Using convolution for event analysis on P pili unfolding data measured by optical tweezers

David Hermann

December 27th, 2006

UmU
Table of contents

1. Introduction .................................................................................................................. 3
2. Theory .......................................................................................................................... 3
   a. Classification of applied evaluations........................................................................ 3
   b. Test function and nomenclature .............................................................................. 4
   c. Evaluations using convolution ............................................................................. 5
      i. General properties and requirements................................................................. 5
      ii. Definition and normalization of the probe function ......................................... 6
      iii. Parameter restrictions of the probe function ............................................... 9
      iv. Capability of analysing noisy data ............................................................... 13
   d. Correlation functions assisting convolution analysis ........................................ 17
3. Analysis of Pili PapA measurement data ................................................................. 20
   a. Preliminary evaluation ......................................................................................... 20
   b. Event analysis using convolution ....................................................................... 22
4. Discussion ..................................................................................................................... 26
5. References .................................................................................................................... 27
6. Appendix ....................................................................................................................... 28
   a. Tables ................................................................................................................. 28
   b. Source code ....................................................................................................... 29
1. Introduction

When interpreting measurement data, one is not only interested in the general behaviour of a physical system, but also in possible (periodic) patterns or special events occurring throughout the experiment, which result in a strong and fast change of amplitude. When auto-correlation changes within a certain interval of the data set, it might be an indicator of an event occurrence. Due to noise it is often not easy to detect this without further investigation. This is why different correlation related functions somehow have to evaluate the measurement data – also denoted as input data. Ideally this evaluation not only reveals periodic patterns, but also occurrence, frequency and strength of events. In this thesis most focus will put on the convolution of the input data with a correlation related function also denoted as evaluation function.

Before this is carried out with real data, evaluation functions are applied on a test function, which has well known and predefined properties. This shows which type of test data can be analysed efficiently with a specific evaluation function and vice versa. So not only capabilities, but also limitations of this analysis are estimated. In addition, this also serves for optimising the evaluation functions for certain kinds of test data.

Then real measurement data is being evaluated by convolution with the correlation functions to extract further information. Real measurement data is taken from force over time measurements at constant or changing elongations within the elongation region 2 of P pilin protein filament subunits PapA, which are expressed by the uropathogenic bacteria Escherichia coli to withstand physical stress like shear forces inside the human body. This data was measured with the optical tweezers method [1-3]. For this thesis test functions are created in Matlab.

2. Theory

a. Classification of applied evaluations

When evaluating data concerning correlation and incidence of events, a certain degree of stationary is preconditioned, meaning that the change in drift, i.e. second derivative of the amplitude versus data points, should become close to zero over a big interval. Otherwise the linear correlation would differ significantly in different intervals of the data, which would make correlation analysis of the whole data set senseless. The functions evaluating the data should therefore create an output that is very similar for evaluations of different parts of the data set. If, however, there are significant changes in output when comparing different data parts this could sometimes be taken as a change in properties of a system or even a change of state.

Evaluations investigating correlation between data can be categorized into different types depending on the type of obtained output afterwards. One can distinguish between outputs which are scalars, functions without direct link to the input data and functions with direct link to the input data. Only the
latter category makes it possible to compare corresponding data points of the input with corresponding points of the output. In other words, output data points whose x-values lie in the vicinity of an input data point serve as a measure whether an event has occurred at this input data point or not and if so how strong it was. Therefore evaluations with this type of output, which is for example obtained by convoluting data with a correlation related function, are of most interest in this thesis.

However the correlation related function has to be defined at first. Unfortunately, there is probably no such thing as one correlation function which always detects events for very different kinds of input data. This is why the evaluation function has to be adjusted by parameters depending on the properties of the input. By properties is mostly meant the expected shape and occurrence frequency of events in the input data. In case these properties are only little known they have to be estimated. This estimation of properties is done by other evaluations, which do not require specific parameter settings before they can be used on input data and produce a scalar or a function without direct link to the original data. Therefore this type of output can’t yield any information about when an event occurs. It is therefore only used to limit possible parameter settings of the correlation related function discussed above.

b. Test function and nomenclature

The correlation functions are tested at a well defined function \( t(x) \) to show their capacities and their limitations, before they are applied on real data. This test function \( t(x) \) has adjustable parameters (length, periodicity, amplitude, noise as well as coordinate, amplitude, duration, frequency and shape of event occurrences). Out of reasons of simplicity \( t(x) \) is from now on abbreviated as \( t \). Unless otherwise stated, the function \( t \) has a periodicity of \( T=40 \) and one kind of event, which can be applied in both positive and negative directions. Exact properties of \( t \) are described in figure 1.

In order to test the capability of the evaluations, this test function \( t \) can be superposed by noise. Uniformly distributed noise (within a given interval), from now on referred as rectangular noise \( r \), or Gaussian distributions \( N(0,var) \) is used here. The maximum amplitude of the rectangular noise is represented by the value \( A_r \). Each test function \( t \) has events with \( A_{ev} \) being their amplitude. However the absolute value of \( A_{ev} \) alone don’t determine how challenging it is to detect events. It is the ratio \( A_r / A_{ev} \) for rectangular noise or the ratio \( var / A_{ev} \) for a Gaussian distribution which is of importance, the larger the ratio gets the harder it becomes to detect any events. If \( t \) is superposed by noise its type and ratio is from now on included as an index of the test function \( t_{r:A_r}/A_{ev} \) for rectangular and \( t_{N(0,var):var/A_{ev}} \) for Gaussian noise. Moreover, \( x_{i, ev} \) is the coordinate of a positive event which increases the amplitude by \( A_{ev} \) where \( i \) is the order number of the event (both negative and positive events are counted). In figure 1 the second event \( i = 2 \) is labelled, i.e. \( x_{2, ev^+} = 10 \). For negative events \( x_{i, ev^-} \) is used. The duration (length) of the \( i \)-th event is denoted as \( l_{i, ev^+} \) if it is a positive event and \( l_{i, ev^-} \) if it is a negative event. The parameter \( i \) is not written when this duration value holds for all events like in the description of figure 1.
Figure 1: The test function $t(x)$ with no noise has the following defined properties: Periodicity $T=40$, occurrence, strength and length of a positive event within a period: $x_{1, ev}=5$, $x_{4, ev}=30$, $A_{ev}=1$ and $l_{ev}=1$. Occurrence, strength and length of negative events: $x_{2, ev}=10$, $x_{3, ev}=25$, $A_{ev}=-1$ and $l_{ev}=1$.

c. Evaluations using convolution

i. General properties and requirements

The output value $f(k_i)$ of an evaluation should give some information about the value $h(x_i)$ of the input data for $k_i = x_i$. This is necessary to spot the exact occurrence of an event. In other words, the evaluation output and the input data can be laid upon each other and the function value of the evaluation provides event related information of the same $x$-value of the input data.

Moreover the vicinity of the investigated $x$-value of the input data should be somehow included in the evaluation in order to be able to consider changes in the behaviour of the input data, which could indicate an event. A possible approach satisfying the above conditions would be the convolution $f(k)$ of a correlation related function $g(x)$, from now on also called probe function, and the input data $h(x)$. The convolution is defined as \[ f(k) = (g^* h)(k) = (h^* g)(k) = \int h(x) \cdot g(k-x) dx = \sum h(x) \cdot g(k-x) \]

The integral is replaced by a sum, because a discrete spectrum is given by P Pili PapA force over time measurements. Graphically spoken, a convolution $f(k)$ produces the overlap of one function $g(x)$ being shifted over another function $h(x)$ [Error! Reference source not found., p.5]. An overlap is the area obtained by the product of two functions $\int g(x) \times h(x) dx$. The convolution produces the amount of overlap of the function $h(x)$ with the reversed (with respect to the x-axis) probe function which in addition is also shifted by the factor $k$ written $g(k-x)$ [Error! Reference source not found.].

In this thesis the probe function $g(x)$ is only used for the convolution with the input data. This is why, the probe function will be reversed in the course of all evaluations as seen in the equation above. Consequently, the probe function is always written and plotted reversed, that is $g(-x)$. This increases readability because then one can directly compute $\sum h(x) \cdot g(k-x)$ to receive the value of the convolution at $f(k)$. 


As a third property, the output $f(k_1)$ of the convolution should be a constant, for example zero or the mean of the input, when there is no event and no drift in the input data $g(x_i)$ for $k_i = x_i$. Otherwise it should clearly differ from this constant value. This requirement is fulfilled, if the integral of the probe function is zero.

Furthermore, the probe function should be created in a way that the convolution $f(k)$ will have extremes when events are occurring. Since the input data is supposed to have no overall drift (only short changes due to events are possible), this can only be achieved if the value of the convolution $f(k_1)$ is only affected by $x$-values of the input data $h(x)$ which are in the vicinity of $k_1$. This means that for $x \cap k_1$ and $x \cap k_i$ $g(x) = 0$ so that these $x$-values won’t distribute to $f(k_i)$. In addition, the convolution $f(k)$ should also exhibit some kind of fast change or even a jump to intensify a change in amplitude of the input data. In other words, the desired convolution $f(k)$ with the probe function $g(x)$ can be roughly described as a graded derivation of the input data $h(x)$.

**ii. Definition and normalization of the probe function**

When convoluting the probe function $g$ with some input data $h$ which exhibits events, one wants these to be represented by a convolution $f$ with sharp peaks each with amplitude equal or at least proportional compared to the amplitude of the observed event. This implies that $g(x)$ should be normalizable, i.e. it must become zero for large $x$. Additionally, this property is also necessary for noise reduction as written in the paragraph before.

In addition, there should be only one $x$-axis intercept of $g(x)$ at $x_a$ in order to weigh all points of $h(x)$ before $x_a$ negatively and after $x_a$ positively. It seems also reasonable that $g(x)$ should be point symmetric with respect to $x_a$ in order to put exact opposite weight on $x$-values of $h(x)$ having the same distance from $x_a$.

The probe function can be characterised by an interval close to $x_a$ and one interval further away from $x_a$. The intervals can have any shape as long as the properties, which were talked about before, are fulfilled and the intervals are continuous to one another. The two $x$-values where both intervals can be applied mark the minimum and maximum value of $g(x)$. For the vicinity of $x_a$ (which is defined by $|x - x_a| \leq n/2$ where $n$ represents a positive integer) step, linear and exponential connections of the minimum and maximum of $g(x)$ are investigated:

\[
g(-x) = 0 \quad \text{step connection}
\]

\[
g(-x) = \frac{2}{n}(x - x_a) \quad \text{linear connection}
\]

\[
g(-x) = \text{sgn}(x - x_a) \frac{\exp(|x - x_a|) - 1}{\exp(n/2) - 1} \quad \text{exponential connection}
\]

For the interval which is more distant to $x_a$, namely $|x - x_a| \geq n/2$, an exponentially decaying function is defined.
The probe function \( g(x) \) has two parameters; an exponential parameter \( a \), which modulates the speed of the decrease in amplitude for points distant to \( x_a \) at \( |x - x_a| \geq n/2 \). The second parameter is a positive integer \( n \), which describes the distance between maxima and minima value of \( g(x) \). Possible probe functions are plotted in figure 2 on page 8.

To speed up numerical calculations all values of \( g(x) < \exp(d) \) are set equal to zero, because they become close to zero and only contribute little to \( f(k) \) for \( d \leq 0 \). Using the equation above gives:

\[
g(-x) = 0 \quad \text{for} \quad \pm \frac{1}{a} \left( x - x_a - \frac{n}{2} \right) < d \quad \text{and} \quad |x - x_a| \geq \frac{n}{2}
\]

For the numerical calculation \( d = -4 \) is used and \( a \) is replaced by the integer parameter \( b = a \cdot d \). The larger \( b \) the slower the exponent of \( g(-x) \) decreases for larger \( |x - x_a| \). In other words, the amount of non-zero elements of \( g(-x) \) are determined by the parameters \( 2 \cdot b + n \) unless a step connection is used reducing them to \( 2 \cdot b \). This is also visualised in figure 2 on page 8.

From now on the probe function will be denoted \( g(x, b, n, s) \) where \( s \) denotes the shape of connection between minima and maxima, which in this thesis can either be linear (lin), exponential (exp) or zero (step). To abbreviate denotation, the dependence of \( x \) is often not written, though it still exists, of course. The probe function \( g \) is point symmetric to \( x_a = b + n/2 \). The result is illustrated in figure 2 on page 8.

As mentioned before, a change in amplitude due to an event should be mimicked by the convolution \( f(k) \). For this \( g(x) \) has to be normalized in a way that

\[
-\int_{0}^{x_a} g(-x) = \int_{x_a}^{2x_a} g(-x) = 1 \quad \text{with} \quad x_a = b + n/2 \quad (1)
\]

This normalisation is not only dependent on the parameters \( b \) and \( n \), but also on the type of connections of the extremes.

\[
N_{\text{step}} = \left( \sum_{x=1}^{x_a-n/2} \exp \left( -\frac{4}{b} \left( |x - x_a| - \frac{n}{2} \right) \right) \right)^{-1} \quad \text{step connection}
\]

\[
N_{\text{lin}} = \left( \sum_{x=1}^{x_a-n/2} \exp \left( -\frac{4}{b} \left( |x - x_a| - \frac{n}{2} \right) \right) + \sum_{x=x_a-n/2+1}^{x_a} \frac{2}{n} \left( x - x_a \right) \right)^{-1} \quad \text{linear connection}
\]
\[ N_{\text{exp}} = \left( \sum_{i=1}^{x,-n/2} \exp\left(-\frac{4}{b} \left( |x-x_a|-\frac{n}{2} \right) \right) + \sum_{x=x_a-n/2+1}^{x} \exp\left(\frac{|x-x_a|-1}{\exp(n/2)-1}\right) \right)^{-1} \exp. \text{ connection} \]

Since there is a discrete spectrum, the integral is replaced by a sum to calculate the normalization constant. Normalization makes it possible to estimate the strength of an event occurring in the input data. Moreover it is necessary to increase comparability between different convolutions, which were calculated with the same input data but with different parameter settings of the probe function.

**Figure 2:** Three normalized, inversed probe functions \( g(-x,20,30,\text{lin}) \), \( g(-x,20,30,\text{exp}) \), \( g(-x,20,30,\text{step}) \) with \( b=20 \), \( n=30 \) and a linear, exponential or step connection, which connects the extremes of \( g \). From now on these functions would be abbreviated as \( g(20,15,\{\text{lin,exp,step}\}) \) where the dependence on \( x \) is not denoted and the curly brackets are used if functions only differ in this parameter.

Extremes of the convolution \( f \) should appear when an event is occurring or ongoing. This happens when the probe function \( g \) is shifted in such a way that the symmetry point \( x_a \) lies upon the input data point where half of the event has taken place

\[ x_a = x_{i,\text{ev}} + l_{i,\text{ev}}/2 \quad (1) \]

For an instant event, that is with duration of \( l_{i,\text{ev}} = 1 \), the extreme of \( f \) should appear at the coordinate when an event is occurring, which is one coordinate before the detected change in amplitude. Consequently, \( f \) has to be shifted to the left, which is instead done by shifting \( g \) to the right.

\[ f(k) = (g \ast h)(k) = \sum_{x=1}^{2b+n} g(k-(x+1),b,n,s) \cdot h(x) \]

To obtain a value \( f(k_0) \), which serves as a meaningful measure of an event, there should be equally many points from both sides of the symmetry point \( x_a \) of the probe function \( g \) contributing to \( f(k_0) \), so that the integral over these set of points for \( g \) becomes zero. To make sure that this is fulfilled in the evaluations, only values of \( f \) are taken into consideration to which the whole domain of \( g \) contributed to. Therefore the first and last \( b+n/2 \) data points of \( f \) are not a good measure for events, because not the whole domain of \( g \) contributes to these points.
iii. Parameter restrictions of the probe function

When convoluting a probe function \( g \) with a test function \( t \), clear and distinct extremes are seen for \( f \), which occur at the same coordinates as the events of \( t \). This is the case if the parameters \( b \) and \( n \) are set the right way like for the dashed function in the upper panel of figure 3 on page 9. In order to be sure to detect the occurrence point of an event, the following restrictions to the parameters have to be made.

Let \( x_{i,\text{ev}} \) denote the coordinate when the \( i \)th event occurs and \( l_{i,\text{ev}} \) denote the length of the \( i \)th event. The coordinate of the maxima is likely to be shifted if not,

\[
n \leq \begin{cases} 
  x_{i+1,\text{ev}} - \left( x_{i,\text{ev}} + \frac{l_{i,\text{ev}}}{2} \right) \\
  x_{i,\text{ev}} + \frac{l_{i,\text{ev}}}{2} - \left( x_{i-1,\text{ev}} + l_{i-1,\text{ev}} \right)
\end{cases}
\]

In case all events are equal in length \( l_{i,\text{ev}} = l_{i+1,\text{ev}} = l_{\text{ev}} \) one gets:

\[
n \leq \left| x_{i,\text{ev}} - x_{i+1,\text{ev}} \right| \frac{l_{\text{ev}}}{2} \quad (2)
\]

![Figure 3: The convolution \( f(k) \) of noiseless \( t \) with the probe functions \( g(3,\{5,8\},\text{lin}) \) (upper panel) and \( g(3,\{15,20\},\text{lin}) \) (lower panel)](image)

Equation 2 yields \( n \leq 4 \) for the test function \( t \). As illustrated by the solid function in the upper panel of figure 3, a slight violation of this restriction doesn’t necessarily have to shift extremes of the convolution for a linear connection of \( g \). But the larger \( n \) becomes the more likely the maxima are
shifted away from the actual event occurrence. For very high \( n \) two events inducing an amplitude change into the same direction, can only be detected as one as showed in the lower panel. Then the coordinate of the extreme of \( f \) is in the middle of the two original events. Therefore, parameter settings of \( g \) are crucial for output and interpretation of the convolution \( f \).

For exponential and step connections of the probe function \( g(x,b,n,\{\text{exp,step}\}) \), similar results are obtained which are, however, more sensitive to parameter settings of \( n \). This sensitivity is because more values of \( g(x,b,n,\{\text{exp,step}\}) \) are close to zero. Since these functions are normalized to fulfil equation 1, this implies that for some \( x \)-values \( g(x) \) differs more significantly from zero than for \( g(x,b,n,\text{lin}) \) as visualised in figure 2 on page 8. When the parameters are varied, \( g(x,b,n,\{\text{exp,step}\}) \) has strong contributions, which are different from zero, at other \( x \)-values than \( g(x,b,n,\{\text{exp,step}\}) \). This means for the convolution \( f \) that points of the input data \( h(x) \) are weighed significantly differently depending on the parameters of the probe function. The sensitivity to variations of \( n \) is also present for convolutions using \( g(x,b,n,\text{lin}) \) but not as eminent.

Another parameter restriction is that the probe function should only cover one event when \( x_a \) lies centred upon the event. If \( l_{i,ev} = l_{i\pm1,ev} = l_{ev} \) this gives:

\[
\frac{1}{2} n + b \leq \left| x_{i,ev} - x_{i\pm1,ev} \right| - \frac{l_{ev}}{2}
\]

Applying the maximum value for \( n \) from equation 2 and solving for \( b \) yields:

\[
b \leq \frac{1}{2} \left( \left| x_{ev} - x_{\pm1, ev} \right| - \frac{l_{ev}}{2} \right)
\]

or

\[
b \leq \frac{1}{2} n_{\text{max}} \quad (3)
\]

The parameter \( n_{\text{max}} \) represents the highest possible value of \( n \) for equation 2. On the example of the test function \( t \), this leads to \( n \leq 4 \) and \( b \leq 2 \). The convolution \( f(g*t)(k) \) is illustrated in figure 4. It can be seen that for \( b > 2 \) of \( g \) the extremes of the convolution \( f \) are slowly being shifted. More noticeable is that the shape of the extremes becomes less sharp and smaller in amplitude. This effect, the broadening and lowering of the extremes of \( f \) for higher parameter values \( b \) of \( g \) depends whether a linear or exponential connections is used. This is because the values of \( g(x_a,b,n,\text{lin}) \) are closer to zero than \( g(x_a,b,n,\text{exp}) \) for \( \left| x_0 - x_a \right| \leq n \). Consequently, for \( g(x,b,n,\text{lin}) \) the convolution \( f \) is more influenced by values of the input data \( h(x) \) in the vicinity of \( x_a \). Since the parameter \( b \) doesn’t influence the vicinity of \( x_a \) as seen in figure 2 on page 8, linear connections are not as sensitive to variations in \( b \) as exponential connections.
Figure 4: The convolution of the noiseless test1 function \( t \) with the probe functions \( g((2,6),4,\text{exp}) \) (upper panel) and \( g((15,20),4,\text{exp}) \) (lower panel).

In practise, the extremes of the convolution \( f \) don’t necessarily have to be shifted if equation 2 or equation 3 is not fulfilled. For example, if there is a function \( h_f(x) \) with two events, one at \( x_{1,ev} = 20 \) and one at \( x_{2,ev} = 30 \), and one chooses the probe function \( g(x,15,1,\text{lin}) \). The value of the convolution \( f(20) \) will then not only be influenced by the first, but also by the second event. Therefore \( b \) is chosen too large. However \( f(20) \) can still be an extreme of \( f \), a minimum in this case, depending on the exact shape of \( h_f(x) \).

If the parameters are set a little too high, they still provide at least qualitative information about events, since the number of extremes usually doesn’t change then. For parameter values much too high, also this information might be lost as illustrated in the lower panel of figure 3 where two events are detected as one.

For events with an event duration of \( l_{ev} \leq 2 \) the extreme of the convolution at \( k_{i,ev} = x_{i,ev} \) can be \( f(k_{i,ev}) = A_{i,ev} \). However for \( l_{ev} \geq 3 \) the amplitude of an extreme of the convolution is \( f(k_{i,ev}) < A_{i,ev} \) unless \( g(x,b,n,\text{step}) \) is used. This is due to the fact that there will be values of \( g(x,b,n,(\text{lin,exp})) \) contributing to \( f(k_{i,ev}) \) while the event takes place and the total change in amplitude has not yet taken place.

When looking at the upper part of figure 5 one notices that the convolution \( f \) of a probe function \( g(x,b,n,\text{step}) \) with the noiseless test function \( t \) seems to form plateaus. These plateaus occur at \( k_{i,ev} \approx x_{i,ev} \). Extremes of the convolution \( f \) should represent an event, but the plateau of \( f \) makes it impossible to detect the event coordinate \( x_{i,ev} \) precisely.
In case the input data is changed to \( t_{r,0.2} \) which shows some noise, the plateaus in \( f \) disappear, which is illustrated in the lower panel of figure 5. Instead sometimes more than one maxima, which is not very distinct, can be seen around \( x_{ev} \), especially for the convolution \( f \) of \( g(2,8,\text{step}) \) with \( t_{r,0.2} \). Then one is not able to conclude whether each extreme represents an event. And if not, it is also unclear which extreme should be counted as an event and which not. In this example the convolution \( f \) actually makes it harder to detect events, because one was able to see events in the input data \( t_{r,0.2} \) but not anymore in \( f \).

![Convolution Graph](image)

**Figure 5:** Convoluting \( g(2,\{4,8\},\text{step}) \) with the noiseless test function \( t \) (upper panel) and \( (2,\{4,8\},\text{step}) \) up on \( t_{r,0.2} \) with uniformly distributed noise having a maximum value \( \lambda_r=0.2 \) (lower panel).

As a result, although the step connection is the only connection which is capable of resembling the exact amplitude of events with a long duration \( l_{ev} \geq 3 \), it is not investigated further and focus is put on a linear and exponential connection, which is less sensitive to noise.

When the input has a constant drift, the mean value of the convolution is eventually shifted by a constant. Consequently, when the drift of the data changes, the output of the convolution shifts like shown in figure 6. The strength of the shift depends on the length of the domain of the probe function. The stronger contributions of \( g \) far away from \( x_{ev} \), the larger the shift becomes, which makes the detection of events especially in opposite direction to the change of drift more difficult.
In conclusion, an exponential connection of the probe function $g(x)$ puts more emphases on points far away from its symmetry point at $x = x_a = b + n/2$ than a linear connection. Mostly the parameter setting of $n$ is crucial for the obtained coordinate of the extremes of the convolution with the input data, whereas different settings of $b$ tend more to change the shape of the extreme rather than shifting it. The suggested restrictions for the parameters in this chapter limit their maximum values. However when it comes to reduce the effect of noise, maybe high parameter settings are more appropriate.

### iv. Capability of analysing noisy data

The effect of noise on the convolution $f$ is investigated. Ideally extremes of the convolution due to noise should be much smaller than these due to events. Therefore, one likes to minimize the extremes of the convolution $f$ caused by noise. In this subsection stress is put on the comparison between the convolution of the probe function $g$ with a pure noise signal and a signal having the same noise distribution, but being superposed by an event. It is investigated how high the noise-event amplitude ratio $A_e/A_n$ can get before correct detection of events becomes unlikely and which values of $b$ and $n$ serve best for this purpose. This means it is attempted to maximize the amplitude differences of the extremes of $f$ between the probe function $g$ and input with or without an event.

In practise, it is impossible to do two experiment runs showing the exact same noise behaviour, only the probability of noise contribution stays equal. So in turn also the outcome between different evaluation trails at the same noise ratio can differ. This is why later on in this section a series of convolutions obtained by noise signals without an event is compared with a series of convolutions obtained by signals having similar noise distributions but exhibiting an event.

A Gaussian distributed pure noise signal $N(0,1)$ is convoluted with the probe function and parameter settings minimizing extremes are searched for. Generally, Gaussian noise can sometimes exhibit a strong contribution, which result in strong amplitude changes of the convolution, and is therefore more difficult to deal with than rectangular noise where the maximum amplitude change is limited.

The higher the value for $b$ and $n$ the larger the domain of $g$ and the more points of the input data are being evaluated for each output coordinate of the convolution. Since the noise is uncorrelated, one
expects to decrease its contribution for higher values of $b$ and $n$ in order to ameliorate the effect of fluctuations.

For visualisation, 3D plots with different values of $b$ and $n$ as x- and y-axis and the maximum absolute value of the convolution $f$ of $g$ with Gaussian noise $N(0, 1)$ as z-axis are used. To shorten nomenclature:

$$\max |f(g(x, b, n, s) * N(0, 1))| := f_{\max}(g(b, n, s) * N(0, 1))$$

As expected, the maximum of the convolution due to noise is minimized for high settings of $b$ and $n$. Most of the suggested parameter setting reduce the absolute maximum of the convolution $f$ of $g$ with $N(0, 1)$ roughly below 0.8 as can be estimated from figure 7.

![Figure 7: Three dimensional plots of the absolute maximum value of the convolution $f$ of Gaussian noise $N(0, 1)$ with the probe function $g$ once with a linear connection (upper panel) and once with an exponential connection (lower panel) dependent on the parameters $b$ and $n$.](image)

It also shows that for higher $b$ the maximum value of the convolution decreases more steadily than into the $n$ direction. This effect is stronger for exponential connections of $g$, because for this connection a change in $n$ mostly shifts the coordinate of the extremes of $g$ but not so much its amplitude. However in order to decrease the influence of noise fluctuations, no input data point should be weighted much stronger by $g$ than others in the convolution $f$.

The three dimensional plots for rectangular distributed noise $r_1$ with $A_j = 1$ look similar, but since the maximum noise has an amplitude of one (unlike for the Gaussian distribution) the absolute
maximum value of the convolution $f_{\text{max}}(g(b,n,s)*r_1) < 0.5$ is lower compared to Gaussian noise for higher values of $b$ and $n$.

After investigating the absolute maximum contribution of the convolution $f$ between $g$ and pure noise, it is compared with input data having an event. Parameter settings of $g$ are searched for which give the strongest difference of the absolute maximum of the convolution between $g$ and pure noise as well as noise superposed by an event. For this purpose the input data $u$ is created, which has a negative event at coordinate $x_{ev} = 40$ with $A_{ev} = -1$ and an event length $l_{ev} = 1$. The function $u$ is superposed either by rectangular noise $r$ with the corresponding noise ratio $A_r/A_{ev}$ or Gaussian noise $N(0,\text{var})$ with $|\text{var}/A_{ev}|$. This is denoted similar to the noise in the test function $t$, $u_{r[A_r/A_{ev}]}$ for rectangular noise and $u_{N(0,\text{var})[A_r/A_{ev}]}$ for Gaussian noise. For example, if $u$ is superposed by rectangular noise with a maximum amplitude of $A_r = 1.5$ it is written as $u_{r,1.5}$.

The difference of the absolute maximum values of the convolution $f_{\text{max}}(g(b,n,\text{lin})*u_{r,0.5}) - f_{\text{max}}(g(b,n,\text{lin})*r_{0.5})$ is plotted over different settings of the parameters $b$ and $n$ in figure 8. Ideally, the values of $b$ and $n$ should be chosen in a way, maximising the difference of the extreme of the convolution between the data, which is purely noisy and the one including an event. Consequently, the maximum value of figure 8 represents the optimum parameter setting.

![Figure 8: Three dimensional plot of the difference between the maximum absolute value of the convolution $f$ between the probe function $g(b,n,\text{lin})$ with $u_{r,0.5}$ and the rectangular noise distribution $r$. The higher $b$ and $n$, the larger this difference becomes.](image_url)

Looking at figure 8 above the following relation holds most of the time

$$f_{\text{max}}(g(b_1,n_1,\text{lin})*u_{r,0.5}) - f_{\text{max}}(g(b_2,n_1,\text{lin})*r_{0.5}) - (f_{\text{max}}(g(b_2,n_2,\text{lin})*u_{r,0.5}) - f_{\text{max}}(g(b_2,n_2,\text{lin})*r_{0.5})) \geq 0 \quad \text{for} \quad b_1 \geq b_2 \quad n_1 \geq n_2$$

In words, the bigger the parameters $n$ and $b$ are chosen, the larger the difference between the absolute maximum of the convolution due to noise and the convolution at the event tends to get. In this case the maximum difference between the two surfaces is situated at the optimized parameters $b_{opt} = 23$ and $n_{opt} = 20$, for which it is easiest to distinguish between noise and event.
Using these optimized parameters $b_{opt}$ and $n_{opt}$ the following amplitude difference ratio between event and non event data is obtained for the example above.

$$\frac{f_{\max}(g(b_{opt}, n_{opt}, \text{lin})*u_{f,0.5})}{f_{\max}(g(b_{opt}, n_{opt}, \text{lin})*r_{0.5})} = \frac{A_{f,\text{ev}}}{A_f} \approx 4.3$$

In this case it is definitely possible to distinguish between pure noise and event superposed by noise, because the absolute maximum of $f$ is for the latter input more than four times as high as compared to pure noise input. It is proposed that an event can be detected if the absolute maximum of the convolution $f$ between the probe function $g(b_{opt}, n_{opt}, s)$ and the input function $u$ is at least 50% stronger than the absolute maximum of the convolution $f$ between the probe function $g(b_{opt}, n_{opt}, s)$ and pure noise.

$$A_{f,\text{ev}} / A_f \geq 1.5 \quad (4)$$

Furthermore, one has to check whether the absolute amplitude maximum of the convolution $f_{\max}(g*u)(k_0) = A_{f,\text{ev}}$ is actually situated at a $k_0$-value which is similar to $x_{ev} + l_{ev} / 2 \approx 42$ given by equation 1 on page 8. This ensures that the absolute maximum of the convolution actually represents the event of the input. This is fulfilled in the example above because the evaluation yields $k_0 = 41$.

In figure 7 and figure 8 it is noticeable that the difference of the absolute maximum value of convolution is sensitive to a change in $n$ given low values of $b$. These low parameter settings for $b$ let only few data points contribute to the convolution, making it sensitive to noise fluctuations in the input data. As a result, the maximum value of $f$ can be obtained because a noise fluctuation of the input data at coordinate $x_0 = x_{ev} \pm n$ in the vicinity of the event emphasises this change of amplitude. The probe function $g(k_0 - x_0 - 1,1,n,s)$ for $k_0 = x_{ev} + x_a + 1$ has an extreme at this coordinate and therefore $h(x_0)$ greatly contributes to $f(k_0)$. In consequence, low parameter setting of $b$ can accidentally be the best one for this specific noise distribution, but probably not for others. To decrease this risk, only parameter settings of $g(x,b \geq 5,n,s)$ are investigated.

The ratio $A_{f,\text{ev}} / A_f$ can vary even between different samples having the same noise ratio $A_f$. Consequently, for each noise ratio nine tests are run to give a probability of a successful detection of the event. The definition of a success is equation 4 and

$$\left| k_{opt} - \left( x_{ev} + \frac{l_{ev}}{2} \right) \right| \leq 5$$

In addition, this procedure tries to reveal patterns for an optimal choice of the parameters $b_{opt}$ and $n_{opt}$. The results are listed in the appendix in table 1 and 2 on page 28. There it can be seen that the success rate for noise ratios $|A_f / A_i| \geq 2$ is very low. Most of the time the event at $x_{ev} = 40$ can’t be detected correctly anymore no matter how the probe function’s parameters $b$ and $n$ are set. Even for lower noise ratios, especially for Gaussian distributed noise, it is not guaranteed that events can be detected. If the ratio of the maximum amplitudes $A_{f,\text{ev}} / A_f$ is equal for Gaussian and rectangular
noise input data, the absolute maximum of the convolution \( f_{\text{max}}(g(b_{\text{opt}}, n_{\text{opt}}, s) * u)(k_0) \) situated at the coordinate \( k_0 \) is more likely to differ from the actual event coordinate \( x_{\text{ev}} \) if there is Gaussian noise.

Furthermore, no clear pattern for the optimum parameters \( b_{\text{opt}} \) and \( n_{\text{opt}} \) can be extracted from table 1 and 2, neither for rectangular nor for Gaussian distributed noise. The parameter values \( b_{\text{opt}} \) and \( n_{\text{opt}} \) vary strongly, often from very low values to high ones. In case of low values the noise might strengthen the amplitude change close to the event as discussed earlier in this chapter. Since the contribution to the amplitude due to noise does not usually strengthen the amplitude change at an event, high values for \( b \) and \( n \) should be favoured. In addition, \( b_{\text{opt}} \leq n_{\text{opt}} \) often holds in the tables 1 and 2 in the appendix. In any case this is postulated by the restrictive equations 2 and 3 on page 9 and page 10.

Up to now it is postulated that events can be detected if \( A_{f,\text{ev}} / A_f \geq 1.5 \). But also for lower ratios, it might be possible to detect events. A histogram which counts the number of extremes of the convolution \( f(g(b_{\text{opt}}, n_{\text{opt}}, s) * h)(k_0) \) of the probe function \( g \) with an input function \( h \) could be used for further evaluation. If there is a drift the histogram is shifted along the \( x \)-axis. More important if there are many events happening (side) maxima might be seen, which are likely caused by events.

In conclusion, unfortunately no method is found which chooses the parameters \( b \) and \( n \) always perfectly. However, higher values of \( b \) and \( n \) and \( b < n \) are more likely to detect events in noisy data correctly. The range of applied parameters of \( g \) can be set as high as possible, as long as the upper parameter restrictions in the chapter before given by equation 2 and 3 is not violated. These restrictions depend on the distance between events and the length of each of them. Those properties are usually unknown at first. Consequently, before a range of reasonable parameters of \( g \) can be set, the distance between events has to be at least roughly estimated, which is done in the following chapter.

**d. Correlation functions assisting convolution analysis**

Since for a successful analysis using convolution the length and the distance between events should be approximately known beforehand, other evaluations have to be applied in addition. The output coordinates of the following functions are not directly linked to the coordinate of the measurement data. Consequently, directly comparing a value of the obtained output with one input data point is not possible. These evaluation functions are being used to investigate global, common properties over the whole data set like drift and periodicity. If there are events, direct identification and detection of them is impossible, but rough estimates about strength \( A_{x_{\text{ev}}} \), length \( l_{x_{\text{ev}}} \) and distance between events \( |x_{i,\text{ev}} - x_{i+1,\text{ev}}| \) can possibly be given. Therefore these functions are mostly used as estimates for limiting the number of feasible parameter settings \( b \) and \( n \) of the probe function \( g \).

The function \( g_{\text{mean}}(l) \) calculates the mean of the absolute difference between input data points \( h(x) \) depending on their lag (correlation) order. For lag order one the difference of each data point is done with its successor, for lag two the difference of each point with its second next point is used and so on.
The integer \( N \) represents the number of data points in \( h(x) \). The function \( g_{\text{mean}}(l) \) works as a measure of stationary, because if \( g_{\text{mean}}(l) \) seems to saturate for higher lags it indicates that the measurement data is stationary, i.e. there is no drift. Vice versa if for high lags \( g_{\text{mean}}(l) \) keeps increasing a drift in the input data is very likely.

When \( g_{\text{mean}}(l) \) is applied on a periodic function, minima are seen when the noise-event ratio is roughly \( A_e / A_{ev} \leq 1 \) as shown in figure 9. These minima indicate periodicity of input data. For stronger noise it becomes highly unlikely to be able to extract information about periodicity.

\[
g_{\text{mean}}(l) = \frac{1}{N-l} \sum_{x=1}^{N-l} |h(x) - h(x+l)|
\]

Again \( N \) denotes the number of data points. When electrical noise is expected throughout the experiment, which is usually very strong in amplitude and very short in length, the obtained max value is most likely to be due to noise. As a consequence, a median of at least five points is used to suppress noise, because one noise point from the input data \( h \) will lead to two noise influenced points in \( g_{\text{max}}(l) \).
\[ g_{\text{max}}(l) = \max \{ \text{med}5 \{ |h(x) - h(x + l)| ; x = \{1, \ldots, N - l - 2\} \} \]  

Figure 10: The function \( g_{\text{max}}(l) \) (using a median of 5) applied on data points of the test function \( t_{r,0.05} \) with the rectangular noise amplitude \( A_r=0,05 \). Since in this example the event length \( l_r=1 \), the obtained values \( g_{\text{max}}(l) \) and \( g_{\text{max}}(2) \) are only due to noise.

When the input data doesn’t exhibit a drift, the function \( g_{\text{max}}(l) \) can approach plateaus for a certain range of lags. The lowest lag value of a plateau can serve as a lower estimate of the length of a series of events all either increasing or decreasing the amplitude of the data. Since the median of five data points is used, the effective length of a series of events is probably \( l_{\text{eff}} = l_n - 2 \) the first lag value \( l_n \) of a plateau minus two. In the case of figure 10 two plateaus are seen. The first plateau has a value of \( g_{\text{max}}(3) \approx 1 \). This means that a single or a series of events has the effective length \( l_{\text{eff}} = 1 \), which corresponds to the actual event length of the test function \( t \). The second plateau begins at lag order \( l_1 = 17 \). This could mean that another event or series of events happen after at least 15 data points. The second plateau is twice as high as the first one, indicating that in the former the event series consists out of twice as many events as in the first one, which in the case of \( t \) is two events. This means that the domain of the probe function \( g \), which is determined by \( b \) and \( n \), should be, if possible, chosen equal or smaller than the minimum length of an event series in order to detect this series properly.

Furthermore, the values of possible plateaus can serve as an estimate for the amplitude of an event, if it is subtracted by two times the maximum expected noise in case there are many events. This is because \( g_{\text{max}}(l) \) detects the maximum change in amplitude due to an event superposed by noise. And for many events the maximum change in amplitude is obtained when the noise accidentally strengthens the event. This occurs if noise influences the data point before an event in the opposite direction than the data point after an event. For example, in figure 10 \( l_n = 1 \) holds for \( t \) and the value \( g_{\text{max}}(3) \) gives either directly an upper estimate for the event’s amplitude or an estimate for the amplitude when the value is subtracted twice by the maximum amount of noise \( 2A_r = 0,1 \), which gives \( A_{ev} \approx 0,95 \). This value is smaller than the actual event strength \( A_{ev} = 1 \) of \( t \), because there were not many events, namely only four, included in the input data.

In figure 10 the periodicity of \( t \) is also seen in \( g_{\text{max}}(l) \). This feature, however, is lost quickly for higher noise values. For some input data \( g_{\text{max}}(l) \) stays approximately constant for high lag values with slight fluctuations, indicating just like \( g_{\text{mean}}(l) \) stationary of the input data. The value of this last plateau
might be due to the maximum amount of amplitude change caused by noise plus event strength. This value, however, can also be superposed by every kind of drift, making it often impractical to interpret.

In conclusion, plateaus of \( g_{\text{max}}(l) \) can be used as estimates for \( l_{ev} \) and therefore help to find a reasonable range of parameters \( b \) and \( n \) of the probe function \( g \). However, one has to keep in mind that using the median of five points usually leads to an effective lag which is \( l_{\text{eff}} = 1\text{-}2 \), but doesn’t necessarily have to.

### 3. Analysis of Pili PapA measurement data

The specified evaluation functions from the chapter before are now applied on real measurement data. This data is taken from mechanical force versus time measurements of PapA, which is a part of the biological macromolecule Pili exposed by the uropathogenic bacteria Escherichia [1].

There are three different elongation regions of Pili each with a different force response. The response within region 1, which is valid for short elongations, is linear similar to a classical spring. The force in region 2 is constant and region 3 shows non-linear behaviour [4].

When PapA is being elongated within region 2 chemical bonds are supposed to open and alpha-helix structures are modified to beta-structures, elongating the molecule and thereby counterbalancing the increased physical stress on the molecule. A bond closing process is vice versa. Both lead to a constant force response in region 2, which enables the bacteria Escherichia coli to withstand shear forces inside the human body. Since these openings and closings are not continuous they can be referred as events. Furthermore, these bond opening and closings are not only present during different elongation speeds, but also during steady state conditions [5-7].

Relatively sharp changes (but little in amplitude) are expected since the time length of bond openings and closings are supposed to be rather short. In principle, these events should be detectable by the event analysis discussed above, if noise is not too strong. To limit reasonable parameter settings of the probe function \( g \), general investigations towards the input data are performed.

#### a. Preliminary evaluation

To find out whether the shape of the input data received by force measurements differs before, while or after an elongation at a certain velocity, the functions above were applied before, while and after elongations within elongation region 2 of Pili. Different behaviour can be caused by a change of properties of the input data like number, periodicity (if there is any at all) and shape of the opening and closing events or by a varying strength of noise. Mostly measurement data series with elongation speeds of \( v = 5,7\mu\text{m/s} \) and \( v = 7,9\mu\text{m/s} \) are evaluated. The elongation speed is denoted as an index of \( v \) as well as whether the data is obtained before (b), while (w) or after (a) elongation. The results look similar for most other elongation speeds, especially before and after elongations.
Figure 11: The function $g_{\text{mean}}(l)$ evaluating force versus time measurement data from PapA Pili before (b), while (w) and after (a) the macromolecule is elongated at a speed of $v=5.7 \, \mu m/s$ and $v=7.9 \, \mu m/s$.

The function of $g_{\text{mean}}(l)$ in figure 11 differs depending on whether measurement data before, while or after elongation was used. This indicates that P Pili behaves differently and/or there is a change in noise for each case. For high lags $g_{\text{mean}}(l)$ seems to become stationary before and after elongation since the values don’t seem to increase, whereas during elongation they still increase slightly. That the values of $g_{\text{mean}}(l)$ are higher after than before elongation most likely originate from an increase in noise which could be due to the fact that after elongation the molecule is longer and vibrations could be stronger. As a result of figure 11, it might be easier to detect events before elongation which in turn most focus is put on.

Figure 12: The maximum difference between data points $g_{\text{max}}(l)$ of a certain lag $l$ before (b) and during (w) an elongation of P Pili with a speed of $v = 5.7 \mu m/s$. The upper panel illustrates lag orders from 1 to 400, whereas the lower focuses on lag orders smaller than 50.

Bond openings and closings are probably also happening during steady state conditions [1]. In figure 12 the function $g_{\text{max}}(l)$ before elongation doesn’t vary significantly from the value one for different lags. This implies that the event length $l_e \leq 3$ is very short as suspected. Otherwise $g_{\text{max}}(l)$ would first
increase for small lags before it reaches a plateau. As a consequence very low parameter settings should be favoured for the probe function $g$. However this does not suppress noise efficiently. Consequently, different values for $b$ and $n$ have to be tested to balance out noise reduction and the ability to visualize single events.

The value of $g_{\text{max}}(l)$ doesn’t increase strongly for higher lag orders before elongation of P Pili unlike during elongation where the value of $g_{\text{max}}(l)$ rises until a lag of approximately $l=50$. The amplitude of events, which are in this case bond openings or closings, is independent of elongation speed. Consequently the increase of $g_{\text{max}}(l)$ during elongation either originates from a drift or from a larger number of same events occurring in series, which in the case of an elongation would be a series of bond openings. In case there is a drift $m$ during elongation, it could be approximated by $m = (g_{\text{max}}(50) - g_{\text{max}}(1)) / 50 \approx (3.5 - 1.0) \text{pN}/50 = 0,05 \text{pN}$. This drift is approximately one order of magnitude weaker than the expected event strength $|A_v| = 0,4 \text{pN}$ for this measurement set up [1], so events could be still detectable if the noise ratio is not too heavy.

**b. Event analysis using convolution**

For the event analysis using the convolution $f$ between the probe function $g$ and P Pili force over time data, measurement data is chosen which seems to be most feasible for an analysis. Due to the results of the chapter before, this is probably the measurement of the steady state system of Pili before an elongation speed of $v = 5,7 \mu m/s$, because there is no drift (figure 12) and the noise level might be smaller than for other measurements (figure 11) so it might be easiest to detect events.

Before the convolution $f$ is calculated, reasonable values for the parameters $b$, the length of one decreasing exponent, and $n$, the distance between the two extremes of the probe function $g$, have to be chosen. From the previous measurement analysis the parameters $b$ and $n$ should be small. However, as concluded from the chapter analysing noisy data, choosing high parameter values are more likely to reduce the effect of noise. As a consequence, parameter values of $g$ are chosen in a moderate range so that between 14 and 25 points of the measurement data points contribute to one point of $f(k)$. In any case equation 3 on page 9, which states $b \leq n/2$, will be obeyed.

Figure 13 shows the input data and the convolution $f$ with different probe functions $g$ having linear connections. Changes for the parameter $b$ don’t have a strong impact on the shape and the coordinates of the extremes of the convolution. The higher $b$ the smaller the extremes $f(k_0)$ become. On the one hand, this is due to noise being more averaged out, which is desirable. On the other hand, decreasing amplitude of an extreme $f(k_0)$ for higher settings of $b$ could give a hint that there is already an opposite event contributing to the convolution at $k_0$, because the domain of $g$ becomes too large and too many data points from the input data are included in the evaluation. Additionally, the convolution becomes more sensitive to drifts for a larger domain of $g$ as illustrated in figure 6 on page 13. Looking at figure 13 convolutions obtained by probe functions with the parameters $b=2$ and $b=5$ but fixed $n$ differ very little from one another so both values of $b$ are probably applicable. Here, for further investigations the higher value for $b$ is taken to decrease the influence of noise.
Figure 13: The convolution of $g((2,5),(10,15),\text{lin})$ with Pili measurement data before an elongation of $v=5.7\mu\text{m/s}$.

Additionally it can also be extracted from figure 13 that the shape of $f(k)$ is more sensitive to the choice of $n$. Different settings lead to different coordinates of the extremes, but it is hard to judge whether some of them are due to events and if so which. Moreover, the number of extremes in $f(k)$ decreases for a higher value of $n$. It is challenging to decide whether in this case $n=10$ or $n=15$ is more appropriate. Since $f(k)$ is strongly influenced by $n$ both possibilities will be considered.

Figure 14: The convolutions $f(k)$ of probe functions $g(5,(10,15),\{\text{lin,exp}\})$ having either a linear or an exponential (solid lines) connection with Pili measurement data before elongation of $v=5.7\mu\text{m/s}$.
Linear and exponential connections of the probe function $g$ being convoluted with the measurement points are compared in figure 14. There it can be seen that the convolution $f(k)$ seems to be less smooth and often stronger in amplitude for exponential connections for $n=15$, which is due to the fact that few values of the measurement data are weighted much stronger than others in $f(k)$. This is why evaluations using probe functions with an exponential connection are probably more sensitive to noise. As a consequence, focus is put on the linear connection.

Figure 15: The convolution of a measurement series of Pili during an elongation speed of $v=5.7\mu m/s$ with $g(5,\{10,15\},\text{lin})$. The labelled point at $f(800)$ is obtained from the convolution using the probe function $g(5,10,\text{lin})$.

Figure 15 illustrates the convolution $f(k)$ of $g(5,\{10,15\},\text{lin})$ with data taken during elongation of Pili at an speed of $v=5.7\mu m/s$. It is noticeable that the measurement data during elongation has a slight drift. In addition, the extremes of the convolution differ a little stronger from zero than for measurement data obtained before elongation in figure 13 and 14. This is either due to more noise or an increased number of equal events happening in a short time period or both. For example, the minimum of the convolution $f(g(5,10,\text{lin})*v_{w,5.7})(800)$ could be caused by more than one event. For the given measurement conditions the event amplitude is supposed to be $A_v \approx ±0,4\ pN$ [1]. In consequence the minimum of $f(800)$ might be influenced by two bond opening events.

After the parameters for the probe function are set to $b=5$ and $n = 10, 15$ with a linear connection, the extremes of the convolution $f$ can further be evaluated with histograms to decide which of them are due to events.

The maxima at positive values and minima at negative values of the convolution $f(k_0)$ are plotted in a histogram in figure 16. The extremes with amplitudes close to zero are certainly due to noise. The stronger the amplitude, the higher the possibility that this extreme is due to an event. Ideally, if there are many events side maxima can be seen which are very likely to come from events. Unfortunately, side maxima are not clearly revealed.
Figure 16: The histogram of the maxima at positive values and the minima at negative values of the convolution $f(k_0)$ with $g(5,10,15,\text{lin})$ and Pili measurement data before elongation of $v=5,7\,\mu m/s$.

However, one could suspect that $|f(k_0)| > 0,5\,pN$ represent events. For each panel in figure 16 one would then count $N_{ev} = 11$ events. To estimate an event frequency, the probability of an event per data point has to be calculated first. The total amount of data points of this measurement are $N = 635$. The convolution has in total $N_f = N - 2 \cdot b \cdot n$ data points, so that for the left panel $N_f = 615$ and for the right panel $N_f = 610$ is obtained. This leads an event occurring probability $P_{ev} = N_{ev} / N_f$ in each case. The force over time measurements were taken at a frequency $\omega = 5khz$. This would lead to an estimated event frequency

$$\omega_{ev,0.5} = P_{ev} \cdot \omega = \frac{N_{ev}}{N_f} \cdot \omega \approx 89hz \quad \text{for} \quad n = 10$$

$$\omega_{ev,0.5} = P_{ev} \cdot \omega = \frac{N_{ev}}{N_f} \cdot \omega \approx 90hz \quad \text{for} \quad n = 15$$

The index of the event frequency $\omega_{ev,0.5}$ denotes that $|f(k_0)| > 0,5\,pN$ is considered as an event. As mentioned before the change in amplitude due to bond openings and closings are expected to be about $|A_v| \approx 0,4\,pN$ for the given measurement set up [1]. So if the estimation of the event frequency is repeated, but now with the expected amplitudes $|f(k_0)| > 0,4\,pN$ being counted, one receives:

$$\omega_{ev,0.4} = P_{ev} \cdot \omega = \frac{N_{ev}}{N_f} \cdot \omega = \frac{21}{615} \cdot 5khz \approx 17hz \quad \text{for} \quad n = 10$$

$$\omega_{ev,0.4} = P_{ev} \cdot \omega = \frac{N_{ev}}{N_f} \cdot \omega = \frac{18}{610} \cdot 5khz \approx 148hz \quad \text{for} \quad n = 15$$

These two different results demonstrate that the choice of the parameter $n$ affects the result of the estimated event frequency. In this analysis it was difficult to determine which extreme of the convolution was caused by noise and which by an event. Depending on the interpretation which extreme should be counted as an event it leads to strongly varying results as seen above. As an outcome, the noise ratio of the evaluated data reaches, if not already surpasses, the possibilities of this event analysis approach.
4. Discussion

Event analysis is possible with the convolution between a normalized, point symmetric probe function and measurement data. In case the noise ratio is twice as weak as the strength of an event, this procedure is always successful not only in detecting occurrences but also in determining strength of events relatively accurately.

For noise ratios until twice as strong as the event, it is still possible to detect events. Then, however, successful detection of events does not only depend on the shape of $g$, but also on the frequency of events. The more often events occur, the more difficult correct parameter settings of the probe function is found to spot each event individually and with the correct amplitude. Different shapes and widths of $g$ were tested. A linear connection between the extremes of $g$ works probably better than an exponential or step connection. For the latter connections certain values of the input data are weighted strongly by the probe function during convolution, making the convolution more sensitive to noise.

Highly noisy data, when the noise is more than twice as strong as an event, is most often not being evaluated successfully when convoluted with the probe function. To receive rough estimates, however, it is possible to run the same procedure for many different parameter settings of the probe function and try to find out similarities. And if possible, the same evaluation should be done with data only showing the same noise distribution. Comparing these two analyses could then give some estimates like frequency or length of events.

It is possible to try more different shapes of the probe function to improve evaluation. However, the convolution should always serve as some kind of graded derivation. Whether there is a lot more potential in this approach is questionable. Stronger improvements in analysing data might be achieved when this method is combined or supplemented with other approaches.
5. References

[1] Fällman E., Schedin S., Jass J., Uhlin B. E., Axner O. 2005. The unfolding of the P pili quaternary structure by stretching is reversible, not plastic. EMBO reports Vol 6, NO 1, 2005


6. Appendix

a. Tables

<table>
<thead>
<tr>
<th>A_r/A_{ev}</th>
<th>I_{ev}</th>
<th>b_{opt}</th>
<th>n_{opt}</th>
<th>A_{I_{ev}}/A_{f}</th>
<th>\lvert k_0 - (x_{ev} + \lfloor l_{ev}/2 \rfloor) \rvert</th>
<th>success</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>3.2</td>
<td>3.8</td>
<td>4.0</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>25</td>
<td>5</td>
<td>3.3</td>
<td>5.6</td>
<td>4.4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>25</td>
<td>2.6</td>
<td>3.4</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>25</td>
<td>1</td>
<td>24</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5</td>
<td>19</td>
<td>5</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>9</td>
<td>25</td>
<td>23</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>5</td>
<td>25</td>
<td>25</td>
<td>7.0</td>
<td>6.3</td>
<td>7.2</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>7.3</td>
<td>6.5</td>
<td>7.5</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>25</td>
<td>23</td>
<td>8.1</td>
<td>7.2</td>
<td>8.2</td>
</tr>
<tr>
<td>1.0</td>
<td>1</td>
<td>25</td>
<td>25</td>
<td>3.5</td>
<td>2.8</td>
<td>2.3</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>2.5</td>
<td>2.1</td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>17</td>
<td>25</td>
<td>4.1</td>
<td>3.3</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>7</td>
<td>25</td>
<td>4</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>9</td>
<td>25</td>
<td>4</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>9</td>
<td>25</td>
<td>2.9</td>
<td>2.8</td>
<td>2.0</td>
</tr>
<tr>
<td>1.0</td>
<td>5</td>
<td>25</td>
<td>25</td>
<td>4.6</td>
<td>4.0</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>2.9</td>
<td>2.8</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>9</td>
<td>25</td>
<td>3.5</td>
<td>3.0</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>7</td>
<td>25</td>
<td>4</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>9</td>
<td>25</td>
<td>4</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>9</td>
<td>25</td>
<td>2.9</td>
<td>2.8</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>7</td>
<td>25</td>
<td>4</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>1.5</td>
<td>1</td>
<td>25</td>
<td>25</td>
<td>2.4</td>
<td>2.9</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>2.9</td>
<td>3.4</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>1.6</td>
<td>1.9</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>7</td>
<td>25</td>
<td>4</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>9</td>
<td>25</td>
<td>4</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>9</td>
<td>25</td>
<td>2.9</td>
<td>2.8</td>
<td>2.0</td>
</tr>
<tr>
<td>1.5</td>
<td>5</td>
<td>25</td>
<td>25</td>
<td>2.2</td>
<td>2.0</td>
<td>2.6</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>25</td>
<td>25</td>
<td>0.7</td>
<td>1.7</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>5</td>
<td>25</td>
<td>2.2</td>
<td>1.6</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>7</td>
<td>25</td>
<td>4</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>9</td>
<td>25</td>
<td>4</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>9</td>
<td>25</td>
<td>2.9</td>
<td>2.8</td>
<td>2.0</td>
</tr>
<tr>
<td>2.0</td>
<td>1</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>11</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5</td>
<td>21</td>
<td>25</td>
<td>4</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5</td>
<td>21</td>
<td>3</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>2.0</td>
<td>5</td>
<td>25</td>
<td>17</td>
<td>25</td>
<td>2</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>21</td>
<td>5</td>
<td>25</td>
<td>2</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>2,5</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>7</td>
<td>5</td>
<td>10</td>
<td>11</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 1: The difference between the convolution \( f(g(b,n,lin)*u) \) and \( f(g(b,n,lin)*r) \) for different noise ratios \( A_r/A_{ev} \) and event lengths \( I_{ev}=1 \) and \( I_{ev}=5 \). For each noise ratio nine similar trails are done. The parameter \( b \) of the probe function is iterated from 5 to 25 in steps of 2 and \( n \) is iterated from 1 to 25 in steps of 1. Using the optimum parameter values \( b_{opt} \) and \( n_{opt} \) leads to the convolution with the biggest maximum amplitude difference ratio \( A_{I_{ev}}/A_{f} \) between an event superposed and pure noisy data. An event analysis is called a success (=1) if \( A_{I_{ev}}/A_{f} > 1.5 \) and the the \( k_0 \)-value of the extreme of the convolution using optimal parameters is at the most five points away from the actual event \( x_{ev} \): \( |k_0 - (x_{ev} + \lfloor l_{ev}/2 \rfloor) | < 6 \).
Table 2: Same analysis like in table 1, but now Gaussian distributed noise is evaluated.

b. Source code

Main script

```
% Calls functions for correlation calculation
clf
close all

% Display menu
disp('1: Mean value of difference')
disp('2: Maximum force difference using a median')
disp('3: Prop for a certain force diff bigger than F diff')
disp('4: Force difference between data and histo')
disp('5: Step function')
disp('6: All')
disp('7: Only plot data'); disp(' ')
disp('8: Update input'), disp(' ')
a=0;
```

29
while (a<1 || a>8 ||
min(class(a)~='double')=1)
a=input('Your choice: ');
end

if a==8
    [dataset, data_char,
    no_datasetpoints, mhz, df, k,
    sf_parameters] = inp;

    a=0;
    while (a<1 || a>7 ||
    min(class(a)~='double')=1)
        a=input('Your choice (1-7): ');
    end
end

switch a
    case {1,2,3,6}
        max_lag=input('Enter maximum
correlation order for 1,2,3,4: ');
    end

switch a
    case 1
        [avg_diff, char_meandiff] = meandiff(dataset,
        data_char, max_lag,
        no_datasetpoints);
        hold on
        %title('Mean value of
difference');
        XLabel('Measurement point or
lag order');
        YLabel('Mean change [pN]');
        plot(1:length(dataset), dataset, 1:length(avg_diff), avg_diff);
        legend(data_char, char_meandiff);
        grid on
    case 2
        m=0;
        while (m<1)
            m=input('Of how many points
should the median be taken off (at
least five): ');
        end
        jumpmax_med=maxdiff(dataset,
        data_char, max_lag,
        no_datasetpoints, m);
        plot_maxdiff
    case 3
        pr=0;
        while (pr<=0 || pr>1)
            pr=input('Set precision
(between 0 and 1) of F diff prop calc: ');
        end
        [jump_min_no, jump_max_no,
        f_diff, fdiffprop_legend] = fdiffprop(dataset,
        data_char, max_lag, no_datasetpoints,
        df, pr, k);
        plot_fdiffprop
    case 4
        order=input('Enter (row)
vector/scalar for correlation orders: ');
        [delta, delta_legend] =
        fdiff(dataset, data_char,
        order);
        plot_fdiff
    case 5
        to=input('to: ');
        [dataset, data_char,
        no_datasetpoints, mhz, df, k,
        sf_parameters] = inp;
        [s1, f, s1_legend, lh, n] = heart
        (dataset, data_char, sf_parameters);
        %plot_heart;
        %figure(1)
        %hold on
        %for i=1:size(s1,3)
        %    plot(s1(:,:,i))
        %end
        %legend(s1_legend,4)
        %grid on
        to=1;
        plot_heart_2inp;
        to=5;
        [dataset, data_char,
        no_datasetpoints, mhz, df, k,
        sf_parameters] = inp;
        [s1, f, s1_legend, lh, n] = heart
        (dataset, data_char, sf_parameters);
        plot_heart_2inp;
    case 6
        m=[5];
        %m=0;
while (m<3)
    m=input('Of how many points should the median be taken off (at least five): ');
end

step=0.05;
step=0;
while (step<=0 || step>1)
    step=input('Set step size (between 0 and 1) of F diff prop calc: ');
end

order=0;
while (min(order)==0 || max(order)>size(dataset,1))
    order=input('Enter vector/scalar for dataset diff. correlation orders: ');
end

figure(1)
disp('Calc mean diff:'); disp(' ');
[avg_diff, char_meandiff] = meandiff(dataset, data_char, max_lag, no_datasetpoints);
%subplot(2,2,1)
plot_meandiff; disp(' ');
hold on

figure(2)
disp('Calc max diff:'); disp(' ');
jumpmax_med = maxdiff(dataset, data_char, max_lag, no_datasetpoints, m);
%subplot(2,2,2)
plot_maxdiff; disp(' ');

figure(3)
disp('F_diff prop:'); disp(' ')
[f_diff, fdiffpropLegend] = fdiffprop(dataset, data_char, max_lag, no_datasetpoints, df, step, k);
%axes('position',[0.05 0.05 0.92 0.45])
%subplot(2,2,3:4)
plot_fdiffprop;
figure(4)
[sl, f, sl_legend] = heart(dataset, data_char, sf_parameters);
plot_heart

figure(5)
[delta, deltaLegend] = fdiff(dataset, data_char, order);
plot_fdiff;
figure(7)
plot(dataset)
if exist('sf_parameters')
a=strcat('Input dataset: ',sf_parameters);
end
else
	title(a);
end
	title('Input dataset')
end
legend(data_char);
grid on

case 7
figure(2)
plot(dataset)
if exist('sf_parameters')
a=strcat('Input: ',sf_parameters);
	title(a);
else
	title('Input: ') legend(data_char);
end
grid on
xlabel('dataset point') ylabel('Amplitude')
grid on
figure(3)
plot(dataset)
if exist('sf_parameters')
a=strcat('Input: ',sf_parameters);
	title(a);
else
	title('Input: ') legend(data_char);
end
grid on
xlabel('dataset point') ylabel('Amplitude')
grid on
v= axis;
v(3:4)=v(3:4)*1.05;
axis(v);
end

clear a i m order step v x_axis f_diff

Input data

%Reads in the input from (Pili)measurements
%input: from files Data0512061, AllData0512132, still1.txt
%output: exp_data(array of measurements), char(name of data),
% df(expected force change), k{distinguishes between different
% rigidities},
% s(# of input files)

function [exp_data, legend, no_datasetpoints, mhz, df, k, sf_parameters] = inp
% defining source data with labeling in alphabetical order (aa, bb, ..)  

sf.parameters = '';  
[sf, sf.char, sf.parameters] = sample;  
% a1 = 'F1(1:1400)'; % drift  
% b1 = 'y28926(1700:2800)';  
% c1 = 'y28926(3202:8105)';  
% d1 = 'y56694(1454:2088)';  
% e1 = 'y79371(1:2500)';  
% f1 = 'y11120(1:2500)';  
% g1 = 'y155568(1300:2600)';  
% h1 = 'y217792(1:2800)';  
% i1 = 'y304914(1000:2800)';  
% j1 = 'y426879(1000:2700)';  
% k1 = 'y597630(1:3000)';  
% l1 = 'still1(1:8000,2)';  
% mhz = 5; % measuring with 5 khz

if exist ('sf.char')  
    legend(1:size(sf.char,1),1:size(sf.char,2))=sf.char;  
    exp.data(1:size(sf,1),1:size(sf,2))=sf;  
    no_data_points(1,1:size(sf,2))=size(sf,1);  
    s=s+1;  
end

for i=1:26 % iter from a till z  
    for j=1:3 % iter from 1 to 3  
        x=strcat(char('a'+i-1),num2str(j));  
        if exist (x) & x(1)~='v' % tests for existence of variable  
            legend(s,1:size(eval(x),2))=eval(x);  
            if (max(legend(s,1:6)~='still1')  
                k=1;  
            end  
            if (min(legend(s,1:6)=='still1') && k==1)  
                k=3;  
            elseif (min(legend(s,1:6)=='still1') && k~=3)  
                k=2;  
            end
        end
    end
end

% load input data  
load Data0512061  
load AllData0512132  
if (exist ('11')==1 || exist ('12')==1)  
    load still1.txt
end

mhz=5; % measuring with 5 khz

if reading in data  
    s=1; k=0; % s coordinate for data files, k for different kappa (rigidity)
if \( k = 2 \)
\[
\kappa(1) = 0.14; \quad \text{\% for every data except still1}
\]
end

if \( k = 2 \)
\[
\kappa(1) = 2.04 \times 10^{-4};
\]
elseif \( k = 3 \)
\[
\kappa(2) = 2.04 \times 10^{-4};
\]
end

\[ x_{AB} = 3.5; \]

\[ df = \kappa \times x_{AB}; \quad \text{\% calc. expected } F_{diff} \text{ due to opening/closing} \]

**Probes function**

\*[generates sample data with certain parameters and defined events]

\*function* \([sf, \text{ sf_legend, sf_parameters}] = \text{ sample}\)

\*\% adjustable parameters\*

\[ T = 200; \quad \text{\% of data points of one period}\]
\[ T_{no} = 1; \quad \text{\% number of periods}\]
\[ \text{noise_ratio} = \text{input}(\text{'noise_ratio : '}); \quad \text{\% noise/(Amp_event): relative value(s) of noise to event strength}\]
\[ \text{noise_ratio(1:3): noise_ratio}; \quad \text{\%1=rectangular, 2=gaussian, c_bw = 0; \% bandwidth constant} \]
\[ A_0 = 0; \quad \text{\% constant amplitude} \]
\[ \text{shift = drift = 0; \% constant drift} \]
\[ d_0 = 0; \quad \text{\% start coordinate of drift} \]
\[ d_1 = T \times T_{no}; \quad \text{\% end coordinate of drift} \]
\[ \text{to = input(}'to: '{);} \quad \text{\% time length of opening event} \]
\[ t_c = 1; \quad \text{\% time length of closing event} \]
\[ o = [100]; \quad \text{\% opening events (ascending order)} \]
\[ c = [1]; \quad \% closing events (ascending order) \]
\[ \text{sh0 = 0; \% time shift of periodicity at the beginning} \]
\[ \text{sh1 = 0; \% time shift of periodicity at the end} \]

\*\% constants (also possible to adjust)\*
\[ o = sh0; \quad c = c + sh0; \quad \text{\% change in } o + sh0; \quad \text{\% change in } c \]
\[ df_o = 1; \quad \text{\% change in Force through opening/closing event} \]
\[ df_c = 0; \quad \text{\% change in Force through opening/closing event} \]
\[ mo = df_o / to; \quad \% slope during event \]
\[ mc = df_c / tc; \quad \% slope during event \]

\[ \text{sf_length} = T_{no} \times T; \quad \% length of sample signal \]
\[ \text{noiseanalysis} = 1; \quad \% \text{input(}'noiseanalysis (1=yes, else=no): ');} \]

\% Adjusting T to length of events if necessary
\[ \text{if max(max(c(:,i)+tc),max(o(:,i)+to))}>T \]
\[ \quad T = \text{max(max(c(:,i)+tc),max(o(:,i)+to))}; \]
\[ \quad \text{sf_length} = T \times T_{no} \times T; \]
end

\% defining a, b, sf
\[ sf(1:T,1:length(noise_ratio)) = 0; \]

\% including events, once not and once overwriting non event data
\[ \text{if noiseanalysis} = 1 \]
if \[ T_{no} = 0 \text{ || length(c) = 0} \]
\[ \quad \text{for } i = \text{size(sf,2)}:1:-1 \]
\[ \quad \text{sf(:,2*i-1)} = \text{sf(:,i)}; \]
end
\[ \]
\[ \quad \] [\[ sf_{ev} = \text{sample_event}(T,c,tc,mc,o,to,mo,noise_ratio); \]
\[ \quad \text{for } i = 1: \text{size(sf}_{ev},2) \]
\[ \quad \text{sf(:,2*i)} = \text{sf}_{ev}(i,:) + \text{sf(:,2*i-1)}; \]
end
end
\[ \]
\% including Bandwidth
\[ \text{if c_bw = 0} \]
\[ \quad \text{[sf] = \text{bandwidth}(sf, c_bw)}; \]
end

\% iter for all periods
\[ \text{for } i = 1: \text{size(sf,2)} \]
\[ \quad \text{for } j = 1:T \]
\[ \quad \text{sf(i*T+j,:) = sf(i*T,:) + sf(j,:)}; \]
end
end

\% including noise
\[ \text{for } n = 1: \text{size(sf,2)} \]
\[ \quad \text{for } i = 1: \text{size(sf,2)} \]
\[ \quad \text{if (noise_distr==1)} \]

```matlab
if (k==2)
    kappa(1)=0.14;  %for every data except still1
end
if (k==2)
    kappa(1)=2.04*10^-4;
elseif (k==3)
    kappa(2)=2.04*10^-4;
end
x_AB=3.5;
df=kappa*x_AB;  %calc. expected F_diff (df) due to opening/closing
```
sf(i,n) = sf(i,n) + (rand(1) - 0.5)*noise_ratio(ceil(n/2))*2;
%rectangular noise
else if (noise_distr==2)
    sf(i,n) = sf(i,n) +
    rand*noise_ratio(ceil(n/2));
%normal distributed noise
end

%including drift
if (i>=d0 && i<=d1)
    sf(i,n) = sf(i,n) +
    drift*(i-d0);
else if (i>d1)
    sf(i,n)=sf(i,n) +
    drift*d1;
end
end

%including constant shift
if A0~=0
    sf=sf+A0;
end

%creating legend
for i=1:length(noise_ratio)
    if noise_ratio(i)==0
        d=strcat('t');
    else if (noise_distr==1)
        d=strcat('(r_',num2str(noise_ratio(i)),')');
    else if (noise_distr==2)
        d=strcat('G_',num2str(noise_ratio(i)),')');
    end
end
if c_bw~=0
    d=strcat(d,'c.bw=');
end
sf_legend(i,1:size(d,2))=d;

%creating string describing parameters
if sh0~=0
    sp=strcat('Startperiod.:',num2str(sh0));
else sp='';
end
if shl~=0
    ep=strcat('end period. T=',num2str(sf_length-shl));
else ep='';
end

pe=strcat(' T=',num2str(T));
if (length(c)~=0)
    cl=strcat(', +event at:',num2str(c),', A+:',num2str(df_c),', t+:',num2str(to));
else cl='';
end
if (length(o)~=0)
    op=strcat(', -event at:',num2str(o),', A-:',num2str(-df_o),', t-:',num2str(to));
else op='';
end
if drift~=0
    dr=strcat('drift:',num2str(drift));
    if d0~=0
        dr=strcat(dr,'starting at:',num2str(d0));
    end
else dr='';
end
sf_parameters=strcat(sp,pe,dr,op,cl);

Test function with event

function [sf] = sample_event(T,c,tc,mc,o,to,mo,noise_ratio)
if (length(c)==0)
    a(1:T,1)=0;
else
    a(1:T,1:length(c))=0;
end
if (length(o)==0)
    b(1:T,1)=0;
else
    b(1:T,1:length(o))=0;
end
for k=1:length(c) %calc. distr. for each closing event within T
    i=c(k);
    for j=1:tc
        a(i+j,k)=a(i+j-1,k)+mc;
    end
    while (i+jc<T)
        a(i+jc+1,k)=a(i+jc,k);
        i=i+1;
    end
end
for k=1:length(o) %calc. distr. for each opening event within T
    i=o(k);
    while (i+jc<T)
        a(i+jc+1,k)=a(i+jc,k);
        i=i+1;
    end
%superposition of closing events
if (length(c)>1)
    for i=2:length(c)
        a(:,1)=a(:,1)+a(:,i);
    end
end

for k=1:length(o)  %calc. distr. for each opening event within T
    i=o(k);
    for j=1:to
        b(i+j,k)=b(i+j-1,k)-mo;
    end
    while (i+to<T)  %including changes of amplitude to all points after event
        b(i+to+1,k)=b(i+to,k);
        i=i+1;
    end
end

if (length(o)>1)  %superposition of opening events
    for i=2:length(o)
        b(:,1)=b(:,1)+b(:,i);
    end
end

%superposition of closing and opening events
for n=1:length(noise_ratio)
    sf(:,n)=a(:,1)+b(:,1);
end

Mean value function (g_{\text{mean}}(l))

%Calculates the mean value between measurement differences with different lags
function [avg_diff, data_char_meandiff]=meandiff(data, data_char, max_lag, no_datapoints)
    for s=1:size(data,2)  %iterating for all source files defined above
        %displaying progress of calculation
        disp('Upcoming data (top-down):')
        disp(data_char(s:size(data,2),:))
        disp('')

        for lag=1:max_lag  %iterating time lag variable
            x=1;  %coordinate of data
            z=0;  %unnormalized data point
            diff
                while (x<=(size(data,1)-lag) && data(x+lag,s)==0)  %iter variable
                    z=z+abs(data(x,s)-data(x+lag,s));  %adding all differences
                    x=x+1;
                end;
            end
            avg_diff(lag,s)= z / (no_datapoints(lag,s)+1);  %calc. the mean difference
        end
    end

data_char_meandiff=strcat('Meandiff',data_char);
return

Max value function (g_{\text{max}}(l))

%Calculates max force change of a certain lag using a median
%output: jumpmax_med
function jumpmax_med=maxdiff(exp_data, data_char, max_lag, no_datapoints, m)
    for k=1:length(m)
        med=-floor((m(k)-1)/2):1:floor(m(k)/2);
        %value and coordinate of F_diff,max
        jumpmax_med(max_lag,size(exp_data,2))=0 ;
        max_coord_med(max_lag,size(exp_data,2)) =0;
        for s=1:size(exp_data,2)  %iterating for all source files defined above
            %displaying progress of calculation
            %disp('Upcoming data (top-down):')
            %
%disp(data_char(s:size(exp_data,2),:))
%disp(' ')
a=(no_datapoints(1,s)-
max(med));

for lag=1:max_lag  %iterating
    %time lag variable
    a=a-1;
    x=-min(med)+1;  %coordinate of data points in exp_data
    while
(x<=(size(exp_data,1)-lag-max(med)))
      & & (exist('exp_data(x+lag,s')~=0) & &
exp_data(x+lag,s)~=0
    c=median(abs(exp_data(x+med,s)-
exp_data(x+lag+med,s)));
    if (x<a & & c >
jumpmax_med(lag,s))
    jumpmax_med(lag,s)=c;
    max_coord_med(lag,s)=x;
    x=x+1;
    end
  end
jumpmax_med=d;

%function [s1, f, sl_legend] = heart
(data,data_char, sf_parameters)
%parameters: many :)
%output: probe function s1 and value of
convolution f(x,b,n) = sl*data
function [s1, f, sl_legend, lh,n] = heart2
(dataset, data_char, sf_parameters)

%generate heart

d=4;  %iterate heart until i/a<d
lh=5:2:25;  %length of one decreasing exponent (t.ex. from heart(x_max) to
hart(x)=0
n=1:25;  %distance between the two exp.

b(1:length(lh))=1;  %shape of
connections between x0(k)-t/2 and
x0(k)+t/2; 1: linear, 2: exponentially,
else: zero
a=[lh(:)/d];  %constant in exp.
decreasing factor (max.length=2)
a=a';
s1=0;
for y=1:size(n,2)
  %n(1:length(lh))=n(1:length(lh),y)
  %changing n from matrix to vector
  for k=1:length(lh)
    sl(1:length(sl))=0;  %resetting
    sl to zero
    x_max=ceil(d*a(k)+1+n(y));
    %coordin(y)ate for which sl is maximum
    sl(x_max)=1;
    sl(x_max-n(y))=-sl(x_max);
    m=2*sl(x_max)/n(y);  %slope
    between x0(k)-1,x0(k)
    x0(k)=ceil(x_max-n(y)/2);  %x0(k)(coordinate where sl >=0)
    if b(k)==1
      %linear connection of the
      two exponents
      for i=1:n(y)-1
        sl(x_max-i)=1-m*i;
      end
      %exponential connection
    elseif b(k)==2
      for i=1:floor(n(y)/2)
        sl(x_max-
        i)=(exp(n(y)/2-i)-1)/((exp(n(y)/2)-
        1)*sl(x_max));
      end
      for i=x_max-
      n(y):floor(x_max-n(y)/2)
        sl(i)=1/(exp(n(y)/2)-
        1)*(exp(abs(i-x_max+n(y)/2))-
        1);
      end
      for i=ceil(x_max-
      n(y)/2):x_max
        sl(i)=1/(exp(n(y)/2)-
        1)*(exp(abs(i-x_max+n(y)/2))-1);
      end
      for i=ceil(n(y)/2):n(y)-1
        sl(x_max-i)=(exp(-
        (n(y)/2-i))-1)/(exp(n(y)/2)+1)*sl(x_max)
    end
end
else
    % zero between two exponents
    for i=1:n(y)-1
        sl(x_max-i)=0;
    end
end

x=x_max;
i=0;
if k==1 && y==1
    sl=transpose(sl); %change line vector to row vector
end
while ((i/a(k))<=d || i==1); % i = coordinate of sl
    sl(x+i)= sl(x_max)*exp(-i/a(k));
    sl(x-i-n(y))= sl(x_max-n(y))*exp(-i/a(k));
    i=i+1;
end

% Normalization of sl
N=0;
for i=ceil(x_max-n(y)/2):length(sl) % iter over positive part of sl
    N=N+sl(i);
end
sl=sl/N;

% calc. convolution f
for s=1:size(dataset,2) % iter for diff. inputs
    f(length(dataset),k,y,s)=0;
    for sh=(lh(k)+ceil(n(y)/2)):length(dataset)-ceil(n(y)/2)-lh(k) % shifting parameter for f
        for i=1:length(dataset(:,s))
            if (i-sh+x0(k)-1)>1 && i-sh+x0(k)-1<=size(sl,l1)
                % sl: -sh moving to the right, +x0(k) shifting to pos.values, -1 shifting sl to the right= f to the
                % left
                f(sh,k,y,s)=f(sh,k,y,s)+sl(i-sh+x0(k)-1)*dataset(i,s);
            end
        end
    end
end

% shifting mean value of f upon datasets mean value
% sh=mean(dataset(:,s))-
% mean(f(:,k,y,s));
% f(:,k,y,s)=f(:,k,y,s)+sh;
end

c(1:length(sl),k,y)=sl; % save values of sl in a vector
end
sl=c;

j=0;
for d=1:length(n)
    for i=1:length(lh)
        if b(i)==1
            c=strcat('g(',num2str(lh(i)),',',num2str(n(d)),',lin)');
        else
            if b(i)==2
                c=strcat('g(',num2str(lh(i)),',',num2str(n(d)),',exp)');
            else
                c=strcat('g(',num2str(lh(i)),',',num2str(n(d)),',step)');
            end
            j=j+1;
            c=strcat(c,'*t');
            sl_legend(j,1:length(c))=c;
        end
    end
end
clear c d j i

Plot of probe function and convolution

clear x y Z 21
lh=lh';
[LH,N]=meshgrid(lh,n);
fig=0;
detectable(1:round(size(f,4)/2),1:round(size(f,4)/2))=0;
if to==1
    f(sh,k,y,s)=f(sh,k,y,s)+sl(i-sh+x0(k)-1)*dataset(i,s);
end
end
end

% shifting mean value of f upon datasets mean value
% sh=mean(dataset(:,s))-
% mean(f(:,k,y,s));
% f(:,k,y,s)=f(:,k,y,s)+sh;
end

c(1:length(sl),k,y)=sl; % save values of sl in a vector
end
sl=c;

j=0;
for d=1:length(n)
    for i=1:length(lh)
        if b(i)==1
            c=strcat('g(',num2str(lh(i)),',',num2str(n(d)),',lin)');
        else
            if b(i)==2
                c=strcat('g(',num2str(lh(i)),',',num2str(n(d)),',exp)');
            else
                c=strcat('g(',num2str(lh(i)),',',num2str(n(d)),',step)');
            end
            j=j+1;
            c=strcat(c,'*t');
            sl_legend(j,1:length(c))=c;
        end
    end
end
clear c d j i
x_fth=100+floor(to/2); %theoretical event detection coordinate (maximum of convolution)
else
x_fth=102;
end

for i=1:size(f,4)/2 %i data without event
    for j=1:size(f,4)/2 %j data with event
        fig=fig+1;
        figure(fig)
        Z(1:size(f,3),1:size(f,2),i)=permute((max(abs(f(:,:,2*i-1))),(3 2 4 1));
        Z1(1:size(f,3),1:size(f,2),j)=permute((max(abs(f(:,:,2*j))),(3 2 4 1));
        %surf(LH,N,Z(:,:,i));
        hold on
        %surf(LH,N,Z1(:,:,j));
        xlabel('b')
        ylabel('n')
        zlabel('Absolute maximum of convolution f')
        %finding coordinates of maximum convolution for all parameter
        [x(i,j),y(i,j)]=find(abs(Z(:,:,i)-Z1(:,:,j))==max(max(abs(Z(:,:,i)-Z1(:,:,j)))));
        b_best(i,j)=LH(x(i,j),y(i,j));
        n_best(i,j)=N(x(i,j),y(i,j));

        [A_n(i),x_n(i)]=max(abs(f(:,:,y(i,j),x(i,j),2*i-1)));
        [A_nev(j),x_nev(j,i)]=max(abs(f(:,:,y(i,j),x(i,j),2*j)));
        %calc. max ampl. ratios of pure noise and event noise
        A_ratio(i,j)=A_nev(j)/A_n(i);
        A_real_ratio(i,j)=abs(min(f(x_nev(j),y(i,j),x(i,j),2*j))/(max(abs(f(:,:,y(i,j),x(i,j),2*j)))/(x_fth));
        if x_dist(j,i)<=5 & A_real_ratio(i,j)>1.5
            detectable(j,i)=1;
        end

end
end

%display only one decimal digit
p=round(10*A_ratio);
A_ratio=p/10;

format compact
b_best, n_best, A_ratio, x_dist, detectable

fig=fig+1;
figure(fig)
if l=1:length(f(:,1));
l0=1:length(dataset(:,1));
k=1;
for i=1:3
    k=k+1;
    for j=1:3
        hold on
        plot(lf,f(:,y(i,j),x(i,j),k),lf,f(:,y(i,j),x(i,j),k));
    end
end
%plot(lf,f(:,y1(1),x1(1),1),lf,f(:,y1(2),x1(2),2),lf,f(:,y1(3),x1(3),3),lf,f(:,y1(4),x1(4),4),...
    %lf,f(:,y1(5),x1(5),5),lf,f(:,y1(6),x1(6),6))
    legend('f(b_o_p_t,n_o_p_t) only noise','f(b_o_p_t,n_o_p_t) with event')
    grid on
    xlabel('dataset point');
    ylabel('Amplitude');

figure((size(dataset,2)/2)^2+2)
plot(dataset)
if exist('sf_parameters')
a=strcat('Input: ',sf_parameters);
else
    legend(dataset_char);
end
grid on
xlabel('data point');
ylabel('Amplitude');
legend(data_char);
v= axis;
v(3:4)=v(3:4)*1.05;
axis(v);

for i=1:size(f,4)/2
    for j=1:size(f,4)/2
        fig=fig+1;
        figure(fig);
surf(LH,N,2(1,:,:,:),j) -  
Z(1,:,:,:,:,:),/Z(1,:,:,:,:,:));  
ev=i;  
xev=j;  
xlabel(strcat('b',',';  
event',num2str(i)))  
ylabel(strcat('n',','; no event  
',num2str(j)))  
end  
end  

%plotting probe functionS  
figure(i+2)  
hold on  
for i=1:size(s1,3)  
   plot(s1(:,:,i))  
end  
legend(s1_legend,4)  
grid on  

%save results in a mat'  
i=find(s1_legend(1,9:11)=='lll');  
for i=1:size(s1_legend,2)-2,  
   if s1_legend(1,i:i+2)=='lin'  
      connect=s1_legend(1,i:i+2);  
   end  
end  
if connect == 'lin'  
   connection='lin';  
else  
   connection='exp';  
end  

data_char2=data_char;  
if data_char2(1,:size(data_char,2)-2)=='.',  
data_char(1,:size(data_char,2)-1)=';',  
data_char(1,:size(data_char,2)-2)='
';  
clear i j  
i(1,1:size(data_char,2),end)=char('G');  
i=(find(data_char(1,:)==-1));  
if isempty(i)  
in='r';  
else  
in='G';  
end  

%save results in .mat of working  
directory  
dataname=strcat('save',data_char2(1,2:size(data_char,2)-1),  
'_to_',num2str(to),'_',connection);  
a=input('Safe mat? (1=yes): ');  
if a==1  
eval('dataname');  
end  

clear x_n x_nev x_dist1 x_dist5 b_best  
n_best A_ratio A_real_ratio1  
A_real_ratioS detectable data_char2  
clear a b i m o t_ev j  

Test function  

%generates sample data with certain  
parameters and defined events  
function [sf, sf_legend, sf_parameters]  
= sample  

%adjustable parameters  
T=40;  
T_no=2;  
noise_ratio=[0.1];  
noise_distr=[1];  
c_bw=0;  
A0=0;  
A=1;  
%bandwidth constant  
drift=0;  
%constant drift  
d0=0;  
%start coordinate of  
drift  
d1=T*T_no;  
%end coordinate of drift  
to=1;  
%time length of  
opening event  
tc=1;  
%time length of  
closing event  
o=[10 25];  
%opening events  
c=[5 30];  
%closing events  
sh0=0;  
%time shift of  
periodicity at the beginning >= 0  
sh1=0;  
%time shift of  
periodicity at the end >=0  

%constants (also possible to adjust)  
o=o+sh0;  
c=c+sh0;  

%change in Force through  
opening/closing event  

%change in Force  
through opening/closing event  
mc=df_c/tc;  
%slope during event  
sf_length=T*T_no;  
%length of sample  
signal  

noiseanalysis=input('noiseanalysis  
(1=yes,else=no): ');  

%Adjusting T to length of events if  
necessary  
if max(max(c(:,)+tc),max(o(:,)+to))>T  
T=max(max(c(:,)+tc),max(o(:,)+to));
sf_length=T_no*T;
end

%defining a,b,sf
sf(1:T,1:length(noise_ratio))=0;

%including events, once not and once
overwriting non event data
if noiseanalysis==1
    if length(o)~=0 || length(c)~=0
        for i=size(sf,2):-1:1
            sf(:,2*i-1)=sf(:,i);
        end
        [sf_ev]=sample_event(T,c,tc,mc,o,to,mo,
noise_ratio);
        for i=1:size(sf_ev,2)
            sf(:,2*i)=sf_ev(:,i)+sf(:,2*i-1);
        end
    end
else
    %including events
    [sf_ev]=sample_event(T,c,tc,mc,o,to,mo,
noise_ratio);
    sf(:,:,)=sf_ev(:,:,);
end

%including Bandwidth
if c_bw~=0
    [sf]=bandwidth(sf,c_bw);
end

%iter for all periods
for i=1:sf_length/T-1
    for j=1:T
        sf(i*T+j,:) = sf(i*T,:) + sf(j,:);
    end
end

%including noise
for n=1:size(sf,2)
    for i=1:sf_length
        if (noise_distr==1)
            sf(i,n) = sf(i,n) + (rand(1)-0.5)*noise_ratio(ceil(n/2))*2;
        %rectangular noise
        elseif (noise_distr==2)
            sf(i,n) = sf(i,n) +
        randn*noise_ratio(ceil(n/2));
        %normal distributed noise
        end
    end
    %including drift
    if (i>=d0 && i<=d1)
        sf(i,n) = sf(i,n) +
drift*(i-d0);
    elseif (i>d1)
        sf(i,n)=sf(i,n) +
drift*d1;
        end
    end
end

%including constant shift
if A0~=0
    sf=sf+A0;
end

%creating legend
for i=1:length(noise_ratio)
    if noise_ratio(i)==0
        d=strcat('t');
    else
        if (noise_distr==1)
            d=strcat('r_');
        elseif (noise_distr==2)
            d=strcat('G_');
        end
    end
end

if c_bw~=0
    d=strcat(d,'c.bw=');
    c_bw=',num2str(c_bw));
end

sf_legend(i,1:size(d,2))=d;
end

%creating string describing parameters
if sh0~=0
    sp=strcat('Startperiod.:',num2str(sh0));
    else sp='';
end

if sh1~=0
    ep=strcat('i;end period. T=',
    num2str(sf_length-sh1));
    else ep='';
end

pe=strcat(' T=',num2str(T));
if (length(c)~=0)
    cl=strcat(' +event at:',
    num2str(c),', A+:',num2str(df_c),',
t+:',num2str(to));
    else cl='';
end
if (length(o) ~= 0)
op = strcat(' - event at: ',
    num2str(o), ', A-', num2str(-df_o), ', t-
    ', num2str(to));
else
    op = '';
end

if drift ~= 0
dr = strcat('',
    drift, num2str(drift));
if d0 ~= 0
    dr = strcat(dr, ', starting at: ',
    num2str(d0));
else
    dr = '';
end

Histogram of f(k_0)

% counting extremes of abs(derivative)
% in other words: counts positive maxima
% and negative minima
clear g
n_coor = 1:2;
b_coor = 1;
in_coor = 1;
leg = [leg1; leg2];

for k = 1:2
    j = 1;
    clear g
    A_posextreme(k) = 0;
    A_negextreme(k) = 0;
    for i = 2:length(f)-1
        if abs(f(i, b_coor, n_coor(k), in_coor)) >
            abs(f(i-1, b_coor, n_coor(k), in_coor))
        &
            abs(f(i, b_coor, n_coor(k), in_coor)) >
            abs(f(i+1, b_coor, n_coor(k), in_coor))
        g(j) = f(i, b_coor, n_coor(k), in_coor);
        if g(j) > 0.5
            A_posextreme(k) = A_posextreme(k) + 1;
        else if g(j) < -0.5
            A_negextreme(k) = A_negextreme(k) + 1;
        end
    end
    j = j + 1;
end

subplot(1, 2, k)
hist(g(:))
xlabel('Amplitude of event')
ylabel('Number')
grid on

A_posextreme
A_negextreme
N_datapoints = length(f(:, 1, 1, 1)) - 2*1h-n
prop_pos = A_posextreme ./ N_datapoints
prop_neg = A_negextreme ./ N_datapoints
prop_mean = (A_posextreme + A_negextreme) ./ (2*N_datapoints)
t = 1 ./ (prop_mean * 5000)
freq = 1 ./ t

xi label('Amplitude of event')
ylabel('Number')
grid on

A_posextreme
A_negextreme
N_datapoints = length(f(:, 1, 1, 1)) - 2*1h-n
prop_pos = A_posextreme ./ N_datapoints
prop_neg = A_negextreme ./ N_datapoints
prop_mean = (A_posextreme + A_negextreme) ./ (2*N_datapoints)
t = 1 ./ (prop_mean * 5000)
freq = 1 ./ t