Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Physics Communications 159 (2004) 225–240

Computer Physics Communications

www.elsevier.com/locate/cpc

MatLab program for precision calibration of optical tweezers ☆

Iva Marija Tolić-Nørrelykke [a,1], Kirstine Berg-Sørensen [a,*], Henrik Flyvbjerg [b]

[a] The Niels Bohr Institute, Blegdamsvej 17, DK-2100 Copenhagen Ø, Denmark
[b] Plant Research Department and Danish Polymer Centre, Risø National Laboratory, DK-4000 Roskilde, Denmark

Received 1 June 2003; accepted 24 February 2004

Abstract

Optical tweezers are used as force transducers in many types of experiments. The force they exert in a given experiment is known only after a calibration. Computer codes that calibrate optical tweezers with high precision and reliability in the (x, y)-plane orthogonal to the laser beam axis were written in MatLab (MathWorks Inc.) and are presented here. The calibration is based on the power spectrum of the Brownian motion of a dielectric bead trapped in the tweezers. Precision is achieved by accounting for a number of factors that affect this power spectrum. First, cross-talk between channels in 2D position measurements is tested for, and eliminated if detected. Then, the Lorentzian power spectrum that results from the Einstein–Ornstein–Uhlenbeck theory, is fitted to the low-frequency part of the experimental spectrum in order to obtain an initial guess for parameters to be fitted. Finally, a more complete theory is fitted, a theory that optionally accounts for the frequency dependence of the hydrodynamic drag force and hydrodynamic interaction with a nearby cover slip, for effects of finite sampling frequency (aliasing), for effects of anti-aliasing filters in the data acquisition electronics, and for unintended "virtual" filtering caused by the position detection system. Each of these effects can be left out or included as the user prefers, with user-defined parameters. Several tests are applied to the experimental data during calibration to ensure that the data comply with the theory used for their interpretation: Independence of x- and y-coordinates, Hooke's law, exponential distribution of power spectral values, uncorrelated Gaussian scatter of residual values. Results are given with statistical errors and covariance matrix.

Program summary

Title of program: tweezercalib
Catalogue identifier: ADTV
Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland.
Program Summary URL: http://cpc.cs.qub.ac.uk/summaries/ADTV
Computer for which the program is designed and others on which it has been tested: General computer running MatLab (MathWorks Inc.).
Programming language used: MatLab (MathWorks Inc.). Uses "Optimization Toolbox" and "Statistics Toolbox".

☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

[*] Corresponding author.
E-mail address: berg@nbi.dk (K. Berg-Sørensen).

[1] Permanent address: Rugjer Bošković Institute, Bijenička 54, HR-10000 Zagreb, Croatia.

0010-4655/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2004.02.012

Memory required to execute with typical data: Of order 4 times the size of the data file.
High speed storage required: None
No. of lines in distributed program, including test data, etc.: 133 183
No. of bytes in distributed program, including test data, etc.: 1 043 674
Distribution format: tar gzip file
Nature of physical problem: Calibrate optical tweezers with precision by fitting theory to experimental power spectrum of position of bead doing Brownian motion in incompressible fluid, possibly near microscope cover slip, while trapped in optical tweezers. Thereby determine spring constant of optical trap and conversion factor for arbitrary-units-to-nanometers for detection system.
Method of solution: Elimination of cross-talk between quadrant photo-diode's output channels for positions (optional). Check that distribution of recorded positions agrees with Boltzmann distribution of bead in harmonic trap. Data compression and noise reduction by blocking method applied to power spectrum. Full accounting for hydrodynamic effects: Frequency-dependent drag force and interaction with nearby cover slip (optional). Full accounting for electronic filters (optional), for "virtual filtering" caused by detection system (optional). Full accounting for aliasing caused by finite sampling rate (optional). Standard non-linear least-squares fitting. Statistical support for fit is given, with several plots suitable for inspection of consistency and quality of data and fit.
Restrictions on the complexity of the problem: Data should be positions of bead doing Brownian motion while held by optical tweezers. For high precision in final results, data should be time series measured over a long time, with sufficiently high experimental sampling rate: The sampling rate should be well above the characteristic frequency of the trap, the so-called *corner frequency*. Thus, the sampling frequency should typically be larger than 10 kHz. The Fast Fourier Transform applied requires the time series to contain 2^n data points, and long measurement time is obtained with $n > 12$–15. Finally, the optics should be set to ensure a harmonic trapping potential in the range of positions visited by the bead. The fitting procedure checks for harmonic potential.
Typical running time: (Tens of) minutes
Unusual features of the program: None
References: The theoretical underpinnings for the procedure are found in [K. Berg-Sørensen, H. Flyvbjerg, Rev. Sci. Instrum. 75 (3) (2004) 594].

PACS: 87.80.-y; 06.20.Dk; 07.60.-j; 05.40.Jc

Keywords: Optical tweezers; Calibration; Power spectrum analysis

1. Introduction

In many applications of optical tweezers in biological physics, the tweezers are used to exert a prescribed force or measure an unknown force. To do this, one must calibrate the tweezers. In some applications it is important to know the force with precision, and in general a good calibration method provides a stringent check that a pair of tweezers work as they are supposed to. A popular calibration method interprets the power spectrum of the Brownian motion of a bead held with the tweezers. The time series of positions of the bead is measured with a photo-diode [2,3], or, in some cases, with an interferometric technique [4]. This power spectrum is a stochastic function, like the bead's position, and is fitted with its theoretical expectation value [1,5]. One parameter thus determined is the *corner frequency* f_c. It describes the ratio between Stokes' friction coefficient γ_0 for rectilinear motion with constant velocity of the bead in the fluid, and the trap's spring constant κ, $f_c = \kappa/(2\pi\gamma_0)$. With f_c and γ_0 known, the latter from Stokes' law (or Faxén's correction to it), so is κ in physical units. Another parameter thus determined is the bead's diffusion coefficient D, which is found in (arb. units)^2/s, where "arb. units" stands for the arbitrary units in which the photo detection system measures position. These units depend on the amplification factor we choose for convenient data acquisition. Since D is already known in physical units through the Einstein relation $D = k_B T/\gamma_0$, the translation between these arbitrary units and the SI unit of length is determined, i.e., one has calibrated the photo-diode. Consequently, the force $-\kappa x$ experienced by the bead as a function of a measured displacement x of the bead in the trap is known.

The theoretical expectation value fitted to the experimental power spectral values can be either the Lorentzian resulting from the Einstein–Ornstein–Uhlenbeck theory for Brownian motion [5], or it can be a more correct theory that takes into account the frequency-dependence of the hydrodynamic frictional force, the possible interaction with a nearby cover slip, possible anti-aliasing filters built into the data acquisition electronics, possible "virtual filtering" caused by the detection system, and possible aliasing caused by finite sampling rate [1]. Any or all of these factors affecting the power spectrum can be taken into account with the tweezer calibration program presented here. A number of outputs, numerical and graphical, allow the program's user to check data and fits for consistency and quality.

2. Data processing

The time series of positions are first Fourier transformed. Each series is transformed as a single, long series, and no "windowing function" [6] is applied before transformation. Instead, noise reduction is done by blocking as described and motivated below. For a coordinate x recorded at intervals Δt for a time T_{meas}, the Fourier transform is defined as

$$\hat{x}_k = \Delta t \sum_{j=1}^{N} \mathrm{e}^{\mathrm{i}2\pi f_k t_j} x_j, \quad f_k \equiv k/T_{\mathrm{meas}} = k f_{\mathrm{sample}},\ k = 1, \ldots, N. \tag{1}$$

Here x_j is the value recorded for x at time $t_j = j\Delta t$, and N is the number of values recorded, $N\Delta t = T_{\mathrm{meas}}$. The experimental power spectrum for x is then $P_x^{(\mathrm{ex})}(f_k) \equiv |\hat{x}_k|^2/T_{\mathrm{meas}}$.

If an interferometric technique is used for position determination, typically a time series for only one coordinate is determined. If a quadrant photo-diode is used as position detector, it measures the position (x, y) orthogonal to the beam axis as combinations of voltages from the four quadrants. If these are numbered I, II, III, and IV like the quadrants of a 2D coordinate system, and their output voltages are denoted V_{I}, V_{II}, V_{III}, and V_{IV}, then changes in the voltage and ratios

$$R_x \equiv (V_{\mathrm{I}} - V_{\mathrm{II}} - V_{\mathrm{III}} + V_{\mathrm{IV}})/V_z \equiv V_x/V_z, \tag{2}$$

$$R_y \equiv (V_{\mathrm{I}} + V_{\mathrm{II}} - V_{\mathrm{III}} - V_{\mathrm{IV}})/V_z \equiv V_y/V_z, \tag{3}$$

$$V_z \equiv V_{\mathrm{I}} + V_{\mathrm{II}} + V_{\mathrm{III}} + V_{\mathrm{IV}} \tag{4}$$

are, to a good first approximation [2,7–9], proportional to changes in the bead's position (x, y, z), z being the coordinate along the laser beam's axis. To a slightly worse approximation, x and y are proportional to V_x and V_y. In the text below, R_x and R_y could be replaced by V_x and V_y. Here, we calibrate the trap in the (x, y)-plane and consider those coordinates only. The general discussion about power spectral analysis applies to the z-coordinate as well, though.

The recorded coordinates may not be entirely independent, while true Cartesian coordinates x and y are. To test for this, i.e., for cross-talk between the recorded channels, the quantity $P_{xy}^{(\mathrm{ex})}(f_k) \equiv \mathrm{Re}(\hat{R}_x(f_k)\hat{R}_y^*(f_k))$ is calculated. If it vanishes over the entire frequency range, R_x and R_y represent independent coordinates. If not, cross-talk is eliminated by a frequency-independent linear transformation to independent coordinates, if such a transformation can be found [1]. This transformation is highly over-determined, but can always be found, is our experience. To judge whether $P_{xy}^{(\mathrm{ex})}$ vanishes or not, $P_{xy}^{(\mathrm{ex})}(f_k)/\sqrt{P_x^{(\mathrm{ex})}(f_k)P_y^{(\mathrm{ex})}(f_k)}$ is blocked as described in the following subsection. It is then plotted before and after the transformation. The remainder of the analysis presented here is done with independent coordinates, i.e., (x, y) refers to the transformed variables in cases where there was cross-talk to eliminate.

2.1. Blocking

Noise reduction is done by “blocking” [1,6]: A “block” of n_b consecutive data points $(f, P^{(\mathrm{ex})}(f))$ of the experimental power spectrum $P^{(\mathrm{ex})}(f)$ is replaced with one point $(\bar{f}, \overline{P}^{(\mathrm{ex})}(\bar{f}))$, where

$$\bar{f} = \frac{1}{n_\mathrm{b}} \sum_{f \in \mathrm{block}} f; \qquad \overline{P}^{(\mathrm{ex})} = \frac{1}{n_\mathrm{b}} \sum_{f \in \mathrm{block}} P^{(\mathrm{ex})}(f). \tag{5}$$

As shown in [1], with n_b sufficiently large, these blocked data points are to a good approximation Gaussian distributed, so standard least-squares fitting applies to the blocked data. Their expectation value is the theoretical power spectrum, $P(f)$, and their standard deviations are easily found:

$$\langle \overline{P}^{(\mathrm{ex})}(\bar{f}) \rangle = P(\bar{f}), \tag{6}$$

$$\sigma\bigl(\overline{P}^{(\mathrm{ex})}(\bar{f})\bigr) = \sigma\bigl(P^{(\mathrm{ex})}(\bar{f})\bigr)/\sqrt{n_\mathrm{b}} = P(\bar{f})/\sqrt{n_\mathrm{b}}. \tag{7}$$

Blocking, as opposed to windowing, can reduce noise by data compression to points with any desired distribution on the first axis. This is the reason we prefer blocking. Equidistant points on a linear and on a logarithmic axis are two particularly useful choices, the former for fitting to data, the latter for display of data in a log–log plot. While windowing will produce the former, it cannot produce the latter. Also, windowing with semi-overlapping windowing functions induces correlations between neighbouring data points.

For ease of notation, we omit the bar in what follows, and let $P^{(\mathrm{ex})}(f)$ denote blocked experimental data points.

2.2. Model-independent experimental test of harmonic trapping potential

The theory for the trapped bead assumes a harmonic trapping potential, hence a Gaussian Boltzmann distribution of positions visited by the bead. To test that one actually has a harmonic potential in the experiment, the program plots a histogram of positions visited by the trapped bead, and a fit of a Gaussian distribution to this histogram.

2.3. Hydrodynamics

Stokes' law, $\gamma_0 = 6\pi\rho_\mathrm{fluid}\mu R$, for the friction coefficient γ_0 of a sphere of radius R moving with constant velocity and vanishing Reynolds number in a fluid with viscosity μ and mass density ρ_fluid, is a low-frequency approximation when used to describe Brownian motion, as it is in the Einstein–Ornstein–Uhlenbeck theory. The correct frequency-dependent hydrodynamical drag force that replaces Stokes Law was also derived by Stokes at vanishing Reynolds number [1,10]. In the computer code, this correct frequency-dependent description is an option. Thus the user can find its effect by comparing results of fits of it with results of fits of the Einstein–Ornstein–Uhlenbeck theory to the same data.

Frequency-dependent or not, the hydrodynamical drag force is increased by nearby surfaces, such as the microscope cover slip, the only surface of relevance here. In the case of Stokes Law for motion with constant velocity parallel to a planar surface, Faxén's formula describes this effect with a truncated power series in R/ℓ, with ℓ the distance from the center of the bead to the cover slip, and R the bead's radius. In the case where the correct frequency-dependent friction is used, harmonic motion parallel to a planar surface at distance ℓ also has a larger drag force than in bulk, though the effect decreases with increasing frequency of the motion. It is described by a formula derived in [11] and given also in [1]. At vanishing frequency, this frequency-dependent formula agrees with Faxén's formula in H.A. Lorentz's original approximation [1,11].

Note that both these formulas for the friction from a nearby planar surface are approximations based on expansion schemes: They are more precise the further away the surface is, and not reliable when R/ℓ approaches 1.

Thus, when frequency-dependent hydrodynamic effects are accounted for, the experimental data are fitted with the theoretical spectrum

$$P^{(\mathrm{hydro})}(f;\,R/\ell)=\frac{D/(2\pi^2)\frac{\mathrm{Re}\,\gamma}{\gamma_0}}{(f_c+f\frac{\mathrm{Im}\,\gamma}{\gamma_0}-f^2/f_m)^2+(f\frac{\mathrm{Re}\,\gamma}{\gamma_0})^2}\tag{8}$$

where

$$\mathrm{Re}\,\frac{\gamma}{\gamma_0}=1+\sqrt{f/f_\nu}-\frac{3R}{16\ell}+\frac{3R}{4\ell}\exp\left(-\frac{2\ell}{R}\sqrt{f/f_\nu}\right)\cos\left(\frac{2\ell}{R}\sqrt{f/f_\nu}\right)$$

and

$$\mathrm{Im}\,\frac{\gamma}{\gamma_0}=-\sqrt{f/f_\nu}+\frac{3}{4}\frac{R}{\ell}\exp\left(-\frac{2\ell}{R}\sqrt{f/f_\nu}\right)\sin\left(\frac{2\ell}{R}\sqrt{f/f_\nu}\right).$$

In these expressions, two new characteristic frequencies appear: $f_\nu\equiv\nu/(\pi R^2)$ and $f_m\equiv\gamma_0/(2\pi m)$, where ν is the kinematic viscosity of the fluid, and m is the mass of the bead. The program asks the user for bead's mass density, and calculates its mass as $m\equiv 4\pi\rho_{\mathrm{bead}}R^3/3$.

When frequency-dependent hydrodynamic corrections are *not* accounted for, the theoretical spectrum fitted to the data is a Lorentzian,

$$P^{(\mathrm{Lorentz})}(f)=\frac{D/(2\pi^2)}{f_c^2+f^2}.\tag{9}$$

If this expression is used, and the distance to the cover slip is small, Stokes' friction coefficient γ_0 should be replaced by Faxén's correction to it.

2.4. Filters

Data-acquisition systems may have built-in filters that affect the recorded power spectrum. The effect of such filters is included in our theory for the experimental power spectrum. The relevant filters are typically anti-aliasing filters with known characteristics. For example, a first-order filter with roll-off frequency $f_{3\mathrm{dB}}$ reduces the power of its input $P_0(f)$ by a factor

$$\frac{P(f)}{P_0(f)}=\frac{1}{1+(f/f_{3\mathrm{dB}})^2}.\tag{10}$$

A typical optical tweezers setup may, in addition to such known electronic filters, contain an unintended virtual filter in the form of a delayed response from the photo-diode position detection system [12]. The form of this virtual filter's characteristic is known [12,13],

$$\frac{P(f)}{P_0(f)}=\alpha^{(\mathrm{diode})2}+\frac{1-\alpha^{(\mathrm{diode})2}}{1+(f/f_{3\mathrm{dB}}^{(\mathrm{diode})})^2},\tag{11}$$

but its parameters, $f_{3\mathrm{dB}}^{(\mathrm{diode})}$ and $\alpha^{(\mathrm{diode})}$, are not, so they are fitted by the program, if this optional virtual filter is included in the theory. In [1] arguments are given why fitting is the optimal way to determine the relevant values of these parameters.

If the Nyquist frequency, $f_{\mathrm{Nyq}}\equiv\frac{1}{2}f_{\mathrm{sample}}$, is not a good deal larger than $f_{3\mathrm{dB}}^{(\mathrm{diode})}$ and $\alpha^{(\mathrm{diode})2}$ is small, as is typically the case, it is difficult to separate $f_{3\mathrm{dB}}^{(\mathrm{diode})}$ and $\alpha^{(\mathrm{diode})}$. They are highly covariant in this case. Fortunately, they can be combined into a single parameter, $f_{3\mathrm{dB}}^{(\mathrm{diode,eff})}$,

$$\frac{P(f)}{P_0(f)}=\frac{1+\alpha^{(\mathrm{diode})2}(f/f_{3\mathrm{dB}}^{(\mathrm{diode})})^2}{1+(f/f_{3\mathrm{dB}}^{(\mathrm{diode})})^2}\approx\frac{1}{1+(f/f_{3\mathrm{dB}}^{(\mathrm{diode,eff})})^2}\tag{12}$$

where we have introduced

$$f_{\mathrm{3dB}}^{(\mathrm{diode,eff})} \equiv \left(1-\alpha^{(\mathrm{diode})2}\right)^{-1/2} f_{\mathrm{3dB}}^{(\mathrm{diode})}. \tag{13}$$

We note that the last expression in Eq. (12) is a simple Lorentzian, as for a first-order filter with $f_{\mathrm{3dB}}^{(\mathrm{diode,eff})}$ as 3dB-frequency. In the program, the user can choose whether $f_{\mathrm{3dB}}^{(\mathrm{diode})}$ and $\alpha^{(\mathrm{diode})}$ are to be fitted separately, or combined into the single parameter $f_{\mathrm{3dB}}^{(\mathrm{diode,eff})}$ which is then fitted. If f_{Nyq} is very much larger than $f_{\mathrm{3dB}}^{(\mathrm{diode})}$, the low-frequency approximation that Eq. (11) is, is insufficient to describe the filtering effect of the diode up to f_{Nyq}. A more elaborate expression is needed [13]. The user can type in and try out expression of increasing complexity, e.g., as described in [13].

2.5. *Aliasing*

The effect of a finite sampling frequency is called aliasing. Even with the use of anti-aliasing filters, it should be accounted for. This is done by summing the theoretical spectrum,

$$P^{(\mathrm{aliased})}(f) = \sum_{n=-\infty}^{\infty} P^{(\mathrm{theory})}(f + n f_{\mathrm{sample}}), \tag{14}$$

where in practice a finite number of terms exhaust the sum. The result, $P^{(\mathrm{aliased})}(f)$, is the theory for the expectation value of the recorded experimental power spectrum. $P^{(\mathrm{theory})}$ is either $P^{(\mathrm{Lorentz})}$ or $P^{(\mathrm{hydro})}$, possibly multiplied by filtering functions as described above.

2.6. *Fits and their support*

The program fits the theoretical power spectrum to the experimental data using general non-linear least-squares fitting routines available with the *Optimization Toolbox* of MatLab (MathWorks Inc.), except for fits of Lorentzians, for which analytical results from [1] are used. In order to judge the quality of a given fit, the resulting value of

$$\frac{\chi^2}{n_{\mathrm{free}}} = \frac{1}{n_{\mathrm{free}}} \sum_{k=1}^{n} \left(\frac{y_k - y_{\mathrm{theory}}(x_k; \text{parameters})}{\sigma_{\mathrm{theory}}(x_k; \text{parameters})} \right)^2 \tag{15}$$

is quoted along with the fitted parameter values. In Eq. (15), n is the number of data points (x_k, y_k) to which we fit, n_{free} is the number of degrees of freedom, i.e., $n_{\mathrm{free}} = n - n_{\mathrm{par}}$ with n_{par} the number of parameters fitted. Also in Eq. (15), y_{theory} is the function fitted, it is the expectation value of the distribution according to which the data scatter according to our theory. The quantity σ_{theory} is the standard deviation of this distribution. For the power spectra, $y_k = P_k^{(\mathrm{ex})}$, $y_{\mathrm{theory}}(x_k; \text{parameters}) = P^{(\ldots)}(f_k)$ where $(\ldots)$ stands for (Lorentz) or (hydro), filtered and/or aliased, while $\sigma_{\mathrm{theory}}(x_k; \text{parameters}) = (n_{\mathrm{b}})^{-1/2} P^{(\ldots)}(f_k)$. In standard least-squares fitting, χ^2 is minimized. As shown in [1, Appendix E], when the parameters fitted appear in σ_{theory} as here, maximum likelihood estimation does not reduce to least-squares fitting, that is, to minimization of χ^2 in Eq. (15). When n_{b} is large, it approximately does so. But in general, one must minimize

$$\chi^2 + 2\sum_{k=1}^{n} \log \sigma_{\mathrm{theory}}(x_k; \text{parameters})$$

where χ^2 is given in Eq. (15). This is done by the program.

Fitting is done after an exact rewriting of the expression for χ^2 [1] to:

$$\chi^2 = \sum_{k=1}^{n} \left(\frac{P(f_k)^{(\dots)-1} - P_k^{(\mathrm{ex})-1}}{\sigma_k} \right)^2 \tag{16}$$

where $\sigma_k = P_k^{(\mathrm{ex})-1}/(n_\mathrm{b})^{1/2}$. When a fit has been obtained, the program computes the *support* for the fit, also known as the goodness-of-fit, or as the *backing* [6,14],

$$P_{n_\mathrm{free}}(> \chi^2) \equiv \frac{1}{2^{(n_\mathrm{free}-2)/2}[(n_\mathrm{free}-2)/2]!} \int_{\chi}^{\infty} x^{n_\mathrm{free}-1} \exp\left(-x^2/2\right) \mathrm{d}x = Q\left(\frac{n_\mathrm{free}-2}{2}, \frac{\chi^2}{2}\right). \tag{17}$$

Here, Q is the incomplete Γ-function and Eq. (17) gives the support irrespective of whether maximum likelihood estimation is nearly the same as least-squares minimization or not, as long as n_b is large enough for the blocked data to be normally distributed. The support is the probability that a repetition of the measurement that produced the data we fitted to, will produce data with a larger value for χ^2 when χ^2 is computed using the theory given by the values already found for the fitted parameters. This interpretation of the support presupposes that data are Gaussian distributed about the theoretical expectation value. As blocking by a large factor guarantees that data are Gaussian distributed (by virtue of the Central Limit Theorem), this condition is satisfied if the fitted theory describes the data's theoretical expectation value. If the support for the fit is good, this is typically the case, but further evidence is provided by direct visual inspection made possible by residual plots produced by the program and described below.

3. Usage

The program is called from the command window in MatLab (MathWorks Inc.). Once the main program is initiated, the user is presented with a separate input window in which all subsequent information is to be entered. Fig. 1 shows the layout of the input window in the test case.

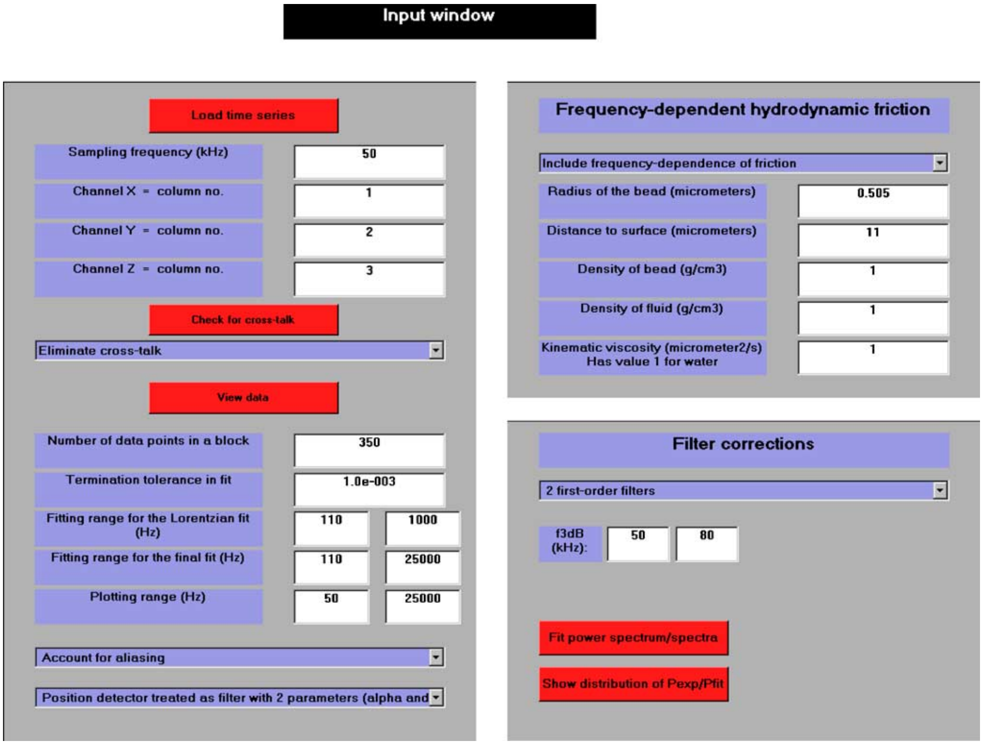
3.1. File specifications

First, the time series of data are read from the file specified by the user. The default format of the data file is

$$\begin{array}{lll} V_x(t_1) & V_y(t_1) & V_z(t_1) \\ V_x(t_2) & V_y(t_2) & V_z(t_2) \\ \vdots & \vdots & \vdots \\ V_x(t_N) & V_y(t_N) & V_z(t_N) \end{array} \tag{18}$$

where the time t_i is $t_i = i\Delta t$, $\Delta t = 1/f_\mathrm{sample}$, and N is the number of data points in the time series. The 2nd and 3rd columns are optional. The user may change the column numbers n_x, n_y and n_z describing which columns contain the data for the x-, y-, and z-channels, respectively. The program reads the number of columns in the data file, and sets as default values $n_x = 1$, $n_y = 2$ (for two or more columns), and $n_z = 3$ (for three or more columns). Setting either n_y or n_z equal to zero indicates, respectively, that no y-signal should be processed or no division by V_z should be done. If a data file has three or more columns, the program's default action is to use $R_x = V_x/V_z$ and $R_y = V_y/V_z$ as the time series for the positions of the bead. These ratios are the signal processed by the program. With $n_y = 0$, the layout of the input window is different from that of Fig. 1: The push-button $\boxed{\text{Check for cross-talk}}$ and the popup-button next to the text *Eliminate cross-talk* do not appear.

The user should change the sampling frequency to the one with which his data were recorded; see Fig. 1. The default value $f_\mathrm{sample} = 50$ kHz is merely the one of our test case.



Sampling frequency (kHz)	50
Channel X = column no.	1
Channel Y = column no.	2
Channel Z = column no.	3

Eliminate cross-talk

Number of data points in a block	350	
Termination tolerance in fit	1.0e-003	
Fitting range for the Lorentzian fit (Hz)	110	1000
Fitting range for the final fit (Hz)	110	25000
Plotting range (Hz)	50	25000

Account for aliasing

Position detector treated as filter with 2 parameters (alpha and

Frequency-dependent hydrodynamic friction

Include frequency-dependence of friction

Radius of the bead (micrometers)	0.505
Distance to surface (micrometers)	11
Density of bead (g/cm3)	1
Density of fluid (g/cm3)	1
Kinematic viscosity (micrometer2/s) Has value 1 for water	1

Filter corrections

2 first-order filters

f3dB (kHz):	50	80

Fig. 1. *The input window* which opens when the main program is initiated from the command window of MatLab (MathWorks Inc.). The initial layout only displays the push-button Load time series. Once the user has activated this button and data have been read, the rest of the window appears. Default values of all descriptive parameters are shown in the fields to be filled by the user or accepted as is. The layout shown here is the one that applies to the test case. Small differences occur in other cases, as described in the text.

3.2. Initial investigation of data

Once the data has been read, the user must decide if cross-talk between channels, i.e., between R_x and R_y, should be eliminated. For this purpose, the user may push the Check for cross-talk button. This displays a figure, **Figure No. 2**, showing blocked data points of $P_{xy}^{(\mathrm{ex})}/\sqrt{P_x^{(\mathrm{ex})}P_y^{(\mathrm{ex})}}$ versus frequency. If this quantity does not vanish over the entire frequency range, elimination of cross-talk is advised.

If the user decides to eliminate cross-talk, the two parameters of the linear transformation that should do this, are determined in a least-squares fit.

Subsequently, the user should inspect the data. When the View data button is activated, three to seven figures appear:

Figure No. 3: Appears if elimination of cross-talk was chosen. The figure displays the quantity $P_{xy}^{(\mathrm{ex})}/\sqrt{P_x^{(\mathrm{ex})}P_y^{(\mathrm{ex})}}$ *before* and *after* the transformation done to eliminate cross-talk. For clarity, data points after transformation are translated slightly towards higher frequencies. If elimination was successful, *after* the transformation $P_{xy}^{(\mathrm{ex})}/\sqrt{P_x^{(\mathrm{ex})}P_y^{(\mathrm{ex})}}$ is identically zero at all frequencies, apart from noise.

Figure No. 4: Histogram of the measured x-coordinates. If cross-talk was eliminated, this histogram shows values *after* elimination. Also shown, as a solid line, a Gaussian fitted to the histogram data, and, as two dotted lines, ±1 standard deviation, assuming binomially distributed counts in individual bins with expectation values equal to the Gaussian fit's integral across each bin. The mean and standard deviation of the Gaussian fit is displayed below the figure with χ^2 and the fit's backing.

The default number of bins in the histograms is $n_{\mathrm{bin}} = 50$, but n_{bin} can be changed for the individual histogram in the box in the upper corner of the figure. If this number is changed and the return key pressed, a new histogram is displayed with a new Gaussian fit.

Figure No. 5: Histogram of the signal for the y-coordinate, if the data file contains data for two coordinates. Number of bins can be changed as in Figure No. 4.

Figure No. 6: Power spectrum/spectra for the x-coordinate (and y-coordinate), log–log plot of data, blocked on the logarithmic frequency axis.

Figure No. 7: Appears if elimination of cross-talk was chosen. It displays the transformed power spectra, after elimination of cross-talk.

Figure No. 8: Power spectrum for the x-coordinate, log–log plot of data, using blocks of equal size on the linear frequency axis. This allows the user to identify possible noise peaks in data.

Figure No. 9: As Figure No. 8, but for the y-coordinate.

Some of the figures corresponding to the test case are shown in Fig. 2.

3.3. Specifications of fit

Before fitting the theory to the data, the user must decide what the theory should include, i.e., which of the several phenomena mentioned above it should take into account. On the left-hand side of the input window (see Fig. 1), the frequency range of data fitted to should be chosen, both for the Lorentzian fit, which is done analytically, and for the more complete theory. The user also decides on the number of data in a block, n_{b}, and on the stopping criterion for the fitting procedure, the *terminal tolerance* [15]. Default values are shown in Fig. 1. Where appropriate, they correspond to the test case discussed in Section 4. The Lorentzian fit is used to determine first estimates for the parameters f_{c} and D fitted in the more complete theory, and the range used for the Lorentzian fit should not extend up to frequencies where filters, frequency-dependence of friction, and aliasing matter. Neither should it extend down to frequencies where beam-pointing instability and other low-frequency effects external to the experiment, contribute to the power spectrum; see [1].

The number of data in a block, n_{b}, should be sufficiently large that the resulting data point is Gaussian distributed to a good approximation. As the original data points are exponentially distributed [1], n_{b} should preferably not be below 100. As a guideline, the number of blocked data points, $n = N/(2n_{\mathrm{b}})$, which are the number of data points fitted to, should lie in the range 50–200.

Aliasing, due to finite sampling frequency, can be accounted for. It *should* be accounted for unless data acquisition with Δ–Σ conversion, or similar, is used. If in doubt, one can simply inspect the power spectrum: If its slope seems to vanish at f_{Nyq}, as in the lower left and right part of Fig. 2, it is aliased, and aliasing must be accounted for in order to fit it.

Virtual filtering by the photo-diode position detection system can also be accounted for. The user can choose to do so by fitting one parameter, $f_{\mathrm{3dB}}^{(\mathrm{diode,eff})}$, or two parameters, $\alpha^{(\mathrm{diode})}$ and $f_{\mathrm{3dB}}^{(\mathrm{diode})}$. The default action is to *not* treat the position detection system as a virtual filter. If accounting for filtering is chosen, the program calculates an initial value for $f_{\mathrm{3dB}}^{(\mathrm{diode})}$ (or $f_{\mathrm{3dB}}^{(\mathrm{diode,eff})}$) by comparing the Lorentzian fit, with or without aliasing as chosen, with the experimental value of the power spectrum at the Nyquist frequency. The initial value for $\alpha^{(\mathrm{diode})}$ is set to $\alpha^{(\mathrm{diode})} = 0.3$. The user may change this initial value in the code.

On the right-hand side of the input window, the user decides whether the frequency-dependent hydrodynamic friction should be used, or only Stokes' law. If frequency-dependent friction is chosen, the user must supply information about the bead, its distance to the nearest surface, and the density and kinematic viscosity of the surrounding fluid. If only Stokes' law is used, the boxes to be filled with the above-mentioned information disappear from the input window.

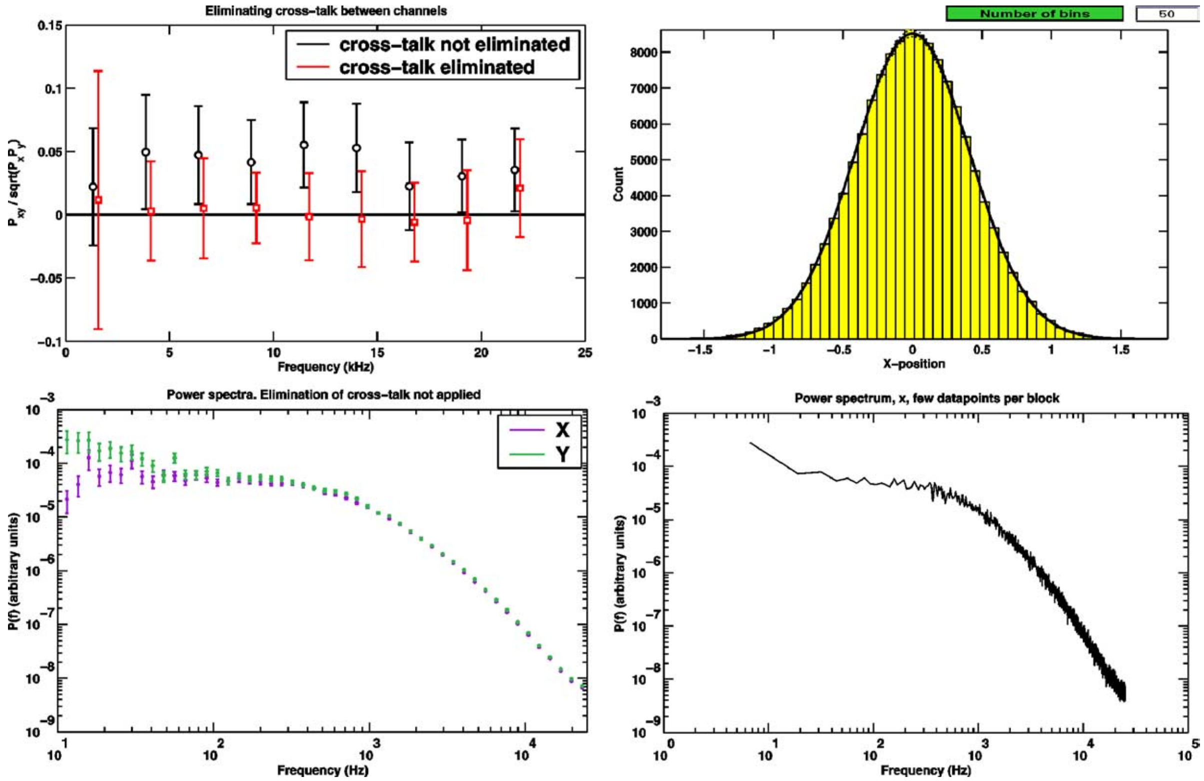


Fig. 2. Four of the seven figures created when activating the $\boxed{\text{View data}}$ button. Example is for the test case discussed later. Upper left: Figure No. 3; $P_{xy}^{(\mathrm{ex})}/\sqrt{P_x P_y}$ before and after elimination of cross-talk. Data points after elimination of cross-talk are slightly translated towards higher frequency for clarity. Upper right: Figure No. 4; Histogram of values for position coordinate x. Lower left: Figure No. 6; Power spectra before elimination of cross-talk. Lower right: Figure No. 8; Power spectrum for x with few data points per block.

Finally, the user defines (a) how many electronic filters should be accounted for in the fitting, and (b) their functional form. The default is none.

3.4. Fitting

To carry out the fitting, the button $\boxed{\text{Fit power spectrum/spectra}}$ should be pushed; see Fig. 1. The program does a non-linear least-squares fit to the data for each coordinate, with data processed as specified by the user, and accounting for the various effects that were chosen by the user.

When the fit has been done, the user is presented with three (six) figures (examples of these figures for the test case discussed later are shown in Fig. 3):

Figure No. 10: $P(f)$ versus f in a log–log plot. Blocked data points and the fitted function are shown. The function represents the theoretical expectation value for the data points, and is shown with a full line. Also shown, as two dotted lines, are ±1 standard deviation of the Gaussian distribution according to which the data points scatter vertically about their expectation value.

Figure No. 11: $P^{(\mathrm{ex})}(f)/P^{(\mathrm{fit})}(f)$ versus f in a lin–lin plot (residual plot). For a perfect fit, this ratio scatters about the value 1 according to a Gaussian distribution. Two horizontal dotted lines denote ±1 standard

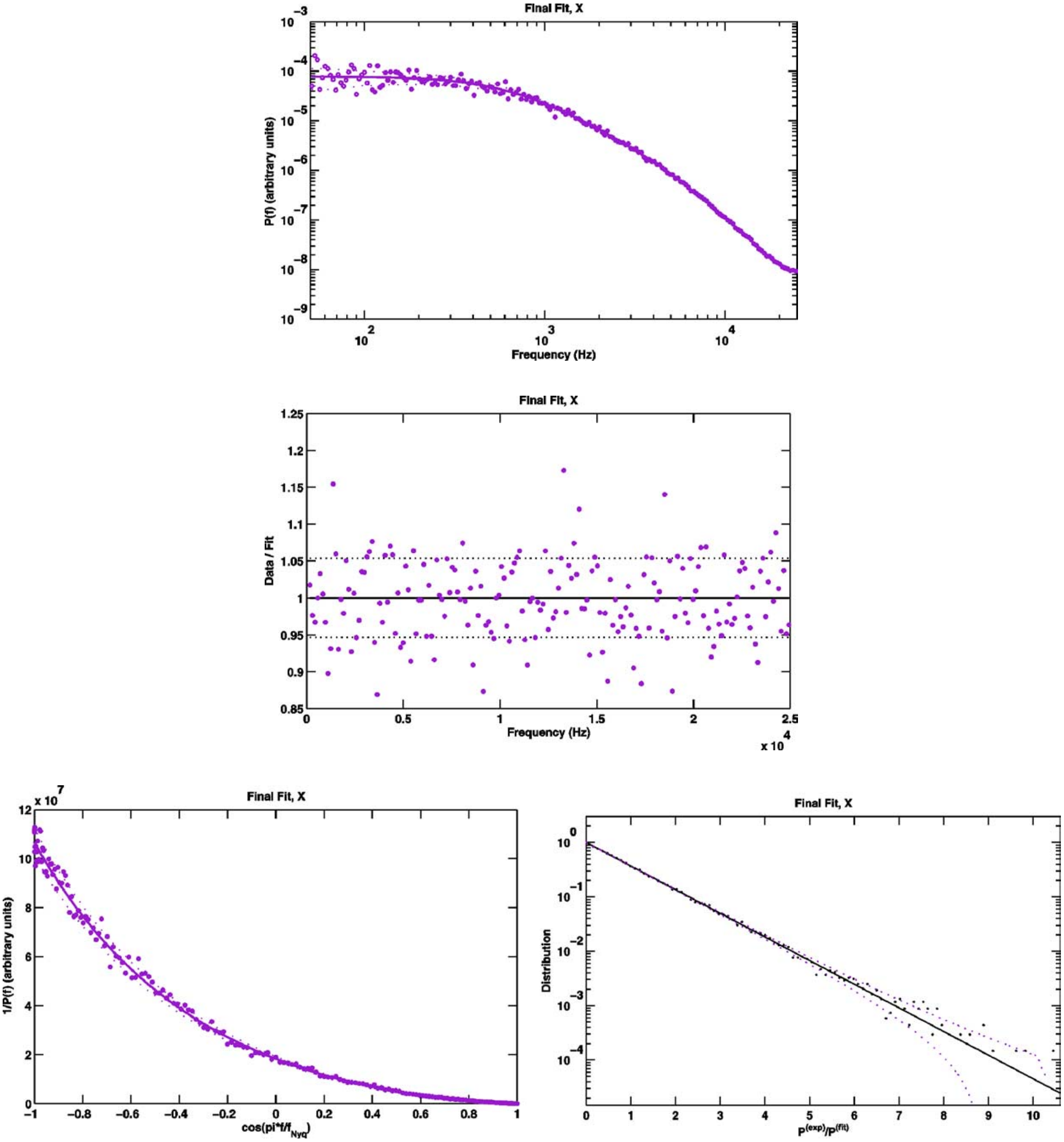


Fig. 3. The figures created when activating the Fit power spectrum/spectra button and the Show distribution of Pexp/Pfit button. Example is for the test run discussed later. Top: Figure No. 10; $P(f)$ versus f in a log–log plot. Solid line is the fit, dotted lines indicate ± one standard deviation (known theoretically). Filled plotting symbols indicate data points fitted to, open plotting symbols are those not included in the fit. Middle: Figure No. 11; $P^{(\mathrm{ex})}(f)/P^{(\mathrm{fit})}(f)$ versus f in a lin–lin plot. Horizontal dotted lines indicate ± 1 standard deviation, according to theory. Lower left: Figure No. 12; $1/P(f)$ versus $\cos(\pi f/f_{\mathrm{Nyq}})$. For a perfect aliased Lorentzian, this would be a straight line. Solid line: Graph of fitted theoretical spectrum, i.e., the theoretical expectation value for the experimental data. Dotted lines: ± 1 standard deviation of data's vertical scatter about their expectation value, according to theory. Lower right: Figure No. 16; Distribution of $P^{(\mathrm{ex})}/P^{(\mathrm{fit})}$ (data points) superposed with the expected exponential distribution (solid line) ± one standard deviation (dashed lines).

deviation of this distribution. In this plot, the user may inspect the quality of the fit: For a perfect fit, 68% $\simeq 2/3$ of the data points fall within the two dotted lines and $1/3$ outside.

Figure No. 12: $1/P(f)$ versus $\cos(\pi f/f_{\mathrm{Nyq}})$. Blocked data points, fitted function, and ±1 standard deviation are shown. For a power spectrum described by an aliased Lorentzian, the data fall on a straight line when plotted as here. As any deviance from a straight line is easily spotted by eye, so is the nature of the data in this plot: It immediately reveals whether a Lorentzian fit is possible or not, and to which part of the data it might be, if not in its entire frequency range.

Figure No. 13: Same as Figure No. 10, for y-coordinate.

Figure No. 14: Same as Figure No. 11, for y-coordinate.

Figure No. 15: Same as Figure No. 12, for y-coordinate.

Then, the user can choose to inspect the supposedly exponential distribution of the unblocked power spectral values about their expectation value. This serves as yet another check of the data. When pushing Show distribution of Pexp/Pfit, one (two) extra figures are produced:

Figure No. 16: Distribution of $P^{(\mathrm{ex})}/P^{(\mathrm{fit})}$, x-coordinate, using the unblocked experimental values. Superposed, the expected exponential distribution (solid line) ±1 standard deviation (dashed lines).

Figure No. 17: Same as Figure No. 16, for y-coordinate.

The resulting parameter values, χ^2, and the support of the fit are given below the figure as well as in the command window of MatLab (MathWorks Inc.). Furthermore, the covariances between the fitted parameters are displayed both places.

In general, the command window displays the progress of the various fits while they are carried out. The displayed parameter values describe the progress of the fitting algorithm, the Levenberg–Marquardt algorithm. The corresponding MatLab help menu (MathWorks Inc.) gives further information about definitions of step size, parameter "Lambda," etc. Also, any error messages or warnings issued by the algorithm during the fitting procedure will appear in the command window. Error messages due to input of improper parameter values will appear in the command window in the format "Error message from program tweezercalib: ..." together with error messages from MatLab caused by the error. The latter have the format "??? Error ...". All final results of fits appear in the relevant plot as well as in the command window.

4. Application

In the MatLab (MathWorks Inc.) command window, enter the command `fitsettings`, then `start_fit`. Then proceed as follows:

(1) Click on the red button Load time series. This opens a file-organizer that shows files in the current working directory of MatLab (MathWorks Inc.). Open the data file to be analyzed, change directory if necessary. The input window now displays more buttons and boxes to be filled in, as shown in Fig. 1. Default numerical values are those applying to the test case.

(2) Choose value for f_{sample}. If needed, redefine column numbers n_x, n_y, and n_z. If $n_z = 0$ the program does not divide V_x and V_y by V_z, but uses them undivided as R_x and R_y, even if a third column with V_z-values is present in the data file. If $n_y = 0$ the program only considers one coordinate (defined as x). In the test case, $f_{\mathrm{sample}} = 50$ kHz, $n_x = 1$, $n_y = 2$, and $n_z = 3$.

(3) If $n_y \neq 0$: Push the red button Check for cross-talk, and inspect the resulting Figure No. 2 in order to decide whether or not to eliminate cross-talk between channels. The default action is to eliminate cross-talk and was applied in the test case.

(4) Click on the red button View data and wait while the data are processed: The program blocks the power spectra, and fits the two parameters, b and c, of the transformation that eliminates cross-talk if this option was chosen. Also, the program calculates histograms of positions, and it fits a Gaussian to each histogram. During this process, a number of windows are opened and the computer seems locked. Then, inspect as follows the four to seven figures that pop up, each in its own window:

Figure No. 3: Check that $P_{xy}^{(\mathrm{ex})}/\sqrt{P_x P_y}$ after elimination of cross-talk is equal to zero modulo error bars.

Figure No. 4 (5): Compare the histogram(s) with the Gaussian fit shown as a solid line with two dashed lines indicating ±1 standard deviation of the scatter of bin-counts about the fitted Gaussian. χ^2, the number of degrees of freedom n_{free}, and the support of the Gaussian fit are all given below each figure. The number of bins in the histogram can be changed by entering a new number in the box in the upper right corner.

Figure No. 6 (7): Blocked power spectra. Inspect for drift/pointing instabilities at low frequencies: Look for excess power at frequencies below $\sim$ 50–100 Hz, depending on the quality of the data, where the power spectrum should be constant if f_{c} is larger than $\sim$ 2–300 Hz. An example of excess power is seen for the y-data in the lower left part of Fig. 2 for frequencies below 40 Hz. This power is due to beam pointing instability. Also, in the same frame, noise at 50 Hz lifts a data-point near that frequency.

Figure No. 8 (9): Inspect the power spectra for noise peaks at 50 Hz, 100 Hz, and at high frequencies, appearing as sharp peaks in power spectral values for distinct frequencies, and excess power at low frequencies. Noisy peaks and excess power from beam pointing instability are demonstrated in Fig. 7 of [11].

(5) If needed, redefine values for number of data points in a block, n_{b}, termination tolerance in fit, fitting range (given by f_{min} and f_{max}) for the Lorentzian fit and for the final fit, and plotting range in final fit. Choose the fitting range of the final fit based on Figures No. 6–9 as described above. Test case:

	$n_{\mathrm{b}} = 350$	
	termination tolerance $= 10^{-3}$	
Lorentzian fit	$f_{\mathrm{min}} = 110$ Hz	$f_{\mathrm{max}} = 1000$ Hz
Final fit	$f_{\mathrm{min}} = 110$ Hz	$f_{\mathrm{max}} = 25\,000$ Hz
Plot	$f_{\mathrm{min}} = 50$ Hz	$f_{\mathrm{max}} = 25\,000$ Hz

(6) Choose whether the position detection system should be treated as a virtual filter or not. If treated like a filter, choose which parameters should be fitted. Test case: Virtual filtering with two parameters. Default: No filtering by position detection system.

(7) Decide whether the hydrodynamical drag force on the bead should be given by Stokes' Law, or by Stokes' frequency-dependent result. Frequency dependent friction is used in test case. In that case, give parameters describing bead and fluid. In test case, bead diameter: 0.505 µm, height above surface: 11 µm, bead and fluid densities: 1 $\mathrm{g/cm^3}$, and kinematic viscosity: 1 $\mathrm{\mu m^2/s}$.

(8) If needed, redefine the number of electronic filters in the data acquisition pathway, and their functional form, if needed. Test case: Two first-order filters with $f_{3\mathrm{dB}} = 50$ kHz and 80 kHz, respectively. Default: No filters.

(9) Click on Fit power spectrum/spectra and wait. Fitting may take some time (tens of minutes) and the computer appears locked while fitting proceeds. For the test case, on a PC running MS Windows 2000 vs. 5.00, with an Intel Pentium 4-processor, 2.00 GHz, 256 MB RAM, fitting took about fifteen minutes.[2] Results of test case:

[2] For users with long data files, faster F90 routines, based on NAGlib fitting packages, may be obtained from the authors. These routines come without a user-friendly graphics interface, though.

x:	y:
$f_{\mathrm{c}} = 643 \pm 14$ Hz	$f_{\mathrm{c}} = 641 \pm 14$ Hz
$D = (6.24 \pm 0.09) \cdot 10^2$ (arb. units)^2/s	$D = (6.48 \pm 0.09) \cdot 10^2$ (arb. units)^2/s
$f_{3\mathrm{dB}}^{(\mathrm{diode})} = 7.5 \pm 0.1$ kHz	$f_{3\mathrm{dB}}^{(\mathrm{diode})} = 7.1 \pm 0.1$ kHz
$\alpha^{(\mathrm{diode})} = 0.26 \pm 0.0$	$\alpha^{(\mathrm{diode})} = 0.26 \pm 0.0$
$\chi^2/n_{\mathrm{free}} = 1.02$	$\chi^2/n_{\mathrm{free}} = 0.96$
backing is 37%	backing is 61%

(10) If wanted, click on Show distribution of Pexp/Pfit and wait.

Appendix A. List of program's subroutines

fitsettings.m Sets path, sets defaults for a number of plotting variables. Most likely, the user should edit this routine, in particular change the path names to match the directories used by the user.

start_fit.m Main program. Creates input window.

load_time_series.m Subroutine that loads the time series. Then calls:

> **input_para1.m** Reads f_{sample}, n_x, n_y, and creates the View data push-button. If $n_y \neq 0$, it also creates the Check for cross-talk push-button. Calls:
>
> > **input_para1_dec.m** Reads whether elimination of cross-talk is chosen.
> >
> > **check.m** Checks that value entered corresponds to the allowed interval (upper end of interval included).
> >
> > **check1.m** Checks that value entered corresponds to the allowed interval (both ends of interval included).
> >
> > **check2.m** Checks that value entered is larger than zero.
>
> **input_para2.m** Reads n_{b}, reads fitting range for analytical Lorentzian fit, for final fit, for plotting range, and tolerance of fit. Also reads how the position detection system should be treated.
>
> **input_para3.m** Reads whether frequency-dependent hydrodynamic friction is chosen. Creates the push-buttons Fit power spectrum/spectra and Show distribution of Pexp/Pfit. Calls:
>
> > **input_para3_hydro.m** Reads the quantities needed for fit accounting for frequency-dependent hydrodynamic friction, R, ρ_{bead}, ρ_{fluid}, ℓ, and ν.
>
> **check_decorr.m** Calculates and creates plot of cross-talk between channels.
>
> **decorr_decision.m** Sets relevant system variable depending on whether elimination of cross-talk between channels is wanted or not.
>
> **min_corr.m** Function to be minimized in order to eliminate cross-talk.

view_data.m Shows histogram(s) of position(s) and power spectrum (spectra) using:

> **plot_histogram.m** Creates histogram plots.
>
> **position_histogram_e.m** Calculates position histogram and fits it with a Gaussian, integrated over each bin.
>
> **caption.m** Makes figure-captions.
>
> **free_erf.m** Gaussian function, integrated over bins, to fit to the position histogram.

free_gauss.m Gaussian function, used for display.
calc_powersp.m Calculate the (yet not blocked) power spectrum of time series.
plot_powerspectrum.m Creates log–log plots of blocked power spectra.
plot_powerspec_lin.m Creates log–log plots of power spectra blocked with few data points per block, and blocked on the linear axis.
decorr_xy.m Performs the elimination of cross-talk between channels, i.e., calculates $P_{xy}^{(\mathrm{ex})}/\sqrt{P_x^{(\mathrm{ex})}P_y^{(\mathrm{ex})}}$ for transformed variables as function of transformation's parameters, and chooses these by fitting $P_{xy}^{(\mathrm{ex})}/\sqrt{P_x^{(\mathrm{ex})}P_y^{(\mathrm{ex})}}$ to zero. Then plots $P_{xy}^{(\mathrm{ex})}/\sqrt{P_x^{(\mathrm{ex})}P_y^{(\mathrm{ex})}}$ before and after transformation and $P_x^{(\mathrm{ex})}$ and $P_y^{(\mathrm{ex})}$ after transformation.
plot_corrxy.m Creates plot of $P_{xy}^{(\mathrm{ex})}/\sqrt{P_x^{(\mathrm{ex})}P_y^{(\mathrm{ex})}}$.

diode_decision.m Sets a system variable depending on whether the position detection system acts as a virtual filter, and whether one or two diode parameters are to be fitted when it does.
alias_decision.m Sets a system variable depending on whether aliasing should be accounted for or not.
g_diode.m Characteristic function of virtual filter of diode. Depends on whether one or two diode parameters are chosen.
hydro_decision.m Sets a system variable depending on whether frequency-dependent hydrodynamic friction is used, or not.
filter_decision.m Sets a system variable and filter function, depending on number of filters chosen.
read_filter_function.m Reads user-defined filter function. Applies only when filters to be accounted for are more complicated than first-order filters.
fit_powerspectrum.m Performs the fit using:

lorentz_analyt.m Estimates initial values for the parameters f_{c} and D, based on analytic formulas for a Lorentzian function [1].
P_hydro.m The functional form of the power spectrum when frequency-dependent hydrodynamic friction is chosen.
P_theor.m The functional form of the power spectrum. Uses P_hydro.m when relevant.
plot_fit.m Creates log–log plot of power spectrum versus frequency, along with the fit.
plot_data_div_fit.m Creates lin–lin plot of data/fit versus frequency.
plot_P_cos.m Creates lin–lin plot of $1/P(f)$ versus $\cos(\pi f/f_{\mathrm{Nyq}})$.
hist_decision.m Reads relevant system variable to determine if a histogram of $P^{(\mathrm{ex})}/P^{(\mathrm{fit})}$ should be calculated and plotted.
spectrum_histogram.m Calls the routine that plots distribution of $P^{(\mathrm{ex})}/P^{(\mathrm{fit})}$.
plot_Phist.m Creates histogram of unblocked $P^{(\mathrm{ex})}/P^{(\mathrm{fit})}$, and plots the normalized distribution.

References

[1] K. Berg-Sørensen, H. Flyvbjerg, Power spectrum analysis for optical tweezers, Rev. Sci. Instrum. 75 (3) (2004) 594–612.
[2] K. Visscher, S.P. Gross, S.M. Block, Construction of multiple-beam optical traps with nanometer resolution position sensing, IEEE J. Select. Topics Quantum Electron. 2 (1996) 1066–1076.
[3] K. Visscher, S.M. Block, Versatile optical traps with feedback control, Methods in Enzymology 298 (1998) 460–489.
[4] K. Svoboda, C.F. Schmidt, B.J. Schnapp, S.M. Block, Direct observation of kinesin stepping by optical trapping interferometry, Nature 365 (1993) 721–727.
[5] F. Gittes, C.F. Schmidt, Signals and noise in micromechanical measurements, Methods in Cell Biology 55 (1998) 129–156.
[6] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes. The Art of Scientific Computing, Cambridge Univ. Press, Cambridge, 1986.

[7] L.P. Ghislain, N.A. Switz, W.W. Webb, Measurement of small forces using an optical trap, Rev. Sci. Instrum. 65 (1994) 2762–2768.

[8] F. Gittes, C.F. Schmidt, Interference model for back-focal-plane displacement detection in optical tweezers, Opt. Lett. 23 (1998) 7–9.

[9] A. Pralle, M. Prummer, E.-L. Florin, E.H.K. Stelzer, J.K.H. Hörber, Three-dimensional high-resolution particle tracking for optical tweezers by forward scattered light, Microscopy Research and Technique 44 (1999) 378–386.

[10] G.G. Stokes, On the effect of the internal friction of fluids on the motion of pendulums, Transactions of the Cambridge Philosophical Society IX (1851) 8–106.

[11] H. Flyvbjerg, unpublished, 2003.

[12] K. Berg-Sørensen, L. Oddershede, E.-L. Florin, H. Flyvbjerg, Unintended filtering in a typical photodiode detection system for optical tweezers, J. Appl. Phys. 93 (2003) 3167–3176.

[13] K. Berg-Sørensen, E.G.J. Peterman, M. van Dijk, C. Schmidt, H. Flyvbjerg, Power spectrum analysis for optical tweezers, II: Laser wavelength dependence of unintended filtering and how to achieve high band-width with precision, 2004, submitted for publication.

[14] N.C. Barford, Experimental Measurements: Precision, Error and Truth, second ed., Wiley, New York, 1986.

[15] MathWorks Inc., MatLab Help Menu, Optional parameters of lsqnonlin, Optimization toolbox.