Machine Learning for Predicting Particle Properties in Optical Tweezers

Lachlan Hamilton (s44792822)

Supervised by Isaac Lenton, Timo Nieminen

Group Members: Lauren McQueen, Oscar Smee

Abstract

This report aimed to use machine learning to predict an optically trapped particle's refractive index and radius. The training data was generated with optical trapping simulations based on a neural network developed by *I.Lenton et al.*[1]. Their 5-DOF neural network predicts the force on a particle based on its position, radius and refractive index. It is much faster than analytical methods, allowing longer simulations to be generated inexpensively. [1]

The project first attempted to analyse the trap stiffness of the 5-DOF simulations. The purpose was to refine the data so a network could extrapolate the radius and refractive index with minimal input data. Some problems were identified with this method. Firstly, only the X and Y axial trap stiffness could be used. This severely limited the amount of unique points in the radius and refractive index trapping landscape. A neural network would only be able to identify the unique points. Therefore, it is not a successful training dataset. The trap stiffness calculations remove too much information for a network to learn.

Instead, *Lauren McQueen* and *Oscar Smee* developed methods for using the forces and positions from each simulation as an input. *Oscar Smee* used a time series method for training a network with all the forces. His network could find the refractive index for a given radius within 0.75% error. *Lauren McQueen* created a network which used a histogram of the position in one dimension as an input. Her network predicted both radius and refractive index with $< 3.5\%$ error.

Both methods were used as to demonstrate it is possible to use machine learning to identify particles. They can now be upgraded to include both position and force datasets to predict radius and refractive index with more accuracy.

Contents

Abstract 1

Aim 3

Introduction 4-10

I1 Optical Trapping 4

I2 Trap Stiffness Calculations 5

I3 ANN Types and the 5-DOF Network 8

Method 11

Results 12-20

R1 Simulation Analysis 12

R2 Trap Stiffness Landscape 16

R3 Network Training 18

Discussion 21-22

D1 Trap Stiffness Data Generation 21

D2 Neural Networks and Optical Tweezers 22

Conclusion 22

References 23

Appendix 24-31

A1 T-Matrix Method 24

A2 Neural Networks 26

A3 Z-Axial Motion 28

A4 Theoretical vs. Simulated κ 29

A5 $\kappa = 0$ Adjusted Landscapes 30

A6 Network Comparison 31

Aim

Optical tweezers are a phenomenal tool for controlling microscopic systems with piconewton precision. Often particles with a known radius and refractive index are used as probes to investigate new systems. This project aimed to do the opposite. To invent a method for finding the radius and refractive index of an unknown particle in an optical trap through machine learning.

This would be a useful tool for identifying particles in systems where other measurements are impossible. Uses for this could be taking measurements inside a cell, or perhaps categorising particles and create a sorting process.

The invention of a neural network to predict optical trapping forces by *I.Lenton et al.* [1] motivates the use of machine learning in the field. Their work created a 5-degree of freedom (5-DOF) network which takes a particles position (x, y, z), radius and refractive index (hence 5-DOF) and outputs the force a particle should experience. This proved to be significantly faster at running dynamical simulations. [1]

This project used the 5-DOF network to simulate the motion of particles with different radii and refractive indices. The aim was to use the motion to train a neural network to output the radius and refractive index of the particle. The first part of the project was analysing the motion and finding the necessary information to teach a network.

Multiple methods of training were attempted. Most analysed the statistics of the particle's motion in the trap. The purpose was to find a unique set of information which describes each particles parameters – and thus train a network.

Machine learning is a unique tool for finding relationships between correlated variables. It is not limited by our analytical capabilities and physical analysis – it can often create links between complicated and otherwise unpredictable variables.

Optical trapping is well suited to machine learning, as there are many trends which have no analytical relationship. Notably, the relationship of radius and refractive index on trap stiffness can only be found by simulations.

There are limitations with machine learning which are discussed in this report. Most significantly the training data must have a unique 1 to 1 map from input to output. Networks must also be generalised to multiple situations to justify the length of time required to train a network.

Introduction

I1 Optical Trapping:

Light has the unique property of being massless but having momentum. A change in momentum must impart a force, and thus light can cause a force on a particle by changing direction. The momentum of light is given by the de Broglie momentum[2].

$$\bar{p} = \frac{h}{\lambda}$$

Optical tweezers are designed to hold a particle in a region or position by designing the light to generate a restoring force when it moves from the equilibrium. The forces on a particle can be calculated a number of ways. Large particles are suited to ray optics, using Fresnel equations [2] for reflection and refraction to determine the asymmetry in incident and scattered light. Particles much smaller than the wavelength of light can be approximated as dipoles, where the restoring force is given by: [2]

$$\boldsymbol{F}_{grad} = \frac{1}{2}\frac{\alpha'}{c\varepsilon_0}\nabla I(\boldsymbol{r})$$

This is called the gradient force – as it follows the intensity gradient. α' is the real polarizability of the particle. This is related to the refractive index and is positive for refractive indices higher than the medium – causing attraction force to the highest intensity. Therefore, particles with a lower refractive index than the medium are not trapped.

For a restoring force, it would be convenient to create a potential giving a Hookean spring relationship.

$$\boldsymbol{F}_{grad} = -\kappa_\rho \rho$$

Where ρ is the particle's distance from the centre of the beam on the $x - y$ plane. Lasers generally propagate in the TEM$_{00}$ Gaussian mode. [3] The Gaussian distribution for $I(\rho)$ gives the desired restoring force, and can be described by: [3]

$$I(\rho) = I_0 e^{-2\rho^2/\omega_0}$$

Where ω_0 is the beam width and I_0 the maximum intensity. Towards the centre, we can take the Taylor expansion to reach:

$$I(\rho \approx 0) = I_0\left(1 - 2\frac{\rho^2}{\omega_0^2}\right) \quad \rightarrow \quad \nabla I(\rho) = -\frac{4}{\omega_0^2}\rho$$

Therefore, our theoretical trap stiffness κ_ρ will be:

$$\kappa_\rho = 2\frac{\alpha' I_0}{c\epsilon_0\omega_0^2}$$

This theoretical stiffness calculation is only valid for the dipole regime, where $r \ll \lambda$.

The most sophisticated method for calculating the forces on a particle is with a field analysis. Various methods exist for comparing the scattered and incident light and extrapolating the forces. A robust method which is fast for spheres is the T-Matrix method. See appx.1 for a detailed description of the T-Marix method and the Optical Tweezer Toolbox (OTT).

I2 Trap Stiffness Calculations:

There are many methods to calculate the trap stiffness of a particle-beam system. The simplest method from a force-displacement relationship (as modelled by the OTT) is demonstrated by *G.Knöner et al.[5].* The equilibrium position is found, and then the force at a slight displacement to either side is recorded (Fig1.2). The force gradient at this point is the linear trap stiffness in that dimension.

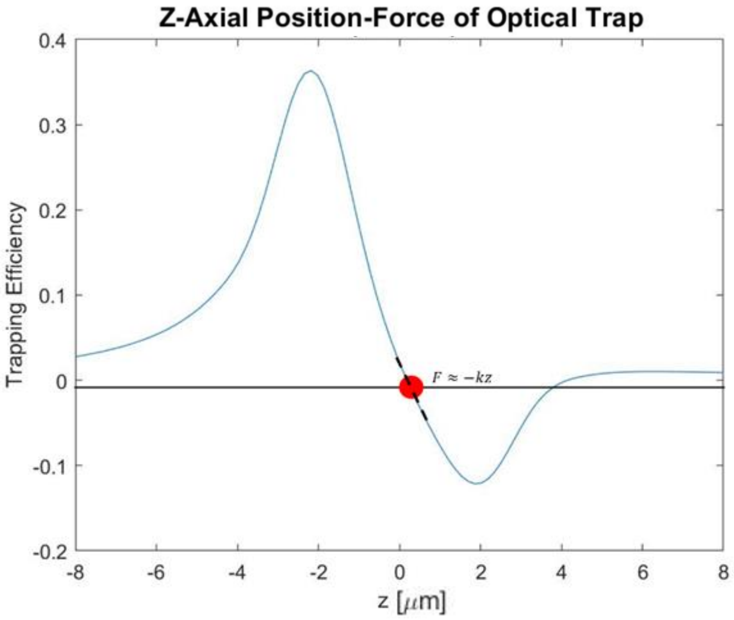


Fig1.2: A demonstration of the linear trap stiffness about the equilibrium of an Optical Trap. The equilibrium is shown in red, and the linear region approximated by the dotted line.

For physical measurements of optical tweezer systems, these force-displacement plots can be difficult to experimentally generate with accuracy.[12] This is mostly because the trap stiffness is only linear around the equilibrium point in all 3 dimension. Quadrant photodiode detectors and acousto-optic modulators systems can measure the force and position of a particle, but only in the focal plane (x, y).[7,12] Therefore there are many trajectories crossing the equilibrium in the measured dimension, but not the others.

M.Sarshar et al.[6] provides a detailed summary of methods for trap stiffness calculations. A micro-particle in an optical trap observes two main forces. The optical force, roughly linear about the equilibrium, and a random Brownian motion force.

The Equipartition Theorem (ET) and Power Spectral Density (PSD) methods [7] are explored in this report. They are both based on analysing the difference in random and optical motion.

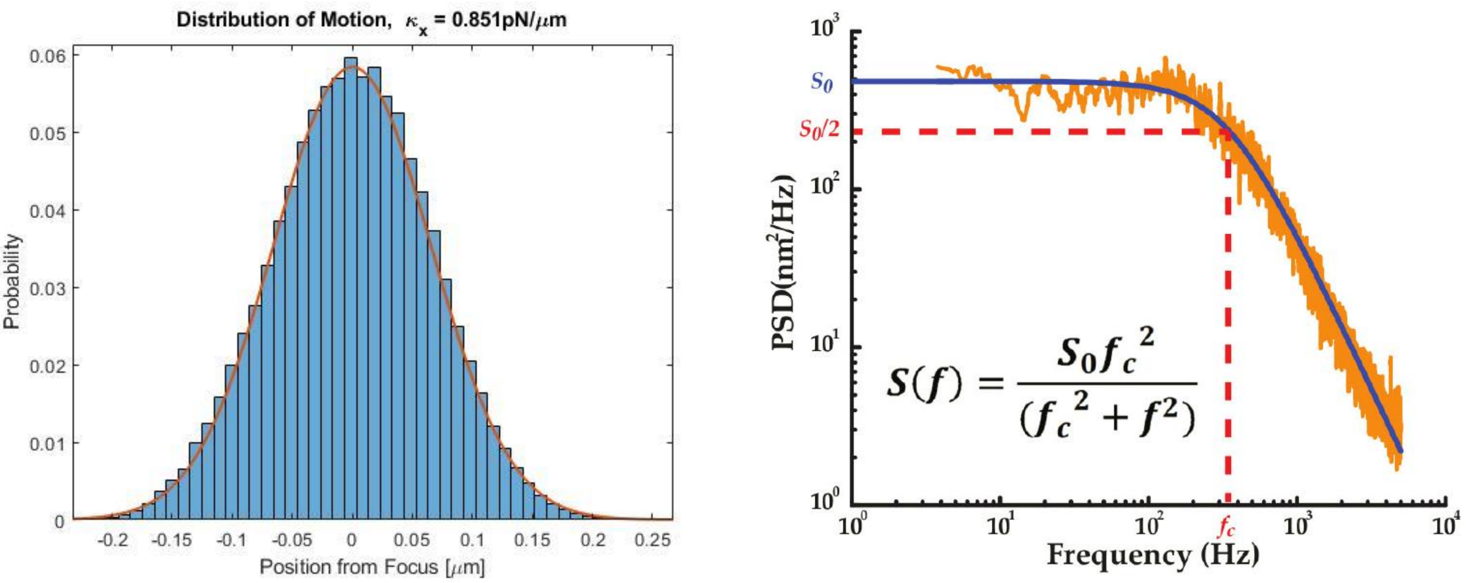


Fig1.3: Comparison of the two methods. Left if the distribution of motion with a Gaussian function fit (orange) calculated from Equipartition theorem. Right shows a power spectral density plot with the Lorentzian fit (blue) and corner frequency marked (red dotted line). Image retrieved from "Nanomanipulation of Single RNA Molecules by Optical Tweezers" [13]

The ET method simplifies the problem by analysing each dimension separately. Statistical mechanics states that the average energy a particle will have in a degree of freedom is given by half the available thermal energy: [9]

$$\langle E \rangle = \frac{1}{2} k_B T$$

If we assume the optical force is linear, and integrate over some displacement, we can obtain an expression for the potential energy at some distance Z:

$$F = -\kappa z, \qquad E = \int F dz$$

$$E = \int \kappa z \, dz = \frac{1}{2} \kappa_z z^2$$

The expectation value of z^2 can be expressed by the variance, provided the mean is set to 0:

$$\sigma^2 = \langle X^2 \rangle - \langle X \rangle^2 = \langle X^2 \rangle$$

Therefore, by equation the expected energy and expected variance:

$$\frac{1}{2} k_B T = \frac{1}{2} \kappa_z \langle z^2 \rangle$$

$$\kappa_z = \frac{k_B T}{\langle z^2 \rangle} \qquad \rightarrow \qquad \kappa_z = \frac{k_B T}{\sigma^2}$$

This method is equivalent to fitting a Gaussian distribution to the distribution of motion. By considering the Boltzmann factor, representing the relative probability of a particle being in a given state, we observe:

$$P \propto \exp\left(-\frac{U(r)}{k_B T}\right) = \exp\left(-\frac{\kappa z^2}{2 k_B T}\right)$$

Considering a Gaussian distribution is written:

$$P(x) = \frac{1}{A} \exp\left(-\frac{1}{2}\left(\frac{x - \bar{x}}{\sigma}\right)^2\right)$$

We obtain the identical result provided $\bar{x} = 0$:

$$\sigma = \sqrt{\frac{k_B T}{\kappa}}$$

The objective of the ET method is thus to measure the variance in the position. It requires the temperature to be known, the trap to be observed for a significant amount of time and only positions about the linear region near the equilibrium to be used.

For the PSD method, we first model the trapped particle with a Langevin equation [8]:

$$\frac{m}{\gamma}\ddot{x}(t) + \dot{x}(t) + \frac{\kappa}{\gamma}x(t) = \left(\frac{2k_B T}{\gamma}\right)^{\frac{1}{2}} \xi(t)$$

The $\xi(t)$ term is the Brownian motion in a random direction, scaled by the Stokes-Einstein diffusion coefficient in a random orientation [8]. κ is the linear trap stiffness and γ is the drag coefficient – a product of the fluid's viscosity η and radius r.

$$\gamma = 6\pi r\eta$$

The Fourier transform of a particle's motion can be analysed to extrapolate the trap stiffness. We can rewrite the Langevin equation in terms of the corner frequency, f_c. This is the highest frequency the optical trap can resist. The frequency of inertia is too high for detection, and thus $\ddot{x}$ can be ignored. [8] This leaves the following equation of motion:

$$\dot{x}(t) + 2\pi f_c x(t) = (2D)^{\frac{1}{2}}\xi(t)$$

The corner frequency can be related to the trap stiffness by this Langevin equation.

$$f_c = \frac{\kappa}{2\pi\beta}$$

This can be measured by taking the Fourier transform of the motion and plotting the power spectral density (PSD). [8] This density can then be fit with a Lorentzian function, which reveals the corner frequency: [8]

$$P(f) = \frac{D}{2\pi^2(f_c^2 + f)}$$

Thus, the PSD method requires the motion to be analysed for a significant amount of time and the viscosity of the fluid to be known.

Finally, it is important to note that the trap stiffness with a spherical particle will be different in all 3 dimensions in a linearly polarised beam (Fig1.4). This is because the linear beam has a two axis of symmetry – but is not rotationally symmetrical.[15,16] The Z-axial trap stiffness is also obscured by the ET method. The experimental trap stiffness will by lower than the theoretical trap stiffness as the scattering force always pushes the particle away from the equilibrium. This creates a slightly non-linear potential. See appx.3 for an example.

Any theoretical trap stiffness measurements will be higher than an experiment or simulation. This is due to the Brownian motion of the particle, and the temperature of the system. Appx.4 discusses this phenomena.

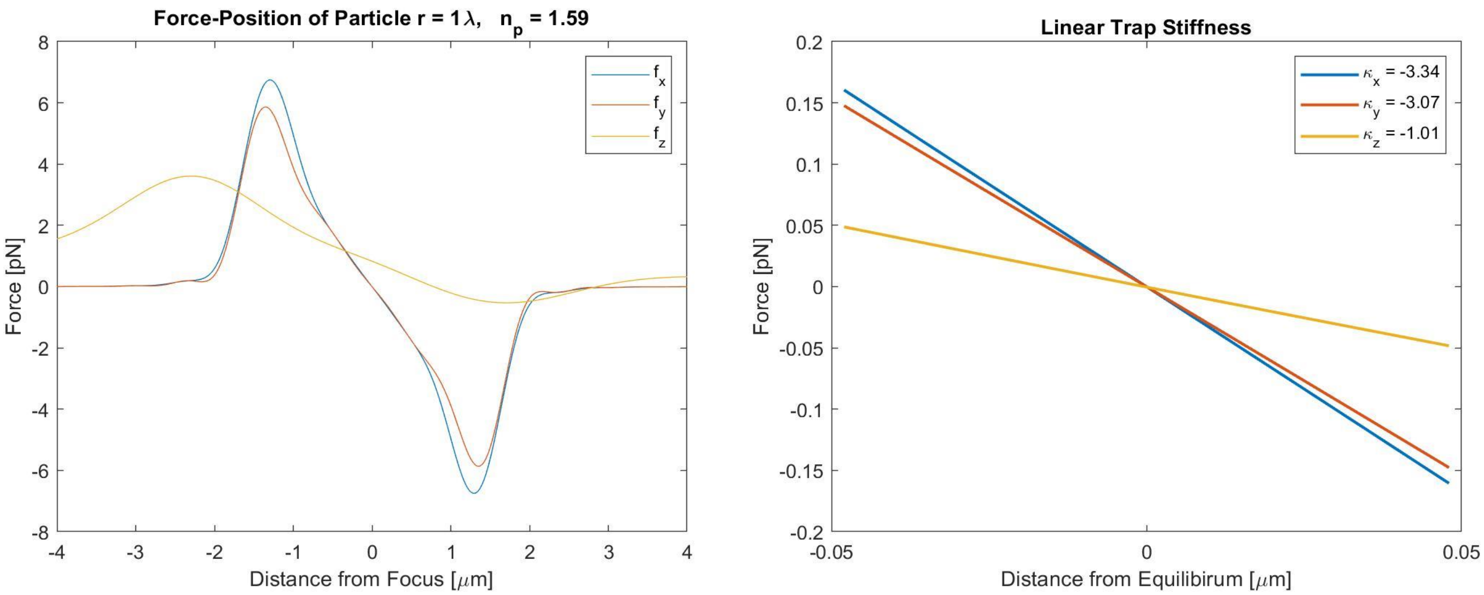


Fig1.4: An example of an ideal trap stiffness calculated by OTT. The left shows the entire force curve for a linearly polarised beam – note that the x and y forces are different. The right demonstrates the linear gradient about the equilibrium. The Z-Axial gradient has been shifted such that the equilibrium is 0. On the actual force-position graph, it is shifted from the focus by the scattering force.

I3 ANN Types and the 5-DOF Network:

For an introduction to artificial neural networks (ANN) and machine learning, see appx.2.

Machine learning can be used to solve a variety of problems. Naturally, some network architectures are better than others. The output, input and method of networking determines the type of network – and some common structures are used for similar problems.

Notably, there are two main types of outputs for supervised ANN. Classification networks take the input data and try to find the most probable output from a list of known results. The output predicts the probability of the input being in each category. [18] Regression networks act as a function mapping each input to a single output or multiple outputs. [18] This requires the training data to form a 1 to 1 map from input to output – as only uniquely described points will be able form a regression 'function'. These networks are often written as functions $N(x_{in}) \rightarrow x_{out}$.

When teaching a network, it is important to determine whether the network is optimally trained. There are two main issues. Underfitting is when the network hasn't observed the training data enough to reach optimum accuracy. [18] Every time the training dataset is passed to the network, and allowed to update the weights, is called an epoch. When a network is underfit, more training epochs are needed.

Overfitting occurs when the network has optimised itself to the training data, but in doing so has made itself less effective at other situations. [18] This is especially a problem if there was a bias in the training data, or structures that were not shuffled. Networks are very effective at correlating events, so if a dataset is not rearranged into a random order the network may predict outputs based on the input's location in the dataset and not its value.

To establish how well a network has been trained, some of the data is removed from the training dataset to be used for validation. The loss on the training data and the loss on the validation data is then measured. It is standard practice to remove $\sim 10\%$ of the data before training is started for validation.

This report was motivated and enabled by the creation of a 5 degree of freedom (5-DOF) regression network for optical tweezers. [1] The network was trained using 10^7 data points generated with the OTT. The aim was to create a regression network which maps:

$$N(x, y, z, r, n_p) \rightarrow (f_x, f_y, f_z)$$

That is, uses the position, radius and refractive index to finds the force. The range of values for radius and refractive index are as follows: [1]

$$0.1 \leq r \leq 1\mu m, \qquad 1.33 \leq n_p \leq 2$$

The network outputs the force with units enabling it to be used for situations with different laser powers and refractive indices. The unit conversion is as follows:

$$F_{NN} = O_{NN}\frac{r^2(n_p - n_m)}{c}P$$

Where O_{NN} is the output of the neural network, n_m is the refractive index of the medium, and P the power of the network. The 5-DOtF network had very good agreement with the OTT data used to train it.

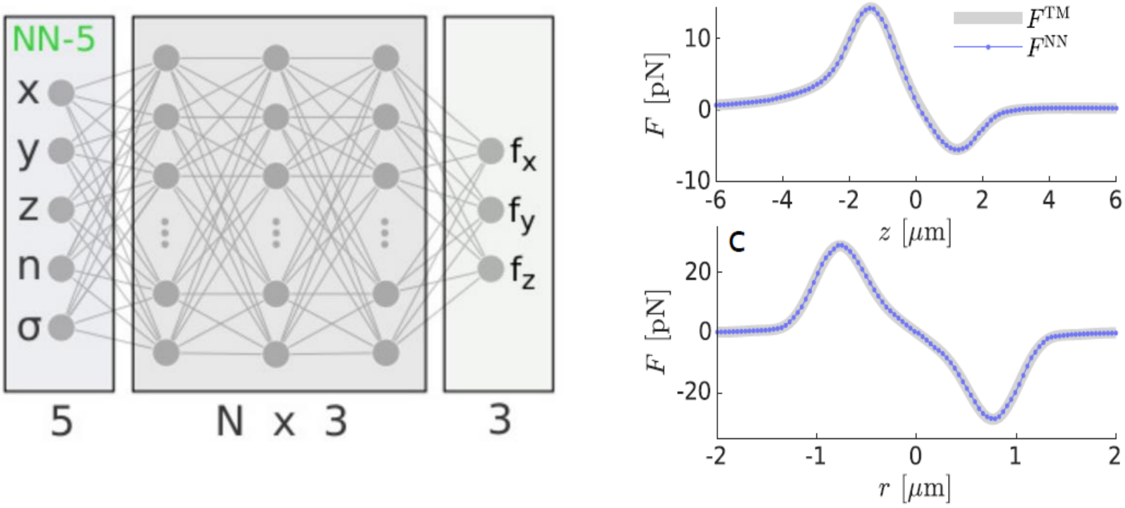


Fig1.6: The architecture of the 5-DOF network (left). Agreement of the force-position curves between the 5-DOF and OTT. Both images from I.Lenton et al.[1].

For creating a neural network to identify radius and refractive index from the particle motion, there are a few possible architectures we can employ. Fundamentally, we must give enough information into the network that it can find both pieces of information from the data.

If we consider the Langevin stochastic equation again:

$$\dot{x}(t) + \frac{\kappa}{\gamma}x(t) = \left(\frac{2k_BT}{\gamma}\right)^{\frac{1}{2}}\xi(t) \qquad \gamma = 6\pi r\eta$$

The Brownian motion noise term depends on the drag coefficient γ, which encodes the radius of the particle and the viscosity of the fluid. The spring constant, κ, is dependent on the radius and refractive index of the particle – but in a way we cannot analytically determine.

It is therefore possible that the trap stiffness alone may give enough information for the refractive index and radius. A regression network taking an input of the trap stiffness in all 3 dimensions may be the simplest architecture. That is:

$$N(\kappa_x, \kappa_y, \kappa_z) \rightarrow (r, n_p)$$

If this is not enough information, then perhaps the entire motion of the particle can be used as an input. The trap stiffness is calculated from the distribution of motion – as it requires the variance σ^2. However, some information is lost here – specifically any skewness or scattering forces. Trap stiffness is calculated without considering the forces, therefore more information is removed.

Instead of finding the variance to fit a Gaussian, as in Fig1.3, *Lauren McQueen* investigated training networks by using a histogram of the motion as an input. This encodes the entire distribution of motion, and the force and position histogram could be used. Let p_i be the probability of the particle being in the ith bin, her network can be described by:

$$N(p_{x1}, p_{x2}, \dots, p_{xn}, p_{y1}, p_{y2} \dots) \to (r, n_p)$$

Oscar Smee used a modified Time Series Classification (TSC) network described by *H.Fawaz et al.*[17]. His network is instead a Time Series Regression (TSR) network. The aim of the TSR network is to allow any amount of data to be input into the network. That is, the position and force at every instance in time can be used. The data could be sampled in any given way, and all the possible information is thus transferred. If $\bar{X}_i$ is the position vector at the ith step and $\bar{F}_i$ the force vector, this could be described by:

$$N(\bar{X}_1, \bar{F}_1, \bar{X}_2, \bar{F}_2, \dots, \bar{X}_n, \bar{F}_n) \to (r, n_p)$$

His network structure uses convoluted layers [17]. The aim of these layers is to find structures in the data, group them together, then pass these structure onward. Initially this was developed to mimic how the human brain processes images.[17] MLP networks are entirely interconnected, which can increase the training time needed to evaluate the weights. The convolution network attempts to decrease this time by grouping outputs together.

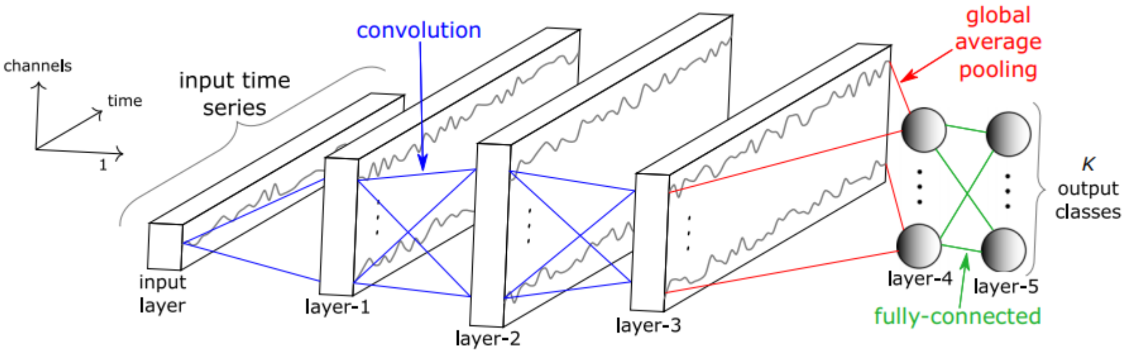


Fig1.7: A diagram of the TSC network from H.Fawaz et al.[17].

Method

The 5-DOF network was used to simulate a particle in an optical trap. The simulation was done on MATLAB based on *I.Lenton et al.*[1] work with the following equation of motion:

$$d\bar{x} = \frac{\bar{F}_{NN}}{\gamma}dt + \sqrt{\frac{2k_bT}{\gamma}dt}\hat{\xi}$$

Where F_{NN} is the predicted force from the neural network and $\hat{\xi}$ is a random direction (either x, y or z). k_B is the Boltzmann constant, T the temperature of the system and dt the timestep. Timesteps must be smaller than Brownian motion phenomena being simulated [14]. Considering a modest experimental setup can have $> 10kHz$ detection, $dt = 10^{-5}s$ will be similar to the physical measurement capabilities. [8,14]

For the simulation, the particle begins outside the trap and falls into it. Once the particle is trapped, the simulation is run for enough time such that the statistics of motion can be analysed. The amount of time necessary for such analysis is discussed later.

After the simulation is finished, the motion of the particle falling into the trap is removed. The co-ordinates are then changed such that the mean position is 0.

The variance is then calculated, and the trap stiffness extrapolated.

The simulation was then run for various amounts of time to discover the variability in the data.

The space of radii and refractive indices used in the simulations should be kept within the values used to train the 5-DOF network. This is because the network will be unable to extrapolate values it has not been trained for. This gives the space as [1]:

$$0.01\mu m \leq r \leq 1\mu m$$

$$1.3 \leq n_p \leq 2$$

For trapping purposes, radius is a more significant measurement – as refractive index can be hard to accurately determine. [5] Radius also has a larger range of values, spanning many orders of magnitudes practically. This must be kept in mind when designing the training dataset.

I.Lenton et al.[1] trained the 5-DOF network with a specific optical trapping system. The force output is generalised and translatable to a variety of laser powers and medium refractive indices, however it was trained on single values for both. To prevent unpredictable behaviour, the power will be kept to the same order of magnitude and the same medium will be used for the simulations. Naturally certain parameters, such as numerical aperture and beam polarisation, cannot be changed in the simulations.

Networks were created and trained in python with Tensorflow[18].

Results

R1 Simulation Analysis:

Before generating the training data, the error in the trap stiffness calculation must be analysed. The accuracy is difficult to ascertain, as any theoretical trap stiffness measurement will not agree with simulations. This is due to Brownian motion and therefore the temperature of the system being in present in simulations but not analysis. Appx.4 demonstrates the difficulty with determining the accuracy of the measurements.

Instead, the precision and uncertainty of the simulation can be assessed, and the accuracy later addressed. By comparing how similar each trial is for simulations of different lengths, we can determine the time needed to get an agreeable value.

To demonstrate the simulation, the following figures have been created.

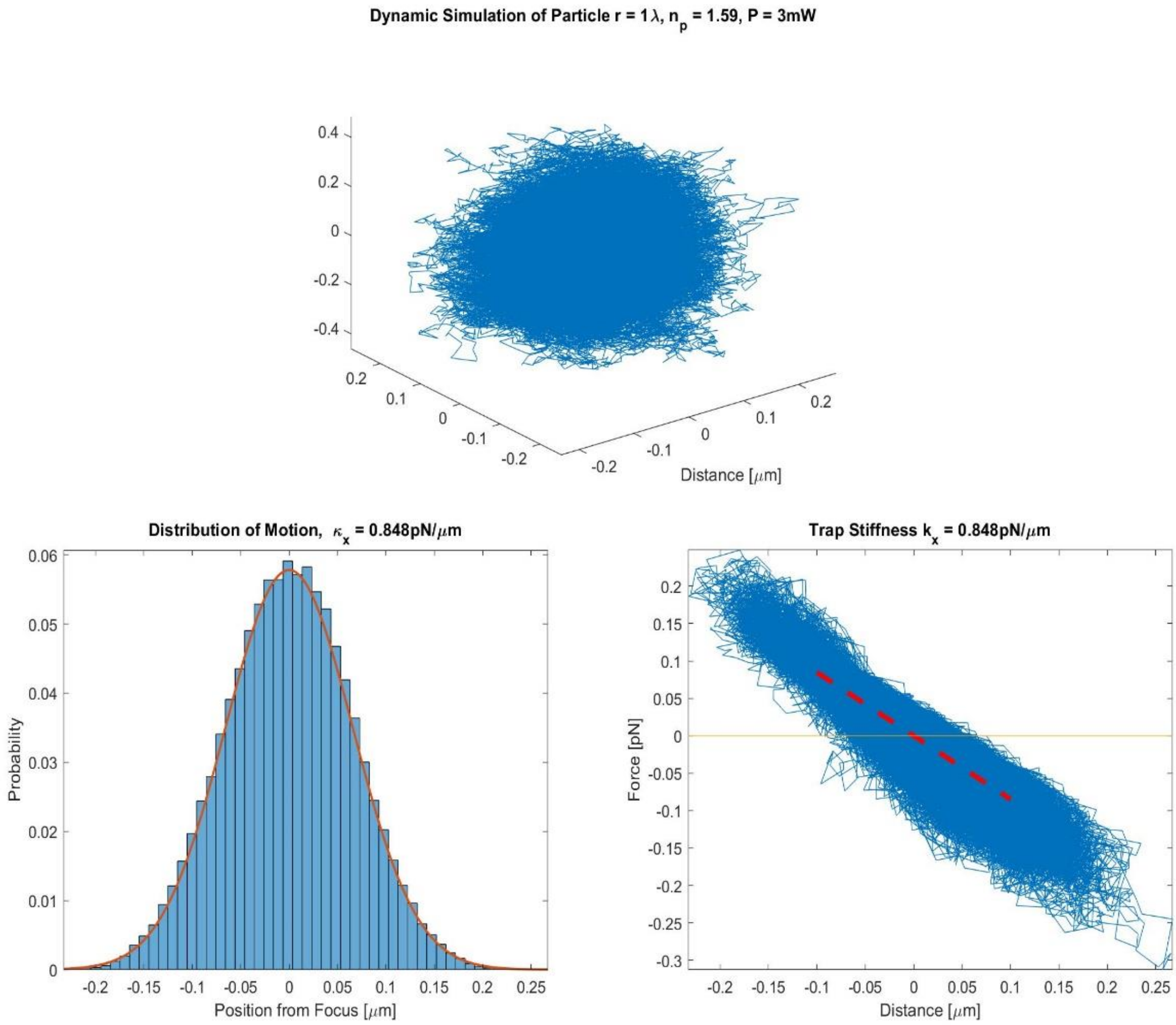


Fig2.1: An example of the simulation. A particle of a particular radius and refractive index has its trajectory simulated in an optical trap with the 5-DOF network (top). The variance of the motion is analysed to find the trap stiffness (left). The trap stiffness around the equilibrium is checked to be linear (right).

It is important to note that the Z-axial trap stiffness will have a skew distribution, as the scattering force is constantly along the axis of propagation.[2] That is, the particle will spend more time beyond the equilibrium than in front. See the appx.3 for such images. Next, we analyse the precision of the measurement as the simulation time is changed.

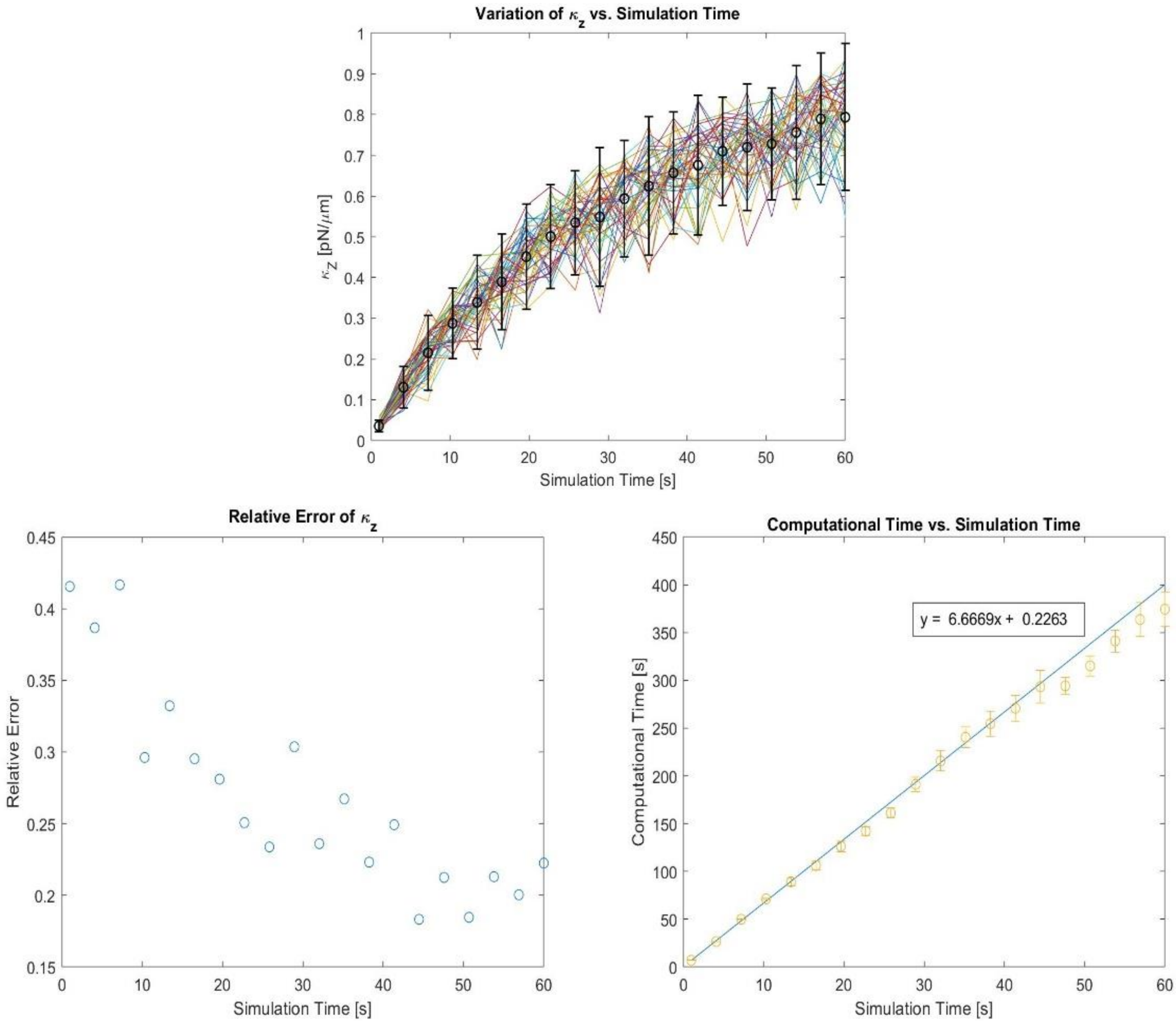


Fig2.2: The top image shows how the measurement of κ_z varies as the simulation time changes. The left image demonstrates how to relative error decreases as the time increases, however always stays above 20%. The right image shows how expensive the simulation time is. It has a slope of: $t_c = (6.7 \pm 0.1)t_s + (0 \pm 1)$

The difficulty with extrapolating the uncertainty in κ_z is due to the scattering force obscuring the gradient force. The low simulation time measurements have a very small κ_z measurement. This is because variance is our measurement and is caused by the Brownian motion. The variance increases as the Brownian motion component is allowed to run for more time. We know from the Langevin model there should be a single value for κ_z. That is, the variance should converge to a single value after a long enough time.

This is why the error bars for shorter simulations are smaller – but the value is least accurate. It is difficult to gauge the accuracy and precision simultaneously.

The relative error appears to still be random with time, and after $\sim 20s$ it plateaus. A linear regression of the computational time demonstrates that each simulation second requires $6.7s$ of computational time. This gives the cost of our precision.

Considering the large amount of data necessary for training, and the possibility that Z-axial trap stiffness has too large an error for training (Fig.2.2) it is perhaps worthwhile to begin with a simulation time of $30s$.

Next, the X and Y-axial stiffness was analysed.

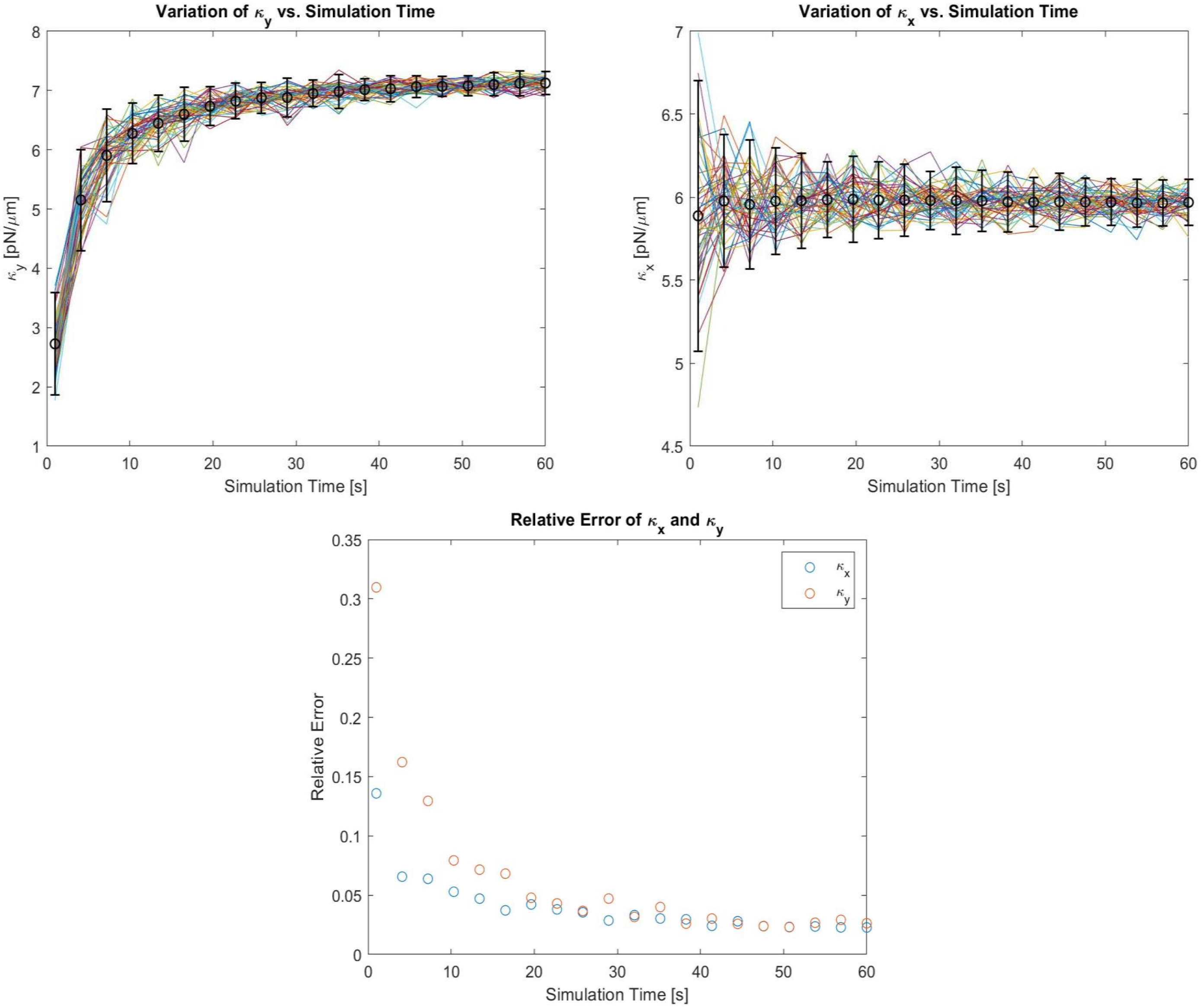


Fig2.3: The top two images show the Y and X-axial trap stiffness measurements of 50 trials with the simulation times given on the x-axis. The bottom image shows the relative error of the measurements based on the uncertainty.

These measurements converge to a single value. This value was compared to the OTT theoretical trap stiffness, however as shown in appx.4 these values do not agree. Again, it is difficult to assess the accuracy of the measurement, however there is precision to $\pm 5\%$ error. This motivates the use of κ_x and κ_y as our network inputs.

To estimate accuracy, we can fit our trap stiffness values to the equilibrium region of the trap.

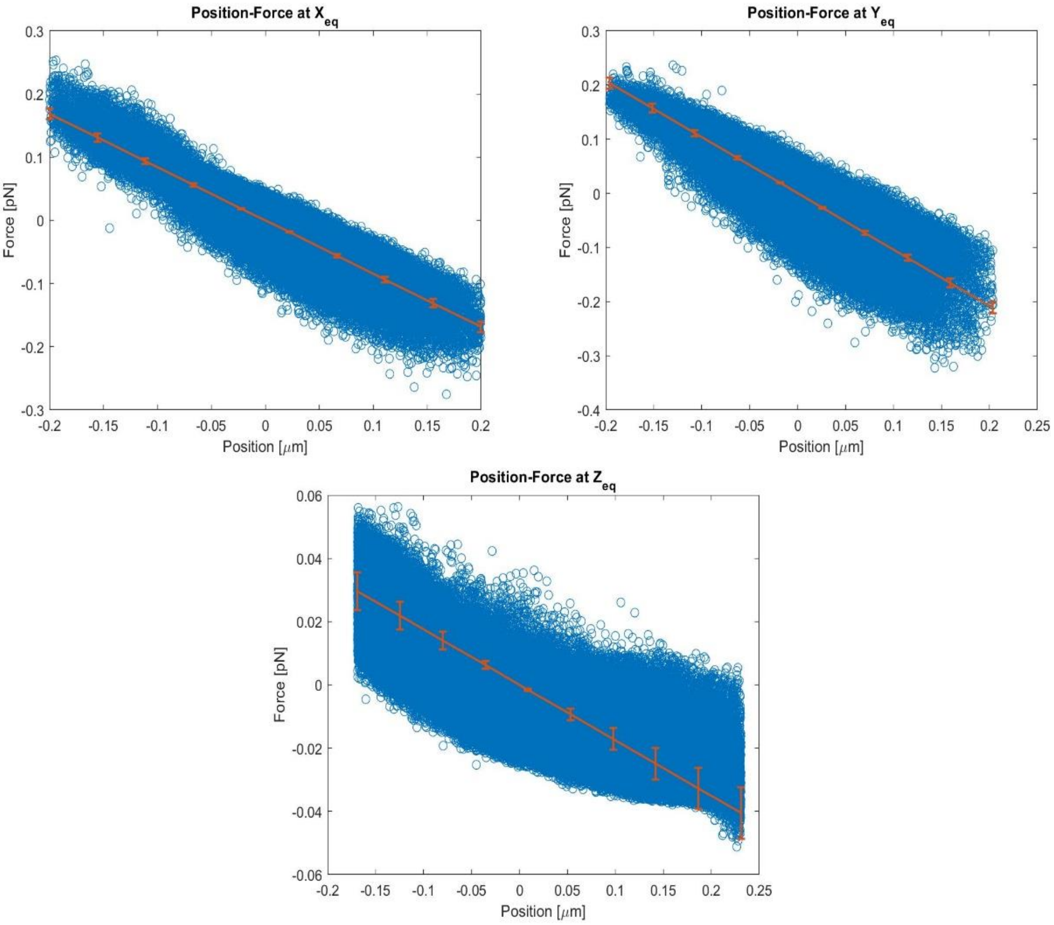


Fig2.4: Comparing the trap stiffness measurement in x, y, z with the equilibrium region. The errorbars are the uncertainty in the measurement, taken from the analysis done in Fig2.2 and Fig2.3 at $30s$.

These plots determine if our linear approximation is valid for the trap in all dimensions. The X and Y-axial trap stiffness values appear to accurately describe the middle of the force-position curve. That is, the equilibrium point in the other two dimensions. This demonstrates that they are well approximated by a spring restoring force. The Z plot however does not appear to agree very well with the trap stiffness. Despite having a much larger uncertainty, the points even as close to the equilibrium as $0.05\mu m$ already are away from middle of the force measurements.

Another more visual argument can be made that the $Z - F_z$ plot is obviously curved at $\sim 0.1\mu m$. While the centre could appear linear from $\pm 0.05\mu m$, it is likely not a good approximation. Appx.3 studies this further.

R2 Trap Stiffness Landscape:

The trap stiffness in all 3 dimensions were calculated across 500 radii evenly spaced from $0.01\mu m - 1\mu m$ and 100 refractive indices from $1.3 - 2$. The X and Y plots should be similar in magnitude and shape. The trap stiffness will be plotted as a colourmap with radius and refractive index on the X and Y axis. This is called a trap stiffness landscape, as it describes the regions of trapping and their magnitude.

The following data was created with $30s$ simulations.

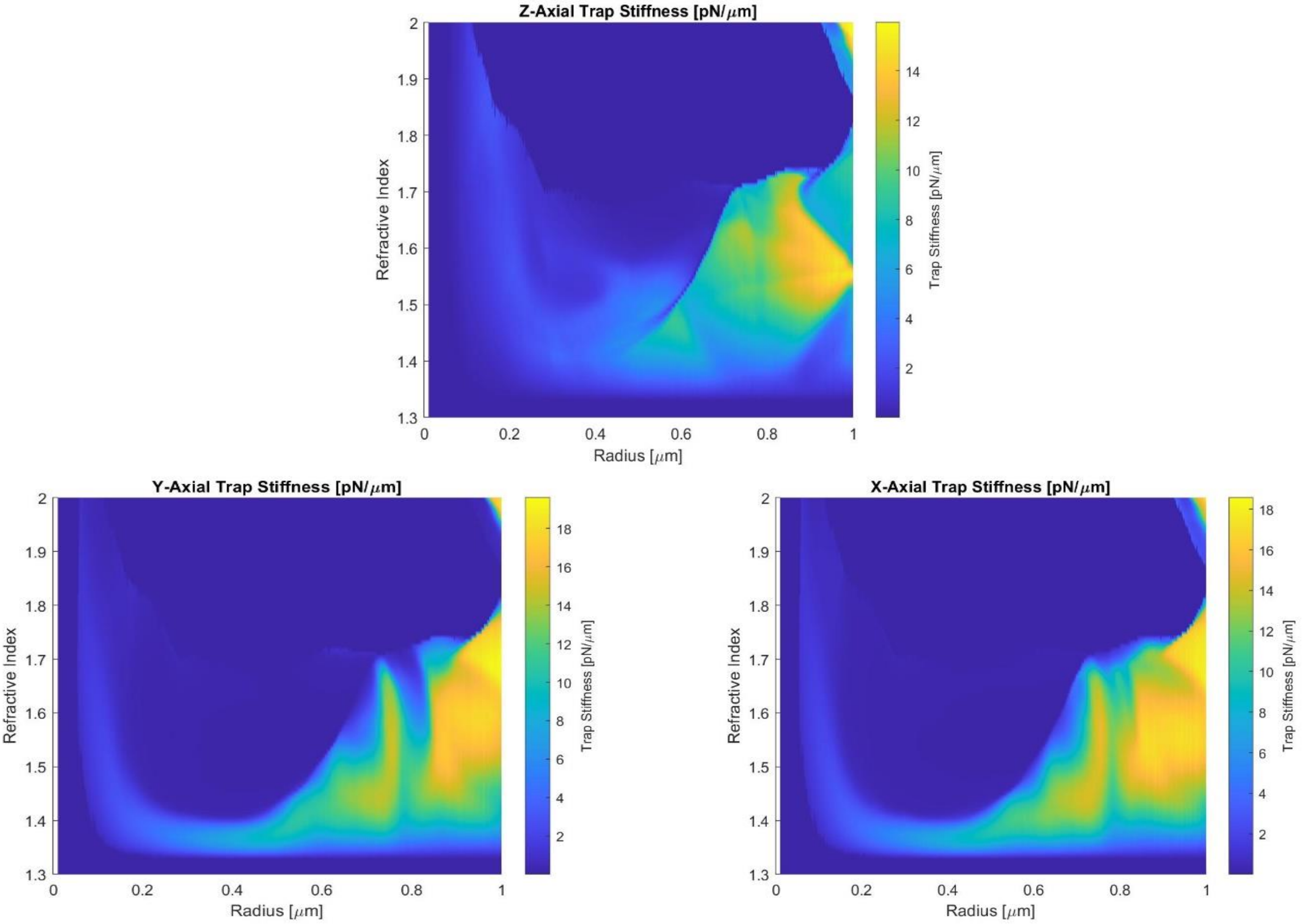


Fig2.5: The trap stiffness κ in all 3 dimensions. The radius is the X axis, the refractive index the Y.

The trap stiffness in $X - Y$ is similar, however there are some differences. Unfortunately, there is a large region of untrapped particles ($\kappa = 0$). This may be physically reasonable, but the largest issue is uniformity.

The data used to train the network must for a 1 to 1 map to each co-ordinate in (r, n_p) space. Even using all 3 trap stiffness measurements, the information is identical in the no-trapping region. All those points in dark blue will be unable to teach the network. This is a limitation of optical tweezers, not this method. It is caused by having a scattering force stronger than the gradient force. [3] That is, the particle will be pushed away from the equilibrium. Therefore, there is no trapping region, and no linear trap stiffness. Appx.5 shows the how this data would be removed when teaching a network.

To further analyse the data, the difference between X and Y trap stiffness can be analysed.

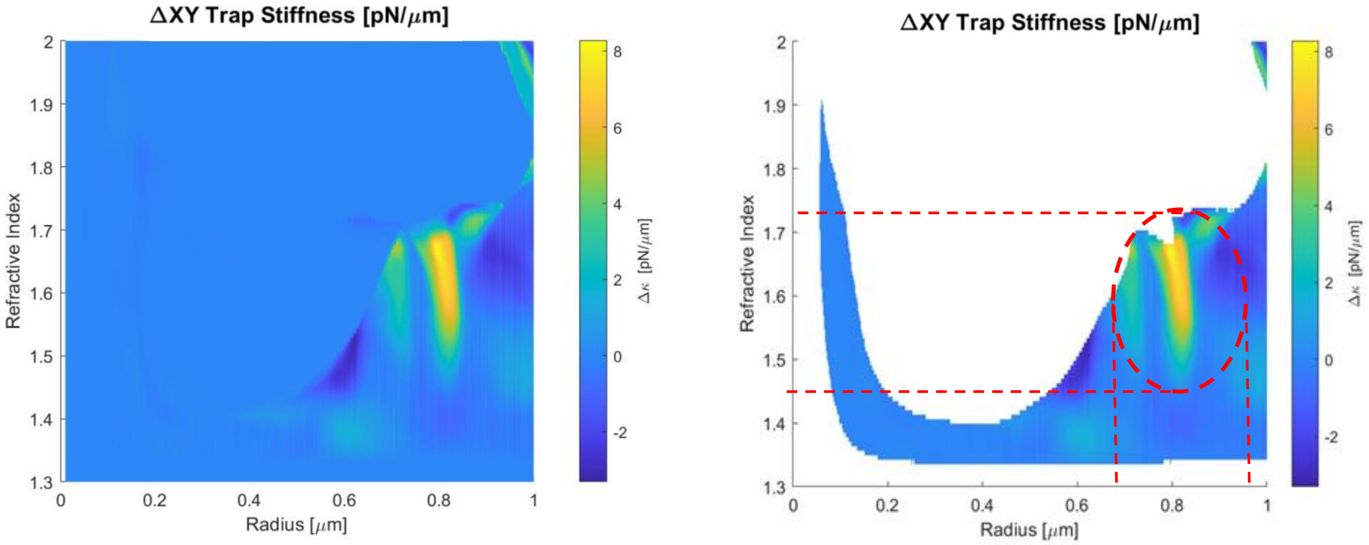


Fig2.6: Difference in X and Y trap stiffness. This is calculated as $(\kappa_x - \kappa_y)$ for each point. The difference has not been made absolute to notice any possible trends, i.e. whether one is consistently higher than the other. The right image has the possible region of uniqueness identified.

The region of most discrepancy also corresponds to the interesting region around $0.8\mu m$ where after $n_p > 1.5$ the stiffness decreases from the maximum (Fig2.5). This plot has also been placed in appx.5 without the untrapped region.

It would appear that trap stiffness can uniquely categorise our space – but only for a small region of points. The Z-axial measurement is not useful for training as it has $\sim 20\%$ uncertainty in any given measurement and requires an invalid approximation that the trap is linear about the equilibrium. See appx.5 for an image of the possible unique points in

Because the 5-DOF network was trained with a linear beam configuration, the ΔXY trap stiffness is not 0 in the trapped regions. This is important, as there are many shared trap stiffness values on a single X or Y trap stiffness landscape. Regions where $\Delta XY \neq 0$ are more likely to be uniquely defined by this representation. Unfortunately, fig2.6 demonstrates that this region is rather small. It has an approximate dimension of:

$$1.45 \leq n_p \leq 1.72, \quad 0.65 \leq r \leq 0.95\mu m$$

Which only a few points with significant difference.

For this reason, training a network with trap stiffness measurements may not be the most efficient method. The discussion will mention possible ways to adjust this measurement. Instead, networks created by *Lauren McQueen* and *Oscar Smee* will be analysed.

R3 Network Training:

Disclaimer: The following analysis has been written entirely by me with only verbal and written correspondence with the research owners. The images, and work required to create these images, are not my own and are attributed to each owner.

Two alternative methods for data generation and training were attempted by *Lauren McQueen (L.McQueen)* and *Oscar Smee (O.Smee)*. First, we will analyse the time series regression network created by *O.Smee.* The aim of his network is to take the positions and forces at each timestep and output the radius and refractive index. As a proof of concept, *O.Smee* created a network to identify the refractive index of a particle for a given radius $r = 0.6\mu m$ with only the forces.

His network can be conveniently described as mapping the following:

$$N(\bar{F}_1, F_2, \dots, \bar{F}_n) \to (n_p)$$

Where $\bar{F}_i$ are the forces in all 3 dimension (f_x, f_y, f_z) at the ith timestep. His simulation had the same structure as outlined in the method and initially created by *I.Lenton et al.*[1]. The particle was trapped, then *O.Smee* sequentially sampled forces at 1000 timesteps to train the network. He simulated a range of 5000 particles with refractive indices from $1.4 \leq n_p \leq 1.75$. His network structure was based on the convolution time series classificaiton network by *H.Fawaz et al.*[17].

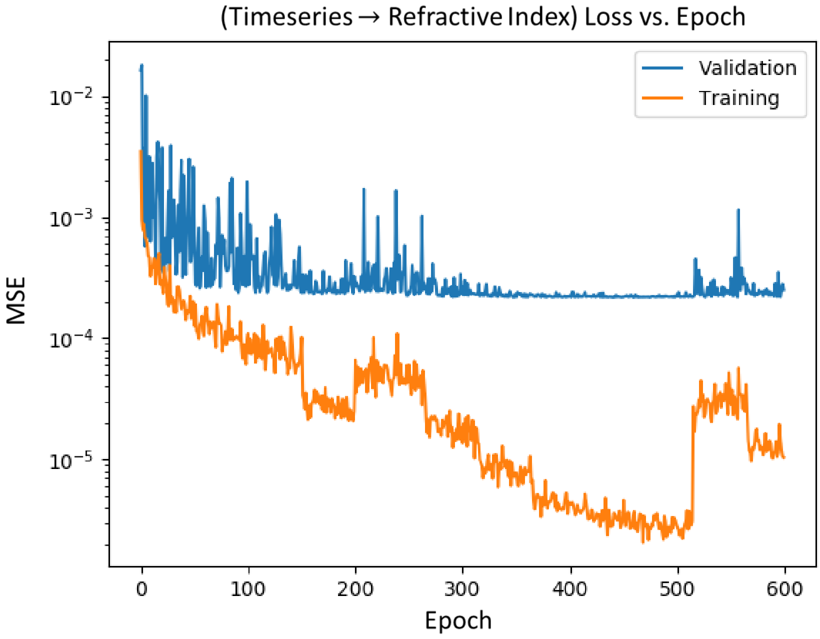


Fig.2.7: Training and validation of the time series network. This image was kindly supplied by Oscar Smee. The Y-axis describes the inaccuracy of the network as a mean squared error (MSE), and the X-axis is the amount of times the training data was run through the network. The orange data is the training dataset, and the blue a separated 10% for validation.

The plot from *O.Smee* demonstrates the increase in accuracy as the network trains. His network quickly increases in accuracy for the training and validation data until 400 epochs. After this point, the training data increases in accuracy with no change to the validation data, which may indicate overfitting.

His network is successful, as it has an error of $< 0.75\%$ after 300 epochs ($\%$ error not shown).

To identify if only a subset of the outputs were causing the error, *O.Smee* created the following plot by finding the error for each refractive index.

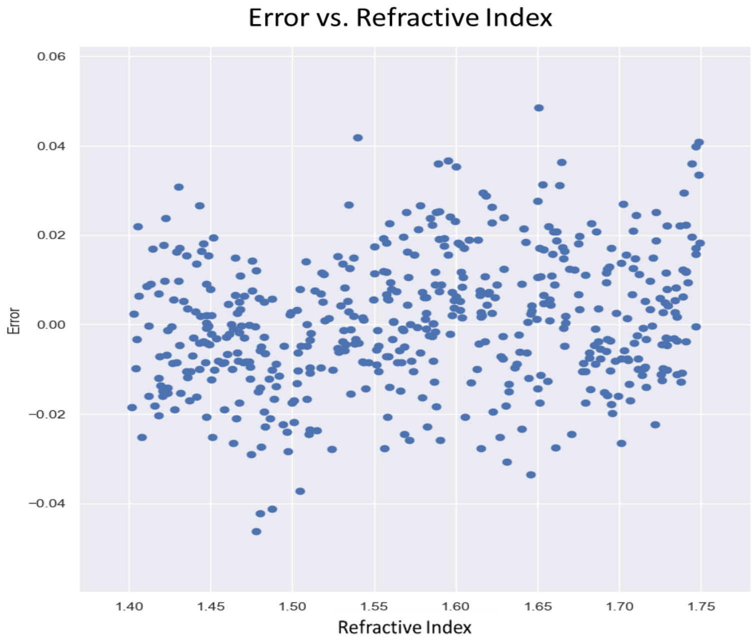


Fig.2.8: Relative error for each refractive index. Kindly supplied by Oscar Smee.

This plot demonstrates that the error in each refractive index prediction is uniform. That is, *O.Smee's* network is not struggling with any refractive indices more than others. This is an important result, as Fig2.6 would suggest that for $n_p > {\sim}1.6$ at $r = 0.6\mu m$ no trap stiffness analysis would work. This is because that region does not allow trapping.

This allows that time series analysis may be able to find refractive index and radius. His next step is including positions in the time series and generating a dataset for both refractive indices and radius.

L.McQueen developed a network to identify refractive index and radius from the probability distribution of the positions. This was done by forming a histogram with 50 bins and using the relative frequency as the input data. For this reason, the network is called a *histcount* network, as it uses only the bin frequency, not its position. To determine suitability, only the X-axial positions were used.

$$N(p_{x1}, p_{x2}, \dots, p_{xn}\ ) \rightarrow (r, n_p)$$

L.McQueen created 5000 simulations with the entire span of the 5-DOF network's space. She used 50 radii from $0.1 \leq r \leq 1\mu m$ and 100 refractive indices from $1.3 \leq n_p \leq 2$. Her network uses two layers of 128 fully interconnected nodes and performs a regression to predict the radius-refractive index pairs used to train the network.

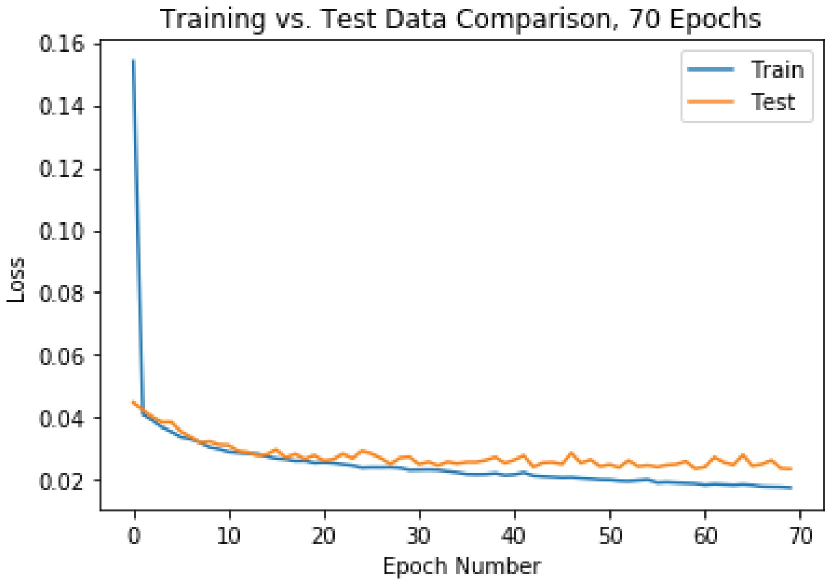


Fig.2.9: Training and validation of the histcount network, in blue and orange respectively. This image was kindly supplied by Lauren McQueen. The loss is defined as the relative error.

This image, created by *L.McQueen*, is another example of successful training. After ~ 10 epochs the validation (test) data is greatly improved, and then gradually improves after this. It reaches a minimum loss of 1.7% for the training data and a comparable 3.2% for the validation data.

Her next step is to include the histcounts in all 3 dimensions, and then the force histcounts. It is remarkable that the X-axial relative frequencies encodes enough information to identify the radius and refractive index alone.

Discussion

D1 Trap Stiffness Data Generation:

This project aimed to create a machine learning method to identify the radius and refractive index of a particle based on its motion. To achieve this, the 5-DOF network created by *I.Lenton et al.*[1] was used to create simulations of particles in an optical trap. The motion of the particle was then analysed and used in neural network training.

The simplest network architecture possible would take only a few inputs – which would therefore need to encode/describe both radius and refractive index. It was hypothesised that trap stiffness would achieve this goal. Fig2.1 demonstrates how trap stiffness is calculated from the variance, and how the simulation is run. A simulation time of $30s$ was deemed sufficient for gather a stable measurement of trap stiffness (Fig2.2 and Fig2.3). However, Fig2.2 also demonstrates that the error in κ_z does not converge. It fluctuates at $\sim 20\%$ even with $60s$ simulation. Appx.3 discusses why trap stiffness is not sufficient in describing the Z-axial motion. Mostly it is due to the scattering force causing a non-linear force, and therefore a skew distribution not described by variance alone.

Fig2.3 identified that with $30s$, κ_x and κ_y have an error of 5% - which is reasonable enough to attempt a network creation. Further analysis by Fig2.4 identifies that the trap is linear in X and Y, which allows that the values are physically reasonable.

The trapping landscapes were then generated in Fig2.5 across the entire space allowed by the 5-DOF network. Appx.5 shows these plots without the untrapped region. By reasoning that the inputs must form a unique 1 to 1 map with the outputs, only two unique pieces of information will yield a result. Fig2.6 demonstrates areas where there is a difference in the X and Y trap stiffness, which is caused by the linearly polarised light used in training the 5-DOF network.

Fig2.6 dissuades the use of trap stiffness as a training parameter, as there is only a small region of $\Delta\kappa_{xy} \neq 0$ This region is approximately enclosed by:

$$1.45 \leq n_p \leq 1.72, \quad 0.65 \leq r \leq 0.95\mu m$$

It is possible that the condition $\Delta\kappa_{xy} \neq 0$ is too harsh, and that having the same X and Y trap stiffness may still describe a unique point. However, it appears that better methods may exist.

Another issue with trap stiffness is discussed in appx.4. The precision of the measurements is determined with uncertainty analysis in Fig2.2 and Fig2.3. The accuracy is difficult to measure, as the theoretical trap stiffness from the OTT does not agree with the simulated trap stiffness.

I.Lenton et al.[1] trained the 5-DOF network with a given laser polarity and numerical aperture. It is also likely that some of the phenomena observed by trap stiffness are resultant from these quantities. Namely, there is a region in Fig2.5 where the Y-axial trap stiffness is 0 and the X-axial stiffness exists. This is entirely due to the linear polarisation of the light. It is not rotationally symmetrical. It is entirely possible that trap stiffness loses too much information to be a useful training parameter.

We would be limited to using trap stiffness networks for only linearly polarised light to ensure $\Delta\kappa_{xy} \neq 0$ and even then, to a very limited region. The purpose of creating a neural network is to allow as general a system as possible to be used. This is why it may be more beneficial to train a network instead of directly studying the phenomena. Otherwise it requires more time training the network than solving/modelling the problem. It may even be possible to describe non-gaussian beams if accurate simulations are done. This would require non-linear trap stiffness, and therefore a more general measurement.

There are refinements possible for the current simulation and calculation. It may be beneficial to do the PSD calculation described in I2. This may give a Z-axial trap stiffness. Adding a condition to terminate if the particle is not trapped would also be useful.

D2 Neural Networks and Optical Tweezers:

Despite the issues with trap stiffness, two networks were trained with alternative inputs. *Oscar Smee* successfully created a network which uses the time series of forces as an input and finds the refractive index for a given radius. Fig2.7 was supplied by *O.Smee* and demonstrates the success in training the network.

This proof of concept indicates that it may be possible to find both refractive index and radius if another piece of information is used. The positions can also be included in all 3 dimensions. However, for this to be generalised to an experiment, only X and Y measurements should be included. This is because they are measurable. His TSR network demonstrates that convolution is a powerful tool for large data inputs.

Lauren McQueen created a regression network which can determine both refractive index and radius with only the histcounts in one dimension. This is a useful result, as it allows that an experiment does not need measurements in multiple dimensions. The network currently has 5000 training datasets, however it is easy to increase this and include more information such as force or other position histcounts.

Both networks are remarkably general. The time series requires only that as much measured data as is recorded is entered. The histcount does not restrict the bin locations, as it does not care about the position, only the relative frequencies. A summary of the two networks is in Appx.6.

It would appear that physical analysis is not necessarily important when using machine learning methods. ANN are incredible correlation machines. In an age where gathering large datasets is possible, they will be the way of the future.

Optical tweezers are well suited to machine learning. There are many phenomena that could not be predicted without intense theoretical analysis or time intensive simulations. An example of this is the trapping landscapes generated in this report. There is no functional relationship between refractive index, radius and trap stiffness. It is a phenomena that cannot be predicted analytically. Yet, if we could teach a neural network to identify correlative trends, a 1 to 1 regression map could be created.

To further introduce machine learning to the field is the future endeavour of this project. More networks will be created to describe particle properties, and perhaps based off the work from *I.Lenton et al.* [1] there will be a shift to using ANN for dynamic simulations.

Conclusion

This project aimed to develop a machine learning method for finding a particles refractive index and radius based on its motion in an optical trap.

The first step was developing a dataset to train the neural network with. This dataset was based on simulations created with a 5-DOF network created by *I.Lenton et al.* [1].

The trap stiffness was identified as a useful parameter for creating a simple input network. Trap stiffness measurements were analysed with respect to the simulation time. It was found that simulations needed to run for $30s$ to have $\Delta\kappa > 5\%$ on the X and Y axis – and the Z-axial trap stiffness error did not converge. The trap stiffness landscape was then generated. It did not contain enough unique information for identification and training.

Two preliminary networks were trained by *Oscar Smee* and *Lauren McQueen* based on more data from the simulations. *Oscar's* network inputs the time series of forces during the motion and was effective at identifying refractive index within $< 0.75\%$ error. In the future, the positions will be included, and it may be possible to find refractive index and radius.

Lauren's network inputs the histogram of X-axial positions without the position label (only relative frequencies) and attempts to find the radius and refractive index. It was also successful, with $< 3.5\%$ error in validation data. In the future, the positions and forces in the other dimension will be used to increase accuracy.

References

[1] Lenton,I., Volpe, G., Stilgoe, A., Nieminen, T., & Rubinsztein-Dunlop, H. Machine learning enables fast statistically significant simulations of optically trapped microparticles (unpublished)

[2] Jones, P., Maragò, O., & Volpe, G. Optical tweezers (2015). Cambridge University Press.

[3] Knöner, G., Nieminen, T., Parkin, S., Heckenberg, N., & Rubinsztein-Dunlop, H. (2006). Calculation of optical trapping landscapes. *Optical Trapping And Optical Micromanipulation III*. doi: 10.1117/12.680114

[4] Nieminen, T., Loke, V., Stilgoe, A., Knöner, G., Brańczyk, A., Heckenberg, N., & Rubinsztein-Dunlop, H. (2007). Optical tweezers computational toolbox. Journal Of Optics A: Pure And Applied Optics, 9(8), S196-S203. doi: 10.1088/1464-4258/9/8/s12

[5] Knöner, G., Parkin, S., Nieminen, T., Heckenberg, N., & Rubinsztein-Dunlop, H. (2006). Measurement of the Index of Refraction of Single Microparticles. Physical Review Letters, 97(15). doi: 10.1103/physrevlett.97.157402

[6] Sarshar, M., Wong, W., & Anvari, B. (2014). Comparative study of methods to calibrate thestiffness of a single-beam gradient-force optical tweezers over various laser trapping powers. *Journal Of Biomedical Optics*, *19*(11), 115001. doi: 10.1117/1.jbo.19.11.115001

[7] Simmons, R., Finer, J., Chu, S., & Spudich, J. (1996). Quantitative measurements of force and displacement using an optical trap. Biophysical Journal, 70(4), 1813-1822. doi: 10.1016/s0006-3495(96)79746-1

[8] Berg-Sørensen, K., & Flyvbjerg, H. (2004). Power spectrum analysis for optical tweezers. Review Of Scientific Instruments, 75(3), 594-612. doi: 10.1063/1.1645654

[9] Schroeder, D. (2018). An introduction to thermal physics. India: Pearson Education Services.

[10] Gulli, A., & Pal, S. Deep Learning with Keras. Packt Publishing Ltd

[11] Kingma, D., Ba, J., Adam: A Method for Stochastic Optimization, ICLR 2015 conference paper

[12] Bui, A., Kashchuk, A., Balanant, M., Nieminen, T., Rubinsztein-Dunlop, H., & Stilgoe, A. (2018). Calibration of force detection for arbitrarily shaped particles in optical tweezers. *Scientific Reports*, *8*(1). doi: 10.1038/s41598-018-28876-y

[13] Stephenson, W., Wan, G., Tenenbaum, S., & Li, P. (2014). Nanomanipulation of Single RNA Molecules by Optical Tweezers. *Journal Of Visualized Experiments*, (90). doi: 10.3791/51542

[14] Volpe, G., & Volpe, G. (2013). Simulation of a Brownian particle in an optical trap. *American Journal Of Physics*, *81*(3), 224-230. doi: 10.1119/1.4772632

[15] So, J., & Choi, J. (2016). Tuning the stiffness asymmetry of optical tweezers via polarization control. *Journal Of The Korean Physical Society*, *68*(6), 762-767. doi: 10.3938/jkps.68.762

[16] Choi, J., & Noh, H. (2015). Investigation of the polarization-dependent optical force in optical tweezers by using generalized Lorenz-Mie theory. *Journal Of The Korean Physical Society*, *67*(12), 2086-2091. doi: 10.3938/jkps.67.2086

[17] Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. (2019). Deep learning for time series classification: a review. *Data Mining And Knowledge Discovery*, *33*(4), 917-963. doi: 10.1007/s10618-019-00619-1

[18] TensorFlow. (2020). Retrieved 29 January 2020, from https://www.tensorflow.org/

[19] Cristelli, M., Zaccaria, A., & Pietronero, L. (2012). Universal relation between skewness and kurtosis in complex dynamics. *Physical Review E*, *85*(6). doi: 10.1103/physreve.85.066108

Appendix

Appx.1 T-Matrix Method:

An electromagnetic wave can be described by the Helmholtz equation[2]:

$$(\nabla^2 + k)\boldsymbol{E} = \boldsymbol{0}$$

The electric field components can be solved in spherical co-ordinates separately as a sum of Hankel functions and vector spherical wave functions (VSWF)[1]. To model the forces on a particle in this electric field, we must consider how the particle changes the incident light. That is, how does the particle scatter the light. Conservation of momentum will allow the force to be found – provided we can describe the transformation of incident field E_i to scattered field E_s.

T.Nieminen et al.[3] described the summed VSWFs concisely with the following equations:

$$E_i = \sum_{n=1}^{\infty} \sum_{m=-n}^{n} a_{nm}\boldsymbol{M}_{nm}^{(2)}(k\boldsymbol{r}) + b_{nm}\boldsymbol{N}_{nm}^{(2)}(k\boldsymbol{r})$$

$$E_s = \sum_{n=1}^{\infty} \sum_{m=-n}^{n} p_{nm}\boldsymbol{M}_{nm}^{(1)}(k\boldsymbol{r}) + q_{nm}\boldsymbol{N}_{nm}^{(1)}(k\boldsymbol{r})$$

Where the VSWFs terms inside $\boldsymbol{M}_{nm}$ and $\boldsymbol{N}_{nm}$can be expanded in terms of the Hankel functions $h_n^{(1,2)}(kr)$ and spherical harmonics $Y_n^m(\theta, \phi)$[3]. The superscript (1) denotes outwardly propagating field and (2) inwardly. Provided the sum for the electric field is convergent for the incident and scattered light, a linear equation can be created[2,4]:

$$\begin{bmatrix} p_{nm} \\ q_{nm} \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} a_{nm} \\ b_{nm} \end{bmatrix}$$

$\boldsymbol{T}$ is the T-matrix and describes the particle which transforms the incident light to scattered light. It is constructed by solving for the boundary conditions of the particle and incident light – along with several symmetry arguments. For a sphere, it is conveniently a diagonal matrix.

The force is then extrapolated by summing and comparing the coefficients – which are describing how much light is in each VSWF mode[4].

$$Q = \frac{2}{P}\sum_{n=1}^{\infty} \sum_{m=-n}^{n} \frac{m}{n(n+1)} Re(a_{nm}^{*}b_{nm} - p_{nm}^{*}q_{nm}) - N_{nm}Re\big(a_{nm}a_{n+1,m}^{*} + b_{nm}b_{n+1,m}^{*} - p_{nm}p_{n+1,m}^{*} - q_{nm}q_{n+1,m}^{*}\big)$$

Q is the trapping efficiency, a normalised force, which is related to the SI quantity by[2,4]:

$$\boldsymbol{F} = \frac{n\boldsymbol{Q}}{c}P$$

Where n is the refractive index of the medium and P the beam power. Numerically the power is described by the magnitude of incident light[3]:

$$P = \sum_{n=1}^{\infty} \sum_{m=-1}^{n} |a_{nm}|^2 + |b_{nm}|^2$$

The Optical Tweezer Toolbox (OTT), created by *T.Nieminen et al.*[4], is a MATLAB library which allows incredible modelling of laser tweezers. The toolbox allows incident beams and particles to be created with a variety of parameters. It then solves the T-matrix and allows forces to be calculated.

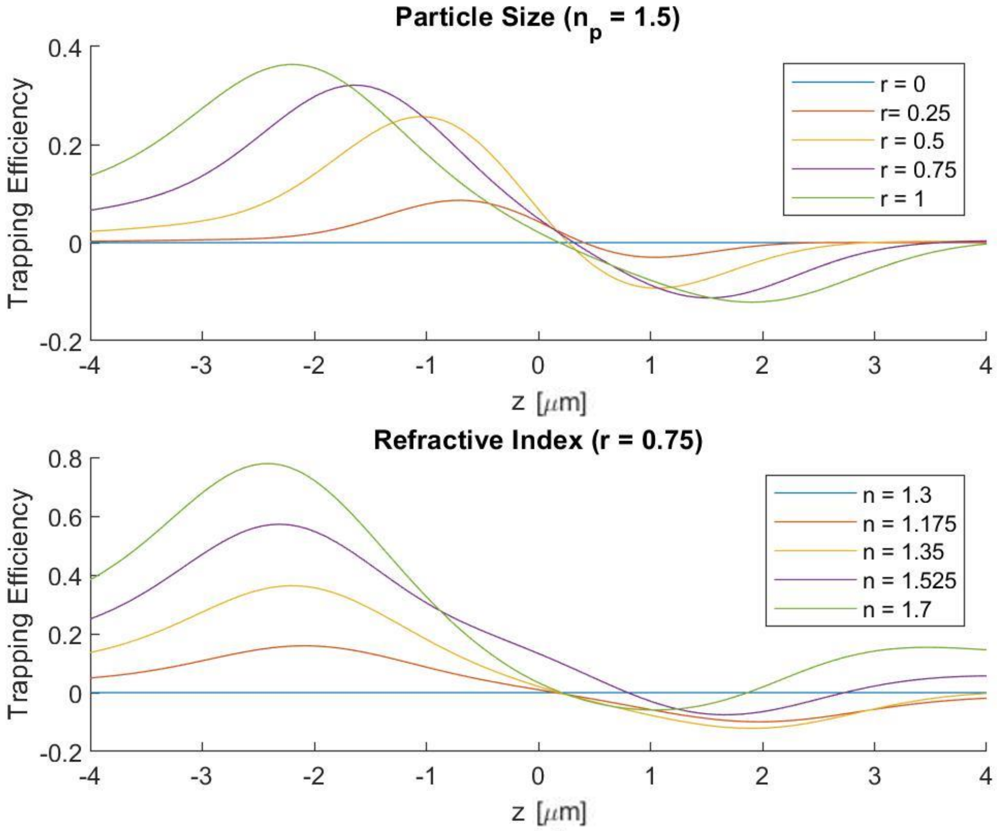


Fig1.1: Example of OTT trapping efficiency calculations for various particle sizes and refractive indices. The force here is along the axis of propagation, i.e. the Z-axis. Lengths are normalised to the wavelength of laser light.

The trapping efficiency curves here demonstrate the forces a trapped particle would observe when moved from the equilibrium position. It is worth noticing, the equilibrium position is not the focus of the laser, which is $Z = 0$.

To characterise the relative strengths of these traps, the linear region near each equilibrium can be studied and the trap stiffness extrapolated.

Appx.2 Neural Networks:

An Artificial Neural Network (ANN) at its most basic level takes N inputs and returns M outputs by mapping $N \rightarrow M$. The network guesses the most likely output based off what it was taught. The structure of each network is different however, they all rely upon linked perceptron nodes. A perceptron node takes an input x, and depending on its weight w_i and bias b outputs a number[9]. The simplest type of perceptron is a binary node [10]:

$$f(x) = \begin{cases} 1, & \sum_i (w_i x + b) > 0 \\ 0, & otherwise \end{cases}$$

This function is called an activation function, as it determines whether the perceptron 'activates' and passes a signal to the next layer. Each perceptron is allowed to determine the best possible weight without any control from the user. They are assigned by giving the network labelled data and initialising in a random state. The network then guesses the output given the input data, and because it is labelled one can measure how accurate the guess was. There are many functions to define the loss, the measure of inaccuracy, and the learning rule. Fundamentally, all perceptron which guessed the correct output are strengthened in their weights, and all incorrect guesses are weakened [9].

Often a system cannot be determined by a single layer of nodes. Multilayer Perceptron (MLP) networks use multiple layers of interconnected nodes to find solutions. Often each node in one layer is connected to every node in the next layer. The relative weights between these layers is adjusted to minimise loss and often structures will form[10] – however the most significant difference to other computational methods is these structures are irrelevant and uncontrolled.

Most MLP use activation functions which allow non-integer outputs [10]. This means that different perceptron outputs can sum to the next layer and give a larger range of signal strength. Two common functions are sigmoidal f_σ and rectified linear unit (ReLU) f_{ReLU} given by [10]:

$$f_\sigma = \frac{1}{1 + e^{-(wx+b)}}, \qquad f_{ReLU} = \max(0, wx + b)$$

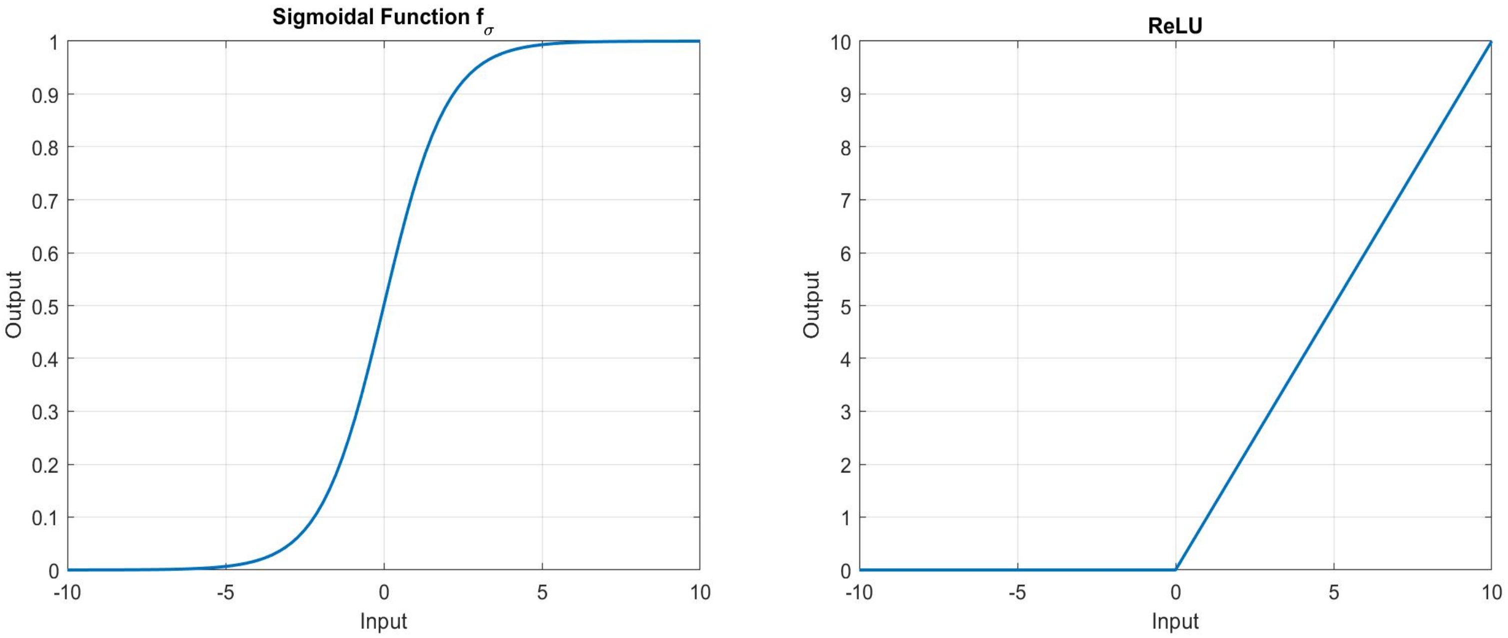


Fig1.5: The two most common activation functions, Sigmoidal and ReLU. The x-axis represents the signal received from the previous layer, and the y-axis is the sent output which will be weight adjusted before the next perceptron.

The sigmoidal function at the limits $\sum_i(w_i x_i + b) \gg \pm 10$ approaches binary outputs – however in the middle there is a continuous region of non-integer values. The ReLU function is 0 when $\sum_i(w_i x_i + b) < 0$, then a simple linear signal after.

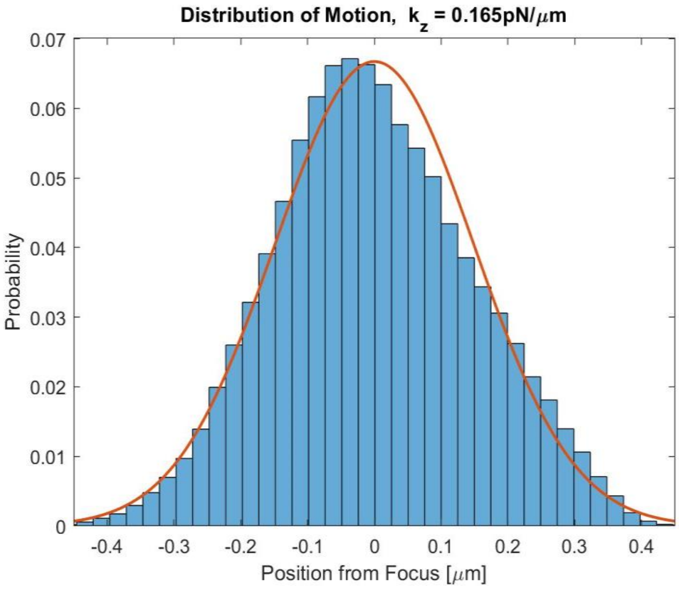
When training the network, a loss function must be defined which will be optimised by the network. This loss needs to include the network output and the labelled known data. Then to learn, the network requires an optimiser which describes how the weights should change given the loss [10].

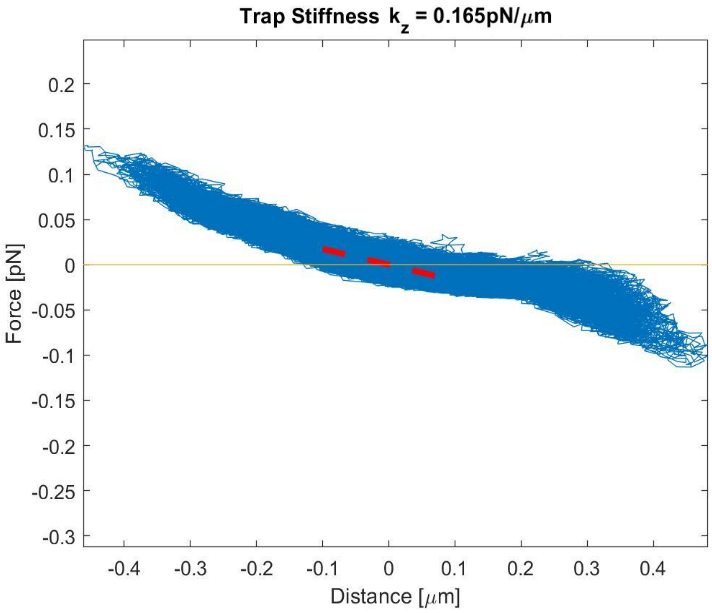
Most simply, the loss function can be mean squared error [10] where Y represents the output of the network, or the real value of the data:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_{network} - Y_{data})^2$$

One of the most commonly used optimiser (learning rule) is adaptive moment estimation (Adam). *D.Kingma et al.* [11] introduce and describe the algorithm in their paper *Adam: A Method for Stochastic Optimization*. Stochastic optimization relies upon taking a gradient of the activation functions with their inputs and weights. It then updates the weights by subtracting a measure for how incorrect each perceptron was, along with a learning rate which is usually fixed. The Adam algorithm takes the gradient between the output and weights and iteratively finds a converging new weight value which optimises the output by changing the learning rate each step [11].

Appx.3 Z-Axial Motion:





The Z-Axial trap stiffness is not well approximated to a linear spring constant. This is evident by the skew distribution (left) when fit by equipartition theorem. It is less apparent in the force-position diagram (right), however when compared to fig2.1 and fig2.4 it becomes more obvious.

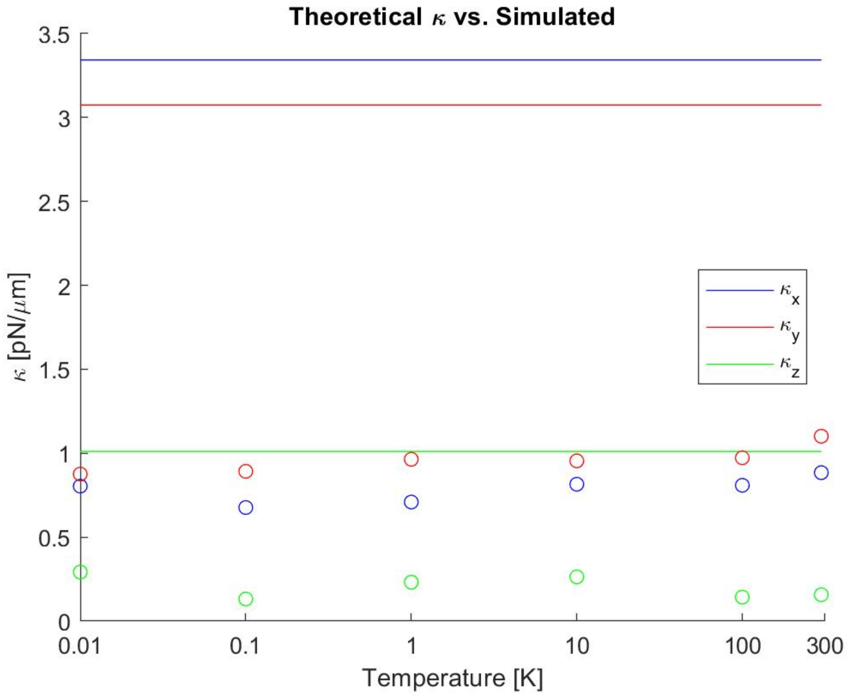
What this means is variance does not fully describe this dimension. Variance is a 2nd order statistic[19] and along with the mean can fully describe a Gaussian distribution (fig.2.1 for example). Skew is a 3rd order variance and describes the bias of the measurements. This would be necessary in describing our non-linear Z $-$axial force. Perhaps the neural network could simply take the skew, variance and mean of the Z positions.

If the skew still did not give enough information for the system, then kurtosis could be used. Kurtosis is a 4th order variance which describes the 'tailing' of a distribution[19]. That is, the extremities of the x-axis.

These statistical measurements describe the data better. They will still miss some unique behaviours of our particle in the trap. That is why it may be beneficial to include the entire motion as an input for training the network.

Appx.4 Theoretical vs. Simulated κ:

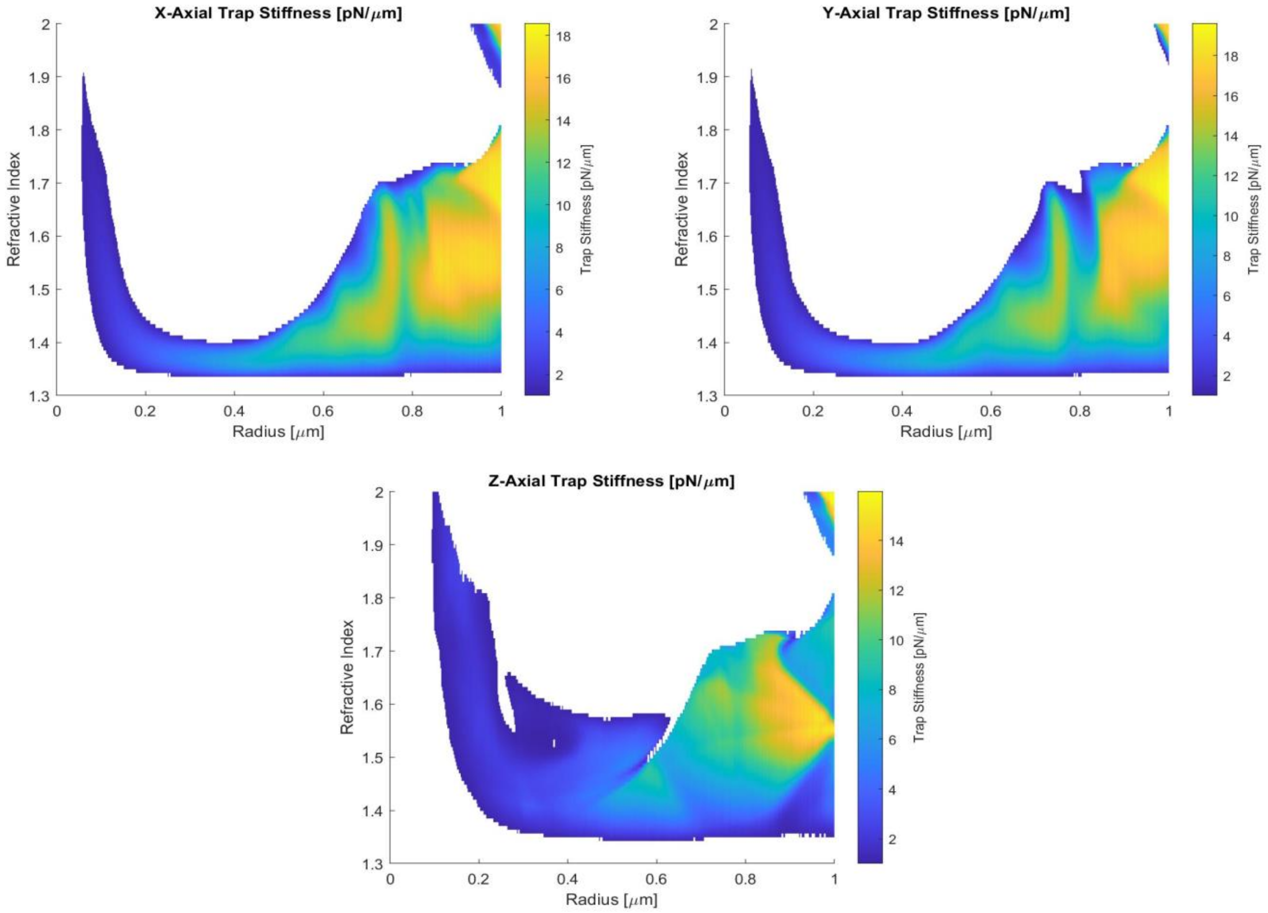
To compare theoretical trap stiffness, the OTT can be used to generate a force-position curve. The gradient about the equilibrium can then be found. Fig1.4 shows this process. These gradients are then compared to a 5-DOF simulation at different temperatures with the same radius and refractive index.



The straight lines are the OTT measurements, and the datapoints are the simulated values. The beam in the OTT was given a linear polarisation.

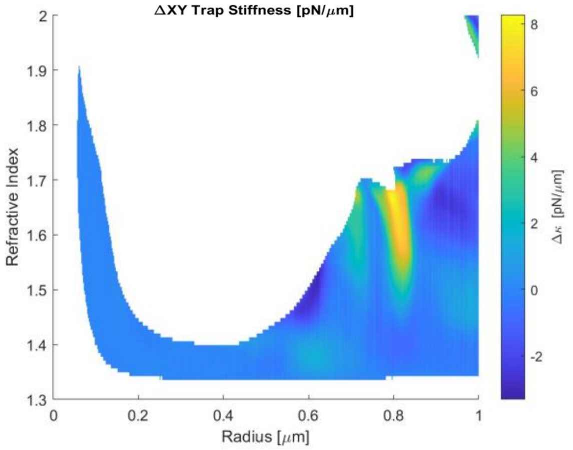
The simulation calculates the force by the T-matrix method – where the scattered light is compared to the incident light. Whereas the 5-DOF simulation extrapolates the force from its training. Fig1.6 and *I.Lenton et al.* [1] demonstrate that the force has agreement between OTT and 5-DOF. Therefore, it is the method of finding the trap stiffness which causes the difference. \

Appx.5 $\kappa = 0$ Adjusted Landscapes:



This plot is fig2.5 with the regions of no-trapping removed. This was done by defining a minimum trap stiffness of $0.1pN/\mu m$ – the theoretical $\kappa = 0$ spots still have a measured variance from the simulation method. For further refinement, the simulation should include a termination if the particle isn't trapped.

It is important to note that only a point with a non 0 measurement in 2 plots will form a 1 to 1 map. This is because two pieces of information are being guessed by the network, and it would be impossible to uniquely define both outputs from a single input via regression.



This plot shows the difference between X and Y in the trapping landscape.

Appx.6 Network Comparison:

Three network structures were attempted in this project. A trap stiffness network by *L.Hamilton*:

$$N(\kappa_x, \kappa_y, \kappa_z) \rightarrow (r, n_p)$$

A convolution time series regression network by *O.Smee*:

$$N(\bar{F}_1, \bar{F}_2, \dots, \bar{F}_{1000}) \rightarrow (n_p)$$

A hiscount network by *L.McQueen:*

$$N(p_{x1}, p_{x2}, \dots, p_{x50}) \rightarrow (r, n_p)$$

The trap stiffness network would require the least amount of input data and train the fastest. However, it includes less data from the particles motion. It also requires processing of the data before feeding it to the network. The trap stiffness was shown in this report to be unsuitable for network training.

The time series regression network by *O.Smee* contains the most information, however required a large amount of time to train (> 300 epochs). The accuracy with the forces alone was very high $< 0.75\%$ - however it has not yet been generalised to find both radius and refractive index. It is easily upgradable to include the position time series.

The hiscount network by *L.McQueen* is an intermediate step. It takes the slightly processed data of the position's relative frequency by assigning it to histogram bins. However, all the positions are used, and therefore a large amount of data is kept. The training time was relatively short (> 70 epochs) and with a reasonable error $< 3.5\%$. This network can be upgraded to include positions in more dimensions and the forces.

O.Smee's network is computationally expensive to train. Once it is trained however, it will be very powerful. The forces and positions calculated during an experiment could be directly fed to the network, and radius and refractive index could be found. It is incredibly general, as it does not require a specific amount of data.

L.McQueen's network is inexpensive to create for specific situations. If particular trap parameters were needed (e.g. different polarisation or NA) then OTT simulations could be used to train the network. This would require less time to train than the TSR network.