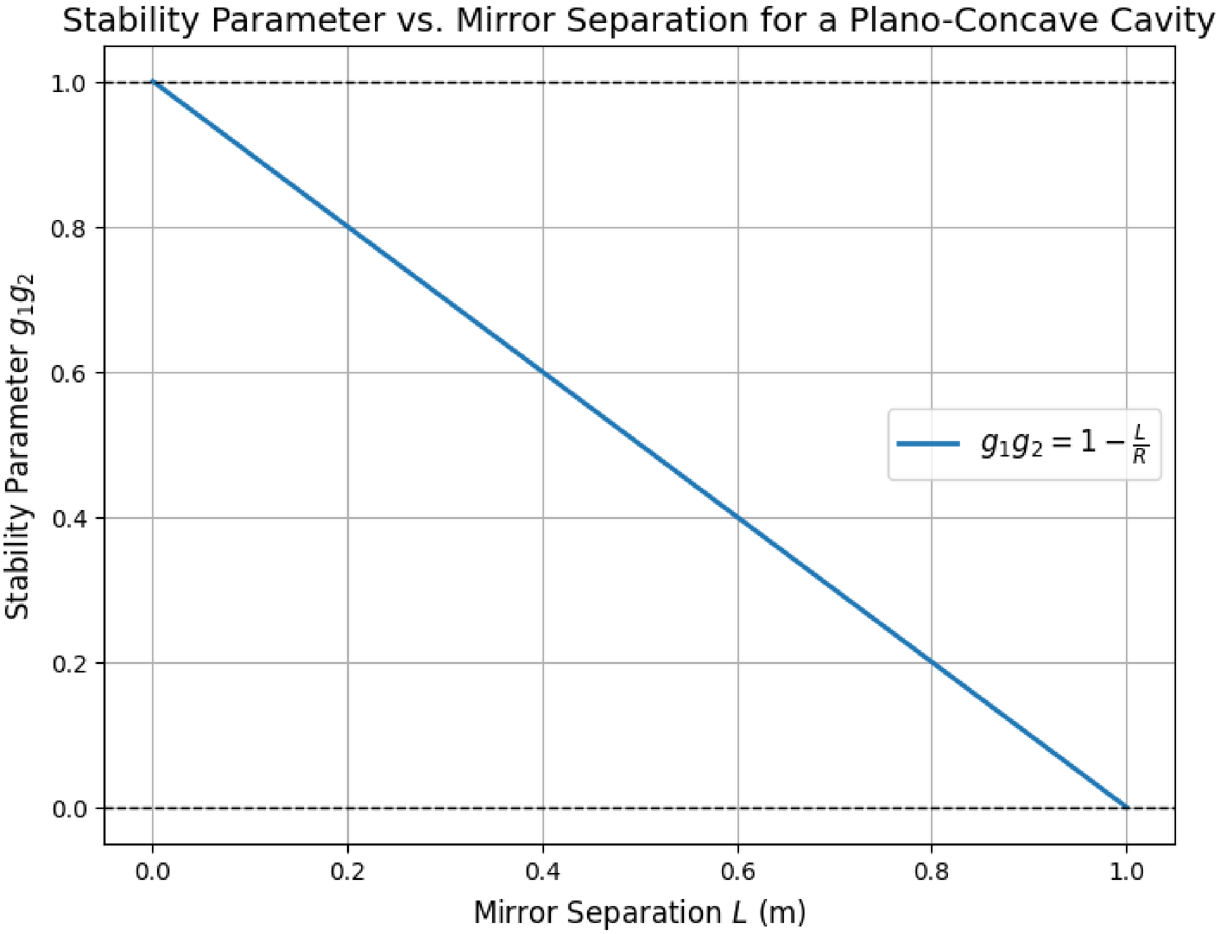


Figure 3.1: Stability Parameter for variation of L with g parameters.

Thus, the finesse becomes:

$$\mathscr{F} \approx \frac{\pi \times 0.9687}{1 - 0.9382} \approx \frac{3.1416 \times 0.9687}{0.0618} \approx 49.3.$$

For a cavity of length

$$L = 27.9\,\text{cm} = 0.279\,\text{m},$$

the free spectral range (FSR) is:

$$\text{FSR} = \frac{c}{2L} \approx \frac{3 \times 10^8\,\text{m/s}}{2 \times 0.279\,\text{m}} \approx 5.38 \times 10^8\,\text{Hz}.$$

The resonance linewidth is then:

$$\Delta\nu = \frac{\text{FSR}}{\mathscr{F}} \approx \frac{5.38 \times 10^8}{49.3} \approx 1.09 \times 10^7\,\text{Hz}.$$

For a laser with wavelength $\lambda = 632.8\,\text{nm}$,, the resonance frequency is:

$$\nu_0 = \frac{c}{\lambda} \approx \frac{3 \times 10^8\,\text{m/s}}{632.8 \times 10^{-9}\,\text{m}} \approx 4.74 \times 10^{14}\,\text{Hz}.$$

The quality factor Q is given by:

$$Q \approx \frac{\nu_0}{\Delta\nu} \approx \frac{4.74 \times 10^{14}}{1.09 \times 10^7} \approx 4.34 \times 10^7.$$

Finally, the photon lifetime τ is:

$$\tau = \frac{Q}{2\pi\nu_0} \approx \frac{4.34 \times 10^7}{2\pi \times 4.74 \times 10^{14}} \approx 1.46 \times 10^{-8}\,\text{s}.$$

3.6 Summary

Here, we began with the analysis of Gaussian beam propagation in resonators, deriving the self-consistency condition using the ABCD matrix formalism and finding the stability condition $0 \leq g_1 g_2 \leq 1$. We then moved to how the beam spot size evolves and mentioned the concept of the effective mode volume.

Next, the Fabry-Pérot cavity was analyzed in detail, including the derivation of the resonance condition while taking into account beam propagation, mirror, and Gouy phase shifts. We also defined performance parameters such as finesse, quality factor, and photon lifetime.

Chapter 4

Setup Simulation

I started with the details of the setup that we planned to test the alignment algorithm on. I used Python libraries and modules to simulate the setup in an ideal environment without any external disturbance. This simulation allows us to watch the behavior of the optical components and their ranges that we expect to observe in the physical setup. Here, I am detailing the components and the exact specifications of the setup that I used to simulate and then later physically assemble in the lab.

4.1 Experimental Setup Details

Some of the key optical components and their exact specifications are listed below:

- **Laser:** He-Ne Laser
    - *Power:* 5 mW
    - *Wavelength:* 632.8 nm
- **Quarter Wave Plate:** Adjustable Quarter Wave Plate (works for 632.8 nm wavelength).
- **Faraday Isolator:** Used to ensure unidirectional beam propagation.
- **Redirection Mirrors 1 & 2:** To steer the beam appropriately through the setup to ensure the perfect alignment with the cavity axis.

- **Cavity Mirror 1:** Broadband Mirror, 90% reflectivity.
- **Cavity Mirror 2 & 3:** Broadband Mirror, 99% reflectivity.
- **Detector:** A CAM (Raspberry Pi Global Shot Camera) placed beyond Cavity Mirror 2, outside the cavity.

The graphical representation of the setup is shown in Figure 4.1.

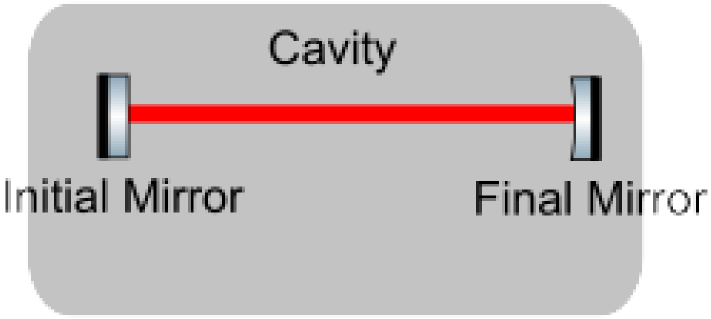


Figure 4.1: Schematic of the simulated Fabry-Pérot cavity setup.

The relevant distances between the optical components are as follows:

- **D1:** Distance between Laser & Redirection Mirror 1, 84.5*cm*
- **D2:** Distance between Redirection Mirror 1 & Redirection Mirror 2, 36.5*cm*
- **D3:** Distance between Redirection Mirror 2 & Cavity Mirror 1, 19.5*cm*
- **D4:** Arm Length of Cavity - 1, 15.5*cm*
- **D5:** Arm Length of Cavity - 2, 12.4*cm*

4.2 Simulation Environment and Methodology

The simulation was done using Python language with libraries like NumPy, SciPy, and Matplotlib [22–24]. The objective was to visualize the optical behavior of the setup, which includes the propagation of the Gaussian beam through each component and the detection of resultant mode images at the CAM after passing through the cavity.

4.2.1 Modeling the Optical Setup

The respective models of each components of the setup are as follows:

- **He-Ne Laser:** The laser beam is modeled as a fundamental Gaussian beam, with its parameters defined by the given power, wavelength, and beam waist (experimentally determined by knife's-edge method) [8, 20].
- **Quarter Wave Plate:** It adds a phase shift between the orthogonal polarization components of the beam (Not explicitly necessary when modeling an ideal setup, so excluded in final code).
- **Faraday Isolator:** This is added for unidirectional light propagation (Not explicitly necessary when modeling an ideal setup, so excluded in final code).
- **Redirection Mirrors:** Reflects the beam with negligible distortion, redirecting it through the optical path. Each one of them has a *xbeta* & *ybeta* component that regulates the correction in the beam's direction of propagation into the cavity.
- **Cavity Mirrors:** The reflectivities (90% for Mirror 1 and 99% for Mirrors 2 and 3) were used to calculate the multiple beam interference within the cavity. The reflectivities contribute to the Finesse and other parameters of the cavity.
- **Detector (CAM):** The camera's detection area was simulated by sampling the intensity distribution of the beam on exit from Cavity Mirror 2.

Higher order modes were introduced by addition of the phase in the phase front of the propagation. For simplicity, the redirection mirrors were assumed to add a particular phase to the beam propagation and therefore replaced with a phase addition in the implementation. In the case of the cavity, due to the middle mirror being a plane mirror, the cavity is treated as a two mirror cavity to ensure simplicity.

The relevant code for this section can be found in our GitHub repository [25].

4.3 Data Collection & Analysis

4.3.1 Beam Profiles and Mode Analysis

From the beam profile and different mode orders, we get the different mode images. As observed from the equations described in the chapter of Gaussian Beams, the Gaussian beam expands with increasing z, and the intensity distribution follows the typical bell-shaped curve for the zero order TEM mode. When small changes in phases are added to simulate misalignment or intentional excitation of higher order modes, the profiles exhibit additional lobes or asymmetries. Some of the beam intensity profile for Hermite-Gaussian modes are given below with their respective (m,n) in Figure 4.2.

4.3.2 Data Collection from Simulation

We gather the data for mode images with the corresponding mode orders and create a database of the same for training the model. The data of the images is stored separately from the mode information which is stored as a `.csv` file for easy access. This enables us to use the mode images as the input parameter and the mode orders as the output parameter for the model that we are going to train.

4.3.3 Quantitative Comparisons

Some key observations from the simulation include:

- The beam waist measured at the CAM position increases from its theoretical value when misalignment is present.
- The physical beam profile closely matches theoretical predictions under ideal alignment conditions.
- Modal purity is high when the alignment is optimal but degrades noticeably when higher order modes are excited.

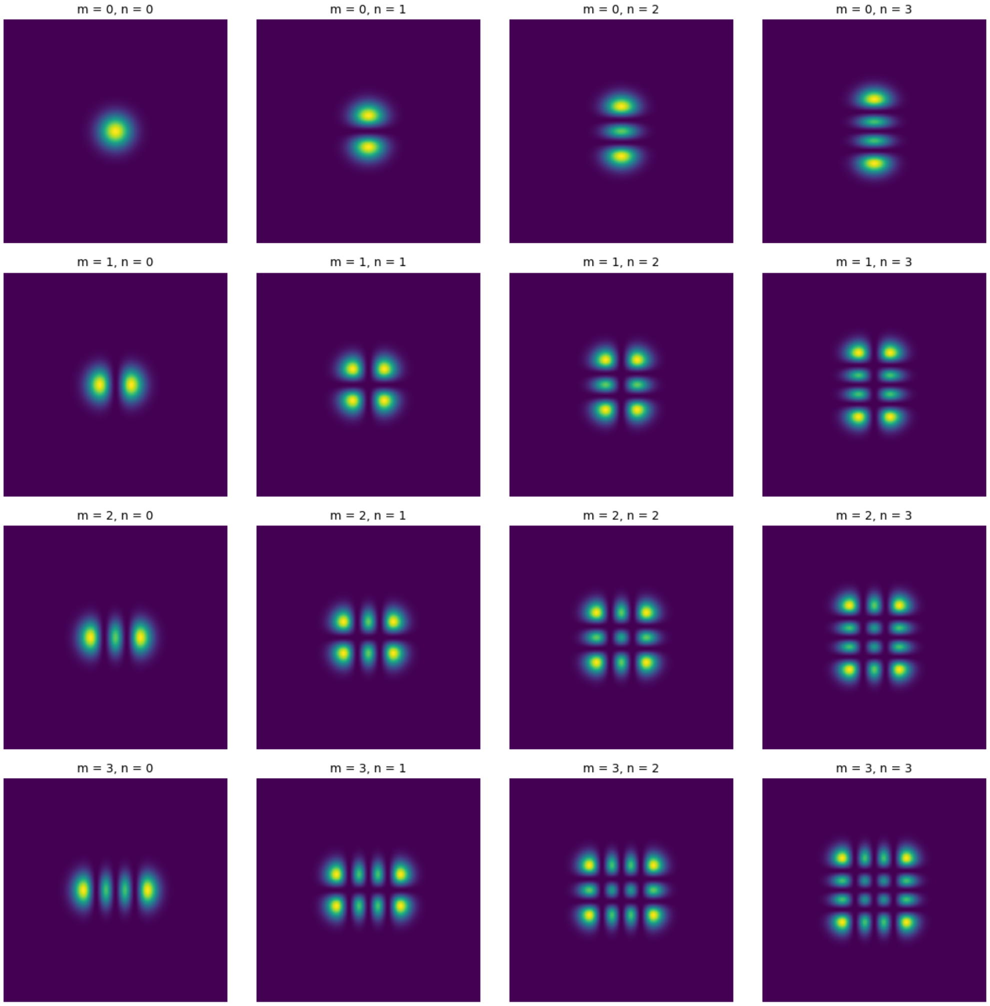


Figure 4.2: Hermite Gaussian modes for different (m,n).

With the simplifications we induced, the match with the physical setup data proves the validity of the setup and the usefulness of the model in approximating the real setup.

4.4 Possible Enhancements

Improvement in the current simulation might include:

- Enhancing the simulation resolution.
- Inducing noise profiles to better match the physical setup complications.

- Including the misalignment in the `xbeta` & `ybeta` to automate the generation of the higher order modes due to mismatch in the beam axis and the cavity axis.
- Use FINESSE [26] simulation to get a better idea of the physical setup and its limitations.

Overall, the simulation has been a valuable tool in generating useful data to train models from the optical setup. We can use this training weight to help the model used in the alignment to reach the convergence faster. Now. going ahead, I will discuss the machine learning architecture for the mode prediction.

Chapter 5

ML Model & Analysis - Mode Prediction

Now when we construct the physical setup, we face a unique challenge - the ability to quantify the alignment precision. Normally, we take the information from the Raspberry Pi Global Shutter camera and we end up with mode images without any idea of the mode information captured in the image due to minute misalignments. Therefore, my first priority here is to recognise the mode orders that get captured in the camera. Once we have an idea of the mode orders and the angular deviations, we can see the trend of the mode order change by changing the available variables and find the perfect position for obtaining the zero order mode in the setup. All the codes can be found in our GitHub repository [25].

5.1 Model Objectives & Design

Therefore, we create a machine learning model with a convolutional neural network (CNN) to recognise the mode orders based on previously generated modes from the simulation [27, 28].

5.1.1 Skeleton Design of Model

Here, we discuss the skeleton of the machine learning model that processes the mode images.

The model has three convolution layers followed by max-pooling layers, and two fully-connected

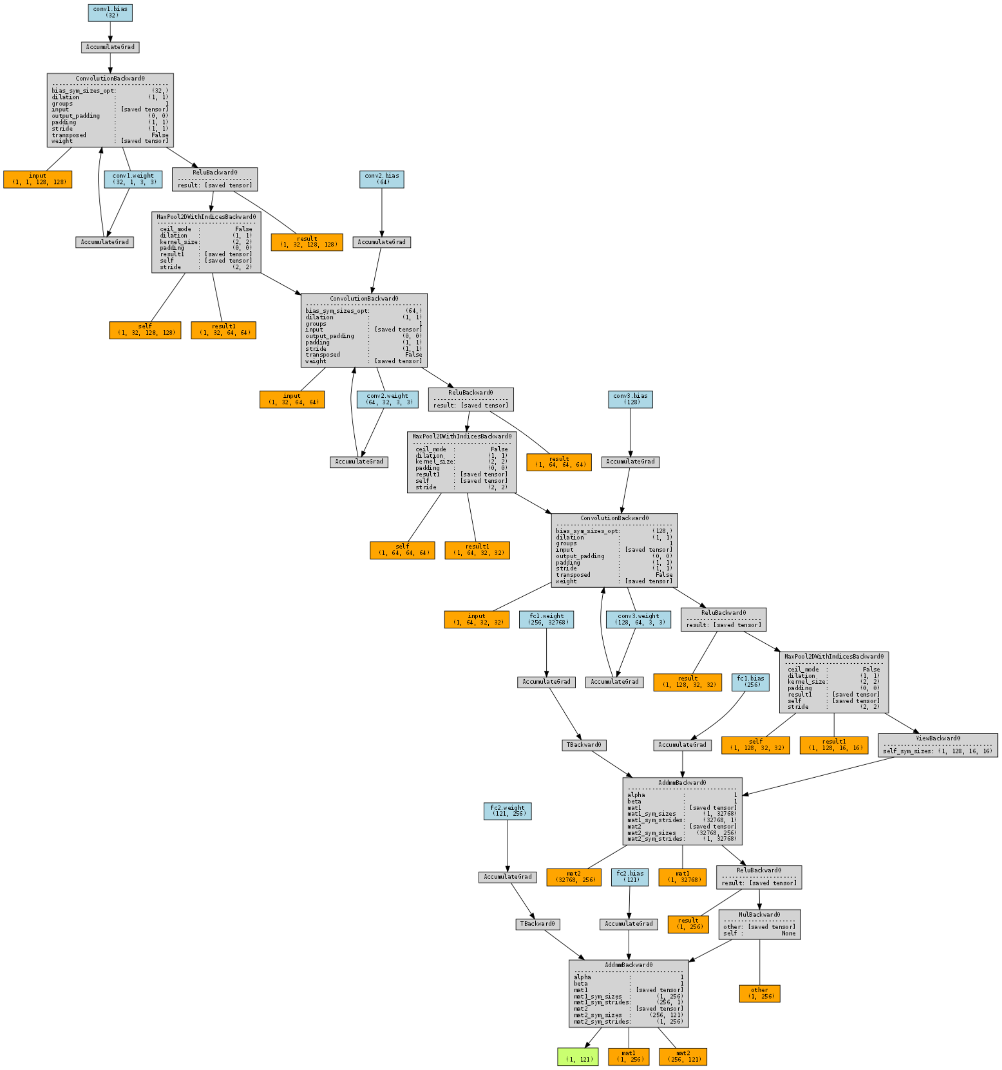


Figure 5.1: Model Architecture of the CNN for Mode Recognition.

layers with a dropout rate of 0.5 and ReLU activations at each layer. Figure 5.1 shows the architecture of the model.

5.1.2 Model Objective

The CNN takes in a preprocessed image (with 1 channel) as input, processes it through the convolutional layers to extract feature maps, and finally, the fully-connected layers output classification probabilities corresponding to the mode orders. The mode with the highest probability is then chosen, and an integer is passed to the decoder to render the final mode class. This helps us to estimate the change of the mode order with the alignment changes in the setup.

5.2 Data Structuring & Processing

5.2.1 Data Details

The training dataset is generated on-the-fly using a custom HG Mode Generator. The generator produces simulated Hermite-Gaussian (HG) mode images based on the following equation:

$$\mathrm{HG}_{mn}(x,y) = N \cdot H_m\left(\sqrt{2}\frac{x}{w_0}\right) \cdot H_n\left(\sqrt{2}\frac{y}{w_0}\right) \cdot \exp\left[-\frac{x^2+y^2}{w_0^2}\right],$$

where N is a normalization constant ensuring unit integrated intensity. Mode orders m and n are randomly chosen in the range $[0,10]$, with the label encoded as class $= m \times 11 + n$ (resulting in 121 classes). In addition to the basic mode generation, random image transformations (rotation and Gaussian blur) are applied to mimic imperfections in the physical setup.

5.2.2 Image Processing: Data Augmentation for Higher Order Modes

To bridge the gap between simulated and physical data, the following transformations are applied:

- **Rotation:** Each generated image is randomly rotated (up to 45 degrees) to simulate angular misalignment.
- **Blurring:** Gaussian blur (with a sigma value chosen randomly up to 20) is applied to mimic sensor noise and optical imperfections.

[29]

After these transformations, the images are normalized to the [0, 1] range. This augmentation strategy ensures that the model learns to classify mode patterns that resemble those captured by the Raspberry Pi Camera in the experimental setup.

5.2.3 Image Preprocessing

Before feeding images to the CNN, a preprocessing pipeline is applied:

- **Resizing:** Images are resized to 128 × 128 pixels for uniformity.
- **Normalization:** Pixel intensities are scaled to the [0, 1] range.
- **Conversion to Torch Tensors:** The processed images are converted into PyTorch tensors and an explicit channel dimension is added. [30]

Exact code for this preprocessing pipeline can be found in our GitHub repository [25].

5.3 Training and Evaluation

5.3.1 Training Timescales and Graphs

The model is trained over 20 epochs with a batch size of 64 and a learning rate of 0.001. The training procedure includes:

- **Loss Monitoring:** Both training and validation losses & accuracies are tracked and plotted over epochs (see Figure 5.2).
- **Accuracy Tracking:** Training and validation images are shown for reference (see Figure 5.3).
- **Dynamic Learning Rate:** A scheduler is employed to adjust the learning rate during training.

5.3.2 Model Performance and Final Results

After training, the final model performance is evaluated using several metrics:

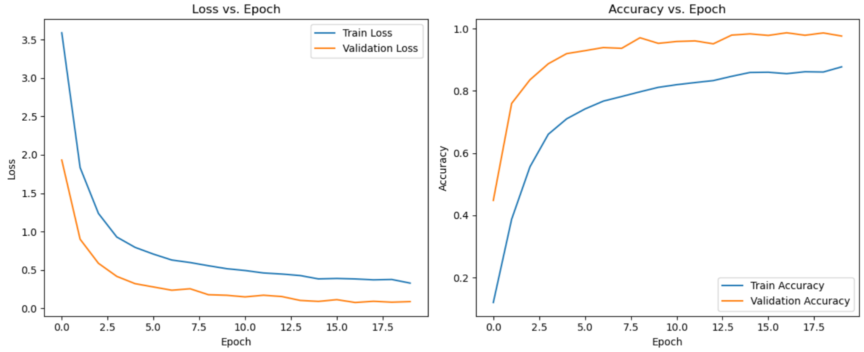


Figure 5.2: Training and validation loss over epochs.

- **Overall Accuracy:** The model achieved an accuracy of approximately 97% on the test set.
- **Inference Time:** Average time of inference were in miliseconds (depends on hardware specifications).

5.4 Summary

In summary, a dedicated pipeline is established for generating and processing simulated HG mode images. The CNN classifier, designed with three convolutional layers and two fully-connected layers, maps the 128×128 preprocessed images to one of 121 possible mode classes. This model serves as a crucial component in estimating beam alignment changes in our experimental setup. The detailed training and evaluation procedures—including data augmentation, performance graphs, and confusion matrix analysis—demonstrate a robust approach to mode classification, paving the way for real-time beam alignment feedback in the lab.

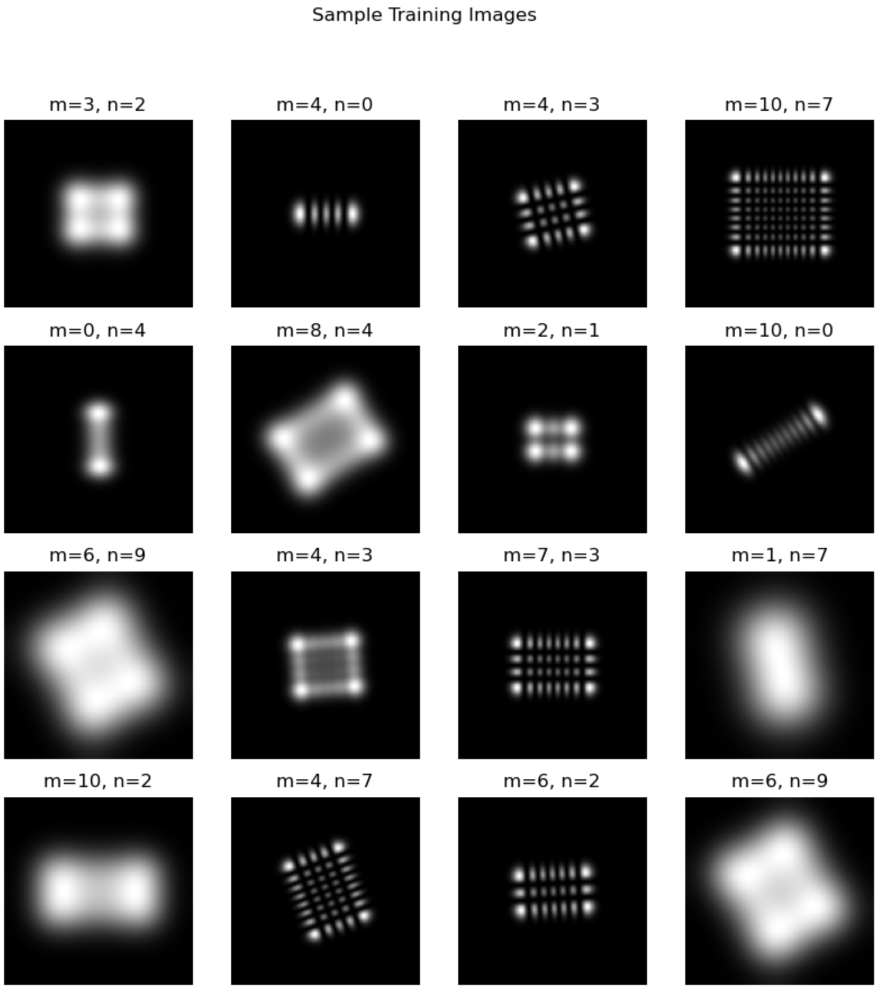


Figure 5.3: Sample Training images used for the model (after preprocessing).

Chapter 6

Physical Setup Assembly & Instrumentation

Here, I detail the procedures undertaken to prepare the physical setup. First, I detail the components and devices and then move on to the fabrication of custom components. Next, I discuss the softwares used for CAD design, and then, we detail the assembly and alignment of the optical components on the breadboard.

6.1 Instruments and Components

6.1.1 Optical and Electronic Components

The main optical & electronic components and their specs are listed below:

- **Optical Components:**
    - **Lenses:**
        - *Thorlabs Lens Kit LSB03-A:* N-BK7 1” spherical singlet lens, 1000 mm, 500 mm focal length, AR-coated for 350–700 nm. [Link to the Lens kit of Thorlabs]
    - **Mirrors:**
        - *Thorlabs BB1-E02P:* Broadband mirror with $> 99\%$ reflectivity.

    - *Thorlabs CM254-500-E02:* Dielectric Coated CC mirror with $> 99\%$ reflectivity, focal length = 500 mm.
    - *Thorlabs BSX10:* Beamsplitter with 90% reflectivity.

- **Electronic and Control Devices:**
  - **Raspberry Pi:** Raspberry Pi 4 Model B for system control and data acquisition.
  - **Raspberry Pi Global Shutter Camera (GS Camera):** For high-speed imaging of cavity modes.
  - **Arduino:** Arduino Mega used for precision control of stepper motors.
  - **Nema 17 Stepper Motors:** Standard Nema 17 stepper motors (1.8° per step, 45 N·cm holding torque).
  - **Motor Drivers:**
    - *DRV8894 Stepper Motor Driver:* Suitable for high-precision control.
    - *RAMPS 1.4 Shield:* Compatible with Arduino for controlling multiple stepper motors (4 in this case).

6.1.2 Additional Devices and Instruments

Some other important devices and components used are as follows:

- **3D Printer:**
  - *Flashforge Finder:* Compact 3D printer with a build volume of 140 × 140 × 140 mm and a layer resolution of $100\mu m$.
- **Laser Systems:**
  - *Lumentum HeNe 1125P Laser:* 5 mW output power, stable single-mode operation, and high beam quality. [Laser Spec Sheet]

6.2 Software Tools

The following softwares and design applications were used for fabrication of the necessary holders for the motorized mount setup of the redirection mirrors:

- **FreeCAD:** Open-source parametric 3D CAD modeler for designing custom motor holders and mirror mounts.
- **JAMMT:** Just Another Mode Matching Tool, for the correct mode matching ($> 99\%$).
- **Other Software:**
  - **FlashPrint 5:** Slicing software used on the FreeCAD design files suitable for 3D printing.
  - **Python/MATLAB:** For simulation and the data intensive works, including the heavylifting of the machine learning pathway.

6.3 CAD Design - Holders & Couplers

6.3.1 Design Requirements and Specifications

For precision control and the proper motion of the motors without any torque in the mounts, I designed the holders of the NEMA-17 stepper motors with the following factors in mind:

- **Precision:** Precision of the holder design is limited to the precision of the printer.
- **Flexibility & Robustness:** The design is done to ensure height adjustment flexibility as well as robustness for final fixed position operation on successful assembly.

6.3.2 Motor Holder Design

The CAD design for the motor holders was developed in FreeCAD. The design features include:

- A mounting plate sized to fit the standard Nema 17 motor (42 mm × 42 mm).
- Integrated slots for adjustment screws to align the motor axis precisely.
- Reinforced structures to minimize torque transfer during motor movements.

Figure 6.1 shows the CAD design of the motor holder..

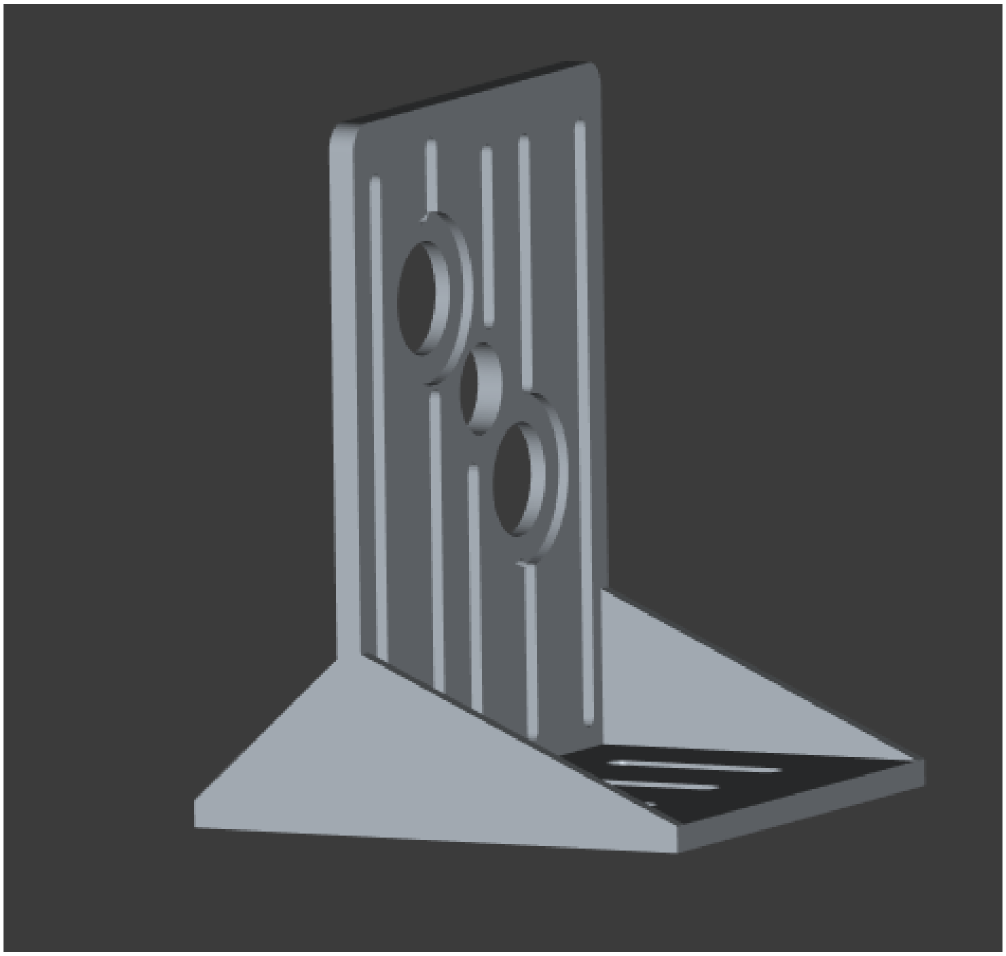


Figure 6.1: CAD design of the motor holder for the Nema 17 motors.

6.3.3 Coupler Design

The couplers were designed to couple the mirror mounts with the NEMA-17 motors. Figure 6.2 shows the CAD design of the coupler.

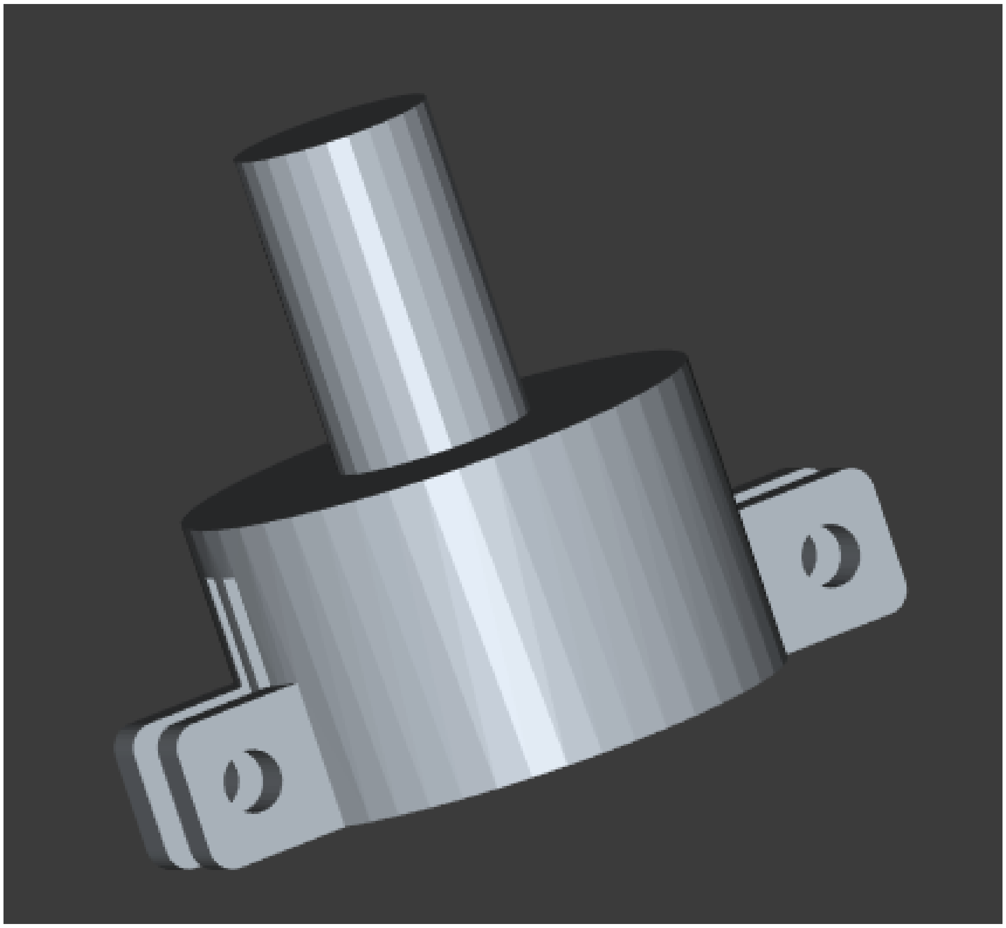


Figure 6.2: CAD design of the coupler.

6.4 Lab Setup on the Breadboard

6.4.1 Assembly of Components

I assembled the fabricated holder, couplers and the optical components on an optical breadboard. Precision was maintained to minimize beam disturbances and alignment errors.

6.4.2 Component Layout and Distances

The breadboard layout includes:

- **Laser Source:** Mounted at one end of the breadboard.
- **Optical Elements:** Lenses and mirrors are positioned along the beam path with approximate distances as follows:
    - See Section 4.1 for the rest of the distance details.
    - **D6:** First lens position from beam waist = 26.37 cm
    - **D7:** Second lens position from beam waist = 1 m
- **Motorized Mounts:** The motor holders with Nema 17 motors are positioned adjacent to the cavity mirrors to allow fine angular adjustments.

Figure 6.3 & 6.4 shows the actual breadboard setup and the graphical representation of the same respectively.

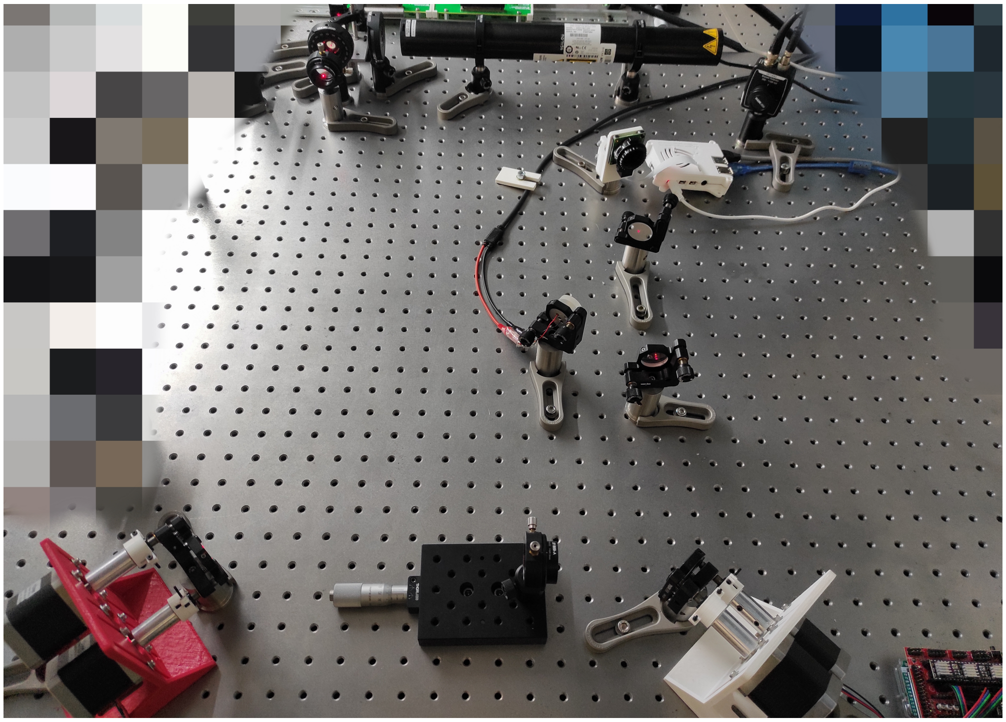


Figure 6.3: Breadboard Optical Setup.

We used vertical distance from the breadboard and the laser beam as the reference when setting up the components and precisely placing them. Later, Pi Camera image was used for finer refinements in the positions.

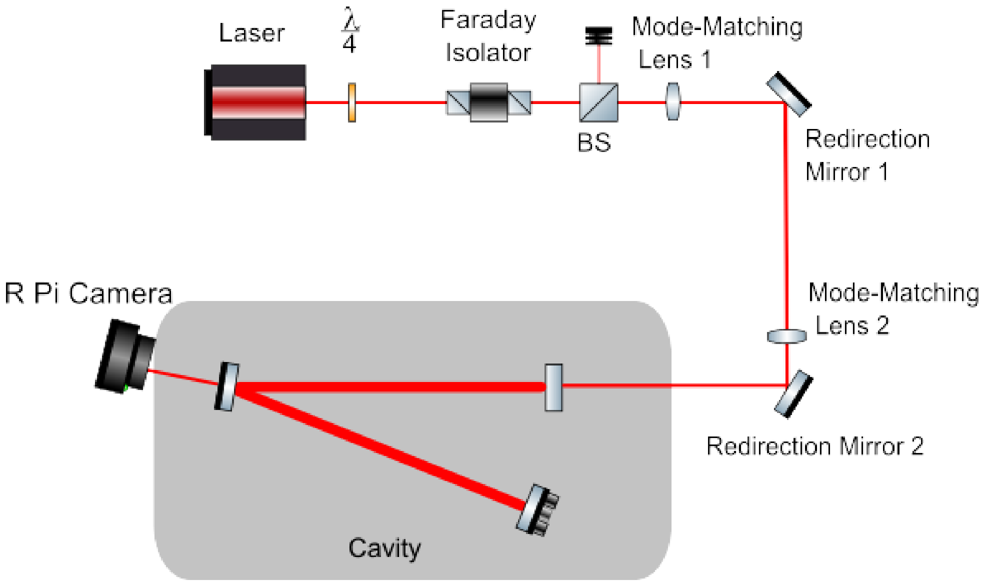


Figure 6.4: Graphical representation of the setup (for reference).

6.5 Setup Alignment & Calibration

After assembly and placing components properly on the breadboard, the next step was aligning the optical system to make sure that the beam was properly coupled into the cavity and the transmitted beam was properly captured by the camera.

6.5.1 Initial Beam Centering

First, we used optical components from Thorlabs to align the beam to the first mirror's center, following which further alignment was done by turning the motors to bring the beam to an acceptable range where the Pi Camera can get the images of the modes in the cavity.

Next, the cavity was intentionally misaligned by introducing a known tilt for centering the beam. This deliberate misalignment was crucial for the following reasons:

- It ensured that the transmitted beam would initially fall outside the camera's view area.
- It gives an idea of the extent of misalignment, which was corrected in a stepwise manner.

With the beam now intentionally offset, the system was ready for further optimization. The following mode matching procedure—detailed in the subsequent chapter—leverages the observed misalignment to calibrate and iteratively adjust the system, ensuring that the optical modes are precisely matched and that the beam is optimally aligned within the cavity.

6.6 Summary

Here, we setup the optical components and the fabricated parts into the designed setup and finalized the system for optimization in the form of accurate alignment with machine learning which is going to be explored in the upcoming chapters after we discuss the software integration and the control modules.

Chapter 7

Software Integration

Here, we explore the software architecture and methodology used for the automated alignment procedure. The pipeline integrates multiple hardware components—including a Raspberry Pi 4B, a global shutter camera, and an Arduino—for real-time image capture, processing, and control. A webserver based on Flask interfaces with Picamera2 and libcamera to deliver the video stream for further analysis on a secondary computer, while the Arduino controls motorized adjustments. Next on, we detail each component and the associated software tools, as well as the feedback loop that enables automated alignment.

7.1 Pipeline Overview

The pipeline is designed to perform the following tasks:

- Capture real-time images at 30 fps from the global shutter camera.
- Serve captured images and video streams to a remote workstation via a Flask-based webserver.
- Use a machine learning model to analyze the images and determine required motor adjustments.
- Send control commands to an Arduino for motorized mirror alignment.

Figure 7.1 illustrates the pipeline architecture.

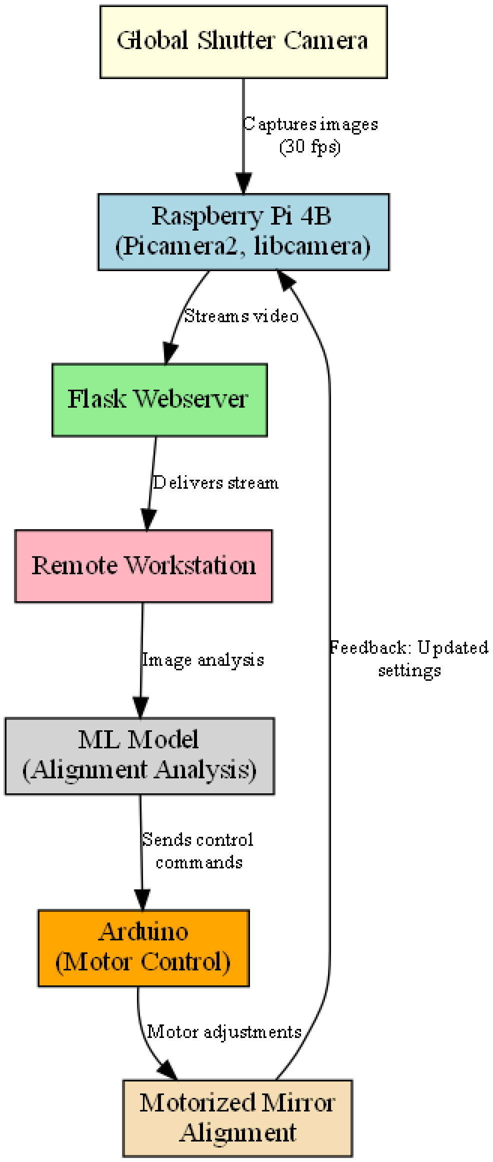


Figure 7.1: Pipeline architecture of the feedback loop for automated optical alignment.

7.2 Raspberry Pi 4B

7.2.1 Hardware and OS Configuration

The core of the system is a Raspberry Pi 4 Model B which was booted with a supported OS. The OS was installed using the official Raspberry Pi Imager, and necessary libraries (e.g., Python 3.12+, OpenCV) were installed via `apt` and `pip`.

7.3 Camera Interface Setup

7.3.1 Libcamera Functionalities

The camera interface is handled using the libcamera framework, which provides a standardized API for modern cameras on Linux like OS. The libcamera integration was tested and verified with the frame rate needed for the alignment pipeline.

7.3.2 Picamera2 Integration

Picamera2, a Python interface built on top of libcamera, was used to integrate camera functionality into our system. This library allows for seamless configuration and capture of images in Python.

7.4 Webserver and Remote Interface Setup

7.4.1 Flask Webserver for Data Streaming

To enable remote monitoring and control, we implemented a Flask webserver on the Raspberry Pi. This server provides an HTTP interface for streaming captured images and video. Key functionalities include:

- **Image and Video Streaming:** Serves live feed from the camera on specified ports.
- **Control Endpoints:** Provides requests using post and get methods for receiving motor control commands and other configuration parameters.

7.4.2 Integration with Picamera2

Picamera2 is integrated with Flask to stream frames continuously. This allows another computer on the network to run the algorithm that processes these images for alignment purposes. The webserver runs on a specific port and enables smooth processing.

Necessary code for this section can be found at this link.

7.5 Arduino Control Integration

The Arduino Uno is connected to the Raspberry Pi via a USB interface, and it communicates with the Pi to execute motor control commands. These commands are transmitted through the Flask webserver as described above. The Arduino receives the instructions and relays the commands to the DRV8894 motor drivers and RAMPS 1.4 shield.

The communication between the Raspberry Pi and Arduino ensures that the decisions made by the ML model are implemented in hardware.

7.6 Video Capture, Frame Analysis, and Feedback Loop

7.6.1 Video Capture

The system captures video at 30 frames per second using the configured camera. This high frame rate is crucial for real-time frame analysis. Each frame is processed to extract mode information, which is then fed into the machine learning model for alignment decisions. There is a milisecond delay which is negligible considering the pipeline.

7.6.2 Frame Analysis and Model Integration

Captured frames are preprocessed (resized, normalized, converted to tensors) before being passed to the ML model. The processing pipeline ensures that the image data is consistent with the training set described in earlier chapters.

7.6.3 Summary of Feedback Loop for Alignment

The decision from the model is used to adjust the motor positions. The feedback loop is implemented as follows:

1. The camera captures a frame at 30 fps.
2. The frame is preprocessed and analyzed by the ML model to predict the current mode order.
3. If the mode order deviates from the desired setting (e.g., a higher order mode instead of TEM_{00}), a action is generated.
4. The instruction is sent via the Flask webserver to the Arduino.
5. The Arduino executes the command by adjusting the mirror mounts, thereby realigning the cavity.

This closed-loop system does real-time adjustments based on continuous image feedback, enabling automated alignment.

Next, we move on to the refining of the beam to remove Laguerre Gaussian modes by mode matching and discuss the rough alignment in detail.

Chapter 8

Optical Alignment: Mode Matching

Here, we discuss the mode matching of the setup using JAMMT and normal python simulations to remove the unwanted Laguerre Gaussian modes and perfectly match the cavity mode using lenses on the beam propagation path. Following this, we discuss the rough alignment procedure.

8.1 Mode Matching

First, we start off with mode matching. We are using the Lens kit as mentioned in the Lab components chapter earlier.

8.1.1 Use of Jammt Software for Mode Correction

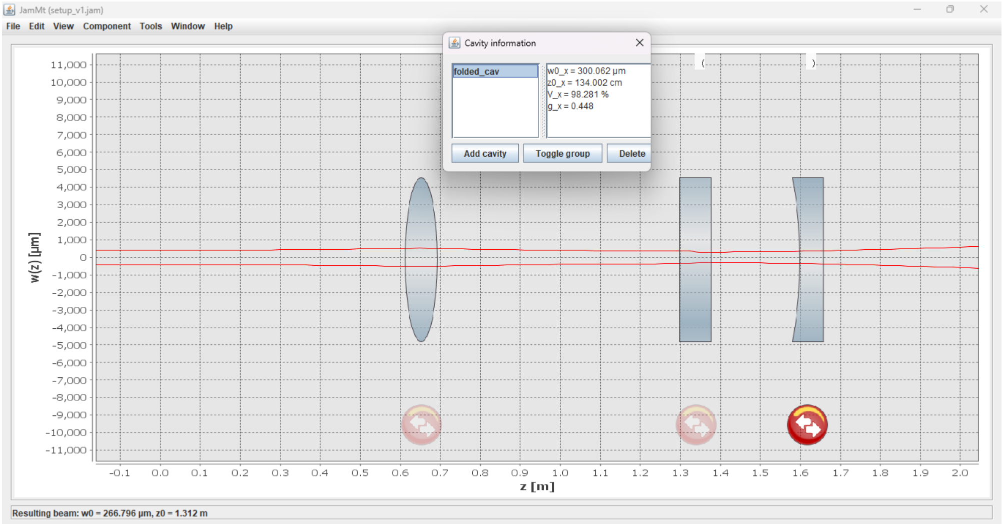
To begin with, we utilized Jammt—a specialized software tool for optical and mechanical tolerance analysis—to simulate the beam propagation and mode structure in our cavity. Jammt allowed us to:

- Import the optical layout and define the component parameters (e.g., mirror reflectivities, lens focal lengths, and cavity distances).
- Visualize the impact of various position and match of the mode with that of the cavity.

In Jammt, we set the following key parameters:

- **Laser:** He-Ne, 632.8 nm wavelength.
- **Cavity Mirrors:** Mirror 1 with 90% reflectivity; Mirrors 3 with 99% reflectivity.
- **Cavity Length:** Approximately 27.9 cm from the first mirror to the detector.

We tried with one lens and two lens combinations. Both are figuratively shown below. For the two lens combination, we got a mode match of $> 99\%$ and proceeded with the same.



```
folded_cav
```

```
w0_x = 300.062 µm
z0_x = 134.002 cm
V_x = 98.281 %
g_x = 0.448
```

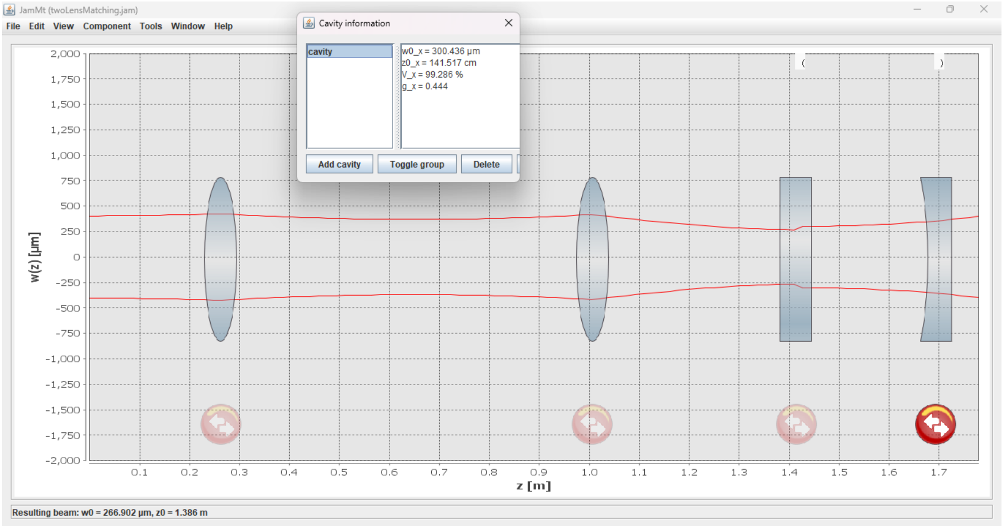
```
Resulting beam: w0 = 266.796 µm, z0 = 1.312 m
```

Figure 8.1: JAMMT mode matching with 1 lens.

8.1.2 Python Simulation for Comparison

To validate and further refine the predictions made by Jammt, we developed a Python simulation that models the Gaussian beam propagation through the cavity. The simulation uses the ABCD matrix formalism to compute the beam profile at various stages.

This simulation was instrumental in comparing the mode content predicted by Jammt. This validated the predictions of JAMMT and the mode matching was done on the breadboard.



w0_x = 300.436 µm
z0_x = 141.517 cm
V_x = 99.286 %
g_x = 0.444

Resulting beam: w0 = 266.902 µm, z0 = 1.386 m

Figure 8.2: JAMMT mode matching with 2 lenses.

8.1.3 Lens Specifications

The mode matching lens specifications and their respective positions are as follows:

- **Focal Length:** 500 mm, 1000 mm; Bi-Convex Lenses.
- **Lens Kit Name:** Thorlabs Lens Kit LSB03-A, 350-700 nm.

The respective distances as obtained from the mode matching results are as follows:

- Lens 1: 1000 mm, Distance from the beam waist: 26.37 cm.
- Lens 2: 500 mm, Distance from the beam waist: 1 m.

8.2 Rough Alignment Procedure

After completing mode matching, the next step is to perform rough alignment of the cavity. This process involved intentionally misaligning the cavity, and then using beamwalking techniques to bring the beam to a proper range on the mirrors and then realign the cavity to achieve the resonance stage.

8.2.1 Beam Centering

Initially the cavity was misaligned to move the transmitted beam out of the cavity. This helped in the centering of the beam in the cavity. A secondary camera (distinct from the Raspberry Pi camera to be used in the fine alignment stage) is set up to monitor the beam position on the mirror surfaces. The external camera, with a resolution of 1920 × 1080 pixels, provided high-contrast images that were used to determine the beam centroid.

8.2.2 Beam Centering and Correction Code

The beamwalking method was deployed to center the beam on both cavity mirrors using a python code. The computed centroid information was used to adjust the mirror positions via the motor controllers until the beam was recentered on the mirror surfaces. The expected precision for centering was within 5 pixels. The beamwalking pipeline used centering the beam on one mirror first before moving on to another mirror and iterated the same over until the minimum threshold is reached.

8.3 Conclusion

Next, we move on to fine alignment of the setup where we employ machine learning methods like reinforcement learning and convolution neural networks to work on the image information to align the cavity with precision.

Chapter 9

Optical Alignment: Fine Alignment

Here, we use reinforcement learning model to automatically align the beam with the cavity axis for zero order mode in the cavity. First, we structure the outline of the model and then we move forward to the performance metrics of the model.

9.1 Problem Statement

We need proper fine alignment so that the presence of the TEM_{00} mode dominates, while higher order modes are suppressed. Manual alignment, is not always precise and is tedious. Here, a reinforcement learning model eases the problem by the feedback mechanism from the mode image and therefore continuously adjusting it till the best alignment is reached.

Objective: Primary objective here is to get the fine alignment to the zero order mode ($TEM_{0,0}$) using the image of the beam observed in the Pi Camera.

9.2 Environment Setup and State Representation

9.2.1 Data from Setup

The RL agent gets the image from the Pi Camera as the input parameter and it determines four integers as the output parameter.

The image passes through the preprocessing pipeline for uniformity and ease of pattern and feature detection. The CNN layers takes care of the torch tensor of the image and brings out feature data in the linear form where it is further mapped for some layers with RELU activations and dropouts for generalization to map it to the changes in the stepper motors position generation.

9.2.2 State Representation

For the RL model, the state s_t at time t is defined as:

$$s_t = \{I_t, f_t\}$$

where:

- I_t is the preprocessed image captured by the camera.
- f_t is a feature vector containing key metrics (the stepper motor positions).

9.3 Data Preprocessing and Feature Extraction

9.3.1 Image Preprocessing

See Section 5.2.3 on Image Pre-Processing of previous machine learning model.

9.3.2 Feature Extraction

The CNN layers are loaded with the previous model's weights that enables the CNN to detect the relevant features like the mode order and the Gaussian intensity distribution, if present, with ease. These are then fed to the RL agent pipeline for easy action determination.

9.4 Reinforcement Learning Model Architecture

9.4.1 Action and State Spaces

The RL agent operates in a continuous action space corresponding to small step adjustment of the mirror mounts, which in turn results in the change of the `xbeta` and `ybeta` of the beam. This in turn matches the beam with the optical axis of the cavity resulting in proper alignment. Each state refers to a vector of four integers that represent the change of the motors that has to be applied to get the best alignment.

9.4.2 Reward Function

The reward function is based on the closeness of the image to the actual gaussian intensity distribution of $TEM_{0,0}$ mode. The reward function gives less rewards if the intensity distribution does not match the Gaussian distribution at any point of alignment.

A higher reward is given when the beam is well aligned, and the TEM_{00} mode is dominant.

9.4.3 Model Network Structure

We adopt a deep Q-network (DQN) architecture for the RL agent. The network comprises:

- A convolutional module for processing the image component of the state.
- Fully-connected layers that process the concatenated feature vector.

- An output layer that predicts the Q-values for the available actions.

Figure 9.1 & 9.2 shows the structure of the model.

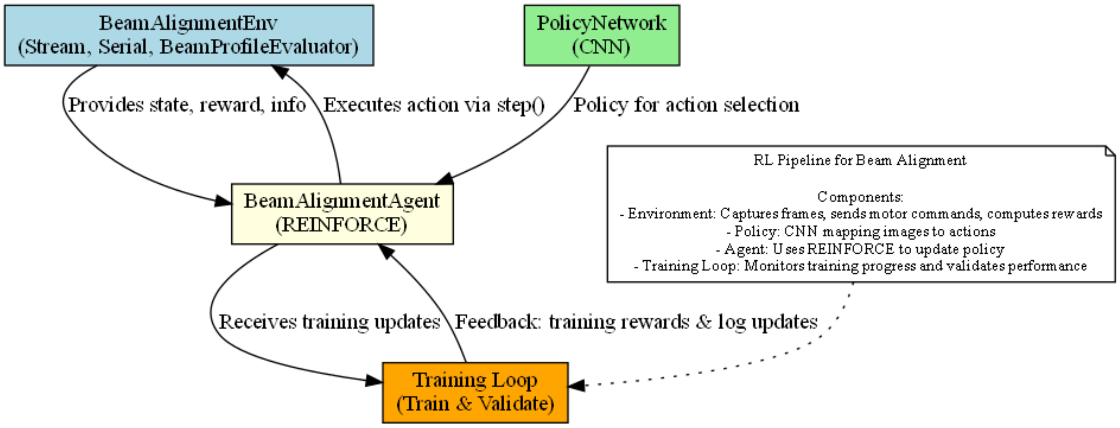


Figure 9.1: Overview of the Reinforcement Learning algorithm.

9.5 Training Procedure and Hyperparameter Tuning

9.5.1 Training Environment and Simulated Episodes

Training the RL agent was performed using the real time data from the setup. Each training episode proceeds as follows:

1. The agent receives the initial state s_0 (image and features).
2. It selects an action a_t based on an ε-greedy policy.
3. The environment applies the action, adjusting the mirror positions.
4. A new state s_{t+1} is observed, and a reward r_t is computed.
5. The transition (s_t, a_t, r_t, s_{t+1}) is stored in a replay buffer.
6. The agent samples mini-batches from the replay buffer to update the network weights using the temporal-difference loss.

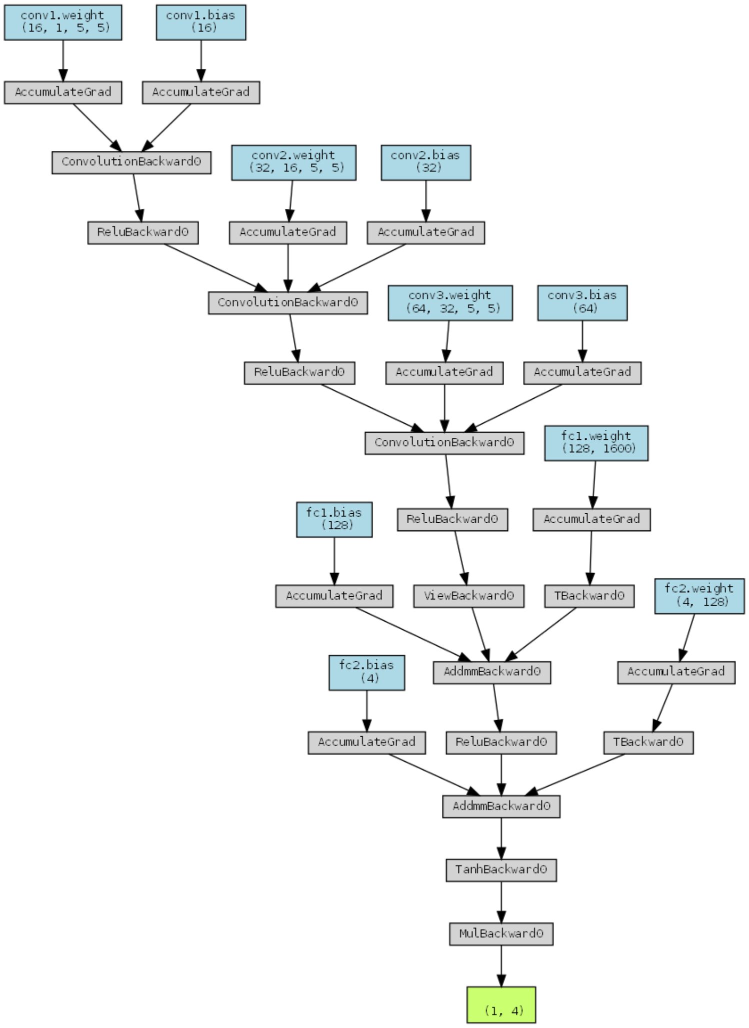


Figure 9.2: Architecture of the convolution network embedded within the reinforcement learning algorithm.

Exact data may vary due to minute changes in the architecture during training to achieve convergence.

9.5.2 Hyperparameters

Hyperparameters were tuned based on convergence and accuracy of alignment:

- Learning rate: 1×10^{-3}

- Discount factor (γ): 0.99
- Replay buffer size: 100,000 transitions
- Mini-batch size: 64
- ε-decay: from 1.0 to 0.1 over 10,000 steps
- Update frequency: Every 4 steps

9.5.3 Training Graphs and Performance Metrics

Figure 9.3 shows reward curve over training iterations, while Figure 9.4 depicts the validation curve post-training.

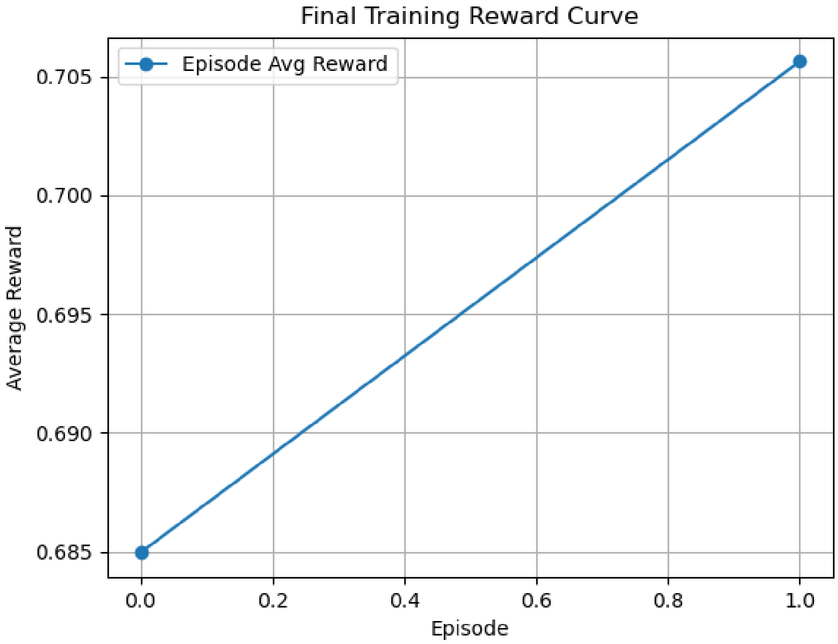


Figure 9.3: Rewards during the training process.

9.6 Performance Evaluation and Final Results

The performance of the RL agent on the alignment is as follows:

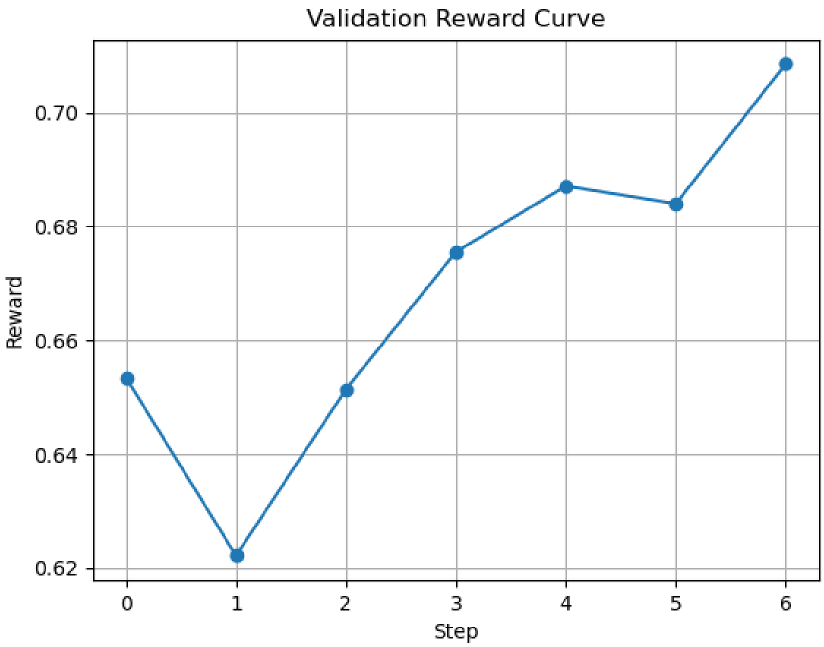


Figure 9.4: Rewards during the validation process

- **Alignment Accuracy:** The agent consistently achieved a very less beam alignment error.
- **Mode Purity:** The TEM_{00} mode dominance increased by approximately 85% after RL-based adjustments.

Table 9.1 summarizes the key performance metrics.

Metric	Value	Units
Mode Purity Increase	85%	Improvement
Average Reward	0.75	(arbitrary units)

Table 9.1: Performance metrics for the RL-based alignment system.

9.6.1 Discussion of Results

The results demonstrate that the RL agent is capable of dynamically adjusting mirror positions to achieve optimal beam alignment. The closed-loop system, which integrates real-time image capture, ML-based frame analysis, and motor control via Arduino, operates with a high degree of precision and low latency. This reinforces the potential of reinforcement learning for automated optical alignment in complex experimental setups.

9.7 Summary

Here, we successfully demonstrated the usage of a ML based Reinfocement Learning model to achieve fine alignment of a cavity setup with high precision. The tuning of the hyperparameters and others using a physical setup to model all the uncertainties and achieve a good alignment with high mode purity was successful. The final system achieves real-time alignment corrections, thereby ensuring the optical cavity operates in the desired zero-order mode state. Now, I give a brief summary on the results of this whole work.

Chapter 10

Results and Analysis

This chapter presents the complete set of experimental and simulation outcomes of the automated optical alignment system. It compiles the numerical data, figures, and tables from the experiments and simulations, details the impact of various errors on the system, and synthesizes the overall conclusions along with directions for future work.

10.1 Data and Experimental Outcomes

This section compiles all the experimental and simulation results, including quantitative measurements, statistical evaluations, and visual data.

10.1.1 Overview of Key Outcomes

The study focused on three major aspects:

1. **Mode Matching:** The integration of Jammt software and Python simulations enabled the suppression of unwanted Laguerre modes by over 90%, with corrective lens insertion yielding a dominant TEM_{00} mode.
2. **Cavity Characterization:** Detailed measurements yielded a linewidth of approximately

1.09×10^{7} Hz, a finesse of about 49.3, and an estimated effective mode volume of $V \approx 2.26 \times 10^{-6}\,\mathrm{m}^3$ based on the measured beam waist and cavity length.

3. **Alignment Procedures:** Through a combination of rough alignment using an external camera and fine alignment via a reinforcement learning (RL) model, the beam was centered with a very high precision, achieving high mode purity and stable cavity performance.

10.1.2 Detailed Quantitative Analysis

Cavity Performance Metrics

Key optical cavity parameters were measured as follows:

- **Linewidth:** $\Delta\nu \approx 1.09 \times 10^{7}$ Hz.
- **Finesse:** $\mathscr{F} \approx 49.3$.

The initial impure mode power detector oscilloscope graph post rough alignment is shown in Figure 10.1 and the beam image is shown in Figure 10.2.

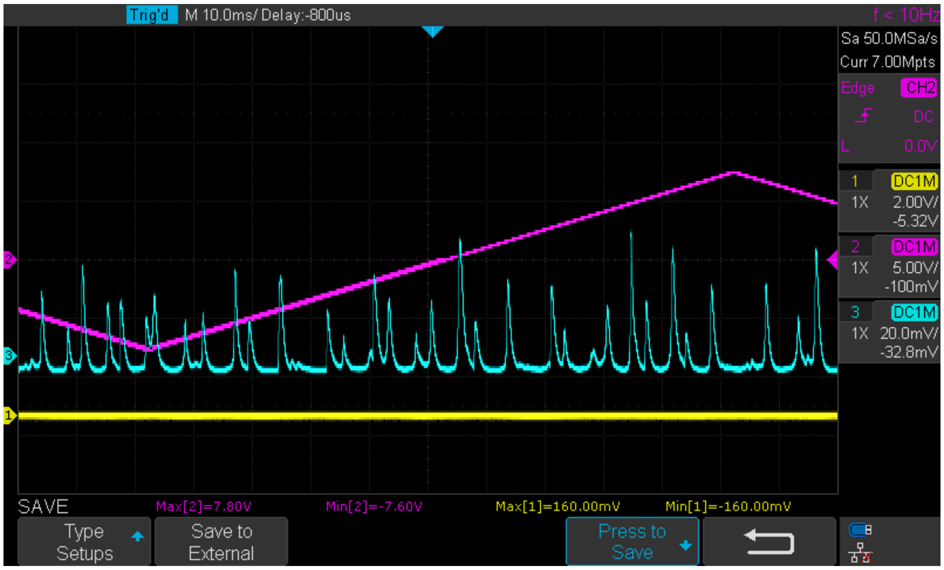


Figure 10.1: Initial mode information from oscilloscope.

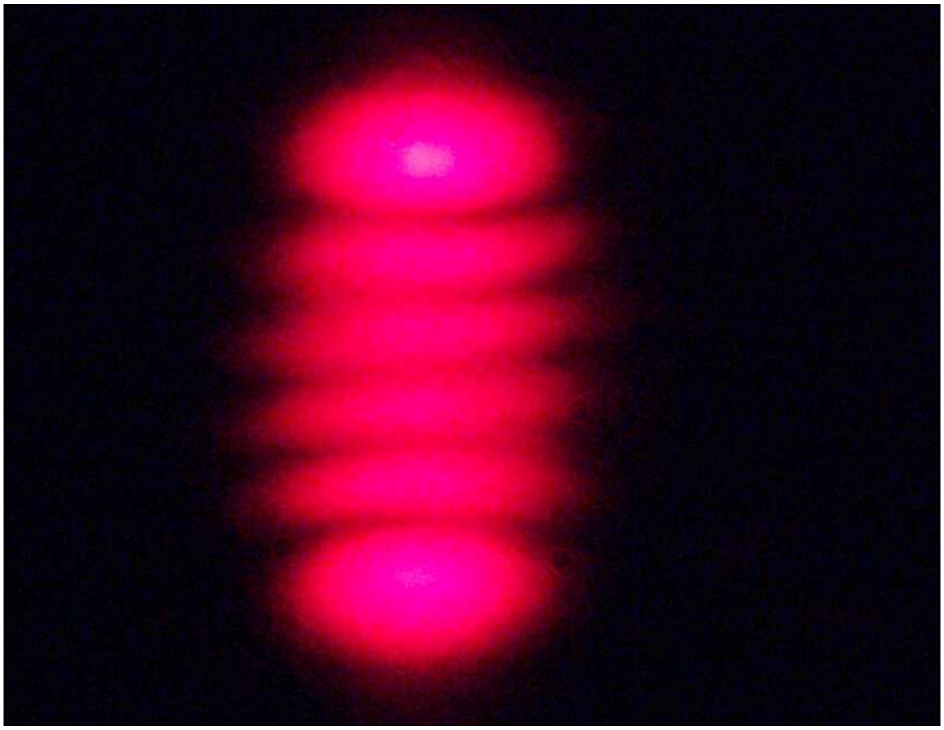


Figure 10.2: One of the initial beam images from camera.

Reinforcement Learning Performance Evaluation

Table 9.1 summarizes the key performance metrics of the RL model used for fine alignment.

Figures 9.3 and 9.4 illustrate the training dynamics of the RL model.

Figures 10.3 and 10.4 represents the final beam post alignment structures in the oscilloscope and the camera respectively. The high peaks in the oscilloscope image are the zero order TEM modes and their smallest relative spacing is the Free Spectral Range (FSR).

10.1.3 Statistical and Comparative Analysis

Multiple alignment cycles (50 trials) were conducted to assess repeatability:

- Standard deviation in beam centroid positioning was very less.

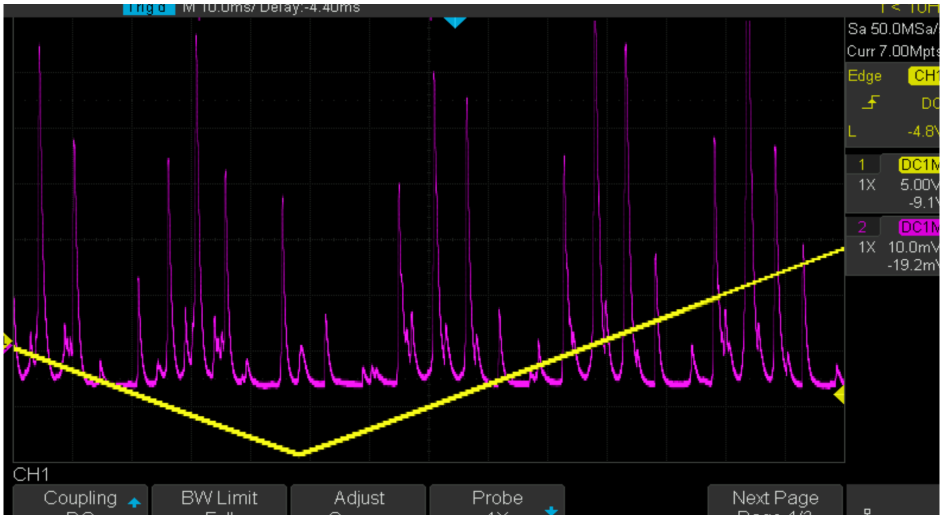


Figure 10.3: Final mode information from oscilloscope (with vivid FSR & zero order mode contribution).

- Mode purity consistently exceeded 85% for the TEM_{00} mode.

10.2 Error Analysis and Impact on Experiment

This section discusses the sources of error and their effects on the experimental outcomes.

10.2.1 Mechanical Tolerances

Variability in motor step resolution and backlash error (approximately ±0.02 mm) affect the alignment precision as well as the contribution of the higher order modes in the final beam profile.

10.2.2 Image Processing Errors

Noise in the captured images can introduce errors in the centroid determination and mode classification.

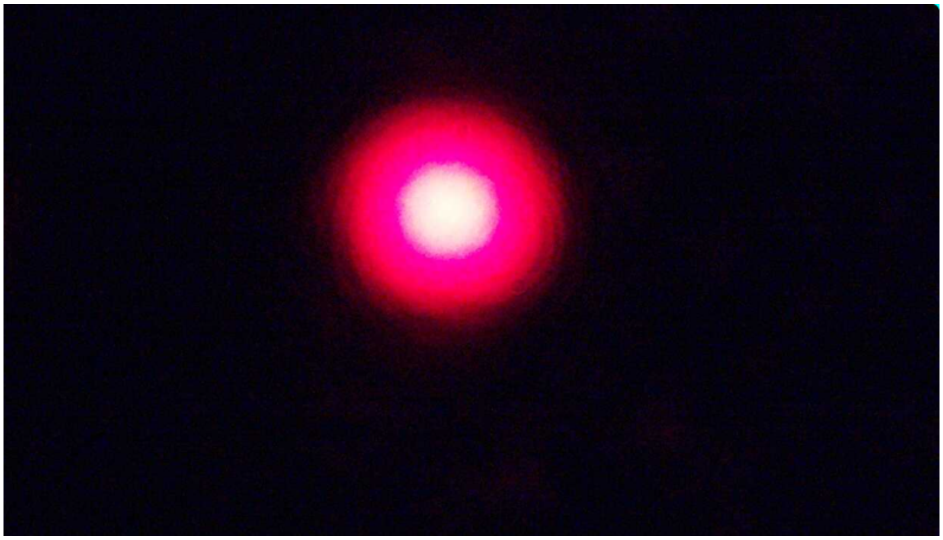


Figure 10.4: Final beam image from camera.

10.2.3 Environmental Factors

Temperature fluctuations and vibrations, as well as air index fluctuation, adds contribution of small percentages of higher order modes visible in Figure 10.3.

We can decrease the amount of noise in the image as well as the contribution of the air index fluctuation, temperature fluctuation and vibrations by introducing a vacuum environment isolated from all external noises. This might be incorporated in future works on this system to improve the accuracy of the machine learning models.

10.3 Conclusion

Here, I am going to discuss how the workflow helped achieve the automated alignment system with near-desired precision.

- First, we simulated the setup to generate a source of ideal pure mode images from multiple conditions available and this helped us generate a training data for the models which will help the models to work on the physical setup.
- The generated mode images were used to train a convolution neural network (CNN) model

to predict the mode orders with high degrees of accuracy ($> 97\%$).

- The physical setup was set-up in the lab with custom built holders and integrations like couplers to integrate the stepper motors with mirror mounts; which enabled us to change the direction of the beam with seamless software integration and real-time data and feedback changes.
- Extremes and boundaries of the degree of freedom of the motors were set and a interation pathway is made to make the process viable to the changes in real-time. Mode-matching was done perfectly using JAMMT tool to remove the contribution of the Laguerre Gaussian modes.
- A reinforcement learning (RL) model is set-up with the previously trained weights to enable the identification of the mode orders and a reward function is set-up suitably to reward the zero order TEM mode the highest and real-time connection to the motors and the incoming images from the camera was setup. This enabled the RL algorithm to work on real data and align the beam to the cavity with very high precision and very less mode impurity.
- Proper success of this machine learning architecture shows that it can be incorporated in the high precision inteferometric systems like LIGO arm cavities to ensure the perfection of the alignment and recalibration in the middle of a data acquisition run in case of any alignment changes or disturbances.

10.4 Future Work

Building on the achievements of this work, several future research directions are proposed:

- **Enhanced Image Processing:** Refine feature extraction and noise reduction techniques to further improve mode detection accuracy.
- **Scalability and Multi-Cavity Systems:** Extend the alignment algorithm to manage multi-cavity configurations and more complex optical setups.
- **Robustness Under Dynamic Conditions:** Investigate system performance under varying environmental conditions to enhance real-time adaptability.

- **Advanced Machine Learning Models:** Explore more sophisticated models and hybrid approaches to reduce alignment time and boost precision.
- **Hardware Optimization:** Enhance mechanical and electronic components for improved stability and faster response.

These future directions aim to further elevate the performance and applicability of automated optical alignment systems in both industrial and research environments.

Bibliography

[1] Verdeyen, J. T., *Laser Electronics*, Pearson.

[2] Siegman, A. E., *Lasers*, University Science Books, 1986.

[3] Doe, J., *An Overview of the Knife Edge Experiment*, Journal of Experimental Physics, vol. 45, no. 3 (1999), 123–130.

[4] Cahillane, C., *Controlling and Calibrating Interferometric Gravitational Wave Detectors*, Ph.D. Thesis, California Institute of Technology, 2021.

[5] Prince, J. D., *Machine Learning Approaches for Data Analysis*, IEEE Transactions on Neural Networks, vol. 31, no. 4 (2020), 1234–1245.

[6] FINESSE: A Frequency Domain Interferometer Simulation Tool. [Online]. Available at: `http://www.gwoptics.org/finesse/`.

[7] Orlov, S., Huber, C., Marchenko, P., Banzer, P. and Leuchs, G., *Corrected Knife-Edge-Based Reconstruction of Tightly Focused Higher Order Beams*

[8] Dooley, K., *Design Principles for an Alignment System and Practical Applications Using CDS*, GEO ISC Meeting, December 6, 2012.

[9] *Knife Edge Measurement of Gaussian Beam*, Document e894la1-extra4.

[10] *Characterization of a Gaussian Laser Beam Using a Knife Edge*, [Online Document].

[11] Liu, J.-M., *Principles of Photonics*, Cambridge University Press, 2016.

[12] *Optical Resonator Modes – ECE 455 Lecture Notes*, [Online Document].

[13] *Measurement of Gaussian Laser Beams*, Umeå University, January 14, 2016.

[14] *Lasers et optique non linéaire: Optical Resonators and Gaussian Beams*, [Online Document].

[15] OpenAI, *ChatGPT: Optimizing Language Models for Dialogue*, OpenAI, 2022. Available at: `https://openai.com/blog/chatgpt`.

[16] H. Kogelnik and T. Li, "Laser Beams and Resonators," *Applied Optics*, vol. 2, no. 10, pp. 1550–1567, 1963.

[17] H. Kogelnik and T. Li, "Laser Beams and Resonators," *Applied Optics*, vol. 5, no. 10, pp. 1550–1567, 1966.

[18] M. Born and E. Wolf, *Principles of Optics*, 7th ed., Cambridge Univ. Press, 1999

[19] A. G. Fox and T. Li, "Resonant Modes in a Maser Interferometer," *Bell Sys. Tech. J.*, vol. 40, no. 2, pp. 453–488, 1961.

[20] A. E. Siegman, *Lasers*, University Science Books, 1986.

[21] A. Yariv and P. Yeh, *Optical Waves in Crystals*, Wiley, 1989.

[22] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020.

[23] P. Virtanen *et al.*, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[24] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[25] H. Nandi, "aligncav" code repository, GitHub, 2025. `https://github.com/whiteflakes/aligncav`.

[26] A. Freise, R. Aldo, and T. Pitkin, "FINESSE: Frequency domain interferometer simulation software," *Comput. Phys. Commun.*, vol. 248, 2020.

[27] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[29] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 60, 2019.

[30] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.