TITLE

TU Delft, Faculty of Applied Sciences,
BSc program Applied Physics

Delft, 25th June 2020

4914058 van Loon, Reinaart
4645197 Sangers, Jeroen

1 Abstract

This report contains our findings on methods of finding the trap stiffness of an optical trap or optical tweezer setup and the conversion of an MATLAB data processing script to Python. Optical trapping is a technique in which a small particle is being held in place by a focussed laser beam, commonly used in the field of biophysics.

The trap stiffness in relation to laser output power was determined by photographing a particle in an optical trap multiple times for a set laser power output setting, then repeating this for multiple laser power output settings. The resulting image stacks were processed by an automated MATLAB script that calculates the movement of the centre of mass of the bead and uses this to calculate the restoring force of the optical trap against the Brownian motion of the particle. The resulting trap stiffnesses were then plotted for each laser power output setting. A linear relation between the trap stiffness and the laser output was fitted, the results are shown below.

Due to the current Corona-virus outbreak we were unable to do the optical trap measurements ourselves so to make up for this we were also tasked with rewriting part of a MATLAB script in Python. Whilst doing this a problem presented itself, there is no efficient way to interpolate a non-evenly spaced data grid in Python. With no knowledge of how to write such a function ourselves we opted to use a function that was able to interpolate at one point at a time which resulted in a slowdown of the data processing speed and allowed errors to creep in. The two other functions that had to be rewritten were the subpixel interpolation function and the symmetry centre finding function, both of these our now working in Python. The presented Python file is therefore capable of roughly following the symmetry centre of the bead in an optical trap.

coefficient for	a_x $[pN/(nm \cdot mW)]$	a_y $[pN/(nm \cdot mW)]$	a_{tot} $[pN/(nm \cdot mW)]$
dataset 1	$1.382 \cdot 10^{-6}$	$1.280 \cdot 10^{-6}$	$1.740 \cdot 10^{-4}$
dataset 2	$1.478 \cdot 10^{-5}$	$3.974 \cdot 10^{-6}$	$1.551 \cdot 10^{-5}$

Table 1: The results of the linear fits for the trap stiffness as a function of laser output power. The shown a-factors satisfy the least squares fit of the function: $k_i = a_i \cdot P$

Contents

2 Introduction

An optical trap, or optical tweezers, is a technique that is frequently used in molecular biology to study particles at the micro- and nanometre scale. By trapping a particle in a focussed laser beam, the particle is limited in movement. This allows the user to study microscopic manipulations and measurements on microscopic particles and therefore proves to be very useful in the field of biophysics. Examples of which are sorting of cells, unzipping of DNA and enzyme interactions [**?**][**?**]. In order to perform quantitative measurements using optical tweezers, it is vital to know what force it exerts on the particle. This force is defined by the trap constant and is dependent on various parameters such as the particle size and laser power. The aim of this report is to find the relation between the laser power and the trap constant. Secondly this report serves to get familiarised with the optical tweezers technique and to investigate its limitations and possibilities. Due to the COVID-19 virus no experiments were carried out for this report and data of previous experiments by other students is used. This data contains images of a trapped bead for laser beams with different powers. For each laser intensity, there is a set of images at fixed time intervals such that the movement of the bead can be studied. For this report, the images are processed by a MATLAB algorithm which calculates the trap constant in two perpendicular directions. The relation between each trap constant and the laser power is found using a distance regression method and the theoretical linear dependence. Another part of this practicum involves designing a PYTHON algorithm that can perform the same calculation as the provided MATLAB code. In this report, the result of our programming is described and suggestions are made for improvement of the PYTHON code and further analysis of the trap constant is different directions. In section 2 the theory regarding the report will be described followed by the experimental method in section 3. The results and discussion can be found in section 4. Lastly the conclusions in section 5.

3 Theory

the optical tweezers technique was used in the experiment that corresponds with the data that is used in this report. The basic principles of optical tweezers and the theory for further calculations is described in this section.

3.1 Optical trapping

To understand the working principle of optical tweezers we consider a spherical dielectric particle, a bead, in a coherent light beam with a symmetrical intensity gradient such as in figure **??**. The light beam will exert a force on the bead in the direction of the highest light intensity of the gradient. To understand this, we need to consider two situations. For the situation in which the dimensions of the bead are much greater than the wavelength of the light we can apply straight forward ray optics. In the situation where the size of the bead is much smaller than the wavelength we can approximate the bead as a dipole that feels Lorentz force due to a gradient in the electric field. For the situation where the dimensions of the bead are much larger than the wavelength of the light beam, we consider that photons can exert a radiation force on the bead. This force is a result of the momentum that photons carry and will be directly proportional to the light intensity. We now consider two rays of light that reach the bead symmetrical with respect to its centre. Due to the bead's spherical symmetric shape, the light rays will be refracted by the dielectric particle at the same rate, but in opposite directions (see figure **??**). Both light rays will, given the change in direction of the light and the third law of Newton, exert a force on the bead. The light ray with the higher intensity will, however, exert a greater force. If the intensity gradient is greater in the centre of the light beam, such as in figure **??**, this would lead to a net force pointing in the direction of the symmetry axis of the light beam. This force would trap the bead to the centre on the beam. In the case of a the beam of light being focussed such as in figure **??**, the bead would not only be trapped in the direction perpendicular to the beam axis, but also in the direction of the axis. This is also a result of the change of the refraction of light exerting a force on the bead (see figure **??**). However, to light scattering, the bead is in the axial direction trapped slight behind the waist of the light beam.[**?**] In the situation where the dimensions of the bead are much smaller of than the wavelength of the light beam, we approximate the bead as a perfect dipole. According to Sheavitz (2006) [**?**], if we also consider the laser to have a Gaussian intensity profile in the plane perpendicular to propagation, the Lorentz Force is given by:

$$F = (p \cdot \Lambda)E + \frac{1}{c}\frac{d\,p}{dt} \times B \tag{1}$$

For continuous wave lasers, which are commonly used for optical trapping, this can be simplified to:

$$\langle F \rangle = \frac{\alpha}{2}\Lambda\langle E^2 \rangle \tag{2}$$

Where α is the induced dipole moment of the bead. Most optical trapping and also the experiment in this report includes beads with the same order of magnitude dimensions as the wavelength of the light beam. The physics of such a system is complicated and somewhat in between the cases explained above. This comprehensive theory will not be discussed in this report. According to Shaevitz (2006), 'for small motions of a bead near the center of an optical trap, the forces acting on the bead approximate a zero rest–length, linear spring at the trapping center.' Therefore, for small motions of the bead, the stored energy in the optical 'spring' is $1/2k_{trap} < x^2 >$ with k_{trap} a constant defining the strength of the optical trap and $< x^2 >$ the variance in the motion. According to the equipartition theory, the energy of the Brownian motion of a particle is given by $1/2k_bT$ with k_b the Boltzmann constant and T the temperature.[**?**] Equating the two energies yields:

$$k_{trap} = \frac{k_BT}{< x^2 >} \tag{3}$$

From this we can conclude that by following the position of the bead over time, it is possible to find the the value of k_{trap}. In the latter definition of k_{trap}, the 3 dimensions of real life are not taken into account. In this report we will only consider the trap constants in the plane perpendicular to the propagation direction of the laser beam. This plane will in this report be addressed as the plane of interest, POI. For the POI we can consider two definitions for k_{trap}. We define $k_{trap,r}$ as the 'average' trap constant and is calculated using only the motion of the bead in the radial direction. $k_{trap,r}$ gives a good indication of the force in any arbitrary direction in the POI. It does, however, ignore the shape of the probability distribution of the bead. In the case where the potential well would be elongated such as in figure **??**, the values for $< x^2 >$ and therefore also k_{trap} can differ depending on the direction. We define the trap constant in an arbitrary direction as $k_{trap,i}$. Note that this is defined as a line in the POI which cuts the expectation value of the bead. To find the value for $k_{trap,i}$ we first realize that for realistic laser beams that are used for optical trapping, we expect a 2-dimensional gaussian intensity profile in the POI [**?**]. The shape of this gaussian profile can be described by a 2 dimensional covariance matrix. The iso-contours and therefore also the variance for such 2-dimensional gaussians are ellipses with their centre at the expectation value [**?**] (see figure **??**. According to Rojas (2009), the eigenvectors of the covariance matrix point in the direction of the axis of an ellipse describing the variance in the POI. The largest of the two eigenvectors, $\vec{v}_1$, will point in the direction of the major axis and the smallest eigenvector, $\vec{v}_2$, in the direction of the minor axis. The magnitude of the semi-major axis and semi-minor axis, a and b respectively, are given by the eigenvalues corresponding to the eigenvectors.[**?**] In order to find the variance in any arbitrary direction in the POI, $< x_i^2 >$, we use the formula for an ellipse in polar coordinates:

$$< x_i^2 > (\theta) = \frac{a\, b}{\sqrt{(b\, cos\theta)^2 + (a\, sin\theta)^2}} \tag{4}$$

Where θ corresponds to the angle of the direction of interest with respect to the major axis and the ellipse centre as origin. Using equation 3 with the value for $< x_i^2 >$ will yield a correct value for $k_{trap,i}$.

3.2 Error calculation

For this report, since we are not fully acquainted with the set-up and the corresponding error, when no error is specified the error is estimated to be half of the finest scale. For example for a size of 1.34 meter, the error would be 0.005 meter.

If Y is a variable which is a function of A,B,C, ... Then the error of Y, $u(Y)$, is given by equation 5.

$$u(Y) = \sqrt{\left(u(A)\frac{\partial Y}{\partial A}\right)^2 + \left(u(B)\frac{\partial Y}{\partial B}\right)^2 + \left(u(C)\frac{\partial Y}{\partial C}\right)^2 + ...} \tag{5}$$

4 Experimental Method

No experiments were carried out for this report, but measurements from previous students are used. The experimental set-up that they used and the computer algorithms that were used for this report will be discussed in this section.

4.1 Experimental set-up

The experimental set-up that was used for this report can be seen in figure 1. According to the practicum manual, the red light from the laser has a wavelength of$\lambda = 658$ nm and passes through a beam expander in order to completely fill the back (back focal plane) of the objective. The beads that are used in the experiment have a diameter of approximately $2\mu m$. The mirrors M1 and M2 are used for compacting the beam path and aligning the beam to the optical axis of the objective. The used objective is meant to be used in an infinity corrected microscope. L4 focusses the light back to an image at focal distance. In the set-up, the L1 and L2 have focal length of respectively 50 and 350 mm. The front focal distance of the objective lens is 1.8 mm and the second objective lens has a focal length of 200 mm. Using simple division and equation 5 we find that the magnification is $M = \frac{200}{1.8} = 1.1 \ \pm 0.3 \cdot 10^2$. Given the pixel size of the camera of $5.2\,\mu$ and equation 5 we find that the conversion factor for pixels to length is given by $l_{pixel} = \frac{5.2}{M} \cdot 10^{-6} \approx 4.7 \ \pm 0.1 \ \cdot 10^{-8} m/pixel$.

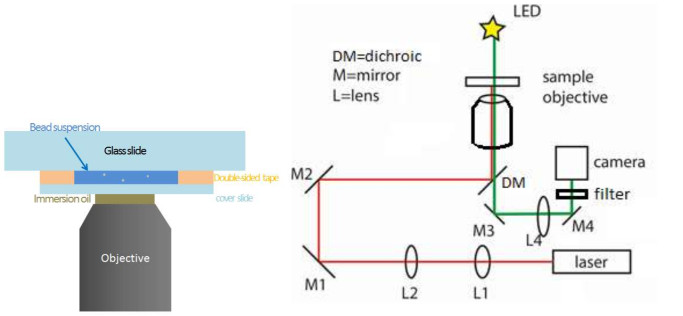

Figure 1: Schematic diagram of the experimental set-up that was used for the experiments corresponding with the treated data in this report. This diagram was taken from the practicum manual [**?**].

Using the discussed set-up, a bead was to be trapped by a laser beam and images were taken at fixed time intervals. This was performed for laser beams with different powers. The involved powers were 0, 5, 10, 20, 30 and 40 mW. For this report, two data sets from different students are analysed.

4.2 Computation

A MATLAB algorithm was provided for this practical in order to calculate the trap constant in a x- and y-direction, $k_{trap,x}$ and $k_{trap,y}$. This algorithm involves noise removal of the images and tracking the bead using a 'quadrant-interpolation (QI) algorithm which makes use of the circular geometry of the diffraction pattern to resample the image on a circular grid.' [1] Subsequently, power spectrum analysis is performed to calculate the values for $k_{trap,x}$ and $k_{trap,y}$. Using this MATLAB code and a linear regression method, the correlation between the laser power and the trap constants are determined for the two data sets.

Secondly, a PYTHON algorithm is used with the same noise cancelling technique, but with a trackpy function to follow the bead. This algorithm subsequently calculates the covariance matrix and the vectors describing the covariance ellipse. Using this PYTHON code, the change of this covariance ellipse is analysed.

5 Results

5.1 Results of the datasets

The results of the experiment were gathered by taking a 1000 photos of the particles in the trap and for each of the laser power output (P) settings and then using an automated MATLAB script to track the location of the centre of the bead for all of the 1000 photos and all of the laser output power settings. Using this information the script was able to calculate the trap stiffness in the x and y direction denoted as k_x and k_y respectively.
The script was not perfect, for most of the image stacks of the first dataset the script was not able to track the centre of the bead, this was the result of a piece of debris in the trap which was like the bead circularly shaped with rings around it, this threw the program off. The corresponding data points have been entered in the table 2 below but the points were not used for fitting a line to the dataset. The second dataset was much better and did not have any faulty measurements, its values are shown in table 3.

Laser Power $[mW]$	0*	10*	20	30*	40	40
k_x $[pN/nm]$	$2.36 \cdot 10^{-5}$	$2.45 \cdot 10^{-7}$	$1.60 \cdot 10^{-5}$	$8.35 \cdot 10^{-7}$	$1.99 \cdot 10^{-5}$	$9.60 \cdot 10^{-5}$
k_y $[pN/nm]$	$8.29 \cdot 10^{-5}$	$2.94 \cdot 10^{-7}$	$1.65 \cdot 10^{-5}$	$9.71 \cdot 10^{-7}$	$1.72 \cdot 10^{-5}$	$8.96 \cdot 10^{-5}$

Table 2: Results of the first dataset. The values are truncated to two decimal places. * denotes a faulty measurement

Laser Power $[mW]$	0	5	10	20	30	40
k_x $[pN/nm]$	$3.64 \cdot 10^{-7}$	$8.24 \cdot 10^{-5}$	$1.08 \cdot 10^{-4}$	$3.03 \cdot 10^{-4}$	$6.14 \cdot 10^{-4}$	$7.55 \cdot 10^{-4}$
k_y $[pN/nm]$	$5.55 \cdot 10^{-7}$	$4.18 \cdot 10^{-5}$	$2.53 \cdot 10^{-5}$	$1.10 \cdot 10^{-4}$	$1.69 \cdot 10^{-4}$	$2.54 \cdot 10^{-4}$

Table 3: Trap results, values are truncated to two decimal places for formatting reasons

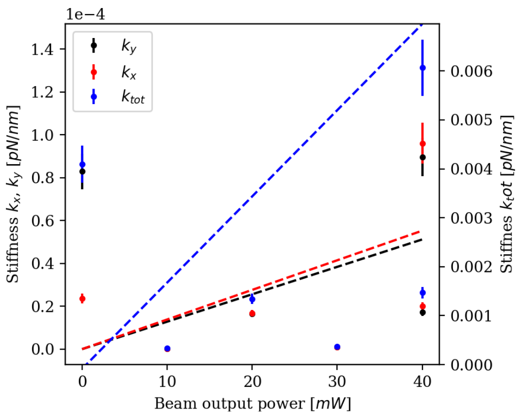

Figure 2: Results of the first dataset plotted and fitted.

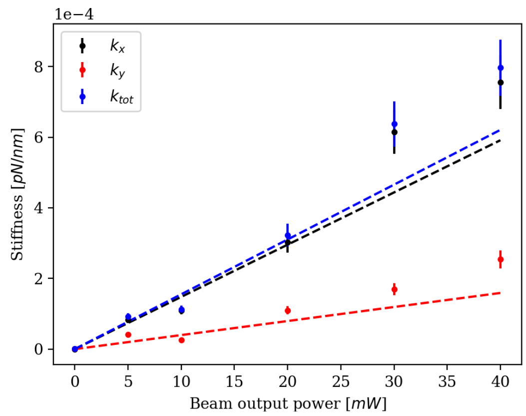

Figure 3: Results of the second dataset plotted and fitted.

The data shown above in tables 2 and 3 are plotted in figures 2 and 3 respectively. In the plots the individual data points and their corresponding error flags are shown as well as the least squares fit of these data points for the k_x, k_y and k_{tot} values. The fit of the k_{tot} values is plotted on a secondary axis.
For the least squares fit a relation of $k = a \cdot P$ was used since it is expected that for zero laser output power there would not be a restoring force acting on the particle. The resulting coefficients are shown below in table 4.

coefficient for	a_x $[pN/(nm \cdot mW)]$	a_y $[pN/(nm \cdot mW)]$	a_{tot} $[pN/(nm \cdot mW)]$
dataset 1	$1.382 \cdot 10^{-6}$	$1.280 \cdot 10^{-6}$	$1.740 \cdot 10^{-4}$
dataset 2	$1.478 \cdot 10^{-5}$	$3.974 \cdot 10^{-6}$	$1.551 \cdot 10^{-5}$

Table 4: The results of the linear fits for the trap stiffness as a function of laser output power.

5.2 Python program results

During this practical we were also tasked with rewriting a MATLAB script in to Python. The piece of script we needed to rewrite was the function that tracks the centre of the bead in the image, this function was comprised of interpolation method and a method that found the symmetry centre of the bead using Fourier transforms. The second part was easily implemented in Python as it was mostly just finding the right Python functions that were equivalent to their MATLAB counterparts. The interpolation function was a lot harder to implement in Python since most of the existing Python interpolation functions did not have the same functionality.

As outlined in section 3.1 a new method for deriving the trap constants was proposed. This method was implemented in a second different python script to try and show its usefulness. The result speaks for itself, it shows a clear connection between the laser output power and the length of the semi-major and semi-minor axis of the ellipses that encircle the points describing the symmetry centres of the beads. The points were fitted with a function with shape $L_{axis} = a \cdot e^{-b \cdot P}$, this seems to fit well as can be seen in figure 4 & 5. This also shows that while the trap stiffness seems to increase linearly with the laser power the area of the ellipse in which we expect a bead to be found decreases exponentially with laser power. In the appendix three more plots showing the spread can be seen.

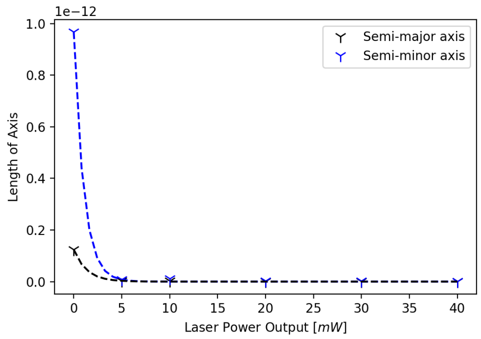

Figure 4: Semi-major and minor axis of the first dataset plotted and fitted.

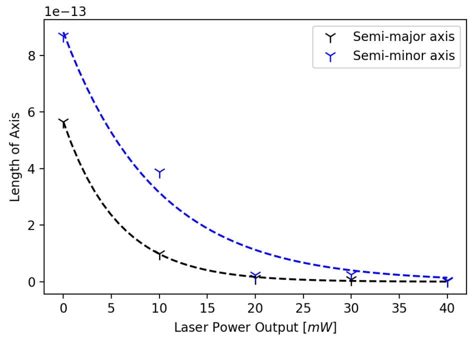

Figure 5: Semi-major and minor axis of the second dataset plotted and fitted.

6 Discussion

In the previous section the results were presented and even though some points of the first dataset were not really usable, a clear relation between the laser output power and the trap stiffness of the optical trap was found. Both of the datasets showed a linear relation between the trap stiffnesses in the x- and y-direction and the laser output power P and between the total trap stiffness k_{tot} and the laser output power.

The task of recreating a MATLAB script in Python was partially successful. The symmetry centre finding function was successfully implemented as well as the subpixel interpolation function. Due to some dissimilarities in the way MATLAB and Python functions interpolate an unstructured set of data we were unable to get the main tracking function to work. The resulting estimates of the symmetry centre location were not far of but were not dead-on either.

7 conclusion

The primary goal of this report was to find a relation between the trap stiffness of an optical trap and the output power of the laser used in that optical trap. The predicted linear relation was measured and found the results are shown in the table below.
Rewriting the MATLAB script using Python was a partial success, the symmetry centre finding function and the subpixel interpolation function were successfully implemented in Python but the tracking function currently uses a slow and not very useful interpolation method which results in an undesirable offset and delay in the predicted symmetry centre. With a properly functioning interpolation function the python script will without a doubt yield the right results.
A second method for deriving the trap stiffness was also tried to be implemented. Using the spread of the symmetry centres of the datasets to try and create an covariance matrix. We were unable to finish this method but were able to show that there is indeed a relation between the ellipse size and the laser power output. This relation was found to be an exponential.

coefficient for	a_x $[pN/(nm \cdot mW)]$	a_y $[pN/(nm \cdot mW)]$	a_{tot} $[pN/(nm \cdot mW)]$
dataset 1	$1.382 \cdot 10^{-6}$	$1.280 \cdot 10^{-6}$	$1.740 \cdot 10^{-4}$
dataset 2	$1.478 \cdot 10^{-5}$	$3.974 \cdot 10^{-6}$	$1.551 \cdot 10^{-5}$

Table 5: The results of the linear fits for the trap stiffness as a function of laser output power. The shown a-factors satisfy the least squares fit of the function: $k_i = a_i \cdot P$

References

[1] M. T. van Loenhout, J. W. Kerssemakers, I. De Vlaminck, and C. Dekker, “Non-bias-limited tracking of spherical particles, enabling nanometer resolution at low magnification,” *Biophysical journal*, vol. 102, no. 10, pp. 2362–2371, 2012.

8 Appendix

8.1 Particle Spread

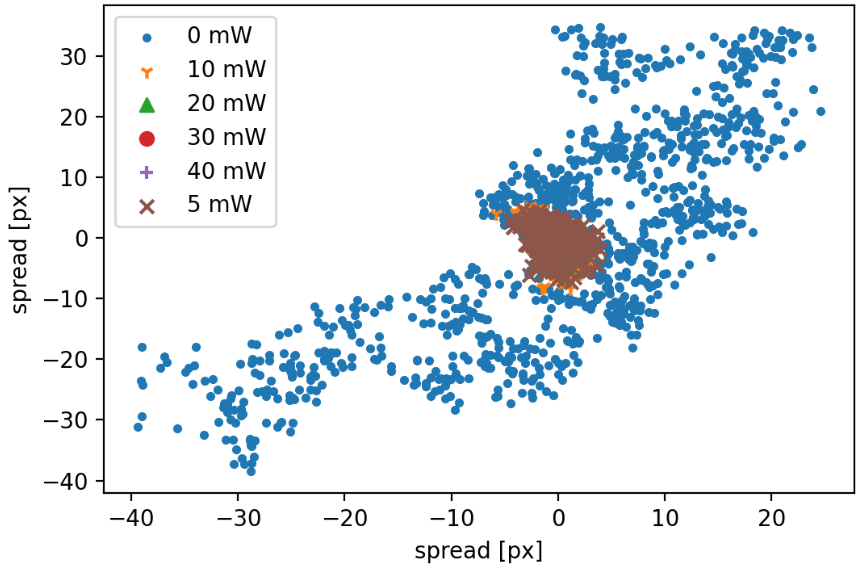

Figure 6: Spread of the first dataset shifted towards origin by average displacement from centre

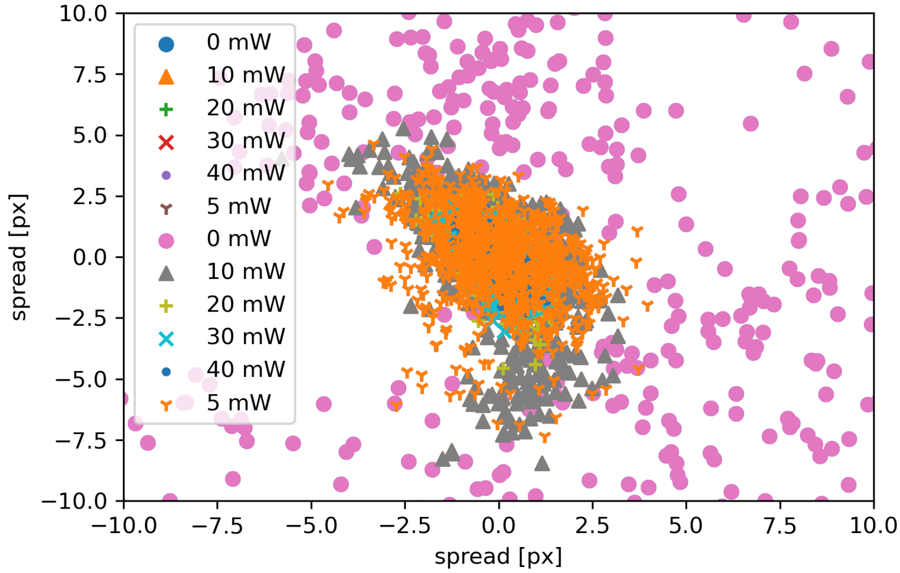

Figure 7: Spread of the first dataset shifted towards origin by average displacement from centre, and zoomed in

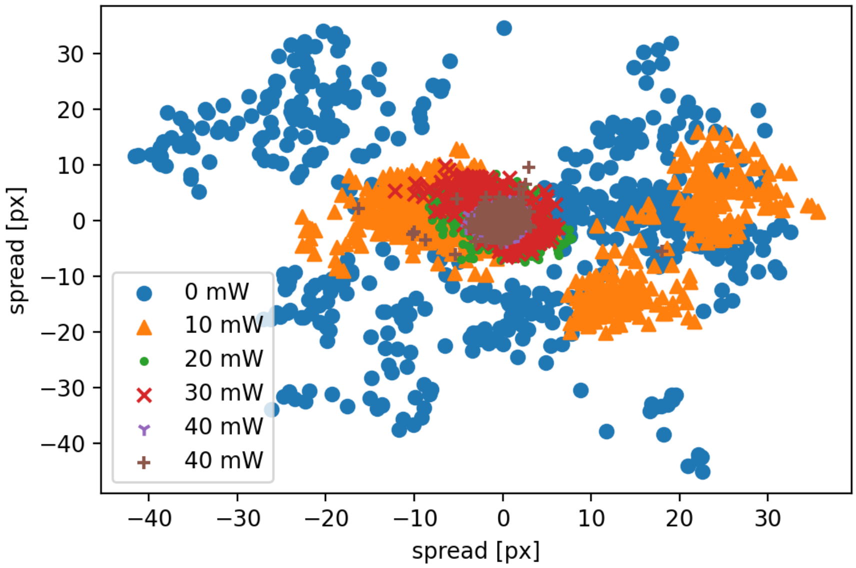

Figure 8: Spread of the first dataset shifted towards origin by average displacement from centre