

Learning-Based Control for Optically Trapped Non-Spherical Particles

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Connor Edlund

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Dr. Murti Salapaka

January, 2025

© Connor Edlund 2025
ALL RIGHTS RESERVED

Acknowledgements

I would like to thank Dr. Salapaka for the opportunity and funding to pursue this research. He afforded me the time and freedom to design and build the optical tweezers system detailed in this thesis as I saw fit. With his support, I was able to learn valuable system and experiment troubleshooting skills. When an opportunity in industry arose during my last year in his lab, he allowed me the flexibility to pursue the industry role while finishing my degree.

My lab mates have been an important source of support throughout my time in the Salapaka lab. My predecessor in the lab, Dr. Rachit Shrivastava, spent an immense amount of time training me on how to run biochemistry optical trapping experiments. He has been and continues to be an important source of advice. Dr. Shrivastava and another lab predecessor, Dr. Shreyas Bhaban, also helped with the construction of the optical tweezers system described in this thesis. They selected and installed the motorized stage, camera, and TIRF system on the microscope of the optical tweezers system. They also handled the procurement of the detection laser and the vibration isolation table. Dr. Bhaban also helped me get the internship that turned into my first industry job.

I would also like to thank Dr. Sivaraj (Shiv) Sivaramakrishnan and his lab members for supporting my biochemistry experiments. While my biochemistry work ended up taking a back seat to my iterative learning control work, it was none the less an enriching experience.

While finishing my research I started my industry job at KLA Corp. Working on my degree and getting up to speed at this job required a huge amount of flexibility from my colleagues at KLA. I would like to thank my manager, Wilfred Ghonsavles, for the flexibility that enabled me to complete this degree. Additionally, my coworkers Ashwin, Alex, and Mike have all helped me balance my workload at KLA.

Lastly I would like to thank my family and my wife for their unwavering support.

Dedication

Dedicated to my wife Eliza Herrero who is my indispensable partner in all aspects of life.

Abstract

The physical momentum of light, while negligible in most situations, becomes significant to particle dynamics when the particle size is on the micron-scale or smaller. An optical tweezers system harnesses the momentum of light to manipulate these small particles by trapping them near the focal point of a sharply focused laser beam. Since the inception of optical trapping in 1986, it has enabled many scientific discoveries at the small scale. Many optical trapping applications involve trapping spherical objects such as cells or spherical beads attached as handles to biological samples. Thanks to the symmetry of a sphere, models for a spherical particle's dynamics in an optical trap can be easily derived. This has enabled the application of modern, model-based control methods to the trapping of spherical particles resulting in significant improvements in experimental outcomes.

The trapping of non-spherical particles, for example rod-shaped bacteria or engineered microstructures, is also an important use case for optical trapping. However, the more complex interactions of a non-spherical particle with the potential well of an optical trap do not lend themselves to closed-form models suitable for real-time control applications. This is a significant hurdle for the dynamic control of optically trapped, non-spherical particles and consequently, existing approaches for manipulating these particles rely on trial-and-error or approximating the particle as spherical. In this work the challenge of applying automatic dynamic control to optically trapped non-spherical particles is overcome through the use of iterative learning (IL) control. This is a model-free approach that seeks to learn the dynamic beam position pattern that achieves a desired manipulation on the particle. As an instantiating example, the out-of-plane rotation of a micron-scale cylinder is taken as the control goal.

To develop the IL controller design and verify its performance both dynamic simulations and experiments are performed. The simulation results prove the IL controller design concept viable through the successful learning of a beam pattern that rotates a cylinder. Before experiments are carried out, a purpose-built optical tweezers system is constructed and programmed to implement the IL control design. The system design, programming, and calibration are discussed in detail. Experiments carried out on this system show the success of the IL controller design as it reliably and repeatedly learns a beam pattern that successfully and automatically rotates the cylinder. This unlocks new potential in both the productivity and capability of non-spherical optical trapping applications.

Contents

Acknowledgements	i
Dedication	ii
Abstract	iii
List of Tables	vii
List of Figures	viii
List of Abbreviations	xii
1 Introduction	1
1.1 Optical Trapping Mechanism	2
1.1.1 Geometric Optics Regime	2
1.1.2 Rayleigh Regime	4
1.1.3 Wavelength-Scale Regime	4
1.2 Particle Dynamics	8
1.2.1 Brownian Motion	8
1.2.2 Langevin Dynamics	10
1.3 Spherical Particles: Applications & Dynamic Control	13
1.4 Non-Spherical Particles: Applications & Control Challenges	15
1.4.1 Beam Shaping for Orientation Control	16
1.4.2 Utilizing Particle Properties for Manipulation	17
1.4.3 Indirect Manipulation Using Handles	18
1.4.4 Direct Manipulation Using Multiple Traps	19
1.5 Statement of Purpose	20

2	ILC Simulation	23
2.1	Simulation Framework	23
2.1.1	Particle Dynamics	23
2.1.2	Numerical Simulation	25
2.2	Control Strategy	26
2.2.1	Closed-Loop Controller	27
2.2.2	Iterative Learning Controller	28
2.2.3	Controller Sample Rate	30
2.3	Results	30
2.4	Simulation Conclusions	34
3	Instrumentation	37
3.1	Optical System Design	37
3.1.1	Detection System	39
3.1.2	Microscope Equipment	40
3.2	Control Hardware	40
3.2.1	NI cRIO Controller	41
3.2.2	AOD Electronics	41
3.2.3	QPD Amplifier	43
3.3	LabVIEW Programming	43
3.3.1	Real-Time VI	45
3.3.2	Iterative Learning Controller Programming	48
3.3.3	FPGA VI	50
3.3.4	Closed-loop Control Rate & Coordinate System	56
3.4	Calibration	58
3.4.1	Flattening Optical Power Across the AOD Scan Range	58
3.4.2	AOD Actuation Calibration	60
3.4.3	Detection System Calibration & Sensitivity	62
3.5	Estimating Trap Stiffness	65
3.6	Conclusion	68
4	ILC Experiments	69
4.1	Introduction	69
4.1.1	Methods	69
4.1.2	Micro-Rod Position and Orientation Detection	71
4.2	Controller Design	72
4.2.1	Iterative Learning Controller	73

4.2.2	Feedback Controller	77
4.2.3	Overall Controller Design	78
4.3	Results	79
4.4	Conclusion	84
5	Conclusion & Future Directions	87
	References	88

List of Tables

3.1	Estimated Trap Stiffness	68
-----	------------------------------------	----

List of Figures

1.1	Geometric Optics Optical Trapping Ray Tracing Diagram	3
1.2	EM Scattering for a Sphere	9
1.3	Linear Approximation of Optical Trap Potential	13
1.4	In-Plane Rotation vs. Out-of-Plane Rotation	17
1.5	ILC Concept Diagram	21
1.6	Iterative Learning Control Goal: Out-of-Plane Cylinder Rotation	22
2.1	Rotations of the particle and its coordinate system.	24
2.2	Control Diagram	27
2.3	Defining the cylinder's orientation using $\hat{z}$'s direction angles relative to the X , Y , and Z axes.	28
2.4	Reference Angle Trajectory $\alpha_{r,x}(t)$	29
2.5	Results for iterations 10 and 20 of the IL process. For each iteration there are four plots arranged horizontally; the leftmost plots show the motion profiles of the two beams' focal points used to generate the results in the three plots to the right.	32
2.6	X -position vs. beam multiplexing duty ratio for iteration 20.	33
2.7	50 runs using ILC generated feedforward input.	36
3.1	System Diagram	38
3.2	QPD Detection System Concept	39
3.3	RF Driver	42
3.4	QPD Amplifier	44
3.5	LabVIEW Real-time and FPGA Architecture Diagram	45
3.6	LabVIEW Real-time Software Diagram	47
3.7	ILC State Machine: Run State	49
3.8	ILC State Machine: Process State	49
3.9	LabVIEW FPGA Firmware Diagram	50
3.10	LabVIEW FPGA Controller Diagram	51

3.11	Optical Tweezers Coordinate System	58
3.12	Optical Power for AOD Scans without RF Power Compensation	59
3.13	Measuring the optical power across the AOD scan range	60
3.14	Detection System Alignment	63
3.15	Example Oscilloscope Readout During Alignment	64
3.16	Detection system sensitivity to a 150 <i>Hz</i> , 0.75 <i>nm</i> amplitude sinusoid	65
4.1	Scan Data a single micro-rod. Sub-figures (a) and (b) show the results of scanning the micro-rod shown in sub-figure (d) along the <i>X</i> and <i>Y</i> axes, respectively. Sub-figure (f) shows the same micro-rod's orientation when the beam is multiplexed ± 2000 <i>nm</i> along the <i>Y</i> -axis on either side of the micro-rod. <i>Y</i> -axis scan data for the micro-rod when held with the multiplexed laser is included in sub-figure (c). Sub-figure (e) shows the micro-rod held in the horizontal orientation with two traps near its ends. The beam visible in the center of the horizontal micro-rod is the detection laser whose trapping strength is negligible.	72
4.2	ILC concept diagram. The Plant block, shown in yellow, represents the micro-rod manipulations and feedback control that occur in real-time during an iteration. The dark blue blocks and arrows, including the ILC block, represent calculations performed offline in-between iterations.	74
4.3	The reference trajectories for the IL processes. Sub-figure (a) contains the reference trajectory for the <i>XZ</i> -plane IL process. Sub-figure (b) contains the reference trajectory for the <i>YZ</i> -plane IL process. The shown magnitudes of the reference trajectories as a function of time are the nominal values. Both reference trajectories are scaled during the calibration process to match the detection system.	76
4.4	Feedback Control illustrations. In sub-figure (a), the issue of the micro-rod sliding towards one of the two beams is illustrated. The use of altering the beam multiplexing duty ratio to counter-act this is shown in sub-figure (b). Sub-figure (c) shows the diagram of the feedback controller that accomplishes this duty-ratio alteration in real-time.	77
4.5	Complete Controller Diagram. The blocks in dark blue represent the calculations performed in between iterations. The blocks in yellow represent the calculations happening in real-time while the micro-rod is manipulated during an iteration.	78

4.6	Detailed results for one XZ -plane IL trial. In the first sub-figure, the feedforward (F.F.) input (Beam 1 X -Pos.) for all iterations are plotted super-imposed over each other. In each iteration the F.F. input is applied to the micro-rod, with the resulting QPD_x signal plotted in the second-subfigure. A color gradient is used to indicate the iteration number of each corresponding F.F. and QPD_x time-series. The reference trajectory $QPD_{x,ref}$ is super-imposed over the QPD_x signals in the second sub-figure. The third sub-figure shows the average percent steady-state (S.S.) error for each iteration. A smoothing spline is fit to this error data with a smoothing parameter equal to 0.5.	80
4.7	Detailed results for one YZ -plane IL trial. The format is the same as used for Figure 4.6 but with Beam 1 Y -position, QPD_y , and $QPD_{y,ref}$ in place of Beam 1 X -position, QPD_x , and $QPD_{x,ref}$. The color gradient indicates which iterations the data in the first two sub-figures belong to. As before, a smoothing spline with smoothing parameter 0.5 is fit the average percent steady-state error data. . . .	81
4.8	Three repeat XZ -plane IL trials performed in succession on the same micro-rod. The left sub-figure shows the three feedforward inputs learned by the three trials. The right sub-figure shows the corresponding QPD_x signals that the learned feedforward inputs produced. The reference trajectory $QPD_{x,ref}$ is plotted super-imposed over the QPD_x signals to show that the learned inputs achieve micro-rod rotation.	83
4.9	Three repeat YZ -plane IL trials performed in succession on the same micro-rod. The three learned feedforward inputs are plotted in the left sub-figure, with the corresponding QPD_y signals plotted in the right sub-figure. The super-imposed reference trajectory $QPD_{y,ref}$ shows that the learned feedforward inputs produce micro-rod rotation.	84
4.10	XZ -plane IL trial results from five different micro-rods on five different days. The left sub-figure shows the learned feedforward inputs from the five trials. The corresponding QPD_x signals are included in the right sub-figure. As implied by the overlaid $QPD_{x,ref}$ reference trajectory, all of the five learned inputs achieve micro-rod rotation.	85
4.11	YZ -plane IL trial results from five different micro-rods on five different days. The left sub-figure shows the learned results of the IL trials, the second sub-figure shows the corresponding QPD_y signals which successfully track the $QPD_{y,ref}$ reference. This indicates successful rotation.	85

4.12 Micro-rod rotation test using a learned feedforward input from a previous experiment on a past day. Sub-figure (a) shows the results of applying the pictured learned XZ -plane rotation feedforward input to a micro-rod ten times. In the plot for QPD_x , horizontal dashed lines indicate the QPD_x magnitudes associated with the vertical and horizontal orientations. All ten QPD_x signals reach the horizontal value indicating succesfull rotaiton. Sub-figure (b) follows the same format, but for YZ -plane rotation. The learned feedfoward input is applied ten times, and as seen in the QPD_y plot, rotation is achieved all ten times. 86

List of Abbreviations

Abbreviation	Meaning
ACF	Autocorrelation Function
ADC	Analog-to-Digital Converter
AOD	Acousto-Optic Deflector
ATW	Amplitude Tuning Word
BNC	Bayonet Neill-Concelman (Connector Type)
CL	Closed-Loop
CS	Chip Select
DAC	Digital-to-Analog Converter
DDA	Discrete Dipole Approximation
DDS	Direct Digital Synthesizer
DIO	Digital Input-Output
EBCM	Extended Boundary Condition Method
EM	Electromagnetic
FDTD	Finite Difference Time Domain
FFT	Fast Fourier Transform
FIFO	First-In, First-Out
FPGA	Field Programmable Gate Array
FTW	Frequency Tuning Word
GUI	Graphical User Interface
IL	Iterative Learning
ILC	Iterative Learning Controller
IO	Input-Output
MSD	Mean Squared Displacement
NA	Numerical Aperture
n.d.	Non-Dimensional
NI	National Instruments
PSD	Power Spectral Density
PSU	Power Supply Unit
QPD	Quadrant Photodiode
RF	Radio Frequency
RMS	Root Mean Squared
RT	Real-Time
sCMOS	scientific Complementary Metal Oxide Conductor
Continued on next page	

Abbreviations – continued from previous page

Abbreviation	Meaning
SPI	Serial Peripheral Interface
TDMS	Technical Data Management Streaming
VHDL	Very High-speed integrated circuit Hardware Description Language
VI	Virtual Instrument (LabVIEW)
VSWF	Vector Spherical Waveform
w/v	Weight-by-Volume

Chapter 1

Introduction

Physical intuition can often be misleading when considering extreme scales or environments. A prime example of this is the momentum of light. In the typical scale and environments considered in day-to-day life, the momentum of light is trivial. However, in the space or at the micron scale this intuition fails. Within the frictionless vacuum of space or the small masses of micron-scale particles, the momentum of light is no longer negligible. Solar sails demonstrate this in the context of space, while laser light reveals light’s momentum at the small scale. In 1970, this momentum was shown to exert meaningful forces on micron-sized particles by Arthur Ashkin [1]. He unlocked yet more potential when he demonstrated in 1986 that a tightly focused laser can trap micron-scale particles near its focal point [2]. Since then the “optical tweezers” approach has facilitated many discoveries at the small scale. In fact, Ashkin’s work was so influential that he was awarded the 2018 Nobel Prize in Physics.

This thesis intends to advance the state of the optical tweezers field by enhancing the optical manipulation of non-spherical particles through learning-based control. Before presenting this research it is helpful to cover relevant background information needed for the approaches in later chapters. First, the physics of optical trapping is presented which leads into the discussion of numerical methods relevant to this thesis’s simulation work. Then the benefits of applying dynamic control to optical trapping is established by reviewing spherical particle trapping applications. Manipulating non-spherical particles is an important use-case for optical tweezers and presents unique challenges for dynamic control. The concept of iterative learning control is introduced as the solution for the challenges posed by non-spherical particles. Finally, this chapter concludes with the outline of this thesis.

1.1 Optical Trapping Mechanism

The theory behind optical trapping is a fervent field of research in of itself. For traditional optical trapping, the particle must be dielectric and have a refractive index higher than its surrounding medium. The essential mechanism of optical trapping is the creation of a gradient potential directed towards the laser's focal point. The gradient potential will exert a force on a dielectric particle that counterbalances the scattering force of the laser light and random perturbations from the particle's surroundings. This creates a three-dimensional potential well that holds the particle near the laser's focal point. The method for quantifying and explaining the creation of the optical trap is dependent on the size of the trapped particle. The size parameter γ discerns the appropriate approach [3] with:

$$\gamma = 2\pi \frac{r_p}{\lambda_m} \quad (1.1)$$

where λ_m is the wavelength of the trapping light in the trapping medium, and r_p is the principle dimension of the particle. The following three ranges of γ are used to discern the best approach for explaining the optical trapping effect and quantifying its forces:

- **If $\gamma \ll 1$ and $(\eta_p/\eta_m)\gamma \ll 1$** (η_p = refractive index of particle, η_m = refractive index of the medium): In this case the particle is much smaller than the wavelength of the trapping laser. Rayleigh scattering can be used to find the force and torque on the particle.
- **If $\gamma \gg 1$:** geometric optics can be used to determine the force and torque on the particle.
- **If $\gamma \approx 1$:** solving for the force and torque on the particle involves approximations and numerical solutions.

In the following sub-sections, the underlying theory for optical trapping in each γ range is summarized. Particular attention is paid to the case of $\gamma \approx 1$ as this will be relevant in later chapters.

1.1.1 Geometric Optics Regime

The geometric optics regime ($\gamma \gg 1$) is considered first as it lends itself well to a more intuitive explanation of optical trap forces. This involves ray tracing as illustrated for an optical trapping scenario in Figure 1.1. In sub-figure (a) the spherical particle shown in blue is below the focal point of the laser. Sub-figure (b) shows the same particle above the focal point of the laser. The black vectors represent the momentum of the laser light rays. The input momentum represented

by $k_{1,in}$ and $k_{2,in}$ enters the particle which acts as a lens to bend the rays of the laser light. The angle of refraction of the light and its momentum is defined by Snell's law. The momentum after the light interacts with the particle is represented by vectors $k_{1,out}$ and $k_{2,out}$. The refraction of the laser light and therefore its momentum results in a net change in momentum Δk_p . Thanks to the particle having a refractive index higher than its surrounding medium, the rays are refracted in such a way that the net change in momentum Δk_p pulls the particle towards the focal point of the trap. This is true both when the particle is below the focus as in sub-figure (a), and above the focus as in sub-figure (b). The analysis for displacements side-to-side with respect to the focal point is similar, with a restoring force pulling the particle back towards the focus. The three-dimensional restoring force the focused laser exerts on the particle keeps it trapped near the laser's focal point. When paired with actuating optics, the trap can be made to move and the particle will move with it. More detailed reviews of the geometric optics approach can be found in [3, 4].

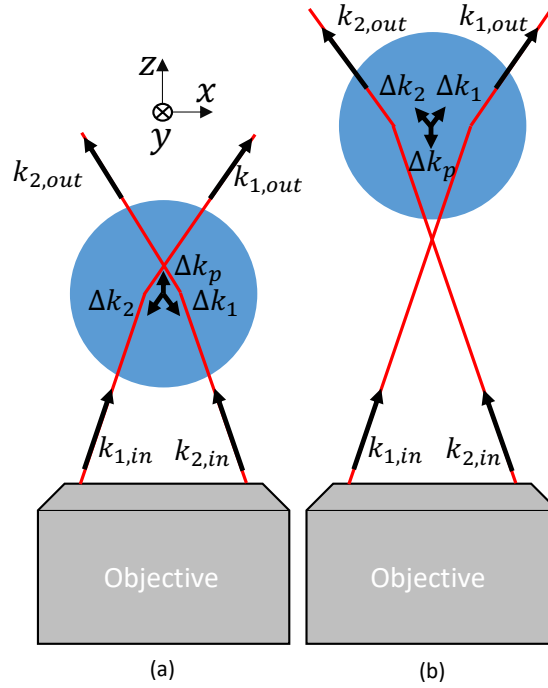


Figure 1.1: Geometric Optics Optical Trapping Ray Tracing Diagram

1.1.2 Rayleigh Regime

A particle with a size parameter $\gamma \ll 1$ can be approximated as a point dipole and therefore Rayleigh scattering can be used to find the forces on it. The oscillating electromagnetic (EM) field of the laser induces an oscillating dipole within the particle. The oscillating dipole will create a scattered EM field and a gradient EM field. The force on the dipole particle from each can be quantified, with the scattering force equation shown below [5]:

$$F_{scatt}(r_d) = \frac{\sigma}{c} n_m I_0(r_d) \quad (1.2)$$

where $r_d \in \mathbb{R}^3$ is the dipole's position, I_0 is the intensity of the incident light, n_m is refractive index of the medium surrounding the particle, c is the speed of light in a vacuum, and σ is the scattering cross section of a sphere, defined as follows:

$$\sigma = \frac{128\pi^5 r^6}{3\lambda^4} \left(\frac{m^2 - 1}{m^2 + 2} \right)^2, \quad (1.3)$$

where $m = n_p/n_m$, r is the sphere's radius, and λ is the wavelength of the trapping laser. The scattering cross section is proportional to the rate at which energy is scattered from the incident wave by the scatterer [3]. The scattering force is pointed in the direction of the EM wave's travel. The gradient force component is calculated as follows:

$$F_{grad}(r_d) = \frac{2\pi\alpha}{cn_m^2} \nabla I_0(r_d), \quad (1.4)$$

where

$$\alpha = n_m^2 r^3 \left(\frac{m^2 - 1}{m^2 + 2} \right) \quad (1.5)$$

is the polarizability of the sphere with $m = n_p/n_m$. The polarizability is the proportionality coefficient between the induced polarization of the sphere and the incident EM wave [3]. When $n_p > n_m$, the polarizability is positive and F_{grad} will be pointed towards the focus of the laser. Note that F_{grad} is proportional to r^3 , while F_{scat} is proportional to r^6 . Therefore, for small enough spheres, the gradient force will be larger than the scattering force [6]. This is the key for forming a stable optical trap.

1.1.3 Wavelength-Scale Regime

The last case is for particles whose principle dimension is approximately equal to the wavelength of the trapping light ($\gamma \approx 1$). This is the most difficult case to quantify the trapping forces

for. Coincidentally, its also the most common regime for optical trapping as trapped cells are typically near this range, as are the beads typically used. In this case, the electromagnetic scattering problem must be solved to find the scattered EM waveform and in turn, the force.

For a homogeneous spherical particle, Mie scattering theory can be applied. Mie scattering theory involves expanding the incident, scattered, and internal (to the particle) waves into infinite sums of vector spherical waveforms (VSWF). Vector spherical waveforms are solutions of the vector Helmholtz equation. The expansion is akin to a Fourier series expansion of a time-domain signal. Note that for the purposes of this introduction, it is assumed the incident waveform is a linear polarized plane-wave. A treatment for focused laser beams can be found in [3, 7]. The expansion of the incident E_i , scattered E_s , and internal E_p waveforms are included below [3]:

$$E_i(r, \hat{r}) = E_i \sum_{l=0}^{\infty} \sum_{m=-l}^l p_{l,m}^{(1)} J_{l,m}^{(1)}(k_m r, \hat{r}) + p_{l,m}^{(2)} J_{l,m}^{(2)}(k_m r, \hat{r}) \quad (1.6)$$

$$E_s(r, \hat{r}) = E_i \sum_{l=0}^{\infty} \sum_{m=-l}^l q_{l,m}^{(1)} H_{l,m}^{(1)}(k_m r, \hat{r}) + q_{l,m}^{(2)} H_{l,m}^{(2)}(k_m r, \hat{r}) \quad (1.7)$$

$$E_p(r, \hat{r}) = E_i \sum_{l=0}^{\infty} \sum_{m=-l}^l w_{l,m}^{(1)} J_{l,m}^{(1)}(k_m r, \hat{r}) + w_{l,m}^{(2)} J_{l,m}^{(2)}(k_m r, \hat{r}) \quad (1.8)$$

where r is the radial distance and $\hat{r}$ is the radial unit vector in spherical coordinates. The l summation is over the radial modes; the m summation is over the azimuthal modes. The functions $J_{l,m}^{(1)}$ and $J_{l,m}^{(2)}$ are the first and second kind J -multipoles. The J -multipoles are functions of VSWFs. The H -multipoles, $H_{l,m}^{(1)}$ and $H_{l,m}^{(2)}$, are also functions of VSWFs. The terms $p_{l,m}^{(1)}$, $p_{l,m}^{(2)}$, $q_{l,m}^{(1)}$, $q_{l,m}^{(2)}$, $w_{l,m}^{(1)}$, and $w_{l,m}^{(2)}$ are the amplitude coefficients of their respective VSWF expansions. The corresponding equations for the magnetic fields B_i , B_s , and B_p are found by substituting the expression of Faraday's law of induction into Equations (1.6) through (1.8).

The amplitude coefficients $q_{l,m}^{(1)}$ and $q_{l,m}^{(2)}$ of the scattered wave are found by applying a boundary condition to the EM fields. For a non-magnetic particle, the tangential components of the internal field must equal the tangential components of the external fields (incident plus scattered) on the surface of the particle. Setting the tangential components of the sum of the external fields equal to the tangential components of the internal field creates a set of integral equations that can be used to solve for the amplitudes of the scattered field. The end results are expressions for $q_{l,m}^{(1)}$ and $q_{l,m}^{(2)}$ as scalar multiples of $p_{l,m}^{(1)}$ and $p_{l,m}^{(2)}$, respectively:

$$q_{l,m}^{(1)} = a_l p_{l,m}^{(1)}, \quad (1.9)$$

$$q_{l,m}^{(2)} = b_l p_{l,m}^{(2)}. \quad (1.10)$$

The scalar coefficients a_l and b_l are the Mie coefficients and are functions of l but not m . There is no coupling between the modes, i.e. $q^{(1)}$ is a function of $p^{(1)}$ only and the same applies to $q^{(2)}$ and $p^{(2)}$. This is thanks to the orthogonality of the modes on the surface of a sphere. The summation of the scattered field's VSWFs in Equation (1.7) is infinite, but can be truncated at the threshold $l = \gamma$ [3]. For small particles, this leads to truncation at $l = 1$, which reduces Mie theory to Rayleigh scattering. Once the scattered waveform is known, the force on the spherical particle can be found by integrating Maxwell's stress tensor over its surface.

Note that the relation between the coefficients for the scattered waveform modes $q^{(1)}$, $q^{(2)}$ and the incident waveform modes $p^{(1)}$, $p^{(2)}$ can be written in a matrix multiplication form:

$$\underbrace{\begin{bmatrix} q_{1,-1}^{(1)} \\ q_{1,0}^{(1)} \\ q_{1,1}^{(1)} \\ \vdots \\ q_{\gamma,\gamma-2}^{(2)} \\ q_{\gamma,\gamma-1}^{(2)} \\ q_{\gamma,\gamma}^{(2)} \end{bmatrix}}_{\mathbf{q}} = \underbrace{\begin{bmatrix} a_1 & 0 & 0 & & 0 & 0 & 0 \\ 0 & a_1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & a_1 & & 0 & 0 & 0 \\ & \vdots & & \ddots & & \vdots & \\ 0 & 0 & 0 & & b_\gamma & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & b_\gamma & 0 \\ 0 & 0 & 0 & & 0 & 0 & b_\gamma \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} p_{1,-1}^{(1)} \\ p_{1,0}^{(1)} \\ p_{1,1}^{(1)} \\ \vdots \\ p_{\gamma,\gamma-2}^{(2)} \\ p_{\gamma,\gamma-1}^{(2)} \\ p_{\gamma,\gamma}^{(2)} \end{bmatrix}}_{\mathbf{p}} \quad (1.11)$$

This is the **T**-matrix formalism for solving EM scattering problems. The linearity of the relationship between the incident and scattered waveforms is owed to the linearity of Maxwell's equations. The matrix **T** depends only on the properties of the scattering particle [8], so once it is known it can be used repeatedly to solve the EM scattering problem of optical trapping for different particle orientations and positions relative to the trap. This is an important efficiency when numerically simulating a particle's interactions with light.

In 1965, Waterman published the extended boundary condition method (EBCM) which expanded Mie theory and the **T**-matrix formalism to homogeneous non-spherical particles [9]. The full derivation is beyond the scope of this thesis but will be summarized here. The full derivation can be found in [9, 10]. The EBCM method uses the fact that the tangential components of the EM fields generated by surface currents on the particle must cancel, much like in Mie theory. The electric fields E_i and E_s are expressed in terms of these field currents. These field expressions in terms of surface currents involve integrating the scalar free space Green's function

$G(\mathbf{r}, \mathbf{r}')$, where $\mathbf{r}$ is the radial position and $\mathbf{r}'$ is a vector extending either into the particle or out to the exterior surroundings. For E_i , $\mathbf{r}'$ extends into the particle while for E_s , $\mathbf{r}'$ extends outward. The fact that the Green's function analytically extends by $\mathbf{r}'$ in/out from the particle surface boundary is the origin of the EBCM's name [10]. Because Green's functions extend in/out of the particle, the expressions of the fields in terms of surface currents do not necessarily have to be evaluated on the non-spherical particle's surface. Instead, the multipole expansion of Green's function for E_i is found on the largest spherical surface that fits inside the particle. Conversely, the multipole VSWF expansion of E_s is found on the smallest spherical surface that contains the entire particle. Solving for the scattered waveform E_s starts by using the multipole expression for E_i and the tangential field boundary conditions on the actual surface of the particle to find E_p , the field internal to the particle. Then, the tangential field conditions are used to put the expression for E_s in terms of E_p which is now known. Then the relation between E_i and E_s is found, and again thanks to the linearity of Maxwell's equations, can be put in the $\mathbf{T}$ -matrix formulism. Its important to note that for a non-spherical particle the $\mathbf{T}$ -matrix will not be diagonal as it is for a spherical particle.

Since the introduction of the EBCM, new methods for finding the $\mathbf{T}$ matrix has been an active area of research. Three common methods include the finite-difference time-domain (FDTD) method, the point matching method, and the discrete dipole approximation (DDA). The FDTD solves the scattering problem by discretizing both the particle volume and its surroundings and stepping Maxwell's curl equations in time. This method is not restricted to homogenous particles or monochromatic light but is computationally intensive [3]. The fields determined by FDTD can be matched to the VSWF expansions of the incoming and outgoing fields to find the $\mathbf{T}$ -matrix [11]. The point matching method is similar to the EBCM except that instead of evaluating surface integrals on the particle, the internal and external fields are equated at many points around the particle [12]. This forms an overdetermined set of equations that can be solved using the least squares method of optimization. The DDA method [13] is a finite element approach involves treating the volume of particle as an ensemble of small dipoles. The expression for the dipole moment of each dipole includes not only the incident electric field but also the re-radiated fields from neighboring particles [3]. This leads to a system of equations that can be solved to find the field at each dipole. Once the dipoles' fields are known, they can be represented as a field emanating from one origin and the VSWF expansion coefficients needed to find the $\mathbf{T}$ -matrix can be found [14]. Point-matching can also be used to take the results of the DDA method and find the particle's $\mathbf{T}$ -matrix [15].

Choosing the appropriate method for finding the $\mathbf{T}$ -matrix is important for accurate and efficient simulations of optical trapping. Qi et al. studied the performance trade-offs of the EBCM, point matching method, and DDA method for cylinders in both the $\gamma \ll 1$ and $\gamma \approx 1$

size ranges [16]. They found that the point-matching method offers the fastest computation time as it avoids the surface integrals of the EBCM while the DDA method is the slowest due to its finite element approach. For their results, the DDA method served as the benchmark for force calculation accuracy on all cylinder sizes after passing a convergence analysis. The point matching method only worked well for cylinders in the $\gamma \approx 1$ range with aspect ratios between 0.5 and 2. The EBCM worked well for $\gamma \approx 1$ scale cylinders provided their aspect ratio was greater than 0.1. In the $\gamma \ll 1$ range, the EBCM worked well but only for cylinders with an aspect ratio greater than 1.4. Overall, DDA was shown to be the most accurate and flexible in regards to particle shape but also the most computationally intensive of the three methods. This will be the method of choice in Chapter 2.

Conveniently there exists a computational toolbox for simulating a particle's interaction with an optical trapping laser [17]. The underlying theory of the toolbox is outlined in [8]. This toolbox offers the aforementioned methods of calculating the **T**-matrix and has built-in functions for calculating the force and torque on a spherical or non-spherical particle. The toolbox accounts for the focusing of the optical trapping laser. An example output for a spherical particle generated for this thesis is included in Figure 1.2. This figure shows the incident and scattered fields, which combine to form the total field. Once the fields are known, the toolbox can calculate the force and torque on the particle. In Chapter 2, this toolbox will be used to calculate the T-matrix for a non-spherical particle and calculate the force and torque on the particle at each simulation time step.

1.2 Particle Dynamics

While the previous section described the forces of the laser light on the particle, these are not the only forces that must be considered. Optically-trapped particles, being of the micron or smaller scale and immersed in a fluid, are subject to random collisions from the molecules constituting their surrounding medium. This results in the particle undergoing random motion that the potential energy well of the optical trap must contain to trap the particle. Here the mathematical formulation for this random motion is summarized followed by the dynamics of a particle undergoing Brownian motion in a potential energy well.

1.2.1 Brownian Motion

The Wiener process is a mathematical formulation for Brownian motion. Defined on a continuous interval of time $t \in [0, T]$, the Wiener process $W(t)$ is a random variable defined to satisfy the following [18]:

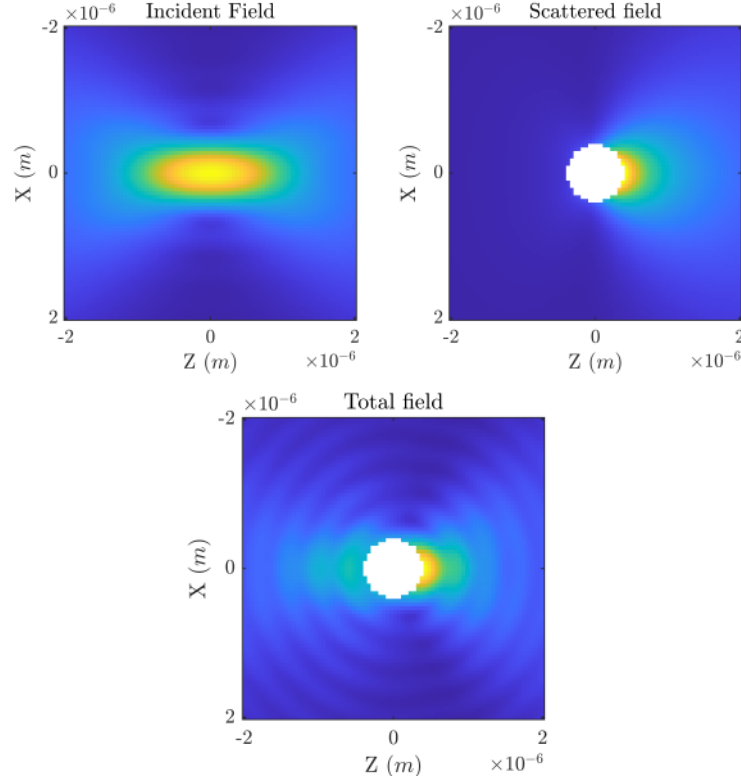


Figure 1.2: EM Scattering for a Sphere

1. $W(0) = 0$ with probability equal to one.
2. For $0 \leq s < t \leq T$, the increment $W(t) - W(s)$ is a random variable with distribution $N(0, \beta^2(t - s))$, where $\beta \in \mathbb{R}$. The distribution can also be expressed as $\beta\sqrt{t - s}N(0, 1)$, where $N(0, 1)$ is the unit normal distribution.
3. Two increments $W(t) - W(s)$ and $W(v) - W(u)$ where $0 \leq s < t \leq u < v \leq T$, are independent.

The β term is the process parameter which is related to the size of the process increments [19]. Condition number two implies that the variance of the increment $W(t) - W(s)$ increases linearly with the time interval length $t - s$. It is important to note the Wiener process is not differentiable and is a self-similar process.

To understand why the Wiener process is an apt mathematical description of Brownian motion, it helps to see how a random walk process converges in distribution to a Wiener process. The full proof is available in [20, 21] and is summarized here. In a sense, a particle colliding

with its surroundings is undergoing a random walk. Consider a random walk in one-dimension with steps at discrete time instances $i = 1, \dots, n$ for some $n \in \mathbb{N}$. Let the random variable S_i represent a step that can go a distance σ to the right or left with equal probability and mean equal to zero. Let $S_0 = 0$. With the step length σ , the variance of S_i is equal to σ^2 . Define the random variable X_N as:

$$X_n = \sum_{i=1}^n S_i \quad (1.12)$$

with $X_0 = 0$. Now define the continuous time stochastic process $Z_t^{(N)}$ as the scaled version of X_n with scaling parameter $N \in \mathbb{N}$

$$Z^{(N)}(t) = \frac{1}{\sqrt{N}} X_{\lfloor Nt \rfloor} \quad (1.13)$$

where $t \in \mathbb{R}^+$ and $\lfloor Nt \rfloor$ is the integer floor of Nt . The scaling term N is interpreted as the number of random walk steps S_i per unit of continuous time t . For large N , $\langle Z^{(N)}(t) \rangle = 0$ and $\text{var}(Z_t^{(N)}) \approx \sigma^2 t$. The Functional Central Limit theorem states that as $N \rightarrow \infty$, $Z(t) \sim N(0, \sigma^2 t)$. Furthermore, $Z(t_2 - t_1) \sim N(0, \sigma^2(t_2 - t_1))$ for $t_2 > t_1$. It can also be shown that $Z(t)$ satisfies conditions one and three for the Wiener process. Therefore, process $Z(t)$ which is the limit of a random walk as the step size becomes infinitesimally small, is a Wiener process with σ as the process parameter. This connects the underlying intuition of Brownian motion as a series of particle collisions to its formal treatment as a Wiener process.

The process parameter β in the Wiener process definition needs to be connected to the physics of the micron-sized particle undergoing Brownian motion. The diffusion equation, as derived by Albert Einstein in 1905 [22], governs the evolution of the probability density function for the position of a micron-scale particle suspended in a liquid:

$$\frac{\partial p(x, t)}{\partial t} = D \frac{\partial^2 p(x, t)}{\partial x^2} \quad (1.14)$$

where D is the diffusion constant from Flick's law. The solution to the diffusion equation is $p(x, t) \sim N(0, 2Dt)$. Comparing this to the distribution of the general Wiener process $W(T) \sim N(0, \beta^2 t)$, it follows that $\beta^2 = 2Dt$ for a micron-scale particle suspended in liquid undergoing Brownian motion.

1.2.2 Langevin Dynamics

Langevin offered another mathematical formulation for Brownian motion that is well-suited for simulating a particle's Brownian motion and interaction with a potential well like that of an

optical trap. Langevin's equation was the first stochastic differential equation and predated the necessary mathematical framework of stochastic calculus by 40 years [23]. The formulation for the Langevin dynamics of a particle with mass m , position $x(t)$, and velocity $v(t) = dx/dt$ is given below [24]:

$$m \frac{dv}{dt} = -\gamma v + L(t) \quad (1.15)$$

where γ is the damping coefficient of the particle and $L(t)$ is a random process with the following properties: $\langle L(t) \rangle = 0$ and $\langle L(t_2)L(t_1) \rangle = \Gamma \delta(t_2 - t_1)$. These properties arise of the physical characteristics of Brownian motion. The term Γ can be shown equal to $2k_B T \gamma$ thanks to the requirement that the mean square velocity of such a particle in equilibrium must equal $k_b T$. Note that $L(t) = \sqrt{\Gamma} \xi(t) = \sqrt{2k_B T \gamma} \xi(t)$ where $\xi(t)$ is a white noise process so Equation (1.15) can be written as:

$$m \frac{dv}{dt} = -\gamma v + \sqrt{2k_B T \gamma} \xi(t). \quad (1.16)$$

The autocorrelation function of $\xi(t)$ being a function of the Dirac delta $\delta(t)$ means that Equation (1.15) is not mathematically well-defined. This can be made apparent by showing that the integral of $\xi(t)$ is the Wiener process $W(t)$ with process parameter $\beta = 1$, which creates a contradiction because the Wiener process is not differentiable [23]. To get around the Langevin equation not being mathematically well-defined, it can be written in its integral form:

$$v(t) - v(0) = -\frac{\gamma}{m} \int_0^t v(s) ds + \frac{\sqrt{2k_B T \gamma}}{m} \int_0^t \xi(s) ds, \quad (1.17)$$

where $\xi(s)$ is a white noise process. With the integral of $\xi(s) ds$ equal to the Wiener process $W(s)$, the term $dW(t) = W(t + dt) - W(t)$ is defined as equal to $\xi(t) dt$ to get:

$$v(t) - v(0) = -\frac{\gamma}{m} \int_0^t v(s) ds + \frac{\sqrt{2k_B T \gamma}}{m} \int_0^t dW(s). \quad (1.18)$$

The rightmost integral is a stochastic integral taken with respect to the Wiener process. Its calculus provides a formal treatment of this stochastic integral which is beyond the scope of this introduction (see [23]). The stochastic integral can be written in the following stochastic differential equation form:

$$dv(t) = -\frac{\gamma}{m} v(t) dt + \frac{\sqrt{2k_B T \gamma}}{m} dW(t). \quad (1.19)$$

For computation via the finite difference method, the stochastic differential equation form is discretized as:

$$v(t_j) = v(t_i) + -\frac{\gamma}{m} v(t_i) \Delta t + \frac{\sqrt{2k_B T \gamma}}{m} (W(t_j) - W(t_i)), \quad (1.20)$$

where $\Delta t = t_j - t_i$ and, from the definition of the Wiener process, $(W(t_j) - W(t_i)) \sim N(0, \Delta t)$, which for computation can be simplified to $(W(t_j) - W(t_i)) \sim \sqrt{\Delta t}N(0, 1)$. This leads to:

$$v(t_j) = v(t_i) + -\frac{\gamma}{m}v(t_i)\Delta t + \frac{\sqrt{2k_B T \gamma \Delta t}}{m}dW_N(t_i), \quad (1.21)$$

where $dW_N(t_i) \sim N(0, 1)$. The Wiener increment being in terms of the standard normal distribution is convenient for simulations where most computation toolboxes have built-in standard normal Gaussian random number generators.

For optical trapping experiments, the inertial term of Equation (1.15) can be dropped if the particle's dynamics are in the diffusive regime. The momentum relaxation time $\tau_m = m/\gamma$ can be used to discern if this simplification applies [3]. For extremely short momentum relaxation times, the effect of the particle's inertia quickly dissipates in comparison to the time constant of the optical trap's restoring force. Therefore in these cases, the inertial term can be dropped, leading to a overdamped Langevin equation:

$$v = \sqrt{2D}\xi(t), \quad (1.22)$$

where the Einstein relation has been used to simplify the white noise coefficient term. The Langevin equation can also accommodate the force $F(x(t))$ from the optical trap on the particle:

$$v = \frac{1}{\gamma}F(x(t)) + \sqrt{2D}\xi(t). \quad (1.23)$$

For a spherical particle, the optical trap will form a three-dimensional Gaussian potential well with $F(x(t)) = -dU(x(t))/dx$. This potential can be approximated as linear for small displacements as illustrated in Figure 1.3. The linear stiffness coefficient K_x must be determined empirically, as will be outlined in Chapter 3. Then the Langevin equation becomes:

$$v = -\frac{K_x}{\gamma}x + \sqrt{2D}\xi(t). \quad (1.24)$$

This is the most common form used of the Langevin equation for optical trapping applications. For computations it takes the form:

$$x(t_j) = x(t_i) - \frac{K_x}{\gamma}x(t_i)\Delta t + \sqrt{2D\Delta t}dW_N(t_i). \quad (1.25)$$

In Chapter 2, the Langevin equation will be put to use to simulate the dynamics of an optically trapped cylinder. Since a cylinder is non-spherical, the Langevin equation for rotational dynamics will be introduced. For a non-spherical particle like a cylinder, the linear approximation shown in Figure 1.3 does not apply. This has significant implications for the real-time dynamic control of non-spherical particles. In simulation where the restriction of real-time

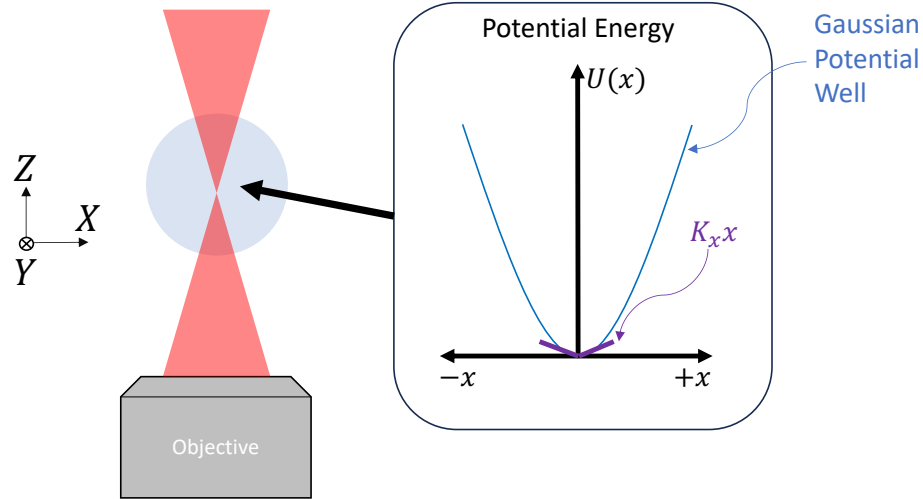


Figure 1.3: Linear Approximation of Optical Trap Potential

computation does not apply, the force and torque on a non-spherical particle can be solved numerically using the aforementioned computational toolbox. This will be done in Chapter 2. However, for real-time applications there are no closed-form analytical solutions for the force and torque on the particle so model based control cannot be applied. The remaining sections of this chapter will establish the success of real-time control when applied to spherical particles, the challenges of applying it to non-spherical particles, and this thesis's aim of overcoming those challenges.

1.3 Spherical Particles: Applications & Dynamic Control

Optical tweezers have seen broad applications in science and engineering [3, 25, 26]. In physics they are used to artificially cool arrays of atoms for quantum science experiments [27]. In the field of nanotechnology, optical tweezers are used to assemble nanostructures [28]. Perhaps the most pertinent field is biology where optical tweezers have facilitated break-through single-molecule biophysical studies [29, 30, 31]. In these experiments micron-sized polystyrene beads act as handles for protein molecules. When a handle is placed on each end of the protein, the folding and unfolding of the protein can be studied by moving the bead handles with optical tweezers. Another experiment setup studies molecular motor proteins that when in vivo, carry cargo around the cell along tracks. In vitro molecular motor experiments typically involve attaching a polystyrene, micron-sized bead to the molecular motor as its cargo. The cargo is then held in the optical trap. The molecular motor will attempt to pull the cargo along a microtubule

or actin filament placed on the sample chamber floor. The bead position data is then used to study its pulling force and stepping behavior.

For those motor-protein studies, the pulling force of the motor protein is in the piconewton range, with step sizes measured in nanometers [32, 33]. With such minute dynamics, the performance of the optical tweezers system is crucial. To improve the signal-to-noise ratio of optical tweezers for these experiments and maintain isometric conditions during the experiment, feedback control is extensively applied. While in 1977 Ashkin demonstrated the use of feedback to maintain the height of an optically levitated particle along the beam axis [34], feedback control in the plane perpendicular to the optical axis did not rise to prominence until the 1990's. The perpendicular-plane position feedback approaches used in molecular motor studies can be generally divided into two types: constant position and constant force.

The approach that arose first was constant position feedback [35]. Here the error between the bead's actual position and its desired position is used to adjust the position of the trap in real-time. As the name implies, this makes the position of the bead more constant at the prescribed location which reduces the bead's Brownian motion and therefore noise in its position trace. Since the bead's position is now essentially constant, the occurrence of pulling events is not discerned from movements of the bead, but instead movements of the trap as actuated by the feedback controller. The reduction in Brownian motion means that the trap's stiffness, represented by the approximate linear stiffness k , has increased. This increased stiffness is in itself useful for certain pulling experiments that involve higher pulling forces. It is possible to get the same increase in stiffness and reduction in Brownian noise by instead increasing the power of the trapping laser. However, increasing the laser power has the downside of a lower signal-to-noise ratio. This is because increasing the laser power reduces Brownian noise by $1/\sqrt{k}$ but similarly reduces the magnitude of the displacements from motor pulling by $1/k$ [36]. Therefore, if the laser power is increased, the noise of Brownian motion is reduced but the sensitivity to pulling events is even more greatly reduced. Constant position feedback offers a way to suppress the Brownian motion of the particle and increase trap stiffness without a reduction in sensitivity. It has been used extensively in the study of rabbit skeletal HMM proteins motility on actin [36, 33, 37, 35], as well as DNA pulling experiments [38, 39].

The technological limit of biochemistry studies was expanded further by the introduction of constant force feedback. One difficulty in processive motor studies, where a motor protein attempts to walk along a microtubule or actin filament while pulling an optically trapped cargo, is that the trap stiffness changes as the bead is moved from center. Changing trap stiffness creates changing trap forces and in turn, unsuitable experimental conditions. More specifically, variable trap stiffness makes it hard to discern how the stiffness of the motor changes as it pulls as it cannot be separated from the changing stiffness of the trap. In 1998, Visscher et al. [40]

outlined an optical tweezers setup employing feedback to create constant force on the trapped particle. This is accomplished by maintaining the bead at a constant distance from the trap center. The distance from the trap center is chosen to achieve the desired constant force level. Since the stiffness of the trap does not change, the changes in stiffness observed via the bead's motion can be attributed to what the motor is doing. Additionally, the trap ends up following the processive motor as it walks as opposed to the bead being forced back to center when it reaches a region of the trap that is too stiff. This lengthens the amount of time that steps can be observed. A year later, Visscher et al. applied this system to study the stepping behavior of kinesin [40]. Block et al. extended the constant force clamp to two dimensions in [41, 42].

Optical tweezer feedback control has been further improved further through the use of modern, model-based control methods. One of the key limitations of both constant position and constant force feedback is the bandwidth of the controller. If the bead's motion exceeds the controller's ability to compensate in real-time, performance degrades. For example, faster motors can exceed the bandwidth of the controller, causing the position or force to deviate from the setpoint. This has required experiments to be performed at biochemical conditions that slow the motor and are not exactly reflective of in-vivo conditions. Model-based control methods, using empirically determined model coefficients [43] for the model given in Section 1.2.2, have been shown to improve the controller bandwidth such that the trap position can keep up with faster moving motors. Additionally, they can improve the estimation of the motor force on the bead from the observable data. In 2009 this was accomplished for constant position trapping using proportional feedback control and inversion-based estimation of the motor's force [44]. This was further improved with a robust control approach in 2011 [45]. Constant force optical trapping received the model-based, robust control treatment in [46], which was further improved up on and verified using actual motor proteins in [47].

1.4 Non-Spherical Particles: Applications & Control Challenges

The trapping of non-spherical particles is an important application of optical tweezers. Here the problem becomes more complex as the trapped particle has at least one axis about which it is not symmetrical. Therefore, the effect of the trap on the particle is not only position dependent, but also orientation dependent. Additionally, in application it is necessary to control the particle's orientation as well as its position. As an introduction to the trapping of non-spherical particles, applications and methods are first reviewed. Then, the challenges of extending the previous success of dynamic control for spherical particles to non-spherical particles are discussed.

Non-spherical particle manipulation arose early on the history of optical tweezers when

Ashkin et al. used two beams to trap *E. Coli* cells in 1987 [48]. By trapping the *E. Coli*, they were able to observe its reproduction and also show that using a 1064 nm laser for trapping resulted in less damage to living specimens than seen in prior work. Motility characteristics of *E. Coli* including swimming velocity, swimming force, and reversals of swimming direction were studied in [49, 50]. The trapping and orientation of red blood cells is another biological application [51]. Orientation control of an optically trapped cell is used to improve the number of angles tomographic imaging can be performed at in [52, 53]. Outside of biology, optical tweezers can be used to orient non-spherical particles to create micron-scale assemblies [54]. Further review of the applications of non-spherical particle manipulation are available in [55, 56].

Several methods to achieve orientation control over optically trapped non-spherical particles exist. Most of these methods can be grouped into four categories. The first relies on shaping the cross-sectional profile of the beam such that the particle assumes the desired orientation when trapped. A second approach is to rely on specific properties of the particle, such as birefringence, to orient the particle. The third approach relies on beads attached as handles onto the particle of interest. The last approach, which is the most readily applicable to most optical tweezers systems, is to use more than one optical trap to manipulate the particle directly without handles. Other approaches outside these four categories do exist. For example, non-spherical particles have been manipulated using counter-propagating or near counter-propagating laser beams delivered into the sample chamber perpendicularly to the optical axis via tapered fiber probes [57, 58]. Rotation about an axis perpendicular to the optical axis has also been achieved by tilting the incoming beam itself [59]. However, these approaches require niche hardware and in the case of tapered-fiber probes, can deviate from the central principle considered here of using tightly focused light to trap particles. Consequently, this review is limited to the four approaches of beam shaping, utilizing particle properties, bead handles, and direct multi-trap method as they can be realized on more typical hardware.

For reviewing these methods, it is helpful to divide particle orientation control into two main categories: in-plane rotation, and out-of-plane rotation. These primary types of rotation are illustrated in Figure 1.4. In-plane rotation is any rotation about an axis that is parallel to the optical axis of the tweezers. Out-of-plane rotation is any rotation about an axis perpendicular to the optical axis. In the coming sections regarding the four techniques of non-spherical particle manipulation it will be shown that out-of-plane rotation can be a more challenging engineering problem.

1.4.1 Beam Shaping for Orientation Control

While most optical trapping systems use beams with Gaussian cross-sectional profiles that are symmetric about the optical axis, more complicated cross-sectional beam profiles can be created

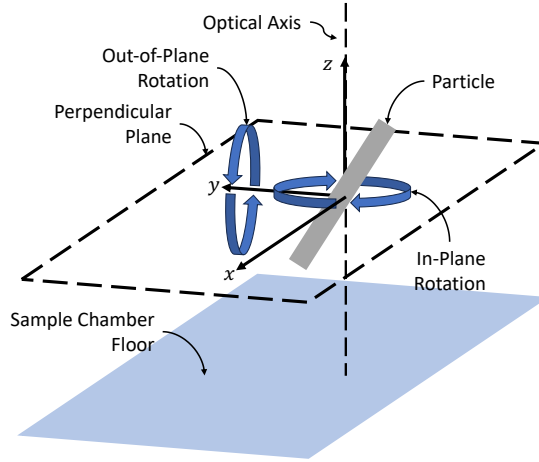


Figure 1.4: In-Plane Rotation vs. Out-of-Plane Rotation

to control the orientation of trapped particles. The aforementioned study of rotating a trapped red blood cell relied on a shaped beam profile produced from Gaussian beam profile by inserting a rectangular aperture into the laser cavity [51]. The focal point of the resulting beam had an elongated cross-sectional intensity profile. It was observed that the elongated dimension of the cell will align with the elongated intensity profile of the beam. If the rectangular aperture in the laser cavity was rotated about the optical axis, the beam profile would rotate and along with it the trapped cell. This alignment of a non-spherical particle with the asymmetric intensity profile of the beam illustrates the underlying mechanism of beam shaping techniques. Another approach to produce a shaped beam is to combine two beams that have different profiles. A combination of a plane-wave and Laguerre-Gaussian beam can be used to induce the in-plane rotation of an optically trapped particle [60]. Out-of-plane rotation was achieved in [61], where a conventional optical trap is combined with an elliptical focal-point optical trap. The orientation of a trapped quartz cylinder was then controlled by adjusting the relative power of the two beams. If the conventional trap power dominates, the micro-rod will be vertical in the trap, if the elliptical trap power dominates, the micro-rod will be horizontal. While these beam shaping methods succeed in controlling the orientation of a particle, they are difficult to apply as most optical tweezers systems employ a single, Gaussian profile laser for trapping.

1.4.2 Utilizing Particle Properties for Manipulation

Orientation control can also be achieved by exploiting the properties of the particle to be manipulated. One trait commonly utilized for orientation control is that of birefringence. Birefringent particles have an index of refraction that depends on the polarization and angle of the incident

light. This can lead to interesting behavior in an optical trap. In 1998, Friesen et. al [62] showed that a birefringent calcite particle can be made to rotate in an optical trap. When calcite particles are in an optical trap, they essentially act as a half-wave plate, altering the polarization of the trapping laser light as it passes through the particle. This changes the net angular momentum of the light, imparting a torque on the particle. They showed that if the incident laser light is plane polarized, the calcite fragment can be held statically at an angle in a plane perpendicular to the optical axis. The calcite fragment can then be rotated through desired angles by rotating the polarization filter of the laser. If the incoming laser light is circularly polarized, the calcite fragment will rotate at a near constant rate. This approach was further refined with angular torque measurements made using photodiodes [63]. In this case, a linearly polarized trapping laser is used and its polarization angle is controlled to apply torque and rotate the particle. The calcite particle splits the polarization into left and right circularly polarized components whose powers are each measured by their own photodiodes. The torque on the particle is proportional to the difference in the measured powers of the left and right polarized light. The torque is made constant by using the torque setpoint error to modify the trapping laser's polarization angle through real-time feedback. The major drawback of relying on particle properties to control orientation is that it is specific to a particular material. Therefore it is not a flexible approach.

1.4.3 Indirect Manipulation Using Handles

Another approach is to use handles held in optical traps to indirectly manipulate a particle of interest. This method is primarily applied to cells that are sensitive to laser exposure. In 2009 Arai et al. manufactured a micro-tool that they proposed could be used to grab and manipulate cells [64]. A new specially fabricated micro-tool was introduced in [65], which achieved the out-of-plane rotation of a yeast cell. The downside of these custom tools is the necessity of advance fabrication techniques to make them. To get around this, readily available beads can be used as handles and were shown to work for the indirect trapping of a particle as early as 1993 [66]. Later a fused pair of beads were used as handle for a bacterial flagellar motor [67]. By trapping the handle, they could exert torque on the motor and control its movement. For the movement of cells, Chowdhury et al. automated the spherical bead handle trapping process by holding several beads in a mathematically defined grabber formation. They implemented a path planning algorithm to navigate the grabbed cell around obstacles to a desired location. With the assumption that both the handles and the cell are spheres, they implemented a visual-based feedback controller to maintain the position of the bead handles while the cell is moved or rotated in the plane perpendicular to the optical axis. This work was later evolved by including the dynamics between the bead and the cell in the controller design and analysis [68]. A stochastic controller design and analysis for this problem that adds feedback control for in-plane rotation

is given in [69], though it assumes rigid contact between the handles and the cell. The spherical bead handle manipulation approaches mentioned thus far have been limited to two dimensional control. It was later extended to three dimensions by the introduction of actuation of the trap position along the optical axis using a deformable mirror [70]. It is important to emphasize that in all of these controller designs, the particle of interest is assumed to be spherical. This is the key requirement for those model-based controller designs and stability analyses. Commentary in [69] mentions that determining the bead handle formation for a non-spherical particle is less easily solved as handle placement for stable grasping is not as obvious.

1.4.4 Direct Manipulation Using Multiple Traps

Applicable to any particle regardless of shape is the method of direct manipulation using more than one optical trap as end effectors. No unusual beam profiles are required, or birefringent particles, or the orchestration of multiple bead handles. This was Ashkin et. al's aforementioned approach in 1987 for holding an E. Coli cell using two traps formed by two lasers. Later this was simplified by using one laser time shared between three trap locations on a bacterium using two scanning mirrors [66]. Tanaka et al. trapped and moved diatoms using a vision-based controller that assigned trapping locations to the diatom and planned a clear path for translating and rotating it in the plane perpendicular to the optical axis [71]. The multiple traps were created using a scanning mirror. The aforementioned example of building micron-scale assemblies used the multiple direct trap method [54], as did the E. Coli swimming studies [49, 50] and the tomographic cell imaging studies [52, 53]. These numerous and broad applications highlight the versatility and relevance of the direct manipulation method.

One of the more challenging control problems that direct manipulation can achieve is the out-of-plane rotation of a particle. This can be thought of as the particle going from horizontal to vertical and back again. The aforementioned cell imaging studies demonstrate the usefulness of out-of-plane rotation [52, 53]. Additionally, many non-spherical particles assume a vertical orientation when picked up in a single trap and out-of-plane rotation is often needed to make them horizontal for experiments. There are a few ways to actuate traps for direct manipulation. Considered first is using a spatial light modulator (SLM), which can actuate multiple traps in three dimensions. Bingelyte et al. accomplished out-of-plane rotation on a fused bead pair using an SLM to actuate two traps in three dimensions, but had to resort to using LaGuerre-Gaussian beams to counteract the tendency of the bead pair to prefer one trap over the other [72]. Horner et al. used a similar approach to control the out-of-plane rotation of multiple E. Coli bacteria at the same time [73]. Lenton et al. studied in greater detail the process of rotating an E. Coli bacterium from the vertical to horizontal position using an SLM [74]. E. Coli rotation was observed to occur at a rate faster than the update rate of their SLM which

posed challenges in stabilizing the rotation. They found that adding an intermediate step to the rotation process in between the beam pattern for the vertical orientation and the beam pattern for the horizontal orientation stabilized the bacterium during rotation. Elsewhere, a real-time, visual-based feedback controller was used to control the out-of-plane angle of a cell, though the cell was approximated as spherical and a linearized dynamic model was used [75].

While the three dimensional actuation of an SLM is convenient, faster trap actuation for out-of-plane rotation can be achieved using trap scanning or trap time sharing, as shown later in this thesis. The downside here is that actuation along the optical axis must be done separately from the actuation in the plane. Using scanning mirrors and actuation of the objective height, Balijepalli et al. successfully rotated a nanowire out-of-plane from horizontal to vertical [76]. Carmon et al. used a scanning mirror to rotate an *E. Coli* bacterium out-of-plane [77]. The actuation of their trap was limited to the plane perpendicular to the optical axis, which relaxes the requirement of being able to quickly actuate the objective height. The trap was scanned rapidly back and forth across the *E. Coli* to rotate it. The out-of-plane angle of the bacterium could be controlled by varying the distance across which the trap was scanned. Yadav et al. tested the orientation and position stability of a bacterium when held using this approach and found it most stable when the traps were at either end of the bacterium [78]. Tanaka and Wakida extended their diatom manipulating tweezers to include out-of-plane rotation using trap scanning and a fast-focus lens for height actuation in [79].

Between the methods outlined here, including beam shaping, utilizing particle properties, trapping handles, and direct manipulation, direct manipulation offers the most flexibility as it does not depend on beam shape, particle properties beyond the refractive index, or the complications of bead handles. Consequently it is the method of choice for this thesis and will be applied in subsequent chapters.

1.5 Statement of Purpose

In the previous section concerning the manipulation of non-spherical particles, the application of dynamic control was shown to be limited to approximately spherical particles or birefringent particles for which actuation was limited to one axis. Consequently, most of the existing approaches are based on empirical observations and trial-and-error. The ability to trap and orient micron and sub-micron particles automatically, particularly using the flexible multi-trap direct manipulation method, would significantly impact biology, physics, and engineering studies at the small scale. However, as mentioned in Section 1.1, the physics of a laser interacting with a non-spherical particle do not lend themselves well to tractable dynamical models and a non-spherical particle's many degrees of freedom make empirical parametric models challenging.

Given the difficulty of realizing tractable models that govern the dynamics of a non-spherical particle, learning-based approaches offer significant potential in automating the movement and orientation of optically trapped, arbitrarily shaped particles.

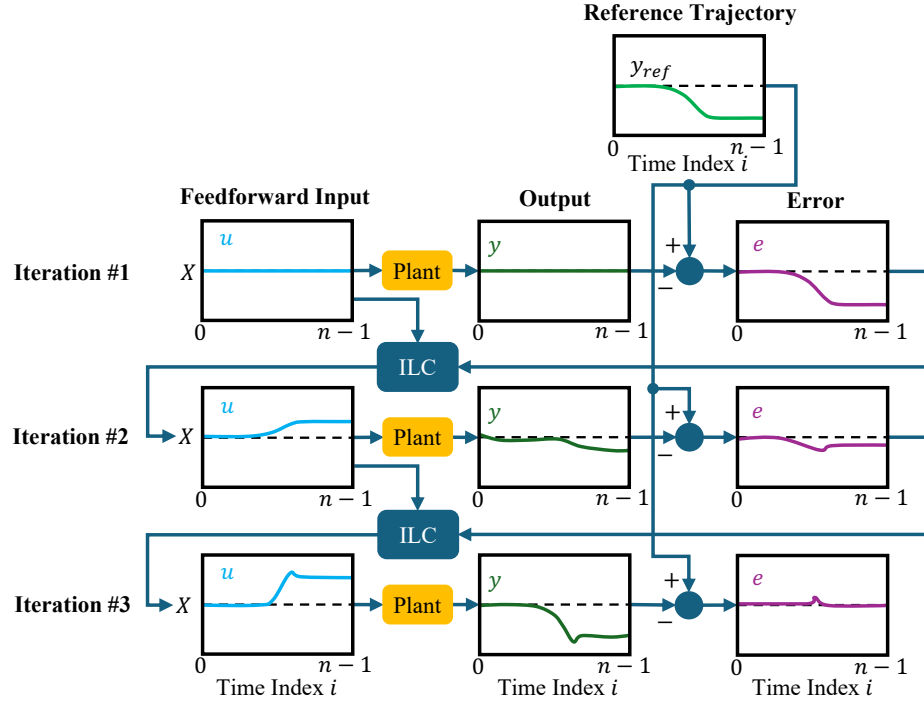


Figure 1.5: ILC Concept Diagram

Iterative learning (IL) control offers an attractive approach to achieving desired orientation control without requiring a model for controller design [80, 81]. As the name suggests, the IL controller works by performing repeat iterations of a control task. The IL control concept is illustrated for a general control case in Figure 1.5. In each iteration, the system represented by the “Plant” block starts in the same initial state. Then a time-series feedforward input $u \in \mathbb{R}^n$ of length n is fed to the plant and the resulting time-series response $y \in \mathbb{R}^n$ is recorded. Every iteration is the same length of time. After an iteration completes, the time-series error data $e \in \mathbb{R}^n$ is calculated as the difference between the system’s actual time-series response y and a time-series reference trajectory $y_{ref} \in \mathbb{R}^n$. The reference trajectory y_{ref} represents the desired system response and is the same for every iteration. In between iterations, the IL controller uses the error e and input u from the last iteration to modify the feedforward input u for the next iteration via pre-defined control law. For the first iteration, the feedforward input u is

the null signal (i.e. equal to zero for all time-instances). Assuming the IL controller is well-designed, over many iterations the feedforward input will be closer to an input that produces a system response matching the reference trajectory. As the feedforward input updates, the error between the system output and the reference trajectory will be made smaller, until it reaches zero. This indicates that the IL controller has learned a feedforward input that completes the desired control task.

The purpose of this thesis is to automate the orientation control of optically trapped non-spherical particles through the application of iterative learning control. The chosen method for optical manipulation is the direct, multi-trap manipulation method. As an instantiating example, the out-of-plane rotation of a micron-scale cylinder is chosen as the control goal. Illustrated in Figure 1.6, this entails rotating a cylinder from a stable, vertical orientation to horizontal orientation aligned with either the X or Y axis via actuating two optical traps along learned trajectories that are confined to the perpendicular XY -plane.

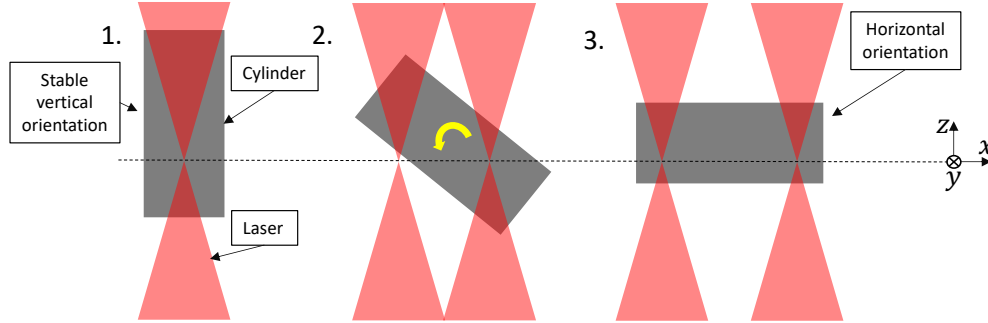


Figure 1.6: Iterative Learning Control Goal: Out-of-Plane Cylinder Rotation

The automated learning of out-of-plane rotation will be pursued first in simulation in Chapter 2 where the concept is shown viable. Next the control design is tested in experiments on actual cylinders. The optical tweezers built for the purpose of these experiments is detailed in Chapter 3. The application of this optical tweezers system to the IL control of cylinders is then recounted in Chapter 4 with results showing the success of the concept. Finally, conclusions are drawn and future directions are discussed.

Chapter 2

ILC Simulation

Toward the learning objective, it is possible to perform the learning iterations via simulation of the particle’s dynamics. The problem of deriving a suitable model for the force and torque on the particle is circumvented by using at each simulation time step an open source software [82, 17] to solve for the force and torque on the particle given its position and orientation. The *in-silico* demonstration presented here provides a first study on the effectiveness of data-driven, learning-based methods for determining manipulation control strategies for non-spherical particles. This work is organized as follows. Section 2.1 outlines the Langevin dynamics model for a non-spherical particle and its discretization for simulation. The IL controller and a closed-loop controller for simulation are developed in Section 2.2. In Section 2.3, the parameters and dimensions of the cylinder trapping problem are defined and simulation results demonstrating the successful out-of-plane rotation of the cylinder are presented.

2.1 Simulation Framework

2.1.1 Particle Dynamics

In this section, a model for a non-spherical particle describing its dynamics under opto-mechanical forces caused by an incident laser is provided. It is assumed throughout that the particle’s translational and rotational motion is uncoupled. The configuration of the particle is determined by the position of its center of mass and its orientation. To track these variables two coordinate systems are considered: an inertial lab coordinate system and a particle coordinate system that is fixed to the particle, both pictured in Figure 2.1. The axes of the lab coordinates are labeled X , Y , and Z . The laser beam is coincident with the Z axis and is focused by an objective towards the positive Z direction. The particle coordinate system axes are defined using unit vectors $\hat{x}$,

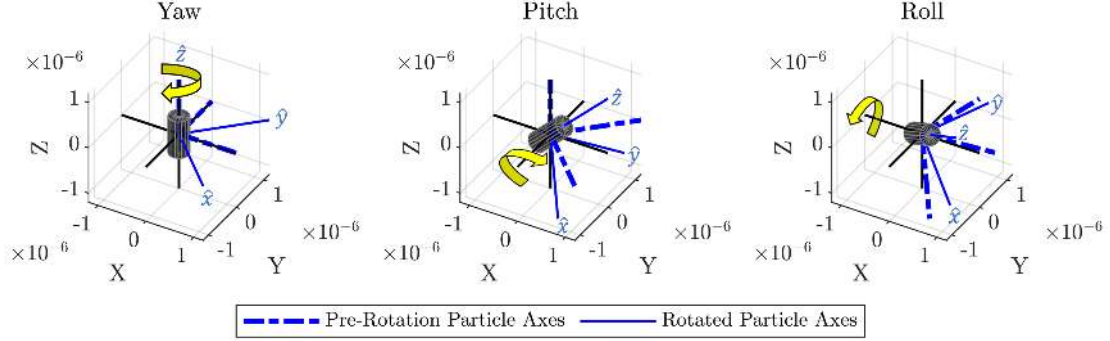


Figure 2.1: Rotations of the particle and its coordinate system.

$\hat{y}$, and $\hat{z} \in \mathbb{R}^3$, which are expressed in terms of laboratory coordinates. The orientation of the particle is tracked using the yaw, pitch, and roll of the particle coordinate system relative to the lab coordinate system (depicted in Figure 2.1). Yaw is a rotation about the Z -axis, pitch is a subsequent rotation about the Y -axis, followed by roll which is a rotation about the X -axis. These three angles are represented by the vector $\theta \in \mathbb{R}^3$, while the position of the center-of-mass is represented by vector $x \in \mathbb{R}^3$.

As mentioned in Chapter 1, dynamic variables x and θ exhibit Langevin dynamics [3]. Here the description of the Langevin dynamics is extended to include the particle's rotational dynamics. Since the time scale of the optical trap's restoring force is much greater than the relaxation time of the particle's momentum, we will assume over-damped Langevin dynamics which are given by (2.1) and (2.2) below:

$$\frac{d}{dt}x(t) = \gamma_x^{-1}F(x(t), \theta(t), \beta(t)) + \sqrt{2D_x}W_x(t), \quad (2.1)$$

$$\frac{d}{dt}\theta(t) = \gamma_\theta^{-1}T(x(t), \theta(t), \beta(t)) + \sqrt{2D_\theta}W_\theta(t). \quad (2.2)$$

Here $W_x(t) \in \mathbb{R}^3$, $W_x(t) \sim N(0, I^{3 \times 3})$, $W_\theta(t) \in \mathbb{R}^3$, and $W_\theta(t) \sim N(0, I^{3 \times 3})$. T is the temperature and k_B is Boltzmann's constant. The matrices $\gamma_x \in \mathbb{R}^{3 \times 3}$ and $\gamma_\theta \in \mathbb{R}^{3 \times 3}$ are the translation and rotation drag coefficient matrices, respectively, which relate the velocity of the particle to the frictional force from the surrounding fluid. Since the particle considered here is a cylinder and therefore axi-symmetric, the drag coefficient matrices are diagonal with non-zero entries along the diagonal [83] and are therefore invertible. Matrices $D_x \in \mathbb{R}^{3 \times 3}$ and $D_\theta \in \mathbb{R}^{3 \times 3}$ are the translation and rotation diffusion matrices of the particle, respectively. These two diffusion matrices quantify the variance of the Brownian motion affecting the particle. They are related to

the particle's drag coefficient matrices by the Einstein relation, which is $D_x = k_B T \gamma_x^{-1}$ for translation and $D_\theta = k_B T \gamma_\theta^{-1}$ for rotation. Values for both the friction coefficient matrices and the diffusion matrices for a cylinder are taken from literature [83]. The function $F(x(t), \theta(t), \beta(t))$ is the force the laser beam exerts on the particle. This force is orientation-dependent due to the particle being non-spherical and is dependent on the location, $\beta(t) \in \mathbb{R}^3$, of the beam. The same is true for $T(x(t), \theta(t), \beta(t))$, which is the torque the beam exerts on the particle.

The particle coordinate system is determined using the diffusion and drag coefficient matrices. Typically the diffusion matrices D_x and D_θ are reported in a particle coordinate system whose origin is the particle's center-of-diffusion. Similarly, the drag coefficient matrices γ_x and γ_θ are defined in a particle coordinate system whose origin is the center-of-reaction (see [83] for definitions of these two centers). Since a cylinder is axisymmetric, both its center-of-diffusion and center-of-reaction coincide with its center-of-mass [83]. Consequently the particle coordinate systems for both the diffusion and drag coefficient matrices are the same. For convenience we take the particle coordinate system $[\hat{x} \ \hat{y} \ \hat{z}]$ (with which we define the orientation of the particle) to coincide with the particle coordinate system of the diffusion and friction coefficient matrices. This results in the axis $\hat{z}$ being coincident with the cylinder's axis of symmetry and the origin coincident with the center of the cylinder, as depicted in Figure 2.1.

2.1.2 Numerical Simulation

For numerical simulation the dynamics in (2.1) and (2.2) are discretized and stepped through time following the procedure outlined in [3]. The discrete-time dynamics of x are given by:

$$x(i+1) = x(i) + \Delta t \gamma_x^{-1} F(x(i), \theta(i), \beta(i)) + \sqrt{2D_x \Delta t} W_x(i), \quad (2.3)$$

where $W_x(i) \in \mathbb{R}^3$ and $W_x(i) \sim N(0, I^{3 \times 3})$. Since D_x is given in particle coordinates the stochastic increment $\sqrt{2D_x \Delta t} W_x(i)$ is determined in particle coordinates then transformed to laboratory coordinates to update x . Transforming from particle to laboratory coordinates is done using the coordinate transformation matrix $\Sigma_{p-l}(i) = [\hat{x}(i) \ \hat{y}(i) \ \hat{z}(i)] \in \mathbb{R}^{3 \times 3}$. Similarly, since γ_x is reported in particle coordinates, we represent the force in particle coordinates using $\Sigma_{p-l}^{-1}(i)$, then scale it by $\Delta t \gamma_x^{-1}$ and transform the result to laboratory coordinates.

To update the orientation as done in [3], we first determine in particle coordinates the increments in yaw, pitch, and roll due to the stochastic term and the torque applied to the cylinder as follows:

$$\Delta \theta_p = \Delta t \gamma_\theta^{-1} \Sigma_{p-l}^{-1}(i) T(x(i), \theta(i), \beta(i)) + \sqrt{2D_\theta \Delta t} W_\theta(i), \quad (2.4)$$

where the subscript p indicates particle coordinates, $W_\theta(i) \in \mathbb{R}^3$, $W_\theta(i) \sim N(0, I^{3 \times 3})$, and $\Sigma_{p-l}^{-1}(i)$ is the aforementioned transformation matrix. We take the rotational perturbation $\Delta\theta_p$ and apply it to the unit vectors of the particle coordinate system expressed in particle coordinates. When expressed in particle coordinates these unit vectors before perturbation are $[\hat{x}_p \ \hat{y}_p \ \hat{z}_p] = I^{3 \times 3}$. The rotated particle coordinate axes are:

$$[\hat{x}'_p \ \hat{y}'_p \ \hat{z}'_p] = R_x(\Delta\theta_{p,x})R_y(\Delta\theta_{p,y})R_z(\Delta\theta_{p,z})I^{3 \times 3}, \quad (2.5)$$

where the superscript $(\cdot)'$ indicates the updated unit vectors and $R_x(\Delta\theta_{p,x})$, $R_y(\Delta\theta_{p,y})$, $R_z(\Delta\theta_{p,z}) \in \mathbb{R}^{3 \times 3}$ are the standard rotation matrices. These rotated particle coordinate axes are then transformed to lab coordinates, which also updates the coordinate transformation matrix as follows:

$$\Sigma_{p-l}(i+1) = \Sigma_{p-l}(i)[\hat{x}'_p \ \hat{y}'_p \ \hat{z}'_p]. \quad (2.6)$$

The force and torque terms in (2.3) and (2.4) are determined at each time step using the Optical Tweezer Toolbox's "forcetorque" function [82, 17]. This function estimates the force and torque on the particle by using the T-matrix method to efficiently solve the electromagnetic (EM) scattering problem. As mentioned in Chapter 1, solving the scattering problem amounts to taking an incident wave and finding the scattered wave created by the particle. The T-matrix method requires the EM scattering problem to be solved once to find the matrix T . Multiple methods exist for solving the scattering problem and the best choice is informed by the constraints and dimensions of the problem. We have selected the discrete dipole approximation (DDA) method based on the guidance in [16]. A built-in function of the Optical Tweezer Toolbox is used to solve for the T-matrix using the DDA method. This algorithm is implemented in MATLAB© with a time step of $\Delta t = 8 \times 10^{-7}$ s, which keeps the dynamics in the over-damped regime [3].

2.2 Control Strategy

An important objective here is to learn the motion profile of a laser's focal point that will result in the desired manipulations of a non-spherical particle. The difficulty of deriving a suitable dynamic model makes learning-based approaches desirable. The problem is instantiated by learning the motion profiles of two laser beams' focal points that result in the out-of-plane rotation of a cylinder. When a cylinder is successfully trapped in a single beam, it will assume a vertical orientation with its axis of symmetry coincident with the beam axis. The learned motion profiles for the two foci need to rotate the cylinder from a vertical stable equilibrium orientation

to a horizontal orientation where its axis of symmetry is aligned with the X -axis. The rotation needs to be achieved without moving either of the two focal points in the Z -direction as is required in some other approaches [73, 75].

It is assumed the system being simulated creates two beams by using an acousto-optic deflector to rapidly multiplex a single beam between two desired beam foci locations along the X -axis. The multiplexing rate is at a much higher bandwidth than the time constant of the particle's dynamics. Consequently, to the cylinder the multiplexed laser appears to be two individual laser beams. To avoid switching the laser between beam focal point positions faster than realistically possible, a minimum time delay between laser shifts is enforced. To accomplish rotation with the movement of the two focal point locations both a closed-loop controller and an IL controller are used, as illustrated in the block diagram of Figure 2.2.

2.2.1 Closed-Loop Controller

The purpose of the closed-loop controller is to ensure that the center of the cylinder stays at the origin of the laboratory coordinate system while it is being rotated to the desired horizontal position aligned along the X -axis. The feedback error signal employed is the difference between the center of the cylinder's X -position and the origin. The controller output is the multiplexing duty ratio δ , which is the fraction of the time the laser beam spends in one position to the time spent in the other. To start the duty ratio is $\delta = 0.5$, where the beam spends an equal amount of time at each of the two focal point positions. Reducing the time spent at one focal point position reduces the effective optical power at that location. Consequently, if the error indicates the cylinder is sliding away from the origin towards one of the two focus locations, the controller reduces the power at that focus location and increases the power at the other by adjusting the duty ratio accordingly. This is accomplished using a simple proportional controller. To avoid unrealistic duty cycles, duty cycles less than 0.25 or greater than 0.75 are set to zero or one, respectively.

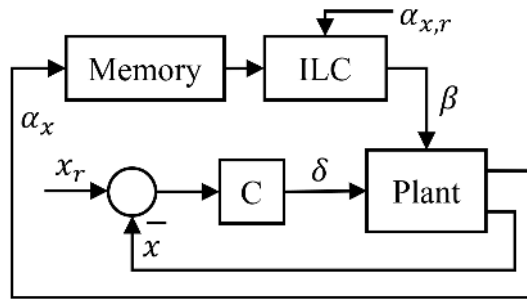


Figure 2.2: Control Diagram

2.2.2 Iterative Learning Controller

Rotating the cylinder is achieved using the motion profiles of the two beam focal points and we learn their motion profiles using an IL controller. For learning, the IL controller relies on multiple trial iterations of the same duration during which the out-of-plane rotation of the cylinder is attempted using a feedforward (i.e. pre-determined) beam motion profile. The time-series error between the desired cylinder trajectory and its actual trajectory is stored in memory during each iteration. In between iterations, the IL controller processes the error and uses it to modify the feedforward beam motion profiles for the next iteration in order to reduce the observed error. Initially the error will be relatively high, but will decrease as the IL controller learns a beam motion profile that achieves the desired rotation. To design the IL controller, the error, the beam motion profiles, and the control law that relates the two must be formally defined.

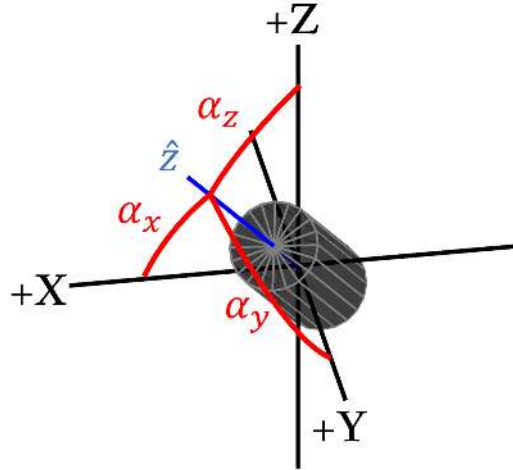


Figure 2.3: Defining the cylinder's orientation using $\hat{z}$'s direction angles relative to the X , Y , and Z axes.

As the goal of the iterative learning controller is to rotate the cylinder, we define the error it operates on using the orientation of the cylinder's $\hat{z}$ -axis, as pictured in Figure 2.3. We track the direction angle, $\alpha_x \in \mathbb{R}$, the $\hat{z}$ -axis makes with the laboratory coordinate system X -axis. In Figure 2.3 the direction angles $\alpha_y \in \mathbb{R}$ and $\alpha_z \in \mathbb{R}$ the $\hat{z}$ axis makes with the Y and Z laboratory coordinate axes, respectively, are also pictured. The angle α_x at time instant k in iteration j is labeled as $\alpha_x^{(j)}(k)$. The chosen reference trajectory, denoted $\alpha_{r,x}(k) \in \mathbb{R}$ for time instant k , for angle α_x is pictured in Figure 2.4. In this figure $\alpha_{r,x}$ goes from $\pi/2$ to 0 indicating a rotation to a horizontal orientation. The transition from $\pi/2$ to 0 in the reference trajectory is not instantaneous as this results in the IL controller producing smoother motion profiles for the

two focal points. With the reference trajectory defined we can now define the error as follows:

$$e^{(j)}(k) = \sin(\alpha_{r,x}(k)) - \sin(\alpha_x^{(j)}(k)), \quad (2.7)$$

where $k = [1, 2, \dots, n_s]$ is the index of the sampling instances of the cylinder's orientation, with n_s being the total number of samples. We take the sine of the reference angle and actual angle so there is no error penalty if the cylinder turns one direction or another to reach the horizontal configuration.

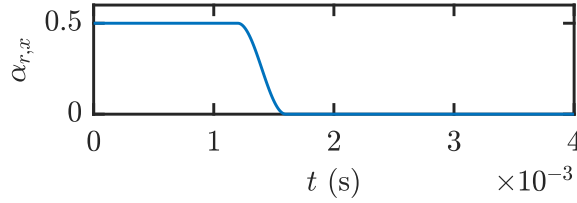


Figure 2.4: Reference Angle Trajectory $\alpha_{r,x}(t)$

With the error defined we now detail the motion profiles the IL controller will generate from it. As mentioned earlier, during each of the learning trials the rotation of the cylinder is attempted using the pre-determined, feedforward motion of the two focal points the laser beam is multiplexed between. The motion profiles for the first and second focal points are denoted $\beta_1^{(j)}(i) \in \mathbb{R}^3$ and $\beta_2^{(j)}(i) \in \mathbb{R}^3$, respectively, for time instant i in iteration j . The two foci are only moved along the X -axis of the laboratory coordinate system where only the X -components, $\beta_{1,x}^{(j)}$ and $\beta_{2,x}^{(j)}$, of the motion profiles are non-zero. Furthermore, the X -position of the second focal point is determined by $\beta_1^{(j)}(i) = -\beta_2^{(j)}(i)$ for all time instants i and iterations j . Consequently the only control variable is the X -component of the first focal point's motion profile, $\beta_{1,x}^{(j)}$, which the IL controller generates. The IL process is initialized with a motion profile that keeps the foci at the origin, i.e. $\beta_1^{(1)}(i) = \beta_2^{(1)}(i) = [0 \ 0 \ 0]^T$ for all time instances i . The IL controller generates the X -component of the first focal point's motion profile, $\beta_{1,x}^{(j+1)}$, for the next iteration $j + 1$ using the last iteration's time-series error $e^{(j)}$ and the X -component of its motion profile for the first focal point, $\beta_{1,x}^{(j)}$.

Exactly how the IL controller will learn from the time-series error $e^{(j)}$ to better determine the X -component of the first focal point's motion profile depends on the style of IL controller employed. Here a proportional-derivative (PD) IL controller is used as described in [84]. The PD-IL controller is introduced first in its most basic form, then afterwards an additional tuning parameter is added. The basic form is:

$$\beta_{1,x}^{(j+1)}(i) = \beta_{1,x}^{(j)}(i) - K_p e^{(j)}(i+1) - K_d(e^{(j)}(i+1) - e^{(j)}(i)), \quad (2.8)$$

for $i \in [1, 2, \dots, n_{dt} - 1]$, where n_{dt} is the number of time-steps. Constants $K_p \in \mathbb{R}$ and $K_d \in \mathbb{R}$ are used to tune the controller, which is typically done by trial-and-error. Note that the j^{th} -trial error at the next time-step $e^{(j)}(i+1)$ impacts the motion profile for the $(j+1)^{th}$ -trial at time i through the proportional term, K_p , and the derivative term, K_d . The resulting action is non-causal.

The ability of the PD-IL controller to preview the error with a longer time horizon is possible using an additional $n_{adv} \in \mathbb{Z}^+$ forward time-steps:

$$\beta_{1,x}^{(j+1)}(i) = \beta_{1,x}^{(j)}(i) - K_p e^{(j)}(i+1+n_{adv}) - K_d(e^{(j)}(i+1+n_{adv}) - e^{(j)}(i+n_{adv})), \quad (2.9)$$

where $i \in [1, 2, \dots, n_{dt} - n_{adv}]$. For time indices $n_{dt} - n_{adv} < i < n_{dt} - 1$, the PD-IL controller reverts to the form in (2.8). After trial-and-error tuning to determine K_p , K_d , and n_{adv} , it was found that a proportional-only IL controller ($K_d = 0$) with $n_{adv} = 300$ works best for this application.

2.2.3 Controller Sample Rate

As mentioned in the preceding closed-loop and iterative learning controller designs, we have accounted for a finite measurement sample rate of the cylinder's position and orientation. For obtaining the cylinder's position and orientation, it is assumed the simulated system has available to it only information that could be obtained from image processing. Image based tracking has precedents in literature [71, 77, 85]. To achieve rotation and position stabilization in the time-scales considered here, a high-speed camera would be used as often done in microfluidics [86]. High-speed cameras are available with frame rates exceeding 100 *kHz*. For these simulations we assume a camera and image processing system can provide a position and orientation sample rate of 10 *kHz*. The angle orientation of the cylinder can be estimated offline since it is only needed in between iterations, while the position can be estimated online in between image samples using threshold pixel value detection. In the results section we discuss the trade-offs of using lower sample rates.

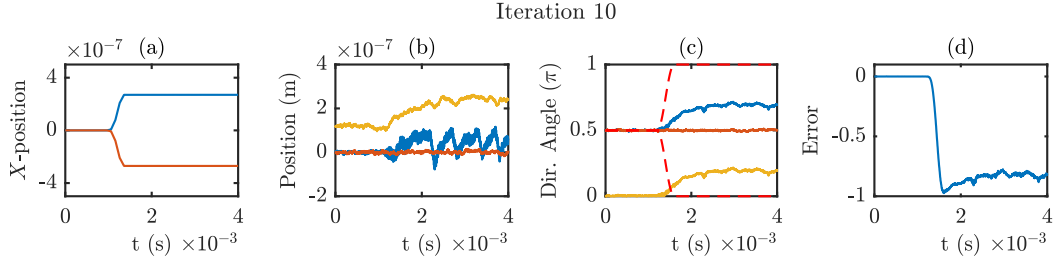
2.3 Results

A series of simulations are performed to confirm that the IL controller learns a motion profile for the two focal points that, together with the closed-loop system, successfully rotates the

cylinder from a vertically oriented equilibrium position to a horizontally oriented position. The learning iterations are performed and their results examined. The resulting two focal point motion profiles are analyzed to provide a rationale for its features from a physical viewpoint. To underscore the utility of this learned motion profile, the IL controller is then replaced with a feedforward controller that executes the final motion profile learned by the IL controller and the ensuing results are assessed.

The simulated optical tweezers system considered uses a 1.2 numerical aperture objective lens with a vertically polarized, 1064 nm, continuous-wave, Gaussian-profile laser beam operating at 50 mW. Temperature in the sample chamber is assumed to be 296 K with a medium whose viscosity is $9.359 \times 10^{-4} \text{ N} \cdot \text{s}/\text{m}$ and refractive index is $\eta_m = 1.33$. The cylinder is 800 nm in length and 400 nm in diameter with a refractive index of $\eta_c = 1.59$. As mentioned in Section 2.1 this cylinder follows the dynamics in (2.1) and (2.2), which are presented in discretized form in (2.3) and (2.4). The Optical Tweezer Toolbox [82, 17] provides the force and torque on the particle.

First the simulation setup for the IL process is detailed. Each iteration has a simulated time duration of 4 ms, during which the learned motion profiles for the two focal points are applied. In each iteration the cylinder is allowed 1 ms to reach an equilibrium before the IL controller action is implemented. For the first iteration both foci are kept at the origin for the entire duration, i.e. $\beta_1^{(1)}(i) = \beta_2^{(1)}(i) = [0 \ 0 \ 0]^T$ for all time instants i . In between the following iterations the IL controller updates the two motion profiles using the error defined in (2.7) of the previous run. Iterations are ceased when the desired performance is achieved.



Results for the first 20 iterations of this learning process are used to evaluate its performance, with the results of the 10th and 20th iterations shown in Figure 2.5. The IL controller takes the first eight or nine iterations to learn a profile that perturbs the cylinder and in iteration 10 these perturbations are noticeable. For both iterations in Figure 2.5 there are four plots arranged horizontally. Plots (a) and (e) contain the X-position of both focal points as a function of time. These motion profiles are the feedforward inputs that the IL controller generates. The focal point motion profiles for iteration 10 and 20 are the outputs of the IL controller after iterations

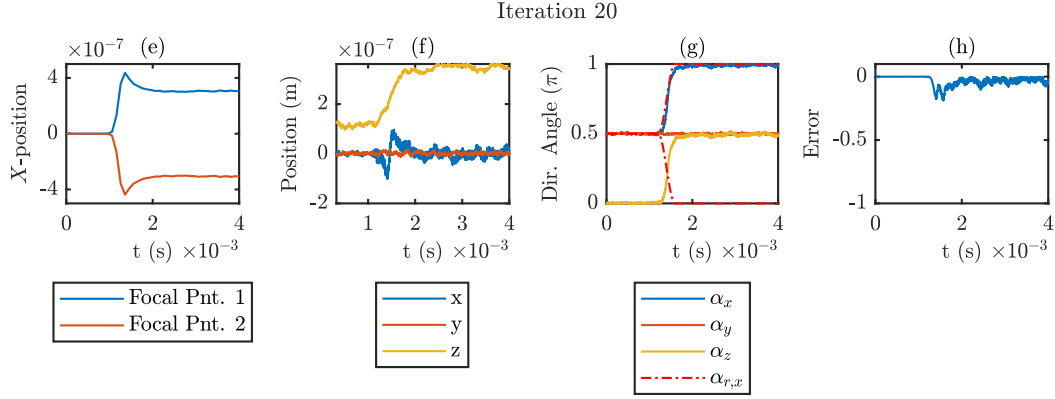


Figure 2.5: Results for iterations 10 and 20 of the IL process. For each iteration there are four plots arranged horizontally; the leftmost plots show the motion profiles of the two beams' focal points used to generate the results in the three plots to the right.

9 and 19, respectively. In iteration 10 the cylinder is subjected to the motion profile in plot (a), resulting in the dynamics plotted in (b)-(d). Similarly, for iteration 20 the motion profile in plot (e) results in the dynamics plotted in (f)-(h). The plots (b) and (f) are the displacements of the center of the cylinder as a function of time. As apparent, the X and Y coordinates oscillate around the origin, as maintained by the closed-loop controller. In the 20th iteration, the Z -position of the center of the cylinder starts at around $Z = 1 \times 10^{-7} \text{ m}$, its equilibrium position when trapped vertically, and moves to a new equilibrium around $Z = 3.5 \times 10^{-7} \text{ m}$.

The plots (c) and (g) show the direction angles $[\alpha_x \ \alpha_y \ \alpha_z]$ the cylinder's axis of symmetry makes with the laboratory XYZ coordinate axes as a function of time. The reference angle trajectory $\alpha_{r,x}$ is also plotted, along with its image mirrored across the line at $\pi/2$; both are plotted as the error defined in (2.7) allows angle α_x to go from $\pi/2$ to π or from $\pi/2$ to 0 as both paths indicate the cylinder reaching the horizontal position. The plots (d) and (h) depict the resulting error, defined by (2.7), as a function of time. The direction angle plot for iteration 10 shows that the cylinder begins to rotate, but stops at an intermediate angle instead of reaching the fully horizontal position. The error plot to the right shows a considerable error for 10th iteration. In iteration 20 we see that α_x closely follows the reference angle trajectory and α_z moves from 0 to $\pi/2$, corroborating a successful rotation to the horizontal position. Meanwhile angle α_y remains at $\pi/2$ indicating the cylinder has rotated to a horizontal position aligned with the X -axis as desired. The error for iteration 20 is small in magnitude with some fluctuations around zero-error while in the horizontal position. This is due to the horizontally trapped position being slightly less stable than the vertically trapped position and the efforts of

the closed-loop controller in keeping the cylinder centered at the origin. The learning process depicted in Figure 2.5 was repeated ten more times and all of these repeat learning processes learned a beam profile effectively identical to the one of iteration 20 in Figure 2.5 and achieved rotation with similar low error.

Intuition as to why the two focal point motion profiles of iteration 20 result in the successful rotation of the cylinder can be gleaned from the iteration's results in Figure 2.5. The two focal points begin to diverge from each other on the X -axis at approximately the 1 ms mark, and as seen in the direction angle plot, the rotation of the cylinder follows soon after. The small duration of focal point actuation before rotation happens is necessary to achieve rotation and the actuation begins before the typical point in time the error used by the IL controller becomes nonzero. A causal IL controller policy would not be effective in determining the necessary pre-actuation from the error. Here $n_{adv} = 300$ is used, as the added error preview allows the IL controller to learn the pre-actuation needed to match the reference angle trajectory. After this small duration of pre-actuation, the diverging focal point positions cause the cylinder to begin tilting at which point one focal point will be closer in X -coordinate to the top and the other will be closer in the X -coordinate to the bottom. This creates opposing influence on the cylinder's ends; one focal point attracts the top of the cylinder, the other the bottom. The focal points continue to diverge until they are just past $\pm 4 \times 10^{-7}\text{ m}$, slightly exceeding the length of the cylinder, which given how the top and bottom of the cylinder are attracted to the focal points, goads the cylinder into a horizontal orientation. The focal points then quickly return to approximately $\pm 3 \times 10^{-7}\text{ m}$, which maintains the cylinder in its now horizontal orientation.

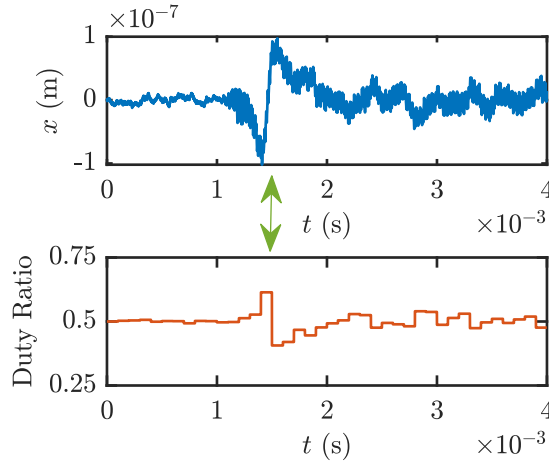


Figure 2.6: X -position vs. beam multiplexing duty ratio for iteration 20.

Meanwhile the closed-loop controller works to maintain the center of the cylinder at the

origin. Figure 2.6 shows again in larger detail the X -position of the cylinder during iteration 20. The bottom plot in Figure 2.6 shows the beam multiplexing duty ratio δ as a function of time. A duty ratio greater than 0.5 increases the amount of time the multiplexed laser beam spends at the focal point location on the positive X -axis, while a duty ratio less than 0.5 increases its time at the focal point location on the negative X -axis. Figure 2.6 demonstrates that the feedback controller acts to maintain the X -position near zero; close to 1.5 ms it is evident that the controller alters the duty ratio to first counter the negative excursion of the X -position, leading to a positive overshoot of the X -position. Then duty ratio is eventually brought back to 0.5; the time constant is seen to be close to 2 ms .

Now that the two focal point motion profiles have been learned, their ability to function as a feedforward input in the absence of the IL controller is tested. Specifically, in the control system shown in Figure 2.2 the “Memory” and “ILC” blocks are removed with the closed-loop controller remaining. The motion profiles of the two focal points used to produce the results of iteration 20 are applied to the closed-loop system as a feedforward input. This exercise is repeated 50 times to ensure the performance is robust. The time-series results from each run are superimposed over one another in the same plots and included here as Figure 2.7. Each of the position and orientation variables in this figure are plotted separately so it can be seen there are no outliers. In the top three plots, the X , Y , and Z -positions all exhibit identical behavior seen previously for iteration 20 in Figure 2.5. The same holds for the direction angles. Angle α_x is seen to diverge from $\pi/2$ to either π or 0, indicating that each run is reaching the horizontal orientation even if they may rotate different directions to do so. The α_z angle corroborates this, and angle α_y is equal to $\pi/2$ at all times indicating the horizontal orientation is aligned with the X -axis as desired. The exercise of applying the learned beam focal point motion profiles as a feed-forward input 50 times was repeated using the learned motion profiles of 10 other repetitions of the learning process, with results for all identical to those pictured in Figure 2.7.

All results for this study were generated using a controller sample rate of 10 kHz as mentioned in Section 2.2. In simulation it was seen that rotation can be achieved with sample rates as low as 5 kHz , but at the price of increased noise. With the lower sample rate there are more fluctuations in the position and orientation of the cylinder when in the horizontally trapped position. A sample rate of 10 kHz offered the best balance of performance and practicality.

2.4 Simulation Conclusions

The results of this work demonstrate via simulation the successful and repeatable out-of-plane rotation of a sub-micron cylinder via an automated iterative learning control process. Both the iterative learning and closed-loop controllers used do not require a model of the particle’s

dynamics for design. This circumvents the aforementioned issue of finding a dynamic model of the particle-laser beam interactions suitable for real-time use, which has restricted previous manipulation strategies to empirical methods on restricted particle shapes. The work here serves as a first proof-of-concept for the broader goal of automating the micro-manipulation of any arbitrarily shaped particle. The next step is to experimentally verify the concept of this IL application. This is carried out in the following chapters, where the construction of an optical tweezers system capable of these experiments is detailed and the experimental results are provided.

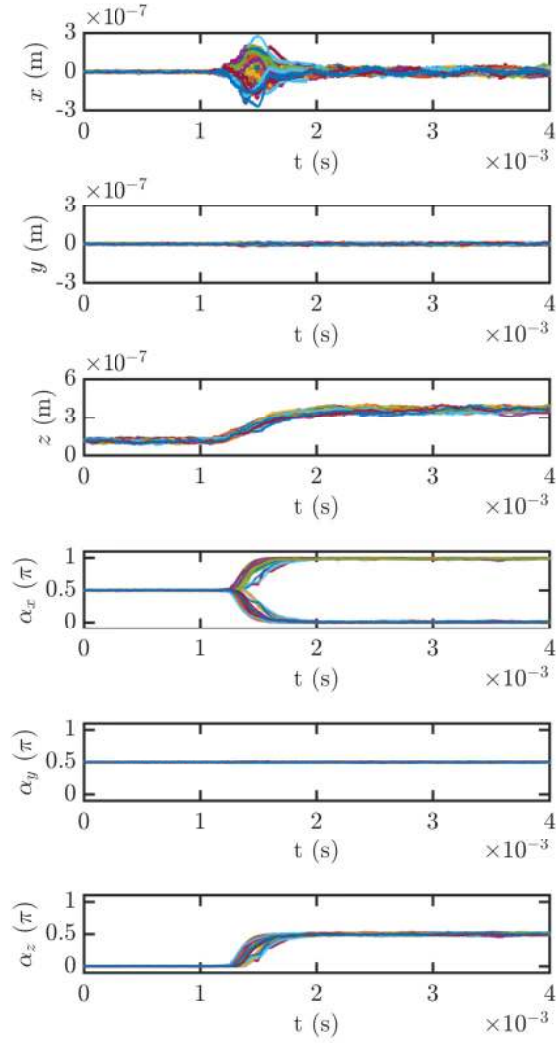


Figure 2.7: 50 runs using ILC generated feedforward input.

Chapter 3

Instrumentation

The simulation work of the previous chapter established the viability of the controller concept. The next step is experimental verification and for this purpose an optical tweezers system was built. The construction of the optical tweezers system is described in this chapter. This includes the optical system design, system electronics, and programming needed to implement the IL controller concept. The calibration of the system is also summarized. The system architecture of this optical tweezers system follows that of Visscher et al. [40] which is used extensively in OT control applications [44, 45, 46, 47]. The architecture includes trap position actuation in the plane perpendicular to the optical axis via an AOD and position detection using a quadrant photodiode.

3.1 Optical System Design

The optical tweezers system is built into a Nikon TE2000-U microscope placed on a vibration isolation optical table. The overall system diagram is presented in Figure 3.1. Two lasers are used, one for detecting the trapped particle's position and one for trapping the particle. The trapping laser is a horizontally linearly polarized, 1064 *nm* wavelength, continuous-wave diode laser with an adjustable power output ranging from 0 to 4 *W* (LaserQuantum Opus 1064). The output power at the laser head is set to 300 *mW* using the power setting on the laser's power supply unit (PSU). This output power was found sufficient for trapping heavier, non-spherical particles. This laser power is put in context in Section 3.4 where the associated trapping stiffness for a spherical bead is estimated. The trapping laser is steered via an Intraaction DTD-274HD6 two-axis AOD. After the AOD, a pair of lenses (l_1 , l_2) are used to both expand the beam and image the exit aperture of the AOD to the back focal plane of a Nikon Apo-TIRF 60x, 1.49 numerical aperture (NA) oil-immersion objective. The beam is expanded enough to slightly

overfill the back aperture of the objective, which is important for forming a strong optical trap [3]. Imaging the exit aperture of the AOD to the back focal plane of the objective ensures that the angular displacement created by the AOD becomes translational displacement in the sample chamber.

The detection laser is an unpolarized, 830 nm wavelength, continuous-wave diode laser with an adjustable power output ranging from 0 to 60 mW (LaserGlow D830601FT). For experiments, the power output at the the laser head is set to 35 mW using the power control on the laser's PSU. This detection laser is passed through a vertical linear polarizing filter (*pol*) and a neutral density filter (*ndf*). These attenuate the detection laser to an optical power too low to trap a particle, ensuring it does not interfere with trapping. The detection laser also passes through the lens pair l_3 and l_4 which expand the beam and also shift its focal point lower in the sample chamber to further ensure it does not interfere with trapping. The detection laser is made co-linear with the trapping laser via a polarizing beam splitter (*pbs*).

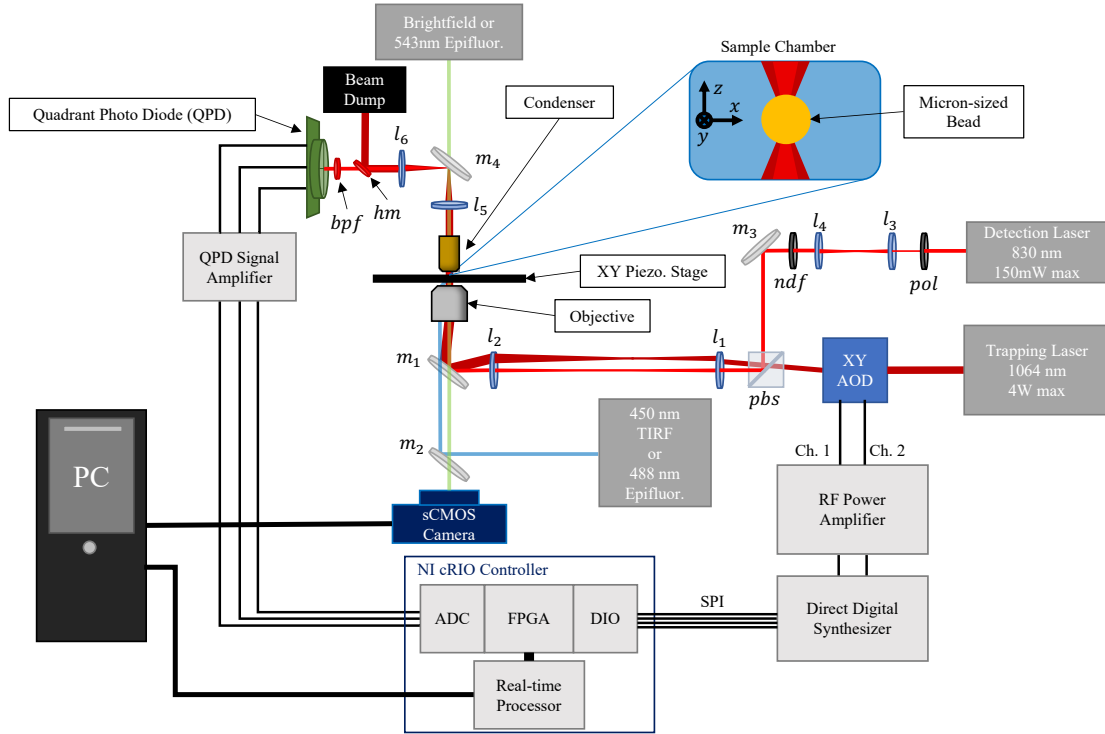


Figure 3.1: System Diagram

3.1.1 Detection System

After passing through the sample chamber, both the detection and trapping lasers are collected by a Nikon Plan Fluor 100 $\times$, 1.30 NA, oil-immersion condenser. An oil-immersion condenser was chosen as the higher NA results in greater sensitivity to changes in the trapped particle's position [87]. Two lenses (l_5 , l_6) are used to image the back focal plane of this condenser to a QPD (First Sensor, QP50-6-42u SD2). Before the trapping laser can reach the QPD, it is redirected into a beam dump via a 1064 nm hot mirror (hm). For further filtering, a 830 nm bandpass filter (bpf) is placed directly in front of the QPD sensor area.

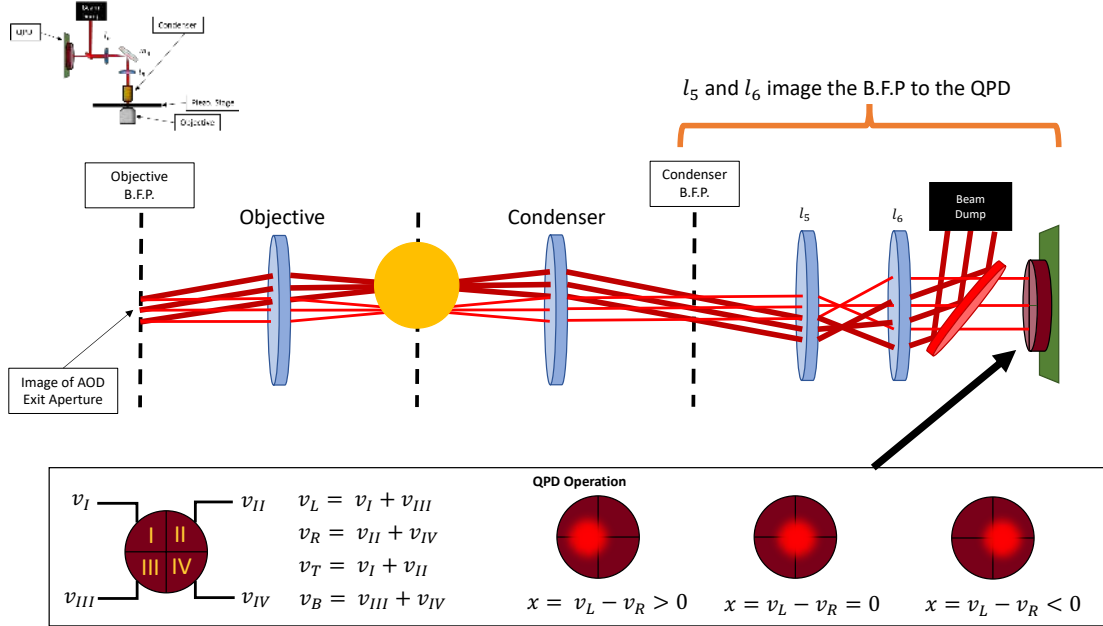


Figure 3.2: QPD Detection System Concept

The concept behind the QPD detection system is illustrated in Figure 3.2. For the purposes of illustration the trapped particle is chosen to be a sphere and the objective and condenser are approximated as thin lenses. Ray traces are shown for both the trapping laser (dark red) and detection laser (bright red). The angular displacement of the trapping beam at the AOD exit aperture is imaged to the back focal plane of the objective and this angle is shown in the diagram. Since the trapping beam passes through the focal point of the objective lens, its angular displacement is converted to a translation in the sample chamber. The detection laser is always passing straight on through the objective's back focal plane, serving as the immovable origin of the experimental coordinate system. When the bead interacts with the detection laser

light, it acts as a lens and bends the light. This creates an interference pattern in the light that is correlated to the bead's position. Lenses l_5 and l_6 image the interference pattern as it appears at the back focal plane of the condenser to the QPD. The back focal plane of the condenser is chosen as the image source as the image will remain at that location even if the objective or condenser positions are disturbed. The boxed sub-diagram in Figure 3.2 shows an example of what this interference pattern looks like for a spherical bead when imaged onto a QPD. For this example, X displacements of the bead are considered but the same operating principles apply to Y displacements. The bright red dot represents an area of higher light intensity that indicates the bead position. The QPD's four quadrants, $v_I, v_{II}, v_{III}, v_{IV}$, each output a voltage that increases with greater incident light power. For a negative X displacement, more light will be incident on the left half of the QPD ($v_L = v_I + v_{III}$) than the right half ($v_R = v_{II} + v_{IV}$). The opposite applies to positive X displacements. Therefore the X position is equal to $\alpha_x(v_R - v_L)$ and the Y position is equal to $\alpha_y(v_T - v_B)$, where α_x and α_y are proportionality constants determined during calibration (see Section 3.4).

3.1.2 Microscope Equipment

Additional hardware includes two illumination sources, a total internal reflection (TIRF) source, an sCMOS camera, and a motorized stage. The motorized stage (Prior Scientific Proscan III H117) is used to move a particle within range of the trap before an experiment commences. During experiments the stage is stationary. Illumination is provided by a brightfield illumination source (ThorLabs MCWHL7 LED paired with the SM1U25A collimation optic) collected by the condenser. Additionally, there is a 488 nm epifluorescence source used during system calibration (ThorLabs M470L5 LED and SM1U25A collimation optic mounted to a Nikon TIRF2 arm). A 450 nm TIRF system is also present, though not used for this project. A PCO Edge 4.2 sCMOS camera is used for imaging during experiments. The camera, motorized stage, and TIRF system were purchased and installed by alumni of the lab group (see Acknowledgments) before the construction of the OT system described here began.

3.2 Control Hardware

Multiple pieces of electronic hardware are needed to run the detection, controller, and trap actuation on the optical tweezers. As shown in Figure 3.1, this includes an amplifier for the QPD signals, an NI cRIO controller, a radio-frequency (RF) driver to synthesize RF signals, and an RF amplifier to amplify the signal sent to the AOD. Getting these systems to work together involves many design choices which are discussed here.

3.2.1 NI cRIO Controller

For logging the QPD signal, running the controller calculations, and actuating the AOD an NI cRIO 9039 controller is used. The NI cRIO-9039 controller combines an Intel Atom E3845 processor running the NI Linux Real-Time OS with a Xilinx Kintex-7 K325T field programmable gate array (FPGA). Both the real-time processor and the FPGA are programmable using LabVIEW software deployed on a desktop PC. Once the program is deployed on the cRIO it can run independently from the desktop PC. The input and output connections of the cRIO are configurable by removable modules that interface with the cRIO's FPGA. The QPD signal is collected via an NI 9223 analog-to-digital (ADC) converter module. The control signal sent to the RF driver is handled by a NI 9401 digital input/output (DIO) module.

3.2.2 AOD Electronics

To explain the electronics needed to run the AOD it is helpful to first briefly review the AOD's operating principles. The AOD consists of two crystals, one for each axis of actuation (X and Y). For each axis, an RF signal of variable frequency and amplitude is sent to the AOD, where it is converted to an acoustic signal that is sent through the crystal. This creates a diffraction grating in the crystal which deflects the laser beam at an angle that is a function of the driving RF frequency. When the driving RF frequency is changed, the diffraction grating spacing changes, and with it the deflection angle of the laser beam. As mentioned in Section 3.1, this deflection angle is converted to translational trap displacement in the sample chamber. By varying the driving RF frequency, the deflection angle of the laser beam changes, and with it, the position of the trap in the sample chamber. By varying the amplitude of the RF signal, the laser power transmitted out of the AOD can be modulated. The input RF frequency range for the AOD used here is 18-36 MHz on both the X and Y channels, with 27 MHz being the center frequency at which the beam passes straight through the AOD. A hertz per nanometers calibration constant is used to convert a desired X, Y position given in nanometers to the corresponding AOD driver frequencies that will achieve it. The determination of this calibration constant is discussed in Section 3.4. Each of the AOD's two axes are controlled by independent RF signals, so X and Y positions of the laser beam can be controlled independently.

The job of the RF driver and the RF amplifier are to create and amplify, respectively, the two independent RF signals needed to drive the AOD. As it turns out, an RF driver setup for use in an optical tweezers system is a niche piece of hardware. The manufacturer that made the popular RF driver of choice for optical tweezers stopped manufacturing it many years ago (IntraAction Corporation). The key requirement is that the RF driver be controllable via a serial peripheral interface (SPI) connection which ruled out other commercially available

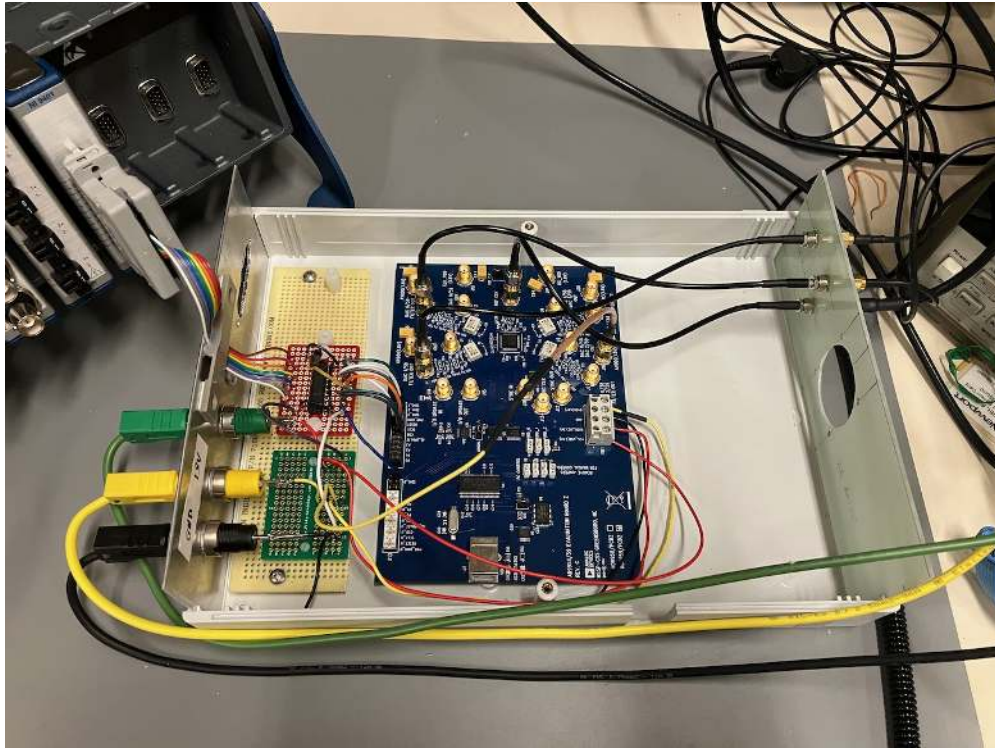


Figure 3.3: RF Driver

alternatives (For example, Isomet Corporation, who could not deliver on SPI support for their RF driver). Consequently a custom solution had to be built in-lab for this optical tweezers system. The result is pictured in Figure 3.3 where the top of the enclosure has been removed to show its contents. The core of the RF driver is an evaluation board for the AD9959 direct digital synthesizer (DDS) chip, a popular choice for RF engineering applications. This is the large blue printed circuit board in the photo. This board requires an external clock signal for its DDS function, which is provided by a Crystek Corporation CPROBS5-0100 100 MHz crystal module. This is multiplied in the AD9959 to 400 MHz using a phase lock loop (PLL). For the SPI connection, the AD9959 uses 3.3 V logic while the cRIO controller's digital outputs use 5 V logic. A Texas Instruments SN74LVC245AN Bus Transceiver is used to step down the cRIO controller's 5 V logic level output to the 3.3 V logic level input of the AD9959. This transceiver chip is located on the small red through-hole breadboard installed in between the cRIO and the AD9959 evaluation board. Across the SPI connection, configuration data is transferred as well as output settings for the DDS. This includes the desired output frequencies and amplitudes for the X and Y channels. The output frequency of each DDS channel is specified using an

unsigned, 32-bit frequency tuning word (FTW) which can be read by the AD9959. The formula for the FTW is given in Equation 3.1:

$$FTW = \frac{2^{32} f_{out}}{f_{clk}} \quad (3.1)$$

where the 2^{32} term arises from the fact that the FTW is 32 bits long, f_{out} is the desired output frequency, and f_{clk} is the clock frequency of the DDS (400 MHz). The formula for the FTW is the same for both the X and Y channels. The output amplitudes are controlled by amplitude tuning words (ATW), which are 10-bit unsigned integers. The formatting of the data sent over the SPI connection is discussed in greater detail in Section 3.3.3.

The AOD represents a 50 Ω load at each of its channels. The maximum root-mean-squared (RMS) RF power output the AD9959 evaluation board can deliver to a 50 Ω load is 0.278 mW. This power output is not enough to drive the AOD to peak diffraction efficiency (the exact amount of power needed is determined during construction calibration, as discussed in Section 3.4). An amplifier for each channel is needed and it was found that the Mini-Circuits ZHL-5W-1+ amplifier with a gain of 40 dB and maximum power output of 5 W can produce enough RF power to drive the AOD to peak efficiency.

3.2.3 QPD Amplifier

While the QPD has built current-to-voltage amplifiers, its output voltage range is small compared to the ± 10 V input range of the NI 9223 ADC module. To amplify the QPD voltage signals to better occupy this input voltage range, a custom two channel amplifier was built in-lab. This amplifier employs two Texas Instruments INA826EVM instrumentation amplifier evaluation boards. These are housed in a custom enclosure shown in Figure 3.4, where the knobs are attached to potentiostats that adjust the gain of the two instrumentation amplifiers. This allows the amplification of the QPD signals to be optimized during the detection system calibration. Note that the enclosure has three BNC connector outputs. Two are for the QPD_x and QPD_y signals, the third is a passthrough for the un-amplified QPD_{sum} signal which is equal to $QPD_x + QPD_y$ and useful in some optical tweezer applications (for example, neural net calibration [88]).

3.3 LabVIEW Programming

The software deployed on the cRIO controller started from a blank LabVIEW project and was built up from there to meet the unique needs of the iterative learning control problem. A LabVIEW project consists of a collection of Virtual Instruments (VI) that are essentially

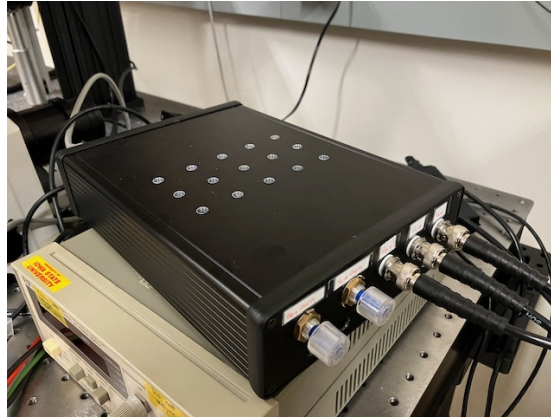


Figure 3.4: QPD Amplifier

graphical programs that upon completion are compiled to run on the cRIO hardware. VI's can be written to target the real-time processor or the FPGA processor. The compilation is handled completely by LabVIEW so that the user does not need to manually program in C or VHDL. This allows fast development and implementation of system software.

Designing a LabVIEW project for a cRIO controller requires that tasks be divided between real-time processor VI's and FPGA VI's. The real-time processor has the benefit of being easily and quickly reprogrammable. It can also run a graphical user interface (GUI) and read/write data files. The FPGA is less flexible as any change requires the FPGA bit file to be recompiled which for this project takes about ten minutes. However, its computational loop time is extremely fast with a 40 MHz clock cycle that unlike the real-time processor, adheres to strict timing. As the FPGA is the connection between the input/output modules, it handles reading and writing outputs. For an optical tweezers system, there needs to be a GUI from which the user runs the instrument, including starting and stopping data recording, setting and reading system parameters, visualizing data, and running calculations for calibration and feedforward control. These tasks are assigned to the real-time processor, as indicated in Figure 3.5. Placing the feedforward calculations in the real-time domain makes sense as these can be done offline and reduces load on the FPGA. Analog input sampling, closed-loop controller calculations, and controlling the RF DDS are all handled by the FPGA. These are all tasks that need to be done quickly with precise timing so the added difficulty of programming it on the FPGA is worthwhile.

Handling the flow of data within a LabVIEW project in real-time is a critical task. Communication between the real-time processor and the FPGA must be defined, as must the communication between separate processes within VI's. The communication between the real-time

processor and the FPGA is pictured in Figure 3.5. First in, first out (FIFO) buffers are used for the lossless transfer of time-series data. The host-to-target FIFO carries time-series feedforward control signals from the real-time processor to the FPGA. Conversely, the target-to-host FIFO carries synced QPD and trap position time-series data from the FPGA to the real-time processor. In either case, the sending side has a “while” loop for loading data into the FIFO while the receiving end has a “while” loop for unloading data. This setup is referred to as a “producer-consumer” architecture [89]. For parameters on the FPGA that need to be read and set at a slower pace, simple read/write commands can be sent from the real-time processor to the FPGA. Within a real-time processor VI, the LabVIEW “queue” blocks can be used to create FIFO buffers between separate processes. To handle simple parameters, “local variables” can be defined to allow the same variable to be accessed by different processes within the same VI. “Global variables” allow separate VI’s to read the same parameters. However, caution must be exercised when using local or global variables as access to them is not synchronized. Within an FPGA VI, a “local FIFO” can be used for inter-process communication. A local shift register is also available with an FPGA VI, which is useful when only the most recent data point is relevant and not the whole time-series history.

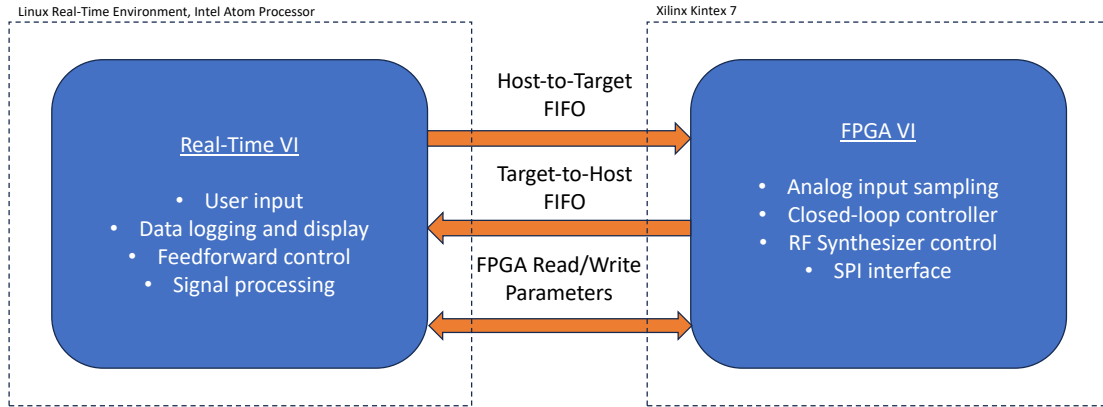


Figure 3.5: LabVIEW Real-time and FPGA Architecture Diagram

3.3.1 Real-Time VI

The real-time VI for the optical tweezers system needs to tackle a broad array of tasks. Including the entire block diagram for this VI here is not feasible due to its size. To communicate its design, the diagram in Figure 3.6 is used. First, the error handling pipeline, indicated in the diagram

with gold arrows, is used to define the sequence of the VI in time from initialization to shut-down. In the leftmost dashed-border box the VI variables, parameters, and the FIFO buffers are all initialized. This includes setting the rate for input sampling, updating the trap position, and data recording. To make the VI easier to modify and maintain, core tasks are separated into their own “while” loops. Note that this is not done to gain any sort of parallel computing advantage. In fact, in the real-time processor, these while loops will not be handled in parallel, but instead the processor will hop between them to tend to their various tasks in a linear fashion. Priority between the loops can be established by intentionally adding execution delays to some loops. Lower priority loops get more delay so they are run less often, freeing up more time for the critical loops.

At the top of the while loop stack in Figure 3.6 is the loop for trap position control. Within this while loop exists a state machine that is used to toggle between several different control modes. This includes a setpoint mode, where the trap position is set to a static point in the XY plane of the sample chamber. Another state, one used during calibration, is a feedforward control mode where the trap is commanded to follow different waveform trajectories such as square wave or triangle wave. There is also a state for executing feedforward control using an input read from a file. The execution of IL control gets its own state and its own sub-VI which creates a separate GUI control panel just for monitoring the IL process. This is discussed further in Section 3.3.2. The state machine structure makes it easy to add more control modes to the software by simply adding more states. Regardless of which control state is chosen, the feedforward trap position data is fed to the same host-to-target FIFO. Before the trap X and Y positions are loaded into the FIFO, they are converted into their corresponding FTW’s using Equation (3.1).

The second while loop from the top in Figure 3.6 is for reading the QPD signal data and trap position data from the target-to-host FIFO. Since this data transfer must be lossless, this loop gets the highest priority among the loops and is allowed to run the fastest. The QPD and trap position data come together in the same packet so that the alignment in time of the two data types is maintained. Once the data is read out of the FIFO buffer, it is fed into the data queue to be read by the data recording and signal processing loops. If any errors occur with the FIFO data transfer such as a timeout or loss of data, there are defined error handling sequences to raise an error flag, flush the buffer, and stop and then restart the FIFO.

Writing the data received from the FPGA to a file is handled by the third loop from the top. In this loop, a packet of data from the data queue is received and a “for” loop is started to write the packet of data into a TDMS (Technical Data Mangement Streaming) format file. The TDMS file format is native to LabVIEW and was chosen here for its writing speed and the availability within LabVIEW of built-in file writing, reading, and formatting features. Also,

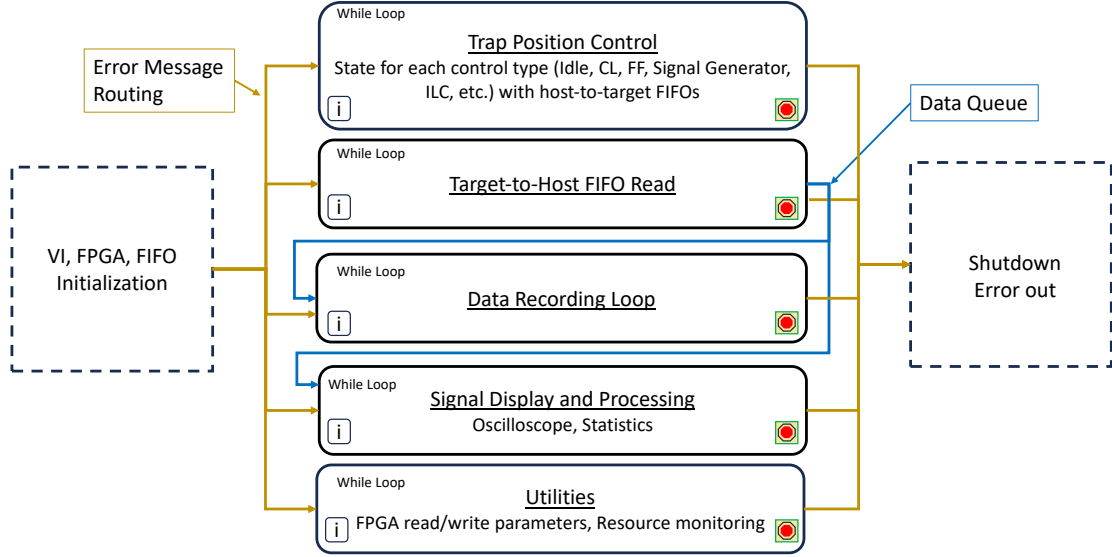


Figure 3.6: LabVIEW Real-time Software Diagram

MATLAB has had built in functions for reading TDMS files into the MATLAB workspace since version 2022a. Start and stop buttons on the real-time GUI are used to control data recording. Additionally, some of the controller functions in the top while loop have built-in functionality to start and stop recording on their own. This includes the IL controller, where having automatic recording stop and starting is necessary to ensure all trials are the same length of time. Lastly, error handling is included in the recorder design to handle any file open or write issues that may be encountered, however unlikely that may be.

The while loop for signal visualization and processing is the second from the bottom of the while loop stack. This handles the oscilloscope display on the GUI. The refresh rate of the oscilloscope display is determined by the packet size delivered from the FIFO read loop to the data queue. A refresh rate of 200 ms was found to work well for displaying the 5 Hz frequency position scans used for calibration. The oscilloscope is one state in a state machine within this while loop. Other states include functions that use the time-series data from the data queue to estimate the trap stiffness. Multiple redundant estimation methods are included to cross validate results. Each method gets its own state in the state machine. The various methods are discussed in Section 3.5.

The last while loop handles the reading and writing of FPGA VI parameters and other utility functions. Key parameters written to the FPGA include the calibration constants for the detection system and AOD actuation, feedback controller gains (for both X and Y directions),

and the power settings for the X and Y channels of the DDS. These parameters are only written to the FPGA if the user changes their values. Parameters read from the FPGA are primarily for the purposes of monitoring and debugging, including the current states of the FPGA state machines and other boolean flags indicating the status of the FPGA VI. These parameters do not need be set or updated quickly so this is the slowest of the while loops, updating at a rate of 2 Hz .

3.3.2 Iterative Learning Controller Programming

Within the control mode state machine of the top “while” loop in Figure 3.6 exists the IL controller state. The IL controller state has two sub-states: “Run” and “Process”. These two sub-states of the IL Control state are illustrated in Figures 3.7 and 3.8. Figure 3.7 shows the IL Control state when the “Run” sub-state is active; similarly, Figure 3.8 shows the IL Control state when the “Process” sub-state is active. In both of these figures, the “while” loop from Figure 3.6 is shown and the underlying machinery of the control mode state machine has been simplified to make the diagram easier to read.

The “Run” state machine, as its name implies, runs the IL iteration. If this is the first call of the “Run” state, for example if the control mode state machine just switched to IL control, or a previous IL iteration just finished processing, the feedforward trap position trajectory for the next iteration will be read into memory from its file. A flag to start recording will be raised. This flag will be seen by the data recording loop (the second-from-top “while” loop in Figure 3.6) which will open a new TDMS file and start recording the target-to-host data. Back in the IL controller, the feedforward trajectory is divided into segments whose lengths are tuned to work well with the depth and transfer rate of the host-to-target FIFO. A segment of the trajectory is loaded into the FIFO in time-series order each time the “while” loop is called. Meanwhile, the FPGA VI receives each segment of the feedforward trap position trajectory and commands the trap to follow it. Once all segments have been loaded into the FIFO, the IL controller switches to the “Process” sub-state.

When the IL controller enters the “Process” sub-state, the FPGA should be nearly done commanding the trap position along the feedforward position trajectory. To check that the FPGA is done running the iteration, the host-to-target FIFO is checked to see if it is empty as this would confirm all data has been consumed by the FPGA. The host-to-target FIFO will be continuously checked for emptiness at each “while” loop iteration until it is confirmed empty. When that is confirmed, the processing of the IL iteration results begins. First, the flag to stop recording is raised and the IL controller waits until the data recording loop confirms it has stopped recording. Then, the IL controller calculations are performed which find the next feedforward position trajectory. As part of the IL controller calculations, the recorded data is

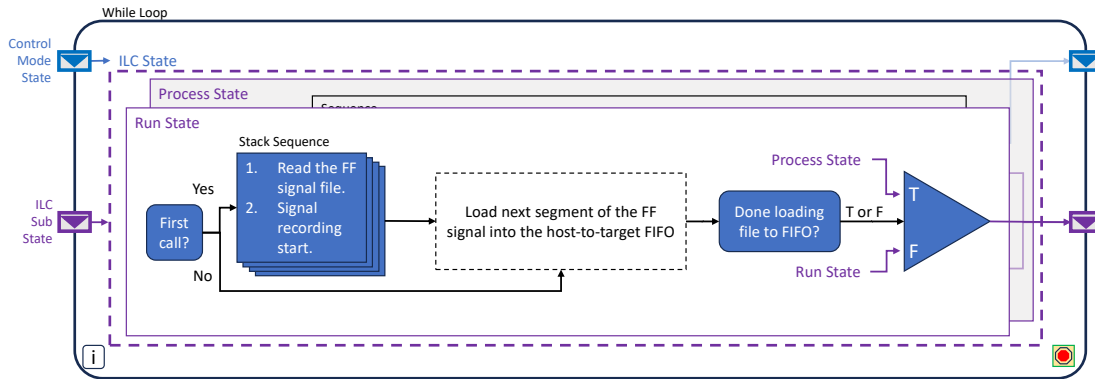


Figure 3.7: ILC State Machine: Run State

aligned and trimmed to ensure every trial is the same length and their recording time stamps align. Once the calculations are done, the new feedforward position trajectory is saved to a TDMS file. If this is not the last iteration of the IL process, the IL controller will return to the “Run” state to run the new feedforward trajectory. If this is the last iteration or the user has commanded the IL process to stop, the control mode state machine will be set to “Idle” and the IL controller state will be exited.

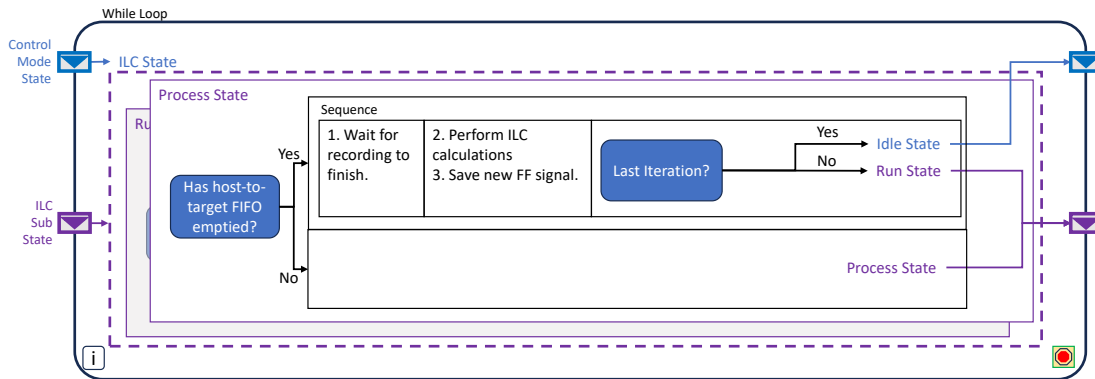


Figure 3.8: ILC State Machine: Process State

As the preceding design details suggest, a lot of the work running the IL process has been automated. The user only needs to provide the file directory path to the chosen reference trajectory, the IL controller tuning parameters, and the limit on the number of iterations to be performed. Once those are set, the IL controller runs the process and the results of each

iteration are displayed on a GUI so that the user can monitor the process for convergence of the error.

3.3.3 FPGA VI

While the real-time processor runs the GUI, data logging, visualization, and feedforward control calculations, the FPGA and its VI's need to maintain reliable input sampling, perform feedback control calculations, and output commands to the RF DDS at a rate of 80 kHz . The overall architecture of the main FPGA VI is shown in diagram form in Figure 3.9. The entire functionality of the VI is contained within the pictured three-pane sequence structure. The sequence structure runs from left to right, and begins in the first pane with initialization of the analog inputs for the QPD signals and the digital outputs for the SPI connection to the RF driver. The rightmost pane is executed last and safely shuts down the FPGA VI. The middle pane is where the input sampling, controller calculations, and SPI connection are ran.

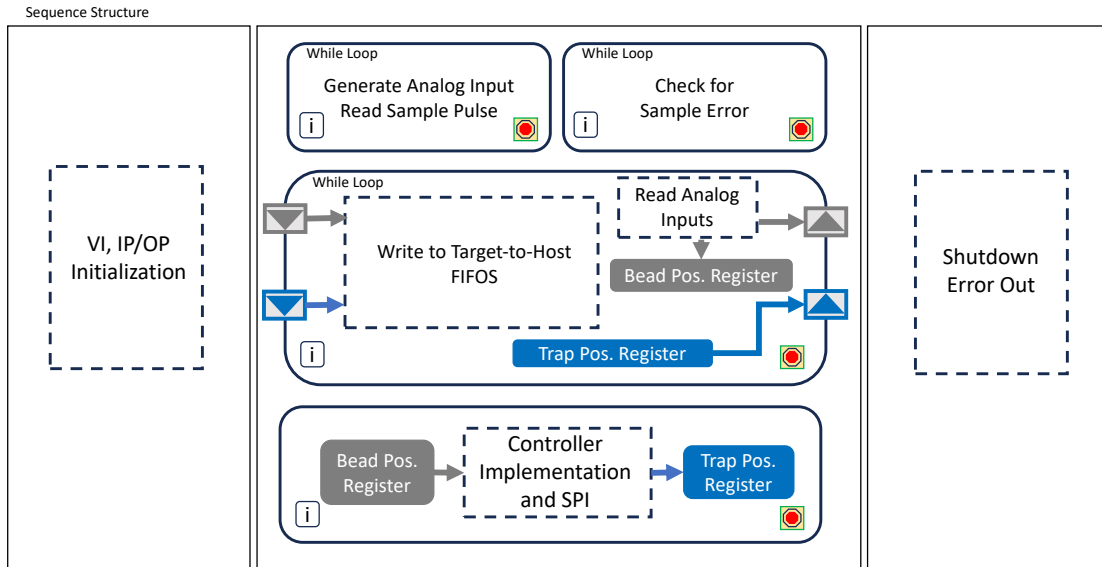


Figure 3.9: LabVIEW FPGA Firmware Diagram

Running all of this at 80 kHz requires careful programming of the FPGA. To reliably sample at this rate, the reading of the analog inputs is manually controlled via three parallel while loops, as opposed to using a built-in LabVIEW function. These three while loops are the top three loops shown in Figure 3.9. The top left loop generates the pulse signal sent to the analog input module. The top right loop checks for sample errors indicated by the ADC module. The loop

below those two handles the actual reading of the samples and loading them into the target-to-host FPGA. To ensure that the sample rate can be achieved (this sampling setup can be ran at up to 1 *MHz*), pipelining is employed. This is the practice of breaking up a linear computing task, in this case reading the analog inputs and then loading them into the target-to-host FIFO, into two parallel tasks connected by shift registers. In the while loop shown, the current analog inputs are read while the previous analog input readings, stored in the shift register, are loaded into the FIFO. Therefore reading and writing can occur in parallel. The price paid is latency, as the analog input being read will not be available to the FIFO until the loop is called again. In this case it is a worthwhile trade-off as the sample rate takes priority in recording. Note that the trap position is read at the same time as the analog input and loaded to its respective shift register at the same time. It is then bundled with the analog input into the FIFO. This ensures that the synchronization between the trap position and the analog input reading is maintained.

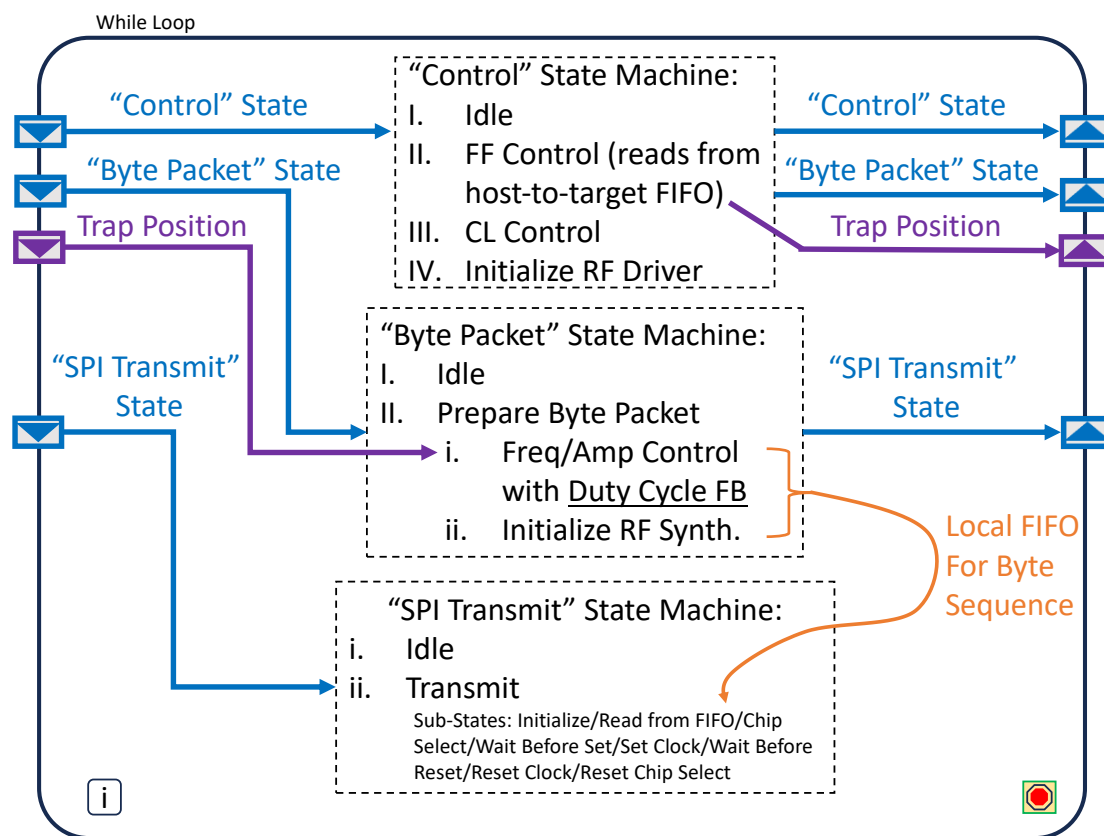


Figure 3.10: LabVIEW FPGA Controller Diagram

The bottom while loop in the FPGA diagram houses the control calculations, as well as the SPI process. A detailed view of this while loop is pictured in Figure 3.10. There are three state machines that operate in parallel and are pipelined together via state shift registers. The states of the three state machines are set in a cascading fashion. The “Control” state machine determines the next state for the “Byte Packet” state machine, and the “Byte Packet” state machine sets the next state for the “SPI Transmit” state machine. Each of these state machines are discussed in detail:

Control State Machine

The “Control” state machine is idle until the control mode is switched to one of three active states. The “Initialize RF Driver” state sets in motion the process of establishing the SPI connection with the RF driver by selecting the “Initialize RF Synth” state for the “Byte Packet” state machine. This only needs to be done when powering up the optical tweezers system. The “CL Control” state is included for future expansion to new controller designs and PID controller calculations have been tested here to ensure it can work within the 80 kHz rate. The “FF Control” state reads the trap position data (X FTW and Y FTW) from the host-to-target FIFO and loads it into the trap position shift register. Note that in the FPGA, the trap position is always expressed using the X and Y FTWs, which are 32-bit unsigned integers. After loading the trap position into the shift register, the “Control” state machine then sets the next state for the “Byte Packet” state machine to “Prepare Byte Packet”.

Byte Packet State Machine

The “Byte Packet” state machine is tasked with preparing the byte packet to be sent to the RF driver via the SPI connection. This byte packet will either be for initializing the RF driver (the “Initialize RF Synth.” state) or commanding the output amplitude and frequency of the RF driver which controls the trap strength and position, respectively (“Freq/Amp Control” state). The formatting of the byte packet is dictated by the requirements of the AD9959 DDS chip in the RF driver. The “Freq/Amp Control” state performs the duty ratio feedback control calculation and uses that to perform the multiplexing of the trap position. The duty ratio feedback controller is implemented in the “Byte Packet” state machine and not the “Control” state machine to reduce latency.

The “Initialize RF Synth.” state is needed on system startup as the AD9959 chip defaults to a single-bit, two wire SPI mode where bi-directional communication between the controller and the AD9959 is carried out at one bit per SPI clock cycle. Transferring one bit at a time is not fast enough for supporting the position update rate of 80 kHz . Thankfully, the AD9959 can be set to use other SPI modes, including a mode for uni-directional communication where

four bits are sent in parallel across four wires from the controller to the AD9959. The primary job of the “Initialize RF Synth” state is to switch the AD9959 from the single-bit SPI mode to the four-bit SPI mode. It also tells the AD9959 what its DDS clock source frequency is (100 *MHz*) and sets the phase lock loop clock rate multiplier to four. A preset byte packet contains the instructions for the AD9959 to make these setting changes:

1. Access the Channel Select Register (CSR) and set the SPI mode.
2. Access the Function Register 1 (FR1) and set the DDS clock frequency.

This byte packet is loaded into the local FIFO and the next state for the “SPI Transmit” state machine is set to “Transmit.” Note that when this byte packet is transmitted to the AD9959, the SPI connection will still be in the default single-bit mode. The AD9959 will remain in the single-bit SPI mode until after the byte packet is done transferring and the “IO Update” pin on the AD9959 is pulsed from high to low. The IO Update pin tells the AD9959 to move the received bytes from its input buffer to its registers. Once that is pulsed from low to high, the AD9959 will be in the parallel four bit SPI mode.

The other of the two “Byte Packet” states is “Freq/Amp Control”, which performs the duty ratio feedback calculations and the trap multiplexing to set the next trap position. It also sets the output power of each RF driver channel. This information is then loaded into a packet of AD9959 commands to send over the SPI connection. To determine the position multiplexing (i.e. sign flip of the X or Y position) at a given time step, the multiplexer looks at the beam position for the past three samples and the duty ratio provided by the feedback controller. The multiplexer position can be considered a binary variable where 0 indicates the negative beam position and 1 indicates the positive beam position. Here are some examples of how the multiplexer decides what the next beam position should be based on what duty ratio is commanded:

- Let the duty ratio be $\delta = 0.25$ and let the last three multiplexer positions be 010. The multiplexer’s goal is to achieve one period of multiplexing with $\delta = 0.25$ in the minimum number of samples. To do this, the multiplexer position is set to the logical NOT of the logical OR of the past three multiplexer positions. This means that the multiplexer position will be 0 so that the time-series becomes 0100. In the next time step, the calculated multiplexer position will be 0 again to achieve 01000, where in the last four samples, one period of the desired duty ratio of 0.25 has been achieved.
- When the commanded duty ratio is 0.5, the multiplexer position is simply the logical NOT of the previous position. Therefore, if the past three multiplexer positions were 001, this would become 0010. Note that for $\delta = 0.5$, the duty period is two samples.

- If the duty ratio is calculated to be 0.75, the next multiplexer position will be the logical NOT of the logical AND of the past three positions. Therefore, if the past three multiplexer positions were 001, the next would be 1 to achieve 0011. In this case, one more time step would be needed to achieve one duty period for $\delta = 0.75$.
- The multiplexer's position choice for $\delta = 0$ or $\delta = 1$ is simple, it is always 0 if $\delta = 0$ and always 1 if $\delta = 1$.

Because only the past three time steps are used to determine the next position, only duty ratios that can be achieved in four samples can be used. As such, the duty ratio provided by the feedback controller is coerced to the set $[0, 0.25, 0.5, 0.75, 1]$ to work with the multiplexer. A larger set of duty ratios could be used, but the additional values would all have duty periods longer than four samples which would reduce the multiplexing rate. For example, the duty ratio of 0.125 would have a duty period of 8 samples, halving the effective multiplexing rate. Note that duty ratio feedback control and multiplexing is only applied to one axis of actuation, X or Y , at a time but this could easily be changed in future work.

Once the beam X and Y positions are decided, they are first loaded into the local FPGA VI shift register to be read into the target-to-host FIFO by the data transfer loop. The output power of the AD9959's X and Y channels are set independently using two ATWs. The ATW values to be used are selected at each time step from a position-dependent look-up-table. The amplitude look-up-table is used to adjust the AD9959's power output for each channel independently to flatten the optical power in the trap across the AOD's X and Y scan ranges. The look-up-table is determined during the build process for the optical tweezers, the process of which is discussed in Section 3.4.

Once the FTWs and ATWs are known, they are packed in a byte packet that is loaded into a local FIFO to be read by the "SPI Transmit" state machine in the next loop cycle. The byte packet contains the tuning words and the following instructions to the AD9959 DDS chip:

1. Access the Channel Select Register (CSR).
2. In the CSR, select the target output channel (Ch. 0 for X , Ch. 1 for Y).
3. Access the Channel Frequency Tuning Word (CFTW) register and write the FTW.
4. Access the Amplitude Control Register (ACR) and write the output power, which is the 10-bit ATW packed into three bytes.
5. Repeat steps 1-5 for the second output channel.

Once this byte packet is loaded into the local FIFO, the next state for the "SPI Transmit" state machine is set to "Transmit."

SPI Transmit State Machine

This state has the sole purpose of running the SPI connection to the AD9959 chip in the RF driver. The SPI processes is implemented as a state machine where states are executed at the FPGA clock rate of 40 *MHz*. The SPI clock phase is set to zero which means that the controller loads data to its digital output pins when the clock is low and the AD9959 reads this data on the rising clock edge. The SPI clock polarity is also zero which means that when the chip select bit is idle (idle = high), the clock is low. The entire SPI data transfer process is programmed as follows:

1. **Initialize:** The SPI connection is initialized by ensuring that both the clock and IO update pins are low.
2. **Read from FIFO:** The local FIFO containing the byte packet is read to obtain the next byte to be sent.
3. **Chip Select:** If this is the first call for a given a byte packet, the next step is to set the Chip Select (CS) pin to active (low) and load the first four bits of the current byte to the four digital output pins.
4. **Wait Before Set:** Now there is a delay before the clock is set to high. A counter counts down 40 *MHz* clock cycles for a designated number of counts. The number of counts specified controls the SPI clock rate.
5. **Set Clock:** The clock is set high. On the AD9959's end, the four bits are read.
6. **Wait Before Reset:** The same number of wait cycles used when waiting for set are counted down here.
7. **Reset Clock:** The clock is reset to low. If the entire byte packet has been written, the IO Update pin is toggled to high and the next state is set to Reset CS. Otherwise, the next four bits are loaded into the digital output pins. If these are the last bits of a byte and more bytes remain to be sent, the next state is set to Read from FIFO. If these are not the last bits of the current byte, the next state is Wait Before Set.
8. **Reset CS:** If there are no more bytes in the bytes packet to transfer, chip select is returned to idle (high) and the IO Update pin is toggled back down to low.

This entire SPI transfer happens fast enough to support the trap position update rate of 80 *kHz*. The role of the SPI connection in limiting the closed-loop control rate is discussed in Section 3.3.4.

3.3.4 Closed-loop Control Rate & Coordinate System

The characteristics of the system hardware and programming introduce limitations that affect and define key controller parameters. This includes the closed-loop control rate, the controller latency, and the controller coordinate system. Just how the hardware and programming determines these parameters is discussed here in this sub-section, starting first with the closed loop control rate.

Closed-loop Control Rate

The entire signal pipeline, from sensing to actuation, needs to be considered when discerning the achievable closed-loop control rate. Here the maximum theoretical update rate for each component is calculated and the component that constricts the update rate is identified. Starting with the sensor, the QPD has a bandwidth of 250 kHz and its amplifiers have a 1 MHz bandwidth. The NI 9223 ADC module and the FPGA programming can support the sampling of the QPD signals at 1 MHz . Within the FPGA, the SPI programming is the rate limiting process. The sequence to transfer the 22 byte long command packet that sets the frequencies and amplitudes of both the X and Y RF driver channels takes a minimum of 180 clock cycles at a clock rate of 40 MHz . This means that the programming implementation of the SPI connection can theoretically support a position update rate of 222 kHz . The NI 9401 DIO module that runs the cRIO side of the SPI connection has a per channel maximum switching frequency of 5 MHz across all eight output channels, so with four channels for writing bits over SPI the byte rate could reach 2.5 MBps and therefore it could transmit command packets at a rate of 114 kHz . The SN74LVC245A bus transceiver has an output transition time of 3.8×10^{-9} seconds. Based on the design guideline that a bus transceivers output transition time not exceed 30% of the input bit rate period, each of the bus transceiver channels could support up to 80 Mbps . With four channels dedicated to transferring the 22 byte long command packet, the bus transceiver can support a position update rate of approximately 1.8 MHz . On the RF driver side of the SPI connection, the serial port clock rate is limited to 200 MHz . With four channels receiving bits in parallel, receiving the 22 byte command packet could occur at maximum rate of 4.5 MHz . Within the RF driver, changes in the FTW and ATW for a channel take 29 DDS clock cycles to reach that channel's digital to analog converter (DAC) output after the IO update pulse is issued. One IO update pulse triggers the update on both channels. With a 400 MHz DDS clock rate, this amounts to 72.5×10^{-9} seconds, corresponding to an update rate of 13.8 MHz .

Once a new RF signal reaches the AOD, the associated change in the AOD's diffraction grating will take some time to complete. This is referred to as the "access time" and it is the amount of time it takes the updated signal to pass through the entire diameter of the laser beam. The formula to calculate the access time is the diameter of the beam divide by the velocity of

the acoustic waveform in the AOD crystal. Here, the access time of the AOD for the trapping laser which is 1.85 mm in diameter is 2.92×10^{-6} seconds. This means that each channel of the AOD could theoretically be updated at a rate of 342 kHz.

In practice, the SPI connection is the rate limiting process. This is not due to the SPI programming but instead the physical characteristics of the SPI connection which further limit the achievable closed-loop bandwidth down to 80 kHz. This is because SPI connections at high bit rates and longer run lengths are prone to capacitive effects and noise interference that can lead to bit errors. The SPI connection here can be run fast enough to support a closed-loop bandwidth up to the 111 kHz limit imposed by the NI 9401 DIO but bit errors occurred roughly at an hourly rate and required the RF driver to be rebooted each time. Consequently the SPI connection was slowed down such that transmitting the command packet takes 444 SPI clock cycles. This results in a much more reliable connection and the closed-loop control rate of 80 kHz.

Controller latency

There is a latency period between the NI cRIO receiving a QPD measurement and the trap position in the sample chamber being updated. The key contributors here are the FPGA programming, the RF driver, and the AOD. The relevant FPGA programming is illustrated in Figure 3.10. When the QPD signal is read into the FPGA VI, it is immediately loaded into a local register to be read by the feedback controller whose output is then loaded into the command packet. This takes one loop iteration to complete (12.5×10^{-6} s). In the next loop iteration, the new command packet will be available to the SPI state machine. The command packet will take 444 SPI clock cycles at a clock rate of 40 MHz to transmit, totaling 11.1×10^{-6} seconds. The NI 9401 DIO module has a maximum output propagation delay of 100×10^{-9} seconds. Once received by the RF driver, it will take the aforementioned 72.5×10^{-9} seconds for the changes to reach the output of the RF driver's two DACs. When the new RF signals reach the AOD, the two crystals will take 2.92×10^{-6} seconds to update. This equates to a total latency of 26.7×10^{-6} seconds. In Chapter 4, it will be shown that this is an acceptable controller latency as the feedback controller works as intended.

Coordinate System

The actuation axes of the AOD define the X and Y axes of the coordinate system, as illustrated in Figure 3.11. The Z axis is defined as coincident with the optical axis of the detection laser. The focal point of the detection laser defines the origin of the (X, Y, Z) coordinate system. The QPD X and Y axes are coincident with the AOD X and Y axes of actuation. The actuation

range of the trapping laser within the sample chamber is approximately $17\ \mu\text{m}$ in both the AOD's X and Y directions.

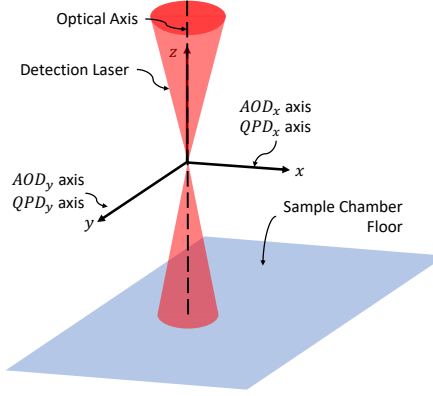


Figure 3.11: Optical Tweezers Coordinate System

3.4 Calibration

Calibration of the optical tweezers system is needed on several fronts. First, the RF drive power compensation table needs to be found to flatten the optical power across the AOD scan range. Then, actuation of the trap must be calibrated such that it can be reliably commanded in nanometers. Lastly, the detection system needs to be calibrated such that the trapped particle position is reported in nanometers. The application of these calibrations to this optical tweezers system is outlined here.

3.4.1 Flattening Optical Power Across the AOD Scan Range

If the controller commands a constant power RF driving signal while sweeping the driving frequency from 18 to 36 MHz across either AOD axis, the optical power observed in the trapping laser after exiting the AOD will vary across the scan range. This is shown for both AOD_x and AOD_y scans in Figure 3.12. The variation is due primarily to nonlinearities in the RF driver and the AOD. Coupling between the laser power and trap location is highly undesirable. For small displacements in the trap position this effect will be negligible, but for IL experiments the entire scan range will be used.

To combat this, the respective drive powers of the two RF driving signals are varied as a function of their respective RF drive frequencies in real-time to ensure flat optical power in the trapping laser beam across the scan range. For each axis, the correct RF drive power (expressed

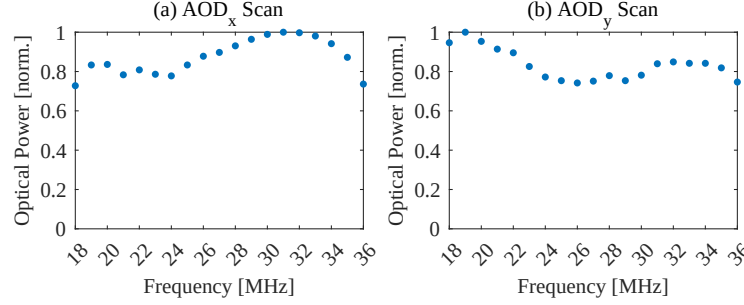


Figure 3.12: Optical Power for AOD Scans without RF Power Compensation

as an ATW) for a given frequency is pulled from a trap-position dependent look-up-table that is loaded into the FPGA. The look-up table only needs to be found once during construction of the system or after hardware changes. The following process is used to build the look-up table, starting with the AOD_x axis:

1. Before the lenses l_1 and l_2 shown in Figure 3.1 are installed, an optical power meter is placed in the path of the trapping laser after it passes through the AOD. This is illustrated in Figure 3.13. Note that the AOD produces not just one beam but actually a pattern of laser beams, i.e. diffraction orders, and only the (1,1) diffraction order is used for trapping. The rest are diverted to a beam dump in the final system assembly. For this calibration, the optical power meter is placed in the path of the (1,1) diffraction order. The power meter output is fed to an operational amplifier circuit which connects to an oscilloscope.
2. The X scan range from 18 MHz to 36 MHz is stepped through in one mega-hertz increments. At each driving frequency, the RF driver power is adjusted upwards until the optical power in the trap begins to saturate. When the optical power saturates and does not increase with greater RF driver power, peak diffraction efficiency has been achieved. After performing this process at each frequency, the result will be a look-up-table that ensures optimal diffraction efficiency at each driving frequency, but uneven optical power in the trap across the scan range.
3. Now the frequency with the lowest optical power is picked from the table. For a flat optical power response, this is the limiting case. The RF drive power at every other frequency will be adjusted down until the optical power at that frequency matches that lowest optical power. Now the look-up-table will ensure flat optical power across the scan range at the expense of diffraction efficiency at some frequencies.

4. Steps one through three are repeated for the AOD_y axis to find its look-up-table.

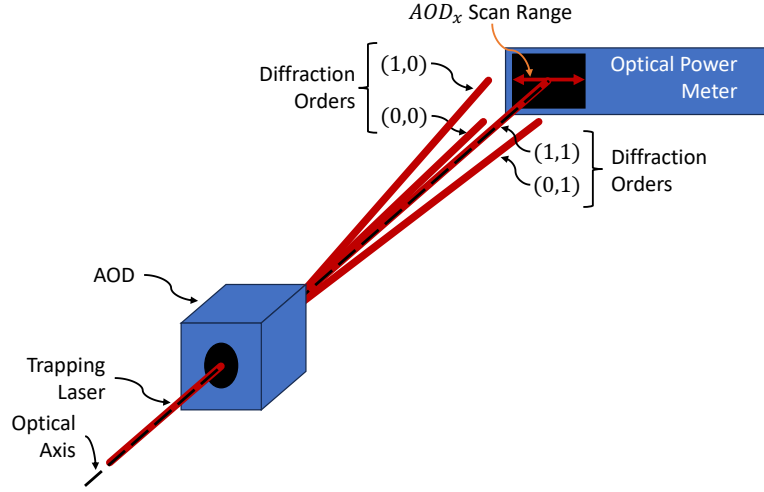


Figure 3.13: Measuring the optical power across the AOD scan range

Once the look-up-tables for both the AOD_x and AOD_y drive power are found, they are saved as arrays in the real-time LabVIEW VI. The bottom “while” loop in Figure 3.6 loads the arrays to the FPGA via the FPGA set/get LabVIEW function. Then in every subsequent call, the “while” loop checks if the user has made any changes to the arrays, and if yes, sends the new arrays down to the FPGA. The FPGA accesses the look-up-table arrays in the “Byte Packet” state machine, as described in Section 3.3.3.

3.4.2 AOD Actuation Calibration

The trap position needs to be commanded in nanometers but the AOD is commanded in hertz. Consequently, a calibration constant is needed to convert a desired position in nanometers to the AOD input frequency in hertz that corresponds to that position. There will be one constant for converting the X position and another for the Y position. Calculating these constants only needs to be done once during construction of the optical tweezers system or after optical hardware changes. In general the process here follows that published in [90]. The first step for finding these constants is to determine the number of pixels per micron in the microscope camera image. This is done by imaging a stage micrometer, which is a microscope slide with micron-scale increments etched into its surface. The slide used here (ThorLabs R1L3S2P) has $10\ \mu m$ increments that are accurate to $\pm 0.5\ \mu m$, which is within the industry standard for such a micrometer. To get the number of pixels per micron, the center-to-center pixel distances between the demarcations

in the slide image are counted for all marks within the field of view. These measurements are averaged to find the number of pixels per micron. For this optical tweezers system, the average pixels per micron is 9.3 with an error of ± 0.5 .

Now that the $pixel/\mu m$ coefficient is known, the $Hz/pixel$ coefficient needs to be found. This is done by scanning a trapped bead through the entire AOD scan range by linearly ramping the input frequency on one channel from 18 to 36 MHz and back again. Specifically, the trap is commanded to follow a 0.1 hertz triangle wave with minima and maxima at the 18 and 36 MHz AOD positions, respectively. This sweep is done separately on the X and Y axes and is recorded for 30 to 40 periods to enable averaging of the results. Each scan video is processed using the Trackmates plugin in ImageJ [91] to trace the bead path and turn it into X, Y pixel coordinates. This Trackmates data is then imported to a MATLAB processing script specially written for this optical tweezers system build. The AOD X and Y axes of actuation are not perfectly aligned with the camera's X and Y pixel axes and in the MATLAB script an optimal frame transformation is found and applied to align the coordinate systems. The X (Y) position is plotted as a function of time, and the slope of the X (Y) position data in pixels per second is found at each ramp up and down. The slope is then converted to pixels per mega-hertz using the known frequency ramp rate in mega-hertz per second. For this optical tweezers system, the coefficient for the X -direction is 8.60 pixels per MHz with a standard deviation of ± 0.03 . In the Y -direction, the coefficient is 8.70 pixels per MHz with a standard deviation of ± 0.04 . The desired $MHz/pixel$ coefficients are then found by inverting these results.

With both the $pixel/\mu m$ and $MHz/pixel$ coefficients known for both X and Y , the desired Hz/nm coefficients can be found. For the X -direction, the coefficient is $1.08 \times 10^3 Hz/nm$ with an error of $\pm 0.06 \times 10^3 Hz/nm$, and for the Y -direction, it is $1.06 \times 10^3 Hz/nm$ with an error of $\pm 0.06 \times 10^3 Hz/nm$. Note that when LabVIEW commands the RF driver to output a certain frequency to the AOD, it is sending the command as a FTW, not a frequency in mega-hertz. Consequently, for the actual programming in LabVIEW the coefficients of interest are the FTW/nm values for the X and Y AOD axes. These are found by plugging in the Hz/nm coefficients into Equation 3.1 and getting their corresponding FTW values.

Once this process is completed, the trap position can be commanded in nanometers and the conversion to FTWs is handled automatically by LabVIEW. Again, this only needs to be done when the optical tweezers system is built or modified. The calibration of the trap actuation has been completely independent of the detections system. As such, actuation of the trap can be serve as a ground truth when calibrating the detection system, as discussed in the following section.

3.4.3 Detection System Calibration & Sensitivity

Calibration

The detection system is aligned to the optical axis of the microscope during construction. However, each time a microscope slide is to be placed on the stage, the condenser (Figure 3.1) needs to be raised to allow access and then lowered back down. This creates slide to slide variation in the fine alignment of the detection system. To counteract this, at the beginning of an experiment the alignment of the detection system is finely tuned using a trapped bead and adjustments made in LabVIEW. The goal is to command the bead to follow a time-series trajectory along either AOD axis and have the QPD time-series signal match that trajectory. If that condition is true, the QPD signal will have been calibrated to nanometers for a bead of that size. Two bead trajectory waveforms are used, a 5 Hz , $\pm 150\text{ nm}$ square wave and a 5 Hz , 1400 nm peak-to-peak triangle wave. The waveforms are applied to one axis at a time, i.e. actuation is strictly along the AOD_x axis, and then strictly along the AOD_y axis. The square wave is used to verify the QPD is aligned within its linear detection range (which is approximately $\pm 150\text{ nm}$ in both X and Y), and the triangle wave is used to ensure the QPD is well aligned for its nonlinear detection range. The linear range can be extended into the nonlinear range using a method in literature [88] if the system is well-calibrated to the triangle wave. The digital tuning knobs in LabVIEW used to align these signals are:

- **AOD X and Y position offsets:** The focal point of the detection laser and the focal point of the trapping laser will be closely aligned in the XY -plane but not perfectly so. To improve their alignment, the X and Y position offsets can be applied to the optical trap. This is illustrated in Figure 3.14 where ΔAOD_x and ΔAOD_y are the trap position offsets that make the two coordinate system origins coincident. If the coordinate system origins are not coincident, the moving bead will not be centered on the detection laser and the resulting interference pattern, and therefore QPD signal, will not be symmetrical. The offsets found during calibration are typically in the ranges $\Delta AOD_x = \pm 100\text{ nm}$ and $\Delta AOD_y = \pm 100\text{ nm}$.
- **QPD X and Y scaling:** The X and Y QPD signals output by the QPD amplifier are in the units of volts but must be converted to the units of nanometers. This is done by adjusting the gains applied to each signal. The two gain knobs on the QPD amplifier, as shown in Figure 3.4, are used during system construction to get these voltages near the calibrated range. Then at the beginning of each experiment finer adjustments are made by adjusting the digital gain in the LabVIEW GUI. The gain applied to the QPD_x signal is adjusted until the magnitude of QPD_x matches the magnitude of the commanded AOD_x

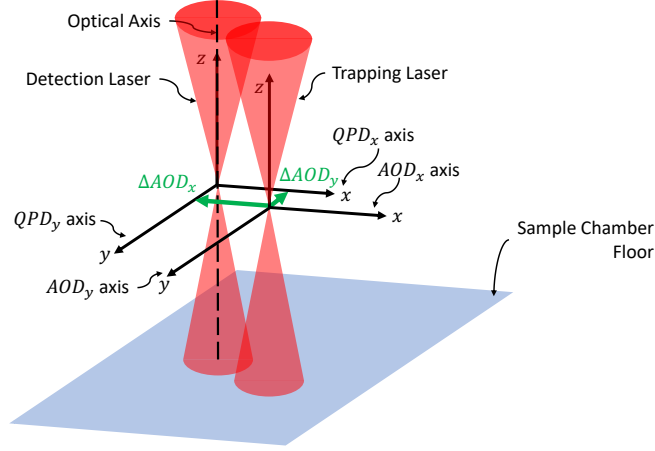


Figure 3.14: Detection System Alignment

scan in the linear region. The same is done for the QPD_y gain so that QPD_y matches the commanded AOD_y scans in the linear regions.

- QPD X and Y offsets: The current-to-voltage amplifier of the QPD itself and the QPD amplifier can introduce offsets in the QPD_x and QPD_y signals. These are compensated for by applying offsets to QPD_x and QPD_y in LabVIEW such that the square wave signals are centered and QPD_x and QPD_y equal zero on average when the bead is held at the origin.

Adjusting these digital tuning knobs such that the QPD signal matches the square wave is straightforward. Achieving good alignment for the triangle wave, however, is less obvious as the nonlinear region of the response cannot be made to match the triangle wave. An example of what the nonlinear response looks like on the LabVIEW GUI's live oscilloscope display is included in Figure 3.15. In this figure, the AOD offsets, QPD scaling terms, and QPD offsets are all shown. They are adjusted with the goal of making the QPD response match the triangle wave well in the linear QPD response regions ($\pm 150 \text{ nm}$), and symmetrical top to bottom in the nonlinear regions, as shown in the figure.

Sensitivity

The sensitivity of the detection system to small movements of the trapped bead is an important system performance metric. The power spectrum of a trapped bead can be used to characterize the detection system's sensitivity [92, 32, 93]. To discern how small of a displacement can be detected by the system, a trapped bead's position is actuated along the AOD_x direction



Figure 3.15: Example Oscilloscope Readout During Alignment

following a 150 Hz sine wave trajectory. The trapping laser power is set to 500 mW at the laser head to better separate the noise characteristics of the system from the Brownian motion of the bead. Recordings of the bead's position are made at increasingly smaller sine wave amplitudes and in post processing the power spectrum of the recorded signal is calculated. The displacement detection limit is then equal to the smallest sine wave peak-to-peak amplitude that is discernible in the power spectrum signal. This process is repeated for the Y axis of actuation and detection. The QPD_x power spectrum for a 1.5 nm peak-to-peak sine wave in the AOD_x direction is included as sub-figure (a) of Figure 3.16, and the QPD_y power spectrum for the same sine wave in the AOD_y direction is included as sub-figure (b).

In both of these figures, the presence of the sine wave displacement is indicated by an arrow and is discernible above the power spectrum noise floor. If the peak-to-peak amplitude of the sine wave is reduced to 1 nm , the sine wave is no longer discernible above the noise floor. Therefore the detection limit of this system at or near the frequency of 150 Hz is approximately 1.5 nm . Further characterization of the detection system sensitivity at other frequencies was not necessary for IL experiments and is left as an area for future work.

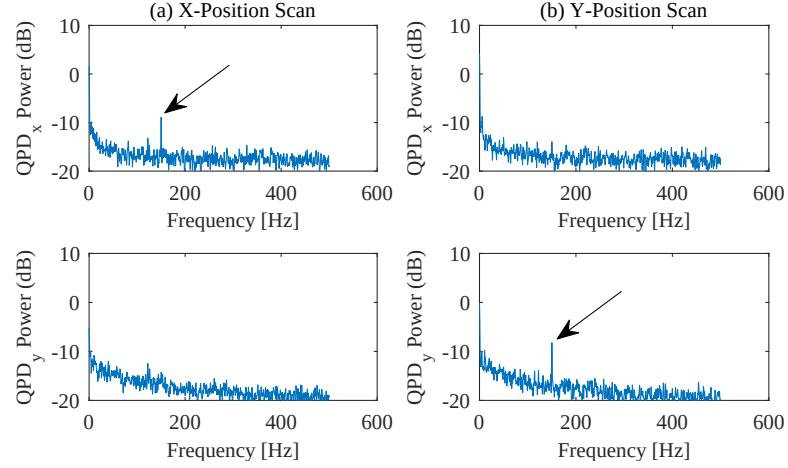


Figure 3.16: Detection system sensitivity to a 150 Hz , 0.75 nm amplitude sinusoid

3.5 Estimating Trap Stiffness

As mentioned in Section 3.3.1, the LabVIEW program GUI offers several redundant methods for estimating the stiffness of the optical trap for a trapped bead in real-time. This includes the equipartition method, the mean squared density method, the autocorrelation method, and the power spectral density method. A MATLAB script was also written to perform the same calculations offline. If all four methods give similar stiffness coefficient results for a trapped bead, it signifies good trapping and detection performance. Since this thesis is focused more on trapping non-spherical particles, the discussion of these methods will be brief and are included mainly to give some context to the strength of the trap used for experiments in Chapter 4. The summaries below follow the formulations provided in [3, 94].

Equipartition Theorem

The least computationally intensive way to find the trap stiffness is the equipartition method. This method relies on the assumption the trapped particle is spherical and that the potential well of the trap is harmonic. With these assumptions, the equipartition theorem can be used to relate the standard deviation of the trapped particle's position to the stiffness of the trap. This is done separately for the X -direction and Y -direction as shown in Equation (3.2)

$$k_x = \frac{K_B T}{\sigma_x^2}, \quad k_y = \frac{K_B T}{\sigma_y^2}, \quad (3.2)$$

where K_B is Boltzmann's constant, T is the temperature of the sample chamber, σ_x^2 and σ_y^2 are the standard deviations of QPD_x and QPD_y , respectively. The terms k_x and k_y are the

harmonic X and Y stiffness of the trap, respectively. In the LabVIEW real-time program this is continuously calculated using 0.2 s segments of the QPD signals to find the standard deviations.

Mean Squared Displacement

The second method implemented both in LabVIEW and MATLAB uses the mean squared displacement (MSD) of the trapped bead. In discrete-time, the mean squared displacement is a function of the time shift $k \in \mathbb{Z}^+$ and is calculated using the following equation:

$$MSD_x(k) = \frac{1}{N-k} \sum_{j=1}^{N-k} (QPD_x(j+k) - QPD_x(j))^2. \quad (3.3)$$

where N is the number of QPD_x samples recorded. This discrete time-series data is then fit to the continuous time expression for the mean-square density of an optically trapped spherical bead:

$$MSD_x(\tau) = 2 \frac{K_B T}{k_x} (1 - e^{-\frac{|\tau|}{\tau_{k,x}}}), \quad (3.4)$$

where $\tau_{k,x}$ is the characteristic time of the optical trap. In MATLAB, the fit is performed using the “fit” function, while in LabVIEW it is performed using the Nonlinear Curve Fit VI. These same functions are used for the fits in the following two stiffness estimation methods as well. Once the fit is performed, the stiffness k_x can be found from the fit coefficients. The process is the same to find k_y .

Autocorrelation Function

Applying the autocorrelation function (ACF) to the QPD signals to find the trap stiffness follows essentially the same procedure as used for the mean square displacement method. The discrete time auto-correlation for QPD_x is calculated as follows:

$$C_x(k) = \frac{1}{N-k} \sum_{j=1}^{N-k} QPD_x(j+k) QPD_x(j) \quad (3.5)$$

This is then fit via least-squares to the auto-correlation function for a trapped sphere (Equation (3.6)) to find k_x . The process to find k_y is analogous.

$$C_x(\tau) = \frac{K_B T}{k_x} e^{-\frac{|\tau|}{\tau_{k,x}}} \quad (3.6)$$

Power Spectral Density

The last method considered here uses the power spectral density (PSD) of the QPD signals to find k_x and k_y . Again, this involves fitting an experimentally determined statistic to its theoretical expression. The theoretical expression for the expected value of the power spectral density of a trapped spherical bead in a harmonic potential is [94]:

$$PSD(f) = \frac{1}{2\pi^2} \frac{D}{f_{c,x}^2 + f^2} \quad (3.7)$$

where f is the frequency, D is the diffusion coefficient of the spherical bead, and $f_{c,x} = k_x/(2\pi\gamma)$ with γ being the spherical bead's drag coefficient.

In practice, discrete-time QPD_x data is taken at time instances $j = 0, 1, \dots, N$ for a total time of $T_s = N\Delta t$, where Δt is the sampling period. In MATLAB, the PSD is calculated from this data using the built-in fast Fourier transform (FFT) function:

$$PSD(f_k) = \frac{1}{T_s} |\Delta t FFT(QPD_x(j))|^2 \quad (3.8)$$

where f_k indicates the k -th frequency point of the FFT. In LabVIEW, the built-in power spectrum VI is used, which will be referred to here as function $P(\cdot)$:

$$PSD(f_k) = T_s P(QPD_x(j)) \quad (3.9)$$

where the multiplication by T_s arises from the underlying definition of the power spectrum VI which is $P(f_k) = |FFT(QPD_x(j))|^2/N^2$.

The power spectral density is measured many times over and then averaged together so that the result can be fit to the theoretical expected value in (3.7). Once the fit coefficients are known, k_x can be found from $f_{c,x} = k_x/(2\pi\gamma)$. The process is the same to find k_y using QPD_y data.

Estimated Trap Stiffness

While the experiments in this thesis concern the trapping of non-spherical particles, knowing the trap stiffness for a trapped spherical bead is helpful for putting the trap strength in context. For the forthcoming IL control experiments, the trapping laser power is set to 300 *mW* at the laser head. For a spherical bead in this trap, the trap stiffness coefficients calculated by the four methods are:

Table 3.1: Estimated Trap Stiffness

	Method			
Trap Stiffness	Equipartition	MSD	ACF	PSD
k_x [fN/nm]	45.98	45.53	46.19	45.92
k_y [fN/nm]	56.17	56.51	56.54	56.14

The four k_x values are in good agreement with one another, as are the four k_y values. The trap stiffness is slightly weaker in the X -direction. This is due to the slightly elliptical potential well of a linear polarized laser [3]. To put these values in context, the trap stiffness coefficients seen here are in the typical 20 to 60 fN/nm trap stiffness range used in biochemistry experiments [95]. Consequently, the trap potential used in Chapter 4 is not unusually strong.

3.6 Conclusion

This chapter has described in detail the construction of an optical tweezers system that will take the simulated IL concept in Chapter 2 to experimental verification in Chapter 4. This included not only the optical design of the system, but also the extensive LabVIEW programming required to make the real-time controller work reliably. Many of the key electronic components had to be custom built in-lab to meet the system requirements. System calibration and verification are also important and those processes were recounted for this system. The results in Chapter 4 will serve as further verification of the system's performance.

Chapter 4

ILC Experiments

4.1 Introduction

In Chapter 2 the concept of applying IL control to the manipulation of a non-spherical particle was proven in-silico using the instantiating example of rotating a cylinder out-of-plane. Experimental verification of this example must follow and in Chapter 3 the construction of an optical tweezers system for experimental verification was detailed. With that hardware, the IL control concept can now be tested. A few details in the experimental approach will differ slightly from the approach taken in Chapter 2. The non-spherical particle considered here is a micro-rod, which is a cylinder with a higher aspect ratio than that considered in Chapter 2. The micro-rod was chosen as it has precedents in literature [96, 97, 98] and is readily available commercially which will make these results more easily replicable by others in the future. The details of this micro-rod are provided in Section 4.1.1. Additionally, the sensing of the micro-rod's position will be carried out differently than assumed in simulation. Instead of image based processing with a high-frame rate camera, the QPD detection system detailed in Chapter 3 is used. The QPD detection approach is more common, less expensive, less complicated, and faster than real-time image processing methods. The use of this system to discern the micro-rod's position and orientation is detailed in Section 4.1.2. The different sensing approach requires updating the IL controller design with new signals and new reference trajectories and this is detailed in Section 4.2. Finally, experimental results are presented.

4.1.1 Methods

The experimental sample consist of two side-by-side sample chambers built using double-sided tape and two number 1.5 glass coverslips. One sample chamber contains a 1:100 dilution of 840

nm diameter fluorescent polystyrene bead stock solution (Spherotech VFP-0856-5, 0.1% w/v) suspended in deionized water. In the second sample chamber is a $1.7 \times 10^{-3}\%$ w/v solution of glass micro-rods (Nippon Electric Glass PF-30S) in deionized water. These micro-rods are nominally $15 \mu m$ in length and $3 \mu m$ in diameter.

To manipulate the orientation of a non-spherical particle, two optical traps are needed to act as end-effectors. The use of two traps to manipulate a non-spherical particle dates back to Ashkin's 1980's work on optical levitation and trapping of non-spherical particles [99, 48]. The requirement of two lasers can be relaxed by time-sharing or "multiplexing" a single laser between two desired trap locations [66, 40]. The multiplexing approach is employed here. The trapping laser in Figure 3.1 is switched between two trap locations at a rate of $80 kHz$ using the AOD. This corresponds to a multiplexing period of $1.25 \times 10^{-5} s$. From diffusion theory [100], the expected drift d of the micro-rod from the trap center during the time period the laser is off, $\tau_{off} = 6.25 \times 10^{-6} s$, is calculated as $d = \sqrt{6D\tau_{off}} = 1.42 nm$, where $D = D_x = D_y = 0.5354 \times 10^{-13} m^2/s$. Compared to the dimensions of the micro-rod, this is an insignificant amount of drift.

Like any optical tweezers system employing interferometric detection, the detection system must be calibrated at the beginning of an experiment. The calibration involves two steps. The first step is the standard detection signal calibration using a trapped bead [40]. The trapped bead is scanned along a known trajectory and the detection system parameters are tuned until the QPD signal matches the commanded trajectory, as described in Section 3.4 of Chapter 3. The trapped bead calibration ensures consistent detection system performance across experiments. Note that since the detection system signal is calibrated using a bead and not a micro-rod, the QPD signals for a micro-rod are in the units of scaled volts. The units of the QPD signal are not important to the IL controller so the QPD signals are considered non-dimensional.

The second calibration step measures the QPD magnitude difference between a micro-rod's horizontal and vertical orientations. A micro-rod is picked up from the chamber floor in the horizontal position using two traps placed at $\pm 6 \mu m$ from the origin, which are near the ends of the nominally $15 \mu m$ long micro-rod. The horizontal position is chosen to be aligned with the rotation plane the experiment will take place in (XZ or YZ). For experiments in the XZ plane, QPD_x is used. For the YZ plane, QPD_y is used. Using the objective height adjustment, the micro-rod is raised to the experimental height ($\approx 11 \mu m$ above the chamber floor) and the QPD signal magnitude is recorded. The QPD magnitude for the vertical orientation is found the same way but with the micro-rod held using only one trap. Then the differences $\Delta QPD_x = QPD_{x,hORIZ.} - QPD_{x,VERT.}$ and $\Delta QPD_y = QPD_{y,VERT.} - QPD_{y,HORIZ.}$ are recorded. These ΔQPD values are used to scale the IL controller reference trajectories such that they match the current QPD calibration. Scaling the reference trajectories is discussed further in

Section 4.2. This only needs to be done at the beginning of the experiment using one micro-rod, barring any drift in the detection system hardware or the objective's z -height.

4.1.2 Micro-Rod Position and Orientation Detection

The type of detection system employed here historically has been used for tracking spherical particles. Its application to non-spherical particles is less common but a few examples do exist. Two examples use the detection system to estimate the force on the particle, but not its position or orientation [97, 50]. Another uses a QPD system to discern the angle two fused beads make in the XY -plane [67]. To show how a detection system can track both the out-of-plane orientation and position of a micro-rod the following study is performed.

To begin, a micro-rod is held in the vertical orientation (approximately aligned with optical axis) using a single trap. An image of this is included in sub-figure (d) of Figure 4.1. While held in this orientation, the micro-rod is moved back-and-forth along the X or Y axis by moving the trap. The QPD_x and QPD_y signals of the X and Y position scans are included in sub-figures (a) and (b) of Figure 4.1. Sub-figure (a) shows that for movement along the X direction, QPD_x is strongly correlated with the micro-rod's X position with some cross-talk seen in QPD_y . For movement along the Y direction, as shown in sub-figure (b), QPD_y shows some correlation to the Y position, but is diminished in magnitude when compared to the cross-talk seen in the corresponding QPD_x signal and also the QPD_x signal seen for the X motion. Additionally, the QPD_y data for the Y position scan has a negative offset. The reduced magnitude and negative offset of QPD_y for the Y position scan is because the micro-rod's equilibrium orientation in a single trap includes a slight tilt about the X -axis, as seen in Figure 4.1. The tilt of a trapped micro-rod in a single optical trap has been well studied both numerically and experimentally [101, 102], as has the alignment of nonspherical particles in linearly polarized light [103]. The tilt can be eliminated by multiplexing the laser ± 2000 nm about the origin along the Y -axis. This changes the shape of the trap's potential energy profile such that the micro-rod is no longer tilted. An image of this is included as sub-figure (f) of Figure 4.1. The results of moving the micro-rod along the Y axis with the tilt eliminated are pictured in sub-figure (c) of Figure 4.1. Now the Y position scan results more closely match those of the X position scan, showing that the QPD_y signal is correlated to a vertical micro-rod's Y position. Additionally, comparing the Y scan results in sub-figures (b) and (c) shows that the QPD signal becomes less correlated with position when the micro-rod is tilted.

In addition to being correlated with the micro-rod's position, the QPD signals are also correlated with the micro-rod's angles about the X and Y axes. The example shown in Figure 4.1 demonstrates the correlation between QPD_x and the micro-rod's angle about the Y axis. For the vertical orientation shown in sub-figure (d), $|QPD_x| = 0.1$ n.d.. For the horizontal

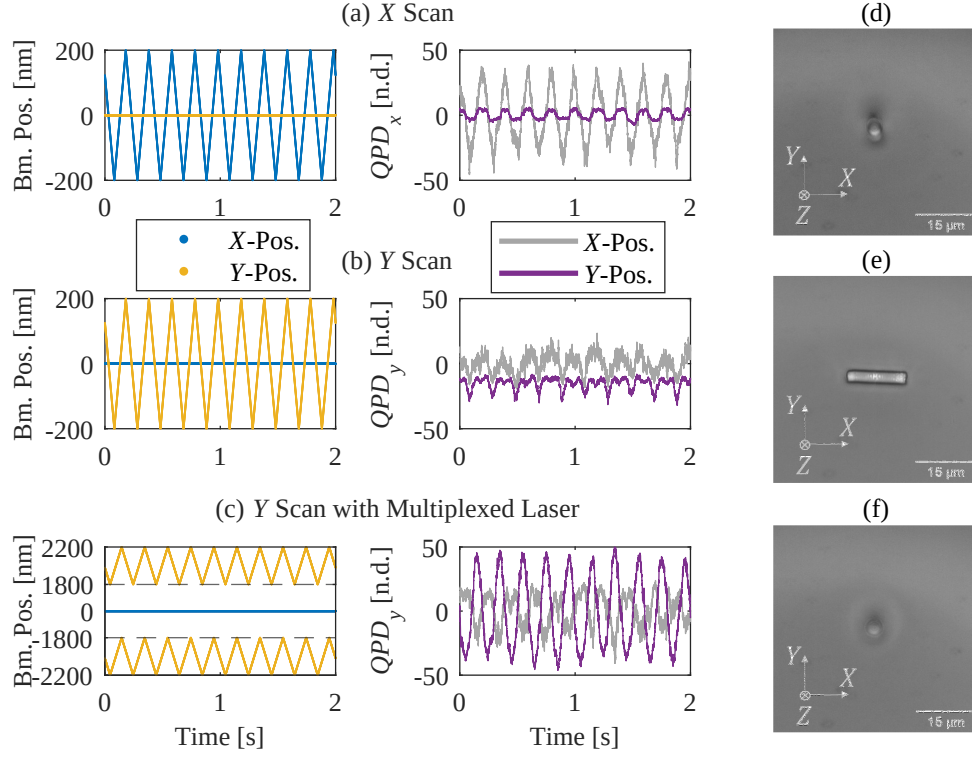


Figure 4.1: Scan Data a single micro-rod. Sub-figures (a) and (b) show the results of scanning the micro-rod shown in sub-figure (d) along the X and Y axes, respectively. Sub-figure (f) shows the same micro-rod's orientation when the beam is multiplexed ± 2000 nm along the Y -axis on either side of the micro-rod. Y -axis scan data for the micro-rod when held with the multiplexed laser is included in sub-figure (c). Sub-figure (e) shows the micro-rod held in the horizontal orientation with two traps near its ends. The beam visible in the center of the horizontal micro-rod is the detection laser whose trapping strength is negligible.

orientation aligned with the X axis shown in sub-figure (e), $|QPD_x| = -35.6$ n.d. Therefore, there is a $\Delta QPD_x = -35.7$ n.d between the micro-rod's vertical and horizontal orientations. The forthcoming results section shows that the same correlation applies to QPD_y when the micro-rod rotates about the X axis.

4.2 Controller Design

The automated learning of non-spherical particle manipulation is accomplished using two controllers. The first is the IL controller, which learns the system input needed to achieve micro-rod rotation. The second is a feedback controller which works in real-time while the micro-rod is

manipulated to stabilize it at the center of the trap. This section outlines how the chosen controller designs achieve these tasks.

4.2.1 Iterative Learning Controller

The application of the IL control concept to the rotation of a micro-rod is illustrated in the diagram of Figure 4.2. In this figure, the time-series reference trajectory is shown as $QPD_{x,ref}$. This represents the desired QPD_x signal associated with a successful rotation. The iteration length n and the choice of reference trajectory are part of the controller design process which is described later in this section. The “Plant” in the diagram represents the feedback-stabilized, optically trapped micro-rod. The inputs to the plant are the two beam X -coordinates, $\beta_{1,x}$ and $\beta_{2,x}$, that the laser is multiplexed between. The output is the QPD_x signal. At the beginning of each iteration the micro-rod is in the vertical position. As shown, for the first iteration the two beam X -coordinates, $\beta_{1,x}$ and $\beta_{2,x}$, are equal to zero for all n time-instances. This null feedforward input is applied to the plant resulting in time-series QPD_x data. Once the first iteration is over, the QPD_x time-series data is subtracted from the reference trajectory time-series data to produce the error time-series data e . The error time-series data and the beam X -position time-series data are fed to the IL controller. The IL controller uses this data to produce the new feedforward beam X -coordinates for iteration #2. As the feedforward input is updated by the IL controller, the micro-rod will begin to rotate and the error will decrease. Convergence to a feedforward input that rotates the micro-rod is illustrated in iteration #3. Note that in the diagram of Figure 4.2 and forthcoming diagrams for this section that the configuration for XZ -plane rotation is shown. Both the IL controller design and the forthcoming feedback controller design are the same for YZ -plane rotation, but with QPD_y , $\beta_{1,y}$, $\beta_{2,y}$ in place of QPD_x , $\beta_{1,x}$, and $\beta_{2,x}$.

The IL control law defines how the IL controller “learns” from its inputs and converges on an effective feedforward signal. Different types of IL control laws exist, with the proportional IL control law [84] chosen here as it does not require a dynamic system model. The proportional IL control law is included as Equation (4.1). In this equation, the beam position $\beta_{1,x}$ for iteration $j + 1$ is defined as a function of the beam position $\beta_{1,x}$ and error e from iteration j :

$$\begin{aligned}\beta_{1,x}^{(j+1)}(i) &= \beta_{1,x}^{(j)}(i) - K_{ilc}e^{(j)}(i + 1 + n_{adv}), \\ \beta_{2,x}^{(j+1)}(i) &= -\beta_{1,x}^{(j+1)}(i),\end{aligned}\tag{4.1}$$

for all time instances $i \in 0, \dots, n - 1$. As shown in the equation, the X -coordinate of the second beam position, $\beta_{2,x}$, is the negative of the first, $\beta_{1,x}$. This suits the symmetry of the control problem as the micro-rod is meant to be maintained at the origin during rotation. The scalar

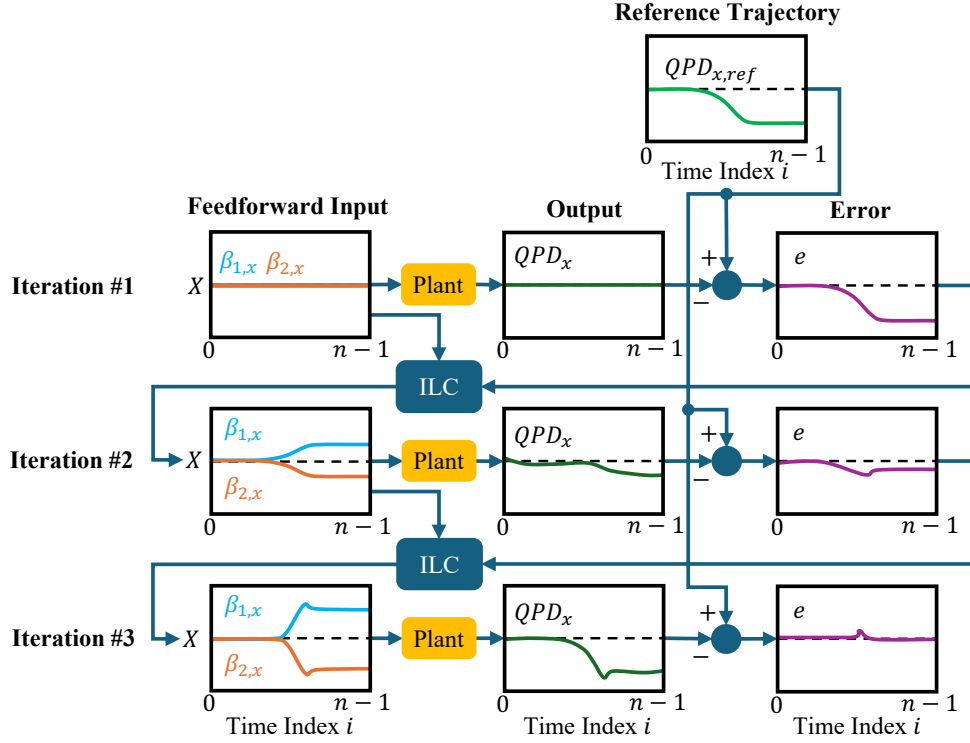


Figure 4.2: ILC concept diagram. The Plant block, shown in yellow, represents the micro-rod manipulations and feedback control that occur in real-time during an iteration. The dark blue blocks and arrows, including the ILC block, represent calculations performed offline in-between iterations.

coefficient $K_{ilc} \in \mathbb{R}$ tunes the proportional action of the controller. This coefficient controls how strongly the input for the next iteration is affected by the error of the previous iteration. For K_{ilc} values that are too high, the Brownian motion of the micro-rod will strongly affect the next iteration's beam position. This can result in learning an input that makes the rotation unstable. For K_{ilc} values that are too low, the learning process will take an unnecessarily long amount of time. Therefore, K_{ilc} is tuned to balance the stability of the learning process's convergence with the amount of time the IL trial takes. It was found that K_{ilc} values in the range of $5 \leq K_{ilc} \leq 7$ achieve a good balance between the stability of the convergence and the trial length. The scalar term $n_{adv} \in \mathbb{Z}^+$ controls what is essentially the amount of error preview. For non-zero, positive values of n_{adv} , the error from later time instances affects the beam position at earlier time instances. This is useful in situations where the system response and therefore the error lags behind the input to the system. As will be discussed in Section 4.3, for YZ -plane rotation error preview is not necessary ($n_{adv} = 0$). This is because the micro-rod's

initial position (as exemplified in Figure 4.1) is aligned favorably for rotation in the YZ -plane. For XZ -plane rotation, there is a lag between the system input and micro-rod rotation due to the initial position not being favorably aligned for XZ -plane rotation. This is addressed by using a value of n_{adv} in the range of $5000 \leq n_{adv} \leq 7000$ ($0.0625 \text{ s} \leq n_{adv} \leq 0.0875 \text{ s}$). Values of n_{adv} in this range produce similar results for the XZ -plane IL process.

Additional key design parameters for the IL controller are the reference trajectory and iteration time length n . At a minimum the iteration time length n must allow enough time for the chosen manipulation to occur, in this case micro-rod rotation. The iteration time length should also include time after the rotation occurs, to show that the new horizontal orientation is stable. It is also good for the iteration time length to include time before the manipulation happens when the micro-rod should be vertical. If the IL controller does not learn any spurious inputs during this initial time period when the micro-rod should be vertical, it indicates good controller performance. For these experiments, a time length of six seconds was found to be sufficient for these goals. The reference trajectory at its core should be a time-series signal that traces a ramp from the vertical signal magnitude, $QPD_{vert.}$, to the horizontal signal magnitude, $QPD_{horiz.}$, which are separated by ΔQPD . However, the reference trajectory will be optimized beyond a ramp shape.

For optimal IL control, it is helpful to have different reference trajectories for the XZ and YZ -plane IL processes. This is due to the asymmetry of the initial orientation of the micro-rod, as shown in sub-figure (d) of Figure 4.1. The initial, slight tilt of the micro-rod favors rotation in the YZ -plane. Consequently, rotation is achieved more easily in this plane and a simple reference trajectory, pictured in sub-figure (b) of Figure 4.3, suffices. This reference trajectory, which represents the desired QPD_y time-series data, is essentially a ramp from $QPD_{y,vert.} = 0$ to a nominal $QPD_{y,horiz.}$ value separated by $\Delta QPD_{y,nom.}$. As mentioned in Section 4.1.1, the nominal reference trajectory is calibrated at the beginning of the experiment by multiplying it by a scalar $\gamma_y \in \mathbb{R}$ such that $\Delta QPD_y = \gamma_y \Delta QPD_{y,nom.}$. The forthcoming IL results will show that the slight hump added to the reference trajectory between $t = 1.97$ and 3.15 seconds results in it better matching the QPD_y signal of a successful rotation. This is an example of an iterative design improvement that can be made across IL trials to the reference trajectory to improve convergence.

The reference trajectory for the XZ -plane trials also has an example of a design improvement made to improve the convergence of the IL process. The initial tilt in the YZ -plane means that if the micro-rod is to rotate within the XZ -plane, it must first be brought into that plane. This causes a delay between feedforward beam position inputs meant to rotate the micro-rod and actual rotation. This can be partly solved by the aforementioned error preview tuning parameter n_{adv} . However, if n_{adv} becomes too large, the IL process will become unstable. Reliance on a

large n_{adv} can be avoided by using a well designed reference trajectory. Pictured in sub-figure (a) of Figure 4.3, the reference trajectory for the XZ -plane IL process features two prominent spikes at times $t = 2.38$ and $t = 2.85$ seconds. These spikes in the reference trajectory will cause the IL controller to learn two spikes in the feedforward beam position, as will be shown in Section 4.3. These spikes will get the micro-rod into the XZ -plane and oscillating before rotation is commanded. The rest of the reference trajectory design is essentially a ramp from $QPD_{x,vert} = 0$ to a nominal $QPD_{x,hORIZ.}$ value. Just as for the YZ -plane reference trajectory, the nominal XZ -plane reference trajectory is scaled by $\gamma_x \in \mathbb{R}$ at the beginning of the experiment such that $\Delta QPD_x = \gamma_x \Delta QPD_{x,nom.}$.

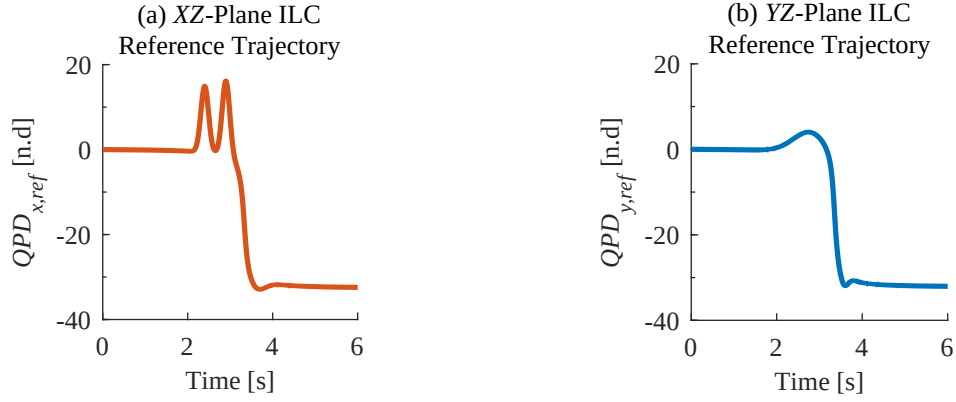


Figure 4.3: The reference trajectories for the IL processes. Sub-figure (a) contains the reference trajectory for the XZ -plane IL process. Sub-figure (b) contains the reference trajectory for the YZ -plane IL process. The shown magnitudes of the reference trajectories as a function of time are the nominal values. Both reference trajectories are scaled during the calibration process to match the detection system.

During experiments, the convergence of the IL process is tracked by calculating the average percent steady state error for each iteration. The steady-state is defined as the time period from $t = 4$ to $t = 6$ seconds when the micro-rod should be horizontal. For XZ -plane IL trials, the average percent steady-state error for an iteration j is defined as follows:

$$e_{ss}^{(j)} = \frac{1}{N_{ss}} \sum_{i=n_4}^{n_6} 100 * \frac{QPD_{x,ref}(i) - QPD_x^{(j)}(i)}{QPD_{x,ref}(i)}, \quad (4.2)$$

where $n_4 = \lfloor 4/\Delta t \rfloor$, $n_6 = \lfloor 6/\Delta t \rfloor$, $N_{ss} = n_6 - n_4 + 1$, and Δt is the sampling period. The equation for the average percent steady-state error is the same as (4.2) for the YZ -plane IL trials but with QPD_y and $QPD_{y,ref}$ in place of QPD_x and $QPD_{x,ref}$, respectively. The IL process has converged when the average percent steady state error of an iteration has reached

zero.

4.2.2 Feedback Controller

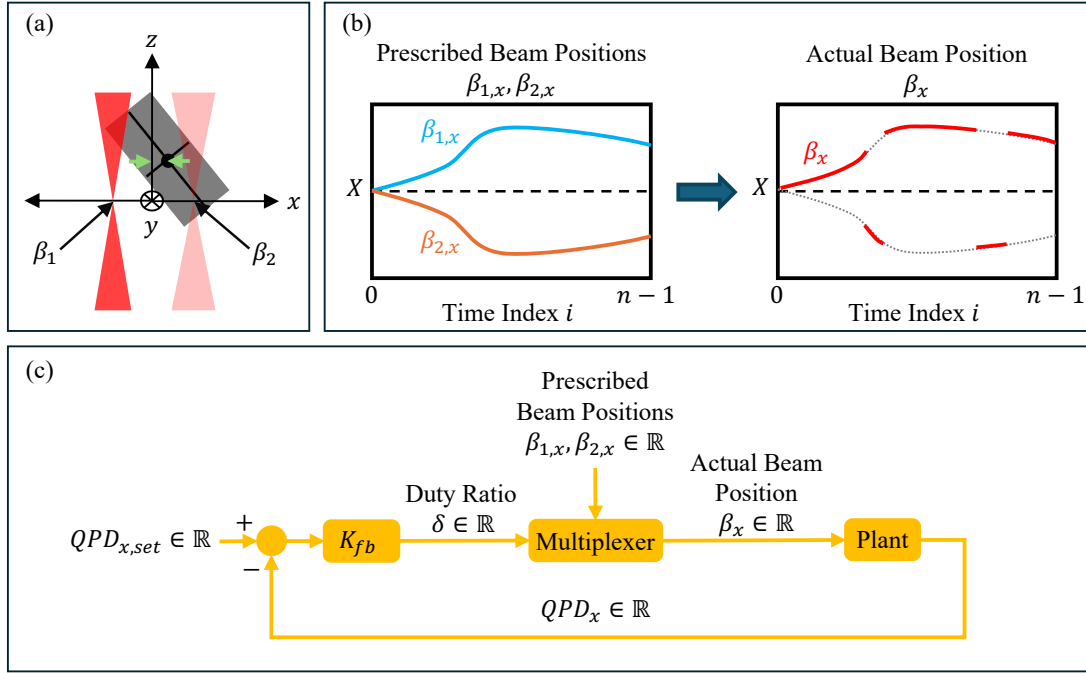


Figure 4.4: Feedback Control illustrations. In sub-figure (a), the issue of the micro-rod sliding towards one of the two beams is illustrated. The use of altering the beam multiplexing duty ratio to counter-act this is shown in sub-figure (b). Sub-figure (c) shows the diagram of the feedback controller that accomplishes this duty-ratio alteration in real-time.

While the iterative learning process is carried out the micro-rod should be maintained at the coordinate system origin. This stabilizes the IL process and ensures there are no unintended flights from the origin. The necessity of feedback control is due to the micro-rod's tendency to slide into one of the two traps while an IL iteration's feedforward signal attempts to rotate it. This situation is illustrated in sub-figure (a) of Figure 4.4. In this illustration, the two beam positions β_1 and β_2 are ramped away from one another along the X -axis. If the micro-rod begins to rotate, it will make greater contact with one of the two beams. In this case, the micro-rod is making greater contact with the beam at location β_2 . This would cause the micro-rod to be pulled to the right until it solely occupies the trap at β_2 . This results in a failed rotation.

The feedback controller counteracts this tendency by adjusting the relative power of the two beams. In the case illustrated in sub-figure (a) of Figure 4.4, the feedback controller would act

to decrease the power of the beam at location β_2 and increase the power of the beam at location β_1 . This would pull the micro-rod away from the beam at β_2 and back to the origin. The correction of the two beam powers is accomplished by manipulating the beam multiplexing duty ratio $\delta \in \mathbb{R}$. The alteration of the duty ratio between the two beam positions is illustrated in sub-figure (b). Adjusting the duty ratio is done in real-time via proportional feedback control. The diagram for the feedback controller is included in sub-figure (c). For the case of XZ -plane rotation experiments, the feedback signal is the QPD_x signal. As discussed in Section 4.1.2, this signal corresponds to the micro-rod's position X -coordinate when its in the vertical or near vertical orientation. The position error is calculated as the difference between the set point $QPD_{x,set} \in \mathbb{R}$ and $QPD_x(i) \in \mathbb{R}$ for each time instance i . The error is multiplied by the feedback gain K_{fb} to get the multiplexing duty ratio $\delta(i) \in \mathbb{R}$ at time instance i . For the case of YZ -plane experiments, the feedback signal is the QPD_y signal with the setpoint $QPD_{y,set}$. In both the XZ -plane and YZ -plane rotation experiments, $K_{fb} = 0.009$ was found to stabilize the micro-rod at the origin.

4.2.3 Overall Controller Design

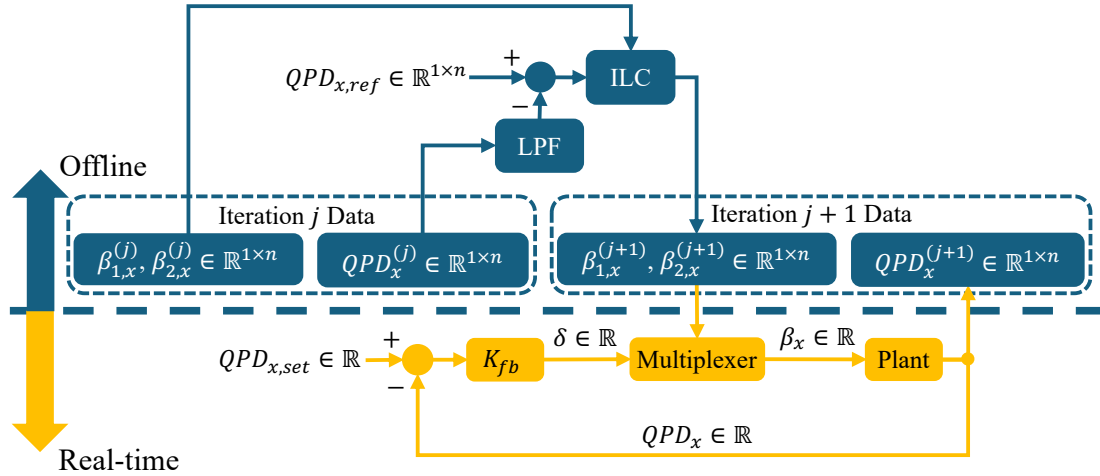


Figure 4.5: Complete Controller Diagram. The blocks in dark blue represent the calculations performed in between iterations. The blocks in yellow represent the calculations happening in real-time while the micro-rod is manipulated during an iteration.

During experiments, the IL controller and the feedback controller work together. A diagram illustrating the overall controller design is included as Figure 4.5. The dashed line separates what is done offline in between IL iterations, and what is performed in real-time while the

iteration is carried out. The offline section of the diagram consists of the IL controller which acts on the beam position $\beta_{1,x}^{(j)}$ and $QPD_x^{(j)}$ data from the past iteration j to produce the beam position data $\beta_{1,x}^{(j+1)}, \beta_{2,x}^{(j+1)}$ data for the next iteration $j + 1$. During the iteration, as shown in the “real-time” section of the diagram, the beam position data is fed to the plant while the feedback controller adjusts the duty ratio δ at every time step. Together these controllers ensure a feedforward beam position input for micro-rod rotation is learned.

4.3 Results

To verify the performance of the controller design there are a four points of verification. The first point of verification is confirming the controller works on an individual trial basis. This is done by examining in detail the results of a single trial. The second point of verification is that the controller performance is consistent when reapplied to the same micro-rod in back-to-back trials. To demonstrate this consistency, three back-to-back trials performed on the same micro-rod are shown. Third, the controller should perform well across all micro-rods of nominal size and perform consistently from day-to-day. To show the consistency across micro-rods and time, results from five different micro-rods each taken on a different day are included. Lastly, the utility of learned feedforward input is demonstrated by applying it to rotate a different micro-rod on a different day. Taken together these results will show the consistent and reliable performance of the controller.

To begin, detailed results for a single IL trial for each direction of rotation are examined. The detailed results for XZ -plane rotation are included in Figure 4.6, with analogous results for YZ -plane rotation in Figure 4.7. In the sub-figures for the feedforward inputs and the resulting QPD signals, results for individual iterations are plotted super-imposed over one another. These superimposed time-series data are plotted on a color gradient that indicates the corresponding iteration number. In both figures, the convergence of the IL process is shown by plotting the average percent steady state error, defined in (4.2), as a function of the iteration number.

The detailed XZ -plane trial results in Figure 4.6 demonstrate the iterative nature of the learning process. The key signals are the X -coordinate of beam one, the QPD_x signal, and the $QPD_{x,ref}$ reference trajectory. Starting at iteration one, beam one (and therefore also beam two) remain at the origin for the entire duration of the iteration. Applying this beam pattern to the micro-rod results in a QPD signal that remains near zero and causes a relatively high steady-state error. This error causes the beam one position for iteration two to grow in magnitude. Each iteration results in an error that spurs the further evolution of the feedforward beam position input. As the trial progresses through more iterations, the magnitude of the error decreases. After iteration 70, the subsequent feedforward inputs become very tightly spaced on

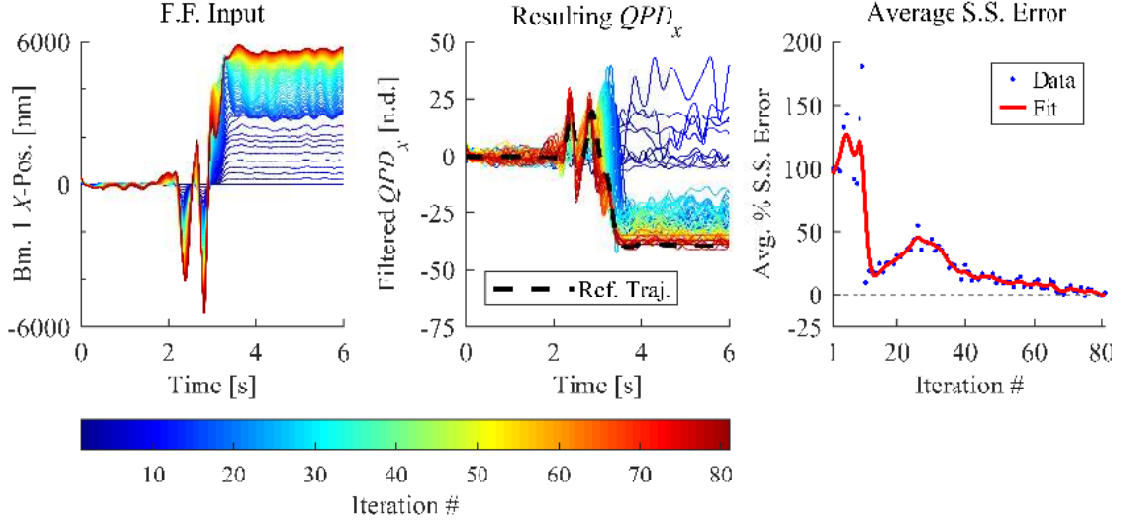


Figure 4.6: Detailed results for one XZ -plane IL trial. In the first sub-figure, the feedforward (F.F.) input (Beam 1 X -Pos.) for all iterations are plotted super-imposed over each other. In each iteration the F.F. input is applied to the micro-rod, with the resulting QPD_x signal plotted in the second-subfigure. A color gradient is used to indicate the iteration number of each corresponding F.F. and QPD_x time-series. The reference trajectory $QPD_{x,ref}$ is super-imposed over the QPD_x signals in the second sub-figure. The third sub-figure shows the average percent steady-state (S.S.) error for each iteration. A smoothing spline is fit to this error data with a smoothing parameter equal to 0.5.

the F.F. plot as the error converges to zero. By iteration 80, the average percent steady-state error has converged to zero. This confirms a feedforward beam position input that results in rotation has been learned.

As shown in the steady-state error plot of Figure 4.6, the progression of error across iterations is not linear. From iteration zero to ten, the result of the feedforward input from $t = 4$ to $t = 6$ seconds is to stand the micro-rod up straight, as exemplified in sub-figure (f) of Figure 4.1. This more upright orientation of the micro-rod results in a QPD_x magnitude greater than zero for $t = 4$ to $t = 6$ seconds and therefore higher error. From iteration 10 to 30, the growing feedforward input causes the cylinder to begin rotating through small angles. However, as discussed in Section 4.1.2, for small angles the QPD signal is still correlated to the micro-rod's position. Therefore QPD_x and in turn, the error, are not a pure representation of the micro-rod's angle during these iterations. Starting from iteration 30 onward, the larger feedforward signal magnitude from $t = 4$ to $t = 6$ seconds achieves larger and larger micro-rod rotation angles. At this point QPD_x is less correlated to the micro-rod's X -position and more reflects

its angle in the XZ -plane. Therefore for iteration 30 onward, QPD_x and the steady-state error more closely match the progression of the micro-rod's angle.

Figure 4.6 also illustrates the utility of the chosen $QPD_{x,ref}$ reference trajectory. The two “spikes” in the reference trajectory at points $t = 2.39$ and $t = 2.89$ seconds create error at those points during early iterations. This error leads to the formation of the two negative spikes in the X -position of beam one at $t = 2.38$ and $t = 2.85$ seconds. The shift forward in time of the formed spikes is thanks to the preview tuning parameter n_{adv} of the IL controller. Since the X -position of beam two is the mirror image of beam one on the X -axis, it also has two spikes. The two spikes in the beams' X -positions work first to get the micro-rod standing up straight, like what is shown in sub-figure (f) of Figure 4.1. Then the action of the spikes gets the micro-rod rotating back and forth in-plane. This pre-rotation action solves the problem of the initial micro-rod orientation, as shown in sub-figure (d) of Figure 4.1, being unfavorably aligned for XZ -plane rotation. With the micro-rod rocking back-and-forth in the XZ -plane, it can quickly respond to rotation inputs. This allows it to quickly rotate to horizontal and in doing so, closely follow the reference trajectory from $t = 3.1$ to $t = 3.7$ seconds. This tight reference trajectory tracking makes the IL process more repeatable.

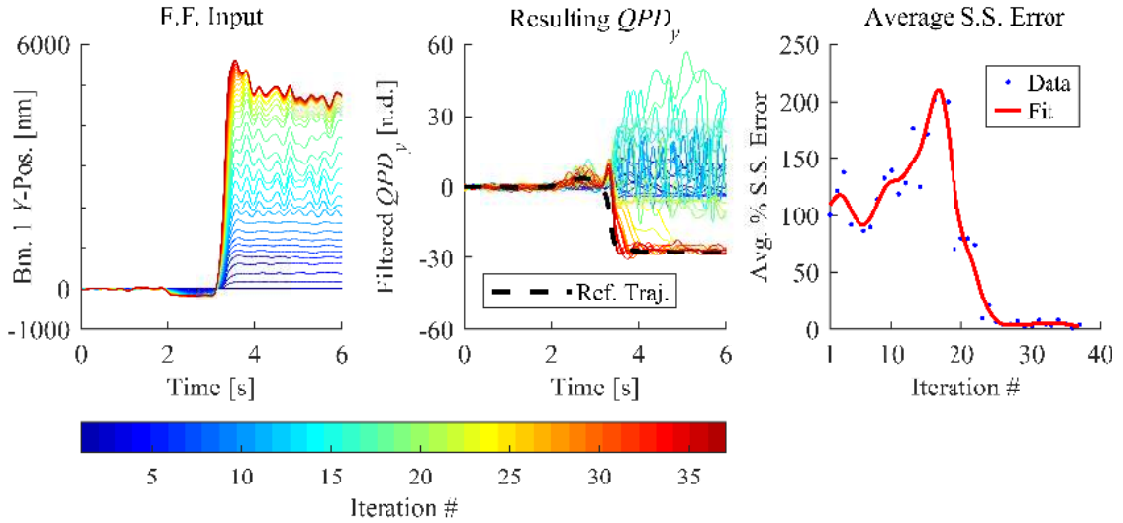


Figure 4.7: Detailed results for one YZ -plane IL trial. The format is the same as used for Figure 4.6 but with Beam 1 Y -position, QPD_y , and $QPD_{y,ref}$ in place of Beam 1 X -position, QPD_x , and $QPD_{x,ref}$. The color gradient indicates which iterations the data in the first two sub-figures belong to. As before, a smoothing spline with smoothing parameter 0.5 is fit the average percent steady-state error data.

For YZ -plane rotation the learning process is more straightforward. The relevant signals

in this case are the Y -coordinate of beam one, the QPD_y signal, and the $QPD_{y,ref}$ reference trajectory. Detailed results for an individual YZ -plane IL trial are shown in Figure 4.7. The format of these results is the same as used for the XZ -plane results in Figure 4.6. Just as before, the feedforward input for the first iteration is the null signal and the micro-rod remains vertical for the duration of the iteration. This error and the error of subsequent iterations grows the feedforward beam position input. As in the XZ -plane case, the steady-state error increases across the early iterations as those feedforward inputs stand the micro-rod more upright. Starting around iteration 20, when the feedforward beam one position exceeds 4000 nm between $t = 4$ and $t = 6$ seconds, the cylinder will rotate to intermediate angles. As the steady-state error plot shows, the transition from these intermediate angles to fully horizontal happens quickly. The horizontal angle is achieved by iteration 26, with tight error convergence reached by iteration 36.

The YZ -plane results in Figure 4.7 show an effective feedforward input being learned in less iterations than needed for the XZ -plane case of Figure 4.6. This is due to the initial micro-rod orientation, pictured in sub-figure (d) of Figure 4.1, being favorable to rotation in the YZ -plane. This favorable initial orientation is also why a more simple reference trajectory is used for the YZ -plane case. The engineered spikes in the reference trajectory for XZ -plane experiments are not needed. Really the only feature of interest is the slight hump in the positive Y -direction between $t = 1.97$ and $t = 3.15$ seconds. As mentioned in Section 4.2, this feature was added to improve convergence. In later iterations when the cylinder is rotating, the resulting QPD_y signals match the shape of this hump. Without the addition of this hump, the mismatch between $QPD_{y,ref}$ and QPD_y would result in poorer convergence of the learned feedforward input between the points $t = 1.97$ and $t = 3.15$ seconds.

While the individual trial results establish the functionality of the controller, repeatability is also important. To show repeatability, the results of three repeat IL trials performed in succession on a micro-rod are compared. Results for the XZ -plane case are presented in Figure 4.8. In this figure, the final results of each of the three trials are shown. This includes the three learned feedforward beam X -position inputs and the corresponding QPD_x signals they produced. All QPD_x signals track the reference trajectory, indicating the micro-rod became horizontal all three times. The learned feedforward inputs are tightly grouped, with those pre-rotation spikes in beam position learned each time. Additionally the QPD_x signals show repeatable results from the feedforward inputs. The same repeatability experiment was performed for the YZ -plane IL process. The results of the three IL trials are included in Figure 4.9. Just as for the XZ -plane results, the three learned feedforward inputs and their corresponding QPD_y signals are tightly grouped. The QPD_y signals track the reference trajectory, indicating successful rotation. Taken together, these results indicate that the controller performance is repeatable for

a given micro-rod.

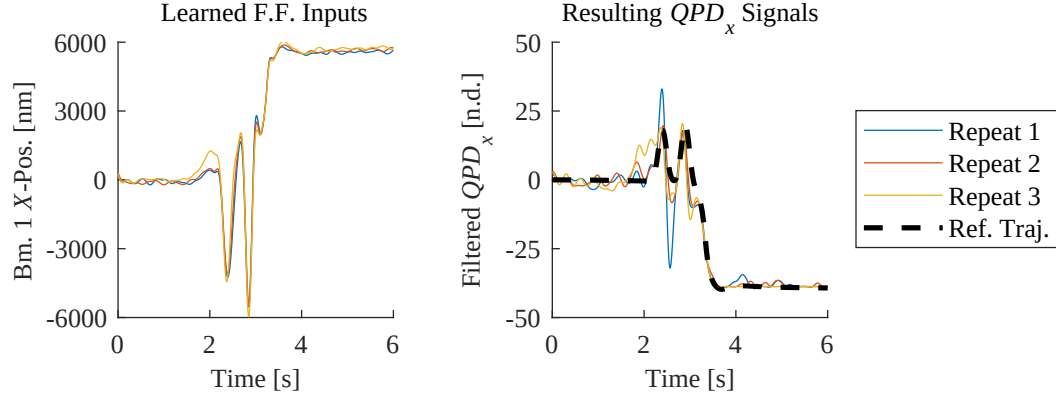


Figure 4.8: Three repeat XZ -plane IL trials performed in succession on the same micro-rod. The left sub-figure shows the three feedforward inputs learned by the three trials. The right sub-figure shows the corresponding QPD_x signals that the learned feedforward inputs produced. The reference trajectory $QPD_{x,ref}$ is plotted superimposed over the QPD_x signals to show that the learned inputs achieve micro-rod rotation.

Repeatability beyond a given micro-rod is also important. The controller should work across experiments and across different days. To this end, IL trial results are shared for five different micro-rods, each taken on a different day. The batch of five results for the XZ -plane case are shown in Figure 4.10, with Figure 4.11 containing analogous results for the YZ -plane case. These figures include the learned feedforward inputs from each trial as well as the corresponding QPD signals the inputs produced. Since the QPD alignment will differ slightly from one experiment to the next, the values of ΔQPD_x and ΔQPD_y will also vary. To facilitate the comparison of QPD signals taken on different days, the QPD signals are normalized such that the steady state horizontal QPD signal is equal to negative one. As apparent in Figures 4.10 and 4.11, on each day a feedforward input was learned that produces a QPD signal closely matching the reference trajectory, indicating successful rotation. The XZ -plane results in Figure 4.10 show that the same profile of feedforward beam input is learned with the two spikes that help prepare the micro-rod for rotation. The feedforward beam inputs learned for the YZ -plane case also closely match one another. In both the XZ and YZ -plane cases, variations in the steady state value of the learned feedforward input can be attributed to variations in the length of the micro-rods. For shorter micro-rods, shorter beam displacements achieve the rotation to horizontal. The other small variations seen in both the learned feedforward beam position inputs and the QPD signals can be attributed to the Brownian motion of the micro-rod and experimental factors such as the feedback tuning on the day of the experiment.

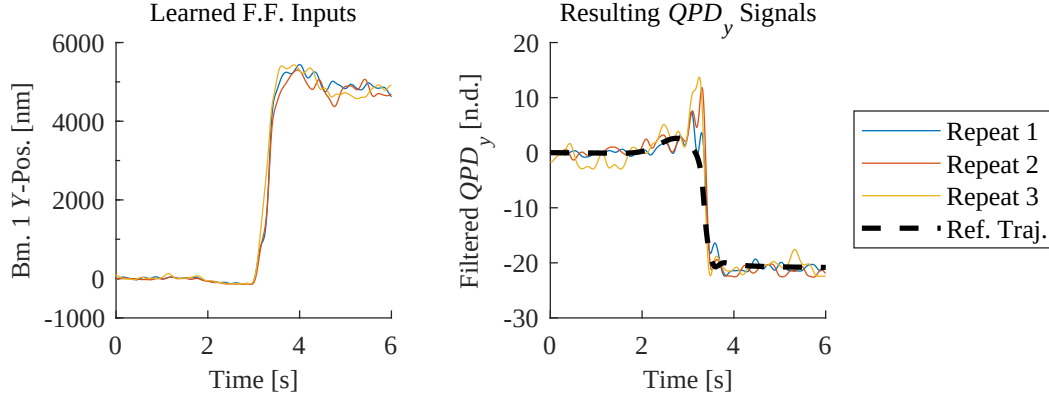


Figure 4.9: Three repeat YZ -plane IL trials performed in succession on the same micro-rod. The three learned feedforward inputs are plotted in the left sub-figure, with the corresponding QPD_y signals plotted in the right sub-figure. The super-imposed reference trajectory $QPD_{y,ref}$ shows that the learned feedforward inputs produce micro-rod rotation.

Having shown the repeatability of the IL process, the final point of verification is the utility of the learned feedforward input. Once learned it should work to rotate other micro-rods on different days. This shown by taking a learned feedforward input from a past experiment on a past day and using it rotate a new micro-rod. Results for this are included in Figure 4.12. In sub-figure (a) are the results of applying the pictured XZ -plane rotation input to a micro-rod ten times. Sub-figure (b) contains the analagous results for applying the pictured YZ -plane rotation input to a micro-rod ten times. In both cases, the QPD signals show rotation to the horizontal position every time. This indicates that the learning process has produced a feedforward input that is a reusable tool for micro-rod manipulation.

4.4 Conclusion

This chapter has shown the successful experimental application of the IL control design. This was accomplished by using the optical tweezers QPD detection system to discern both the micro-rod's position and orientation. Asymmetry in the micro-rod's initial position is shown to not be an issue for the IL controller as successful manipulation beam patterns are learned for rotation about the X axis and the Y axis of optical tweezers' actuation coordinate system. The performance of the IL process was shown to be consistent when applied repeatedly to the same cylinder. Additionally, results across different days on different cylinders show that the IL controller performance is repeatable and robust. Lastly, the utility of the learned manipulation beam pattern is shown by applying it to a new cylinder to successfully and reliably rotate it.

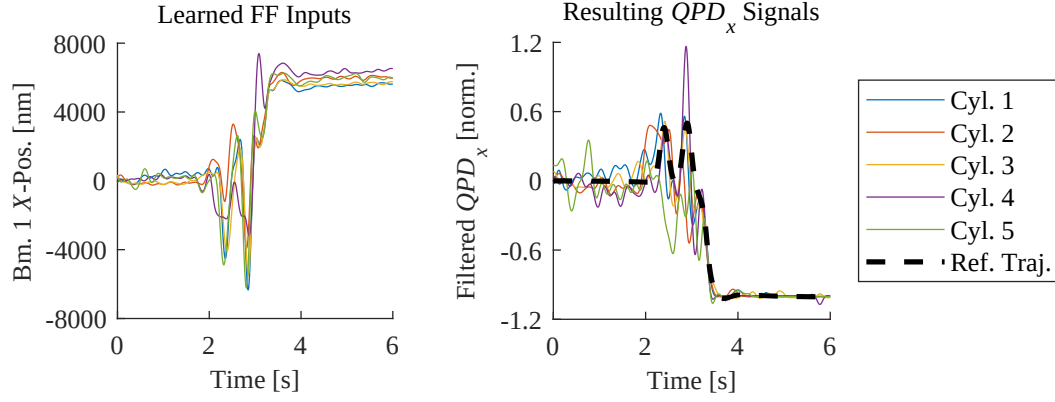


Figure 4.10: XZ -plane IL trial results from five different micro-rods on five different days. The left sub-figure shows the learned feedforward inputs from the five trials. The corresponding QPD_x signals are included in the right sub-figure. As implied by the overlaid $QPD_{x,ref}$ reference trajectory, all of the five learned inputs achieve micro-rod rotation.

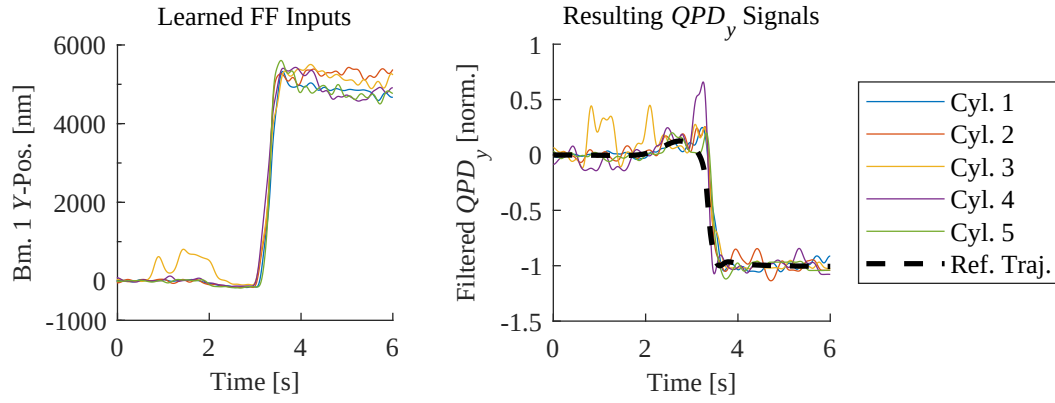


Figure 4.11: YZ -plane IL trial results from five different micro-rods on five different days. The left sub-figure shows the learned results of the IL trials, the second sub-figure shows the corresponding QPD_y signals which successfully track the $QPD_{y,ref}$ reference. This indicates successful rotation.

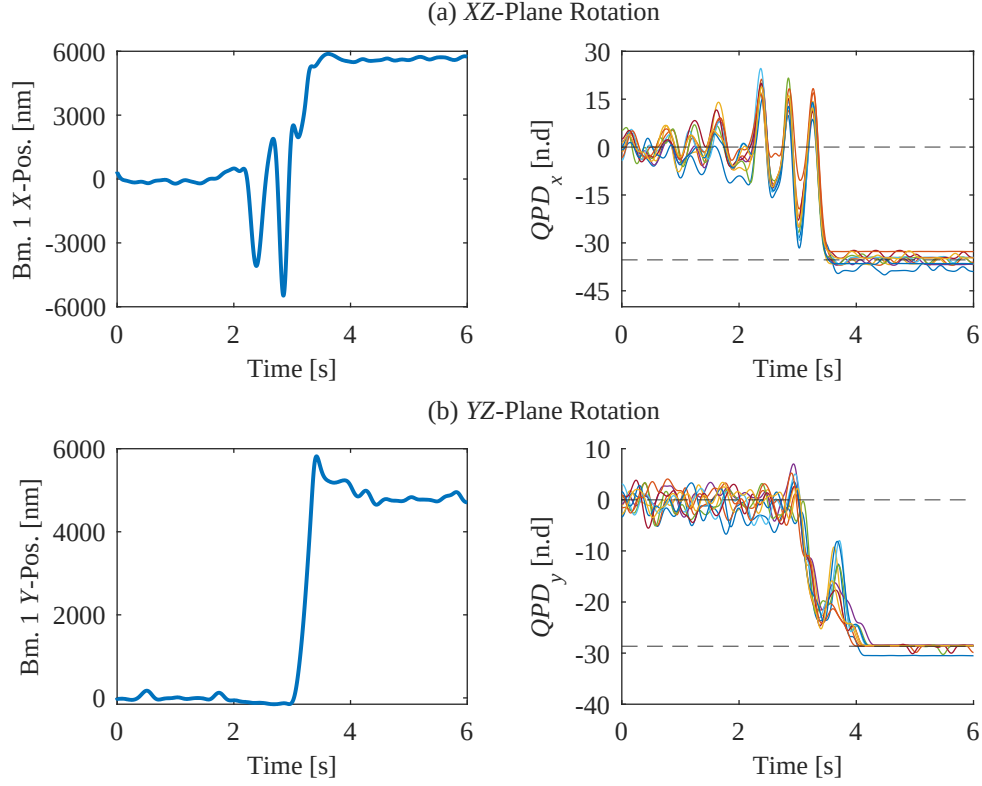


Figure 4.12: Micro-rod rotation test using a learned feedforward input from a previous experiment on a past day. Sub-figure (a) shows the results of applying the pictured learned XZ -plane rotation feedforward input to a micro-rod ten times. In the plot for QPD_x , horizontal dashed lines indicate the QPD_x magnitudes associated with the vertical and horizontal orientations. All ten QPD_x signals reach the horizontal value indicating successful rotation. Sub-figure (b) follows the same format, but for YZ -plane rotation. The learned feedforward input is applied ten times, and as seen in the QPD_y plot, rotation is achieved all ten times.

Chapter 5

Conclusion & Future Directions

The manipulation of non-spherical particles is an important use case for optical tweezers that has, until now, been a difficult subject for modern control methods. Extending modern control's success from optically trapped spherical particles to non-spherical particles has been hampered by the difficulty of deriving a dynamic model suitable for real-time use. Here this problem has been circumvented through the use of iterative learning control. Simulation results showed the promise and viability of the proposed IL controller design in achieving the control goal of out-of-plane cylinder rotation. To extend this verification to actual experiments, an optical tweezers system was built and this process was recounted in detail. The experiments performed on this optical tweezers system successfully verified the performance of the IL controller which reliably and repeatedly learned beam motion patterns to achieve out-of-plane cylinder rotation.

Now that the application of IL control to the manipulation of a non-spherical particles has been proven, there exist many new future directions. A natural next step is to extend this control design to the manipulation of particles that have no axis of symmetry. The key challenge here would be extracting enough information about such a particle's orientation from the QPD data. This leads into the next future direction, which is the application of image-based detection of the particle's orientation in place of QPD-based detection. This would allow the IL controller to learn more complex maneuvers on more complex shapes. Regardless of what direction the work takes next, IL control's potential to improve the productivity and capability of optical trapping applications targeting non-spherical particles is clear.

References

- [1] Arthur Ashkin. Acceleration and trapping of particles by radiation pressure. *Physical review letters*, 24(4):156, 1970.
- [2] Arthur Ashkin, James M Dziedzic, John E Bjorkholm, and Steven Chu. Observation of a single-beam gradient force optical trap for dielectric particles. *Optics letters*, 11(5):288–290, 1986.
- [3] Philip H Jones, Onofrio M Maragò, and Giovanni Volpe. *Optical tweezers: Principles and applications*. Cambridge University Press, 2015.
- [4] Agnese Callegari, Mite Mijalkov, A Burak Gököz, and Giovanni Volpe. Computational toolbox for optical tweezers in geometrical optics. *JOSA B*, 32(5):B11–B19, 2015.
- [5] Keir C Neuman and Steven M Block. Optical trapping. *Review of scientific instruments*, 75(9):2787–2809, 2004.
- [6] Timo A Nieminen, Alexander B Stilgoe, Norman R Heckenberg, and Halina H Rubinsztein-Dunlop. Approximate and exact modeling of optical trapping. In *Optical Trapping and Optical Micromanipulation VII*, volume 7762, pages 509–516. SPIE, 2010.
- [7] TA Nieminen, H Rubinsztein-Dunlop, and NR Heckenberg. Multipole expansion of strongly focussed laser beams. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 79:1005–1017, 2003.
- [8] Timo A Nieminen, Nathaniel du Preez-Wilkinson, Alexander B Stilgoe, Vincent LY Loke, Ann AM Bui, and Halina Rubinsztein-Dunlop. Optical tweezers: Theory and modelling. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 146:59–80, 2014.
- [9] Peter C Waterman. Matrix formulation of electromagnetic scattering. *Proceedings of the IEEE*, 53(8):805–812, 1965.

- [10] Michael I Mishchenko, Larry D Travis, and Andrew A Lacis. *Scattering, absorption, and emission of light by small particles*. Cambridge university press, 2002.
- [11] Vincent LY Loke, Timo A Nieminen, Simon J Parkin, Norman R Heckenberg, and Halina Rubinsztein-Dunlop. Fdfd/t-matrix hybrid method. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 106(1-3):274–284, 2007.
- [12] TA Nieminen, H Rubinsztein-Dunlop, and NR Heckenberg. Calculation of the t-matrix: general considerations and application of the point-matching method. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 79:1019–1029, 2003.
- [13] Edward M Purcell and Carlton R Pennypacker. Scattering and absorption of light by non-spherical dielectric grains. *Astrophysical Journal, Vol. 186, pp. 705-714 (1973)*, 186:705–714, 1973.
- [14] Daniel W Mackowski. Discrete dipole moment method for calculation of the t matrix for nonspherical particles. *JOSA A*, 19(5):881–893, 2002.
- [15] Vincent LY Loke, Timo A Nieminen, Norman R Heckenberg, and Halina Rubinsztein-Dunlop. T-matrix calculation via discrete dipole approximation, point matching and exploiting symmetry. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 110(14-16):1460–1471, 2009.
- [16] Xiaoqiong Qi, Timo A Nieminen, Alexander B Stilgoe, Vincent LY Loke, and Halina Rubinsztein-Dunlop. Comparison of t-matrix calculation methods for scattering by cylinders in optical tweezers. *Optics letters*, 39(16):4827–4830, 2014.
- [17] Isaac CD Lenton, Ann AM Bui, Timo A Nieminen, Alexander B Stilgoe, and Halina Rubinsztein-Dunlop. Optical tweezers toolbox: full dynamics simulations for particles of all sizes. In *Optical Trapping and Optical Micromanipulation XV*, volume 10723, page 107232B. International Society for Optics and Photonics, 2018.
- [18] Desmond J Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM review*, 43(3):525–546, 2001.
- [19] Don S Lemons and Paul Langevin. *An introduction to stochastic processes in physics*. JHU Press, 2002.
- [20] Rabi Bhattacharya, Edward C Waymire, Rabi Bhattacharya, and Edward C Waymire. The functional central limit theorem (fclt). *Random Walk, Brownian Motion, and Martingales*, pages 199–214, 2021.

- [21] Benjamin A Stickler, Ewald Schachinger, Benjamin A Stickler, and Ewald Schachinger. The random walk and diffusion theory. *Basic Concepts in Computational Physics*, pages 271–295, 2016.
- [22] Albert Einstein et al. On the motion of small particles suspended in liquids at rest required by the molecular-kinetic theory of heat. *Annalen der physik*, 17(549-560):208, 1905.
- [23] Crispin Gardiner. *Stochastic methods*, volume 4. Springer Berlin Heidelberg, 2009.
- [24] Nicolaas Godfried Van Kampen. *Stochastic processes in physics and chemistry*, volume 1. Elsevier, 1992.
- [25] Kishan Dholakia, Peter Reece, and Min Gu. Optical micromanipulation. *Chemical Society Reviews*, 37(1):42–55, 2008.
- [26] Paolo Polimeno, Alessandro Magazzu, Maria Antonia Iati, Francesco Patti, Rosalba Saija, Cristian Degli Esposti Boschi, Maria Grazia Donato, Pietro G Gucciardi, Philip H Jones, Giovanni Volpe, et al. Optical tweezers and their applications. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 218:131–150, 2018.
- [27] Adam M Kaufman and Kang-Kuen Ni. Quantum science with optical tweezer arrays of ultracold atoms and molecules. *Nature Physics*, 17(12):1324–1333, 2021.
- [28] Onofrio M Marago, Philip H Jones, Pietro G Gucciardi, Giovanni Volpe, and Andrea C Ferrari. Optical trapping and manipulation of nanostructures. *Nature nanotechnology*, 8(11):807–819, 2013.
- [29] Carlos J Bustamante, Yann R Chemla, Shixin Liu, and Michelle D Wang. Optical tweezers in single-molecule biophysics. *Nature Reviews Methods Primers*, 1(1):25, 2021.
- [30] Marco Capitanio and Francesco S Pavone. Interrogating biology with force: single molecule high-resolution measurements with optical tweezers. *Biophysical journal*, 105(6):1293–1303, 2013.
- [31] Stella Corsetti and Kishan Dholakia. Optical manipulation: advances for biophotonics in the 21st century. *Journal of Biomedical Optics*, 26(7):070602–070602, 2021.
- [32] Karel Svoboda, Christoph F Schmidt, Bruce J Schnapp, and Steven M Block. Direct observation of kinesin stepping by optical trapping interferometry. *Nature*, 365(6448):721–727, 1993.
- [33] Jeffrey T Finer, Robert M Simmons, and James A Spudich. Single myosin molecule mechanics: piconewton forces and nanometre steps. *Nature*, 368(6467):113–119, 1994.

- [34] Arthur Ashkin and JM Dziedzic. Feedback stabilization of optically levitated particles. *Applied Physics Letters*, 30(4):202–204, 1977.
- [35] Robert M Simmons, Jeffrey T Finer, Steven Chu, and James A Spudich. Quantitative measurements of force and displacement using an optical trap. *Biophysical journal*, 70(4):1813–1822, 1996.
- [36] RM Simmonst, JT Finer, HM Warrick, B Kralik, S Chu, and JA Spudich. Force on single actin filaments in a motility assay measured with an optical trap. *Mechanism of myofilament sliding in muscle contraction*, pages 331–337, 1993.
- [37] JE Molloy, JE Burns, J Kendrick-Jones, RT Tregear, and DCS White. Movement and force produced by a single myosin head. *Nature*, 378(6553):209–212, 1995.
- [38] Michelle D Wang, Hong Yin, Robert Landick, Jeff Gelles, and Steven M Block. Stretching dna with optical tweezers. *Biophysical journal*, 72(3):1335–1346, 1997.
- [39] Michelle D Wang, Mark J Schnitzer, Hong Yin, Robert Landick, Jeff Gelles, and Steven M Block. Force and velocity measured for single molecules of rna polymerase. *Science*, 282(5390):902–907, 1998.
- [40] Koen Visscher and Steven M Block. [38] versatile optical traps with feedback control. In *Methods in enzymology*, volume 298, pages 460–489. Elsevier, 1998.
- [41] Matthew J Lang, Charles L Asbury, Joshua W Shaevitz, and Steven M Block. An automated two-dimensional optical force clamp for single molecule studies. *Biophysical journal*, 83(1):491–501, 2002.
- [42] Steven M Block, Charles L Asbury, Joshua W Shaevitz, and Matthew J Lang. Probing the kinesin reaction cycle with a 2d optical force clamp. *Proceedings of the National Academy of Sciences*, 100(5):2351–2356, 2003.
- [43] A Ranaweera and B Bamieh. Modelling, identification, and control of a spherical particle trapped in an optical tweezer. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 15(16):747–768, 2005.
- [44] Hullas Sehgal, Tanuj Aggarwal, and Murti V Salapaka. High bandwidth force estimation for optical tweezers. *Applied Physics Letters*, 94(15):153114, 2009.
- [45] Tanuj Aggarwal, Hullas Sehgal, and Murti Salapaka. Robust control approach to force estimation in a constant position optical tweezers. *Review of Scientific Instruments*, 82(11), 2011.

- [46] Subhrajit Roychowdhury, Shreyas Bhaban, Srinivasa Salapaka, and Murti Salapaka. Design of a constant force clamp and estimation of molecular motor motion using modern control approach. In *2013 American Control Conference*, pages 1525–1530. IEEE, 2013.
- [47] Shreyas Bhaban, Saurav Talukdar, Mingang Li, Thomas Hays, Peter Seiler, and Murti Salapaka. Single molecule studies enabled by model-based controller design. *IEEE/ASME Transactions on Mechatronics*, 23(4):1532–1542, 2018.
- [48] Arthur Ashkin, James M Dziedzic, and T Yamane. Optical trapping and manipulation of single cells using infrared laser beams. *Nature*, 330(6150):769–771, 1987.
- [49] Taejin L Min, Patrick J Mears, Lon M Chubiz, Christopher V Rao, Ido Golding, and Yann R Chemla. High-resolution, long-term characterization of bacterial motility using optical tweezers. *Nature methods*, 6(11):831–835, 2009.
- [50] Declan J Armstrong, Timo A Nieminen, Alexander B Stilgoe, Anatolii V Kashchuk, Isaac CD Lenton, and Halina Rubinsztein-Dunlop. Swimming force and behavior of optically trapped micro-organisms. *Optica*, 7(8):989–994, 2020.
- [51] S Sato, M Ishigure, and H Inaba. Optical trapping and rotational manipulation of microscopic particles and biological cells using higher-order mode nd: Yag laser beams. *Electronics letters*, 27(20):1831–1832, 1991.
- [52] Mor Habaza, Barak Gilboa, Yael Roichman, and Natan T Shaked. Tomographic phase microscopy with 180 rotation of live cells in suspension by holographic optical tweezers. *Optics letters*, 40(8):1881–1884, 2015.
- [53] Yu-chih Lin, Hui-Chi Chen, Han-Yen Tu, Chin-Yu Liu, and Chau-Jern Cheng. Optically driven full-angle sample rotation for tomographic imaging in digital holographic microscopy. *Optics letters*, 42(7):1321–1324, 2017.
- [54] Peter John Rodrigo, Lóránd Kelemen, Carlo Amadeo Alonzo, Ivan R Perch-Nielsen, Jeppe Seidelin Dam, Pál Ormos, and Jesper Glückstad. 2d optical manipulation and assembly of shape-complementary planar microstructures. *Optics express*, 15(14):9009–9014, 2007.
- [55] David G Grier. A revolution in optical manipulation. *nature*, 424(6950):810–816, 2003.
- [56] Darwin Palima and Jesper Glückstad. Gearing up for optical microrobotics: micromanipulation and actuation of synthetic microstructures by optical forces. *Laser & Photonics Reviews*, 7(4):478–494, 2013.

- [57] Moritz K Kreysing, Tobias Kießling, Anatol Fritsch, Christian Dietrich, Jochen R Guck, and Josef A Käs. The optical cell rotator. *Optics Express*, 16(21):16984–16992, 2008.
- [58] Jianbin Huang, Xiaoshuai Liu, Yao Zhang, and Baojun Li. Optical trapping and orientation of escherichia coli cells using two tapered fiber probes. *Photonics Research*, 3(6):308–312, 2015.
- [59] Masatoshi Ichikawa, Koji Kubo, Shizuaki Murata, Kenichi Yoshikawa, and Yasuyuki Kimura. Single cell manipulation by using tilt controlled optical tweezers. In *2007 International Symposium on Micro-NanoMechatronics and Human Science*, pages 316–321. IEEE, 2007.
- [60] Lynn Paterson, Michael P MacDonald, Jochen Arlt, W Sibbett, PE Bryant, and K Dholakia. Controlled rotation of optically trapped microscopic particles. *Science*, 292(5518):912–914, 2001.
- [61] SK Mohanty, R Dasgupta, and PK Gupta. Three-dimensional orientation of microscopic objects using combined elliptical and point optical tweezers. *Applied Physics B*, 81(8):1063–1066, 2005.
- [62] Marlies EJ Friese, Timo A Nieminen, Norman R Heckenberg, and Halina Rubinsztein-Dunlop. Optical alignment and spinning of laser-trapped microscopic particles. *Nature*, 394(6691):348–350, 1998.
- [63] Arthur La Porta and Michelle D Wang. Optical torque wrench: Angular trapping, rotation, and torque detection of quartz microparticles. *Physical review letters*, 92(19):190801, 2004.
- [64] Fumihito Arai, Kazuhisa Onda, Ryo Iitsuka, and Hisataka Maruyama. Multi-beam laser micromanipulation of microtool by integrated optical tweezers. In *2009 IEEE International Conference on Robotics and Automation*, pages 1832–1837. IEEE, 2009.
- [65] Ivan Shishkin, Hen Markovich, Yael Roichman, and Pavel Ginzburg. Auxiliary optomechanical tools for 3d cell manipulation. *Micromachines*, 11(1):90, 2020.
- [66] K Visscher, GJ Brakenhoff, and JJ Krol. Micromanipulation by multiple optical traps created by a single fast scanning trap integrated with the bilateral confocal scanning laser microscope. *Cytometry: The Journal of the International Society for Analytical Cytology*, 14(2):105–114, 1993.

- [67] Teuta Pilizota, Thomas Bilyard, Fan Bai, Masamitsu Futai, Hiroyuki Hosokawa, and Richard M Berry. A programmable optical angle clamp for rotary molecular motors. *Biophysical journal*, 93(1):264–275, 2007.
- [68] Chien Chern Cheah, Quang Minh Ta, and Reza Haghighi. Grasping and manipulation of a micro-particle using multiple optical traps. *Automatica*, 68:216–227, 2016.
- [69] Quang Minh Ta and Chien Chern Cheah. Stochastic control for orientation and transportation of microscopic objects using multiple optically driven robotic fingertips. *IEEE Transactions on Robotics*, 35(4):861–872, 2019.
- [70] Edison Gerena, Stéphane Régnier, and Sinan Haliyo. High-bandwidth 3-d multitrap actuation technique for 6-dof real-time control of optical robots. *IEEE Robotics and Automation Letters*, 4(2):647–654, 2019.
- [71] Yoshio Tanaka, Hiroyuki Kawada, Ken Hirano, Mitsuru Ishikawa, and Hiroyuki Kitajima. Automated manipulation of non-spherical micro-objects using optical tweezers combined with image processing techniques. *Optics Express*, 16(19):15115–15122, 2008.
- [72] V Bingelyte, J Leach, J Courtial, and MJ Padgett. Optically controlled three-dimensional rotation of microscopic objects. *Applied Physics Letters*, 82(5):829–831, 2003.
- [73] Florian Hörner, Mike Woerdemann, Stephanie Müller, Berenike Maier, and Cornelia Denz. Full 3d translational and rotational optical control of multiple rod-shaped bacteria. *Journal of biophotonics*, 3(7):468–475, 2010.
- [74] Isaac CD Lenton, Declan J Armstrong, Alexander B Stilgoe, Timo A Nieminen, and Halina Rubinsztein-Dunlop. Orientation of swimming cells with annular beam optical tweezers. *Optics Communications*, 459:124864, 2020.
- [75] Mingyang Xie, Adnan Shakoor, Yajing Shen, James K Mills, and Dong Sun. Out-of-plane rotation control of biological cells with a robot-tweezers manipulation system for orientation-based cell surgery. *IEEE Transactions on Biomedical Engineering*, 66(1):199–207, 2019.
- [76] Arvind Balijepalli, Thomas W LeBrun, and Satyandra K Gupta. A flexible system framework for a nanoassembly cell using optical tweezers. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 42578, pages 333–342, 2006.
- [77] G Carmon and M Feingold. Rotation of single bacterial cells relative to the optical axis using optical tweezers. *Optics letters*, 36(1):40–42, 2011.

- [78] Amarjeet Yadav, Anindita Dutta, Pramod Kumar, Yuval Dahan, Alexander Aranovich, and Mario Feingold. Optimal trapping stability of escherichia coli in oscillating optical tweezers. *Physical Review E*, 101(6):062402, 2020.
- [79] Yoshio Tanaka and Shin-ich Wakida. Controlled 3d rotation of biological cells using optical multiple-force clamps. *Biomedical optics express*, 5(7):2341–2348, 2014.
- [80] Murray Garden. Learning control of actuators in control systems, January 12 1971. US Patent 3,555,252.
- [81] Masaru Uchiyama. Formation of high-speed motion pattern of a mechanical arm by trial. *Transactions of the Society of Instrument and Control Engineers*, 14(6):706–712, 1978.
- [82] Timo A Nieminen, Vincent LY Loke, Alexander B Stilgoe, Gregor Knöner, Agata M Brańczyk, Norman R Heckenberg, and Halina Rubinsztein-Dunlop. Optical tweezers computational toolbox. *Journal of Optics A: Pure and Applied Optics*, 9(8):S196, 2007.
- [83] José García de la Torre and Victor A Bloomfield. Hydrodynamic properties of complex, rigid, biological macromolecules: theory and applications. *Quarterly reviews of biophysics*, 14(1):81–139, 1981.
- [84] Suguru Arimoto. Mathematical theory of learning with applications to robot control. In *Adaptive and Learning Systems*, pages 379–388. Springer, 1986.
- [85] David B Phillips, David M Carberry, SH Simpson, Henrik Schäfer, M Steinhart, R Bowman, Graham M Gibson, Miles J Padgett, S Hanna, and MJ Miles. Optimizing the optical trapping stiffness of holographically trapped microrods using high-speed video tracking. *Journal of Optics*, 13(4):044023, 2011.
- [86] Jaideep S Dudani, Daniel R Gossett, TK Henry, and Dino Di Carlo. Pinched-flow hydrodynamic stretching of single-cells. *Lab on a Chip*, 13(18):3728–3734, 2013.
- [87] Akbar Samadi and S Nader S Reihani. Role of condenser iris in optical tweezer detection system. *Optics letters*, 36(20):4056–4058, 2011.
- [88] Tanuj Aggarwal and Murti Salapaka. Real-time nonlinear correction of back-focal-plane detection in optical tweezers. *Review of Scientific Instruments*, 81(12), 2010.
- [89] Jeffrey Kodosky. Labview. *Proceedings of the ACM on Programming Languages*, 4(HOPL):1–54, 2020.

- [90] Koen Visscher, Steven P Gross, and Steven M Block. Construction of multiple-beam optical traps with nanometer-resolution position sensing. *IEEE journal of selected topics in quantum electronics*, 2(4):1066–1076, 1996.
- [91] Jean-Yves Tinevez, Nick Perry, Johannes Schindelin, Genevieve M Hoopes, Gregory D Reynolds, Emmanuel Laplantine, Sebastian Y Bednarek, Spencer L Shorte, and Kevin W Eliceiri. Trackmate: An open and extensible platform for single-particle tracking. *Methods*, 115:80–90, 2017.
- [92] Winfried Denk and Watt W Webb. Optical measurement of picometer displacements of transparent microscopic objects. *Applied optics*, 29(16):2382–2391, 1990.
- [93] Lucien P Ghislain, Neil A Switz, and Watt W Webb. Measurement of small forces using an optical trap. *Review of Scientific Instruments*, 65(9):2762–2768, 1994.
- [94] Kirstine Berg-Sørensen and Henrik Flyvbjerg. Power spectrum analysis for optical tweezers. *Review of Scientific Instruments*, 75(3):594–612, 2004.
- [95] James A Spudich, Sarah E Rice, Ronald S Rock, Thomas J Purcell, and Hans M Warrick. Optical traps to study properties of molecular motors. *Cold Spring Harbor Protocols*, 2011(11):pdb-top066662, 2011.
- [96] Samarendra K Mohanty, Pradeep K Gupta, and Ravi S Verma. Self-rotation of an assembly of two or more cylindrical objects in optical tweezers: A simple approach for realization of optically driven micromotors. *Current Science*, pages 695–698, 2007.
- [97] Frederic Català, Ferran Marsà, Mario Montes-Usategui, Arnau Farré, and Estela Martín-Badosa. Extending calibration-free force measurements to optically-trapped rod-shaped samples. *Scientific reports*, 7(1):42960, 2017.
- [98] Quang Minh Ta, Gulam Dastagir Khan, Xiang Li, and Chien Chern Cheah. Laser-actuated multi-fingered hand for dexterous manipulation of micro-objects. *IEEE Access*, 11:29650–29660, 2023.
- [99] Arthur Ashkin and JM Dziedzic. Observation of light scattering from nonspherical particles using optical levitation. *Applied optics*, 19(5):660–668, 1980.
- [100] Howard C Berg. *Random walks in biology*. Princeton University Press, 1993.
- [101] RC Gauthier. Theoretical investigation of the optical trapping force and torque on cylindrical micro-objects. *JOSA B*, 14(12):3323–3333, 1997.

- [102] Robert C Gauthier, Mike Ashman, and Chander P Grover. Experimental confirmation of the optical-trapping properties of cylindrical objects. *Applied optics*, 38(22):4861–4869, 1999.
- [103] Alexis I Bishop, Timo A Nieminen, Norman R Heckenberg, and Halina Rubinsztein-Dunlop. Optical application and measurement of torque on microparticles of isotropic nonabsorbing material. *Physical Review A*, 68(3):033802, 2003.