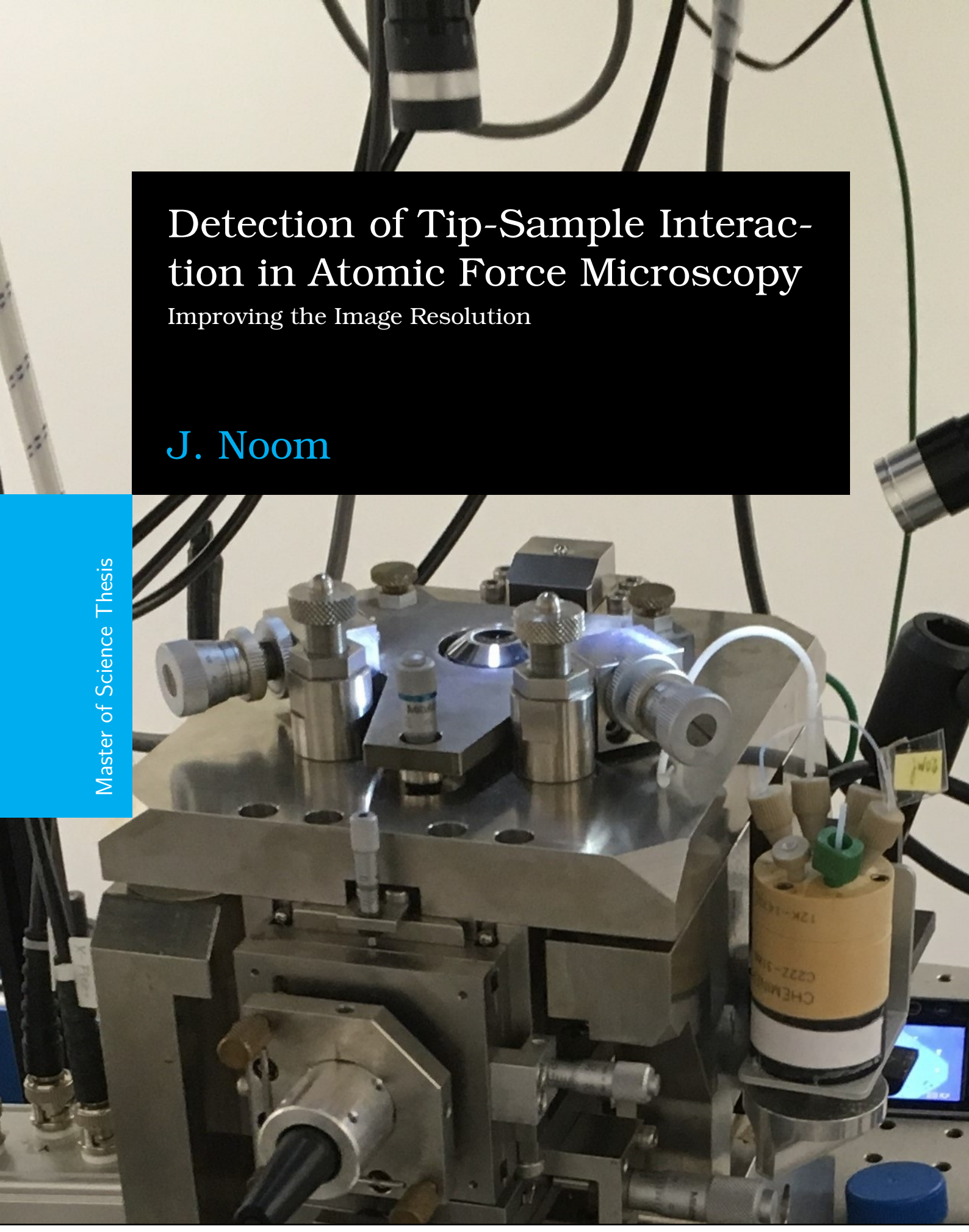


Detection of Tip-Sample Interaction in Atomic Force Microscopy

Improving the Image Resolution

J. Noom

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Detection of Tip-Sample Interaction in Atomic Force Microscopy

Improving the Image Resolution

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

J. Noom

June 19, 2019

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Copyright © Delft Center for Systems and Control (DCSC)
All rights reserved.

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Detection of Tip-Sample Interaction in Atomic Force Microscopy

by

J. Noom

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: June 19, 2019

Supervisor(s):

prof.dr.ir. M. Verhaegen

Reader(s):

dr.ir. C.S. Smith

dr.ir. G. Giordano

dr.ir. F. Alijani

dr.ir. A.J. Katan

Abstract

Currently used imaging methods in Atomic Force Microscopy (AFM) including the use of a Lock-In Amplifier or a Phase-Locked Loop, are suboptimal. In this report, the image resolution in AFM is improved by detecting the tip-sample interaction using complete measurements of the input of the cantilever and its measured deflection.

Two methods are studied while assuming that the tip-sample interaction is sparse, namely a model-based approach and a data-driven approach. Real-life experiments have shown that the model-based approach improves the image resolution with a factor of 7.5 to 0.555 nm compared to the conventional imaging method, according to a metric using Fourier Ring Correlation in which a reference image is unnecessary. The data-driven approach can be used in the model-based approach to further improve the resolution. In addition to improved resolutions, a Linear Time-Invariant model of the mechanically driven AFM-cantilever immersed in liquid – from piezo input to cantilever deflection – has been obtained through subspace identification with a Variance Accounted For of 79.2%.

Recommendations for future research include applying the latter model in detecting the tip-sample interaction, improving the data-driven approach, reducing the computational effort of the model-based approach and implementing algorithms for detecting the tip-sample interaction online.

Table of Contents

List of Figures

List of Tables

Preface & Acknowledgements

This work results from cooperation of the research groups "Control for Scientific Imaging" (CSI) and the "Cees Dekker Lab", both part of Delft University of Technology, though the former at the department "Delft Center for Systems and Control" and the latter at "Bionanoscience". Due to the cooperation, nanosized biological samples can be studied using the latest innovations in imaging systems, so that the Cees Dekker Lab is one step ahead of the competition. Also for CSI the cooperation is beneficial, as the innovated imaging systems are directly faced with all practical issues.

As is my research in Atomic Force Microscopy (AFM). For instance, I came to know the "forest of peaks", which initiated to change plans for obtaining a pragmatic model. For research in image improvement in AFM, this seems a minor problem. Nevertheless, obtaining a pragmatic model is crucial for eventual operation of the innovation in AFM.

The ultimate goal of this graduation project was to find the limiting factor in AFM-imaging and to take action to tackle this limiting factor. Fortunately, there are abundant opportunities – among other things resulting from this study – to improve the image resolution.

I would like to thank Tijmen van Oldenrijk as initiator of studying methods for image enhancement in AFM. Besides, thanks to dr. Allard Katan for cooperation in the project, which consisted of providing access to the AFM-lab, helping during operation of the AFM and thinking along. Thanks to Omkar Patil for his help during the initial phase of the project and last but not at least, I would like to show my gratitude to prof.dr.ir. Michel Verhaegen for his helping hand during the regular meetings.

Delft, University of Technology — J. Noom
June 19, 2019

Chapter 1

Introduction

Understanding life has always been a major curiosity of human beings. Both religion and science are concerned with this issue, although it may be questioned whether life will ever be completely understood at all. Nevertheless, curiosity is not the only reason to investigate life. Namely, understanding life more and more may strengthen humanity, for instance in enhancement in disease control or in food production. Therefore, research should continue investigating life.

A current issue in investigating life is the desire to observe the smallest possible biological structures, such as condensin with its tiny coiled coils [1] which facilitate the folding of DeoxyriboNucleic Acid (DNA). Furthermore, static observations are inadequate. Dynamic observations are more favourable, since the actual operation of those structres should be discovered.

Facilitating dynamic observations of small biological structures entails various technical challenges. Firstly, those biological structures are extremely small, namely in the order of tens of nanometers. So a microscope should be used for this. Secondly, the observations should be done in ambient conditions, as the usual behaviour of the biological structures is preferred to observe. This means that observations in vacuum, but also in highly irradiated environments are futile. Thirdly, the sample to be investigated should be affected as little as possible during observation. In this way, the original behaviour of the biological structure is preserved. Finally, successive static observations should be made as quickly as possible to make the entire observation dynamic.

Atomic Force Microscopy (AFM), as a type of Scanning Probe Microscopy, turns out to have high potential in scanning living biological samples, due to the ability to scan with very high resolution in ambient conditions in air [2] and liquids [3]. However, AFMs are nowadays often used for making single images at a slow rate, while making videos using AFMs is relatively unexplored. Yet, Eeftens et al. [1] succeeded in filming condensin in liquid using the AFM and found that condensin is flexible and dynamic. Nevertheless, certain features of condensin were still undiscovered due to the slow frame rate and unsatisfactory resolution.

Moreover, the obtained video with size 70×75 nm has a frame rate of 10 frames per second, while the dynamics of the coiled coils (or: the arms) of condensin are expected to have much

faster dynamics. Namely, condensin can extrude DNA with a speed up to 1500 base pairs per second (ca. 500 nm/s) [4] and the length of the arms of condensin is approximately 50 nm (implying a step size of $\leq$ 100 nm [5]), so condensin operates roughly with 5 cycles per second. Thus, the dynamics of the arms of condensin cannot be captured appropriately with a frame rate of 10 frames per second. A more suitable frame rate would be between 20 and 50 frames per second.

This gives rise to the idea to investigate methods to improve AFM-imaging in general. More specifically (though still satisfying the general goal), this report is concerned with improving the image resolution, while keeping the frame rate and influence on the sample constant. While a literature study [6] provided research directions in detecting the interaction between the AFM-cantilever and the sample to be studied, this report focuses on the experimental results of the proposed directions. Furthermore, practical experience led to investigating another new method, which is simulated and also experimentally tested. The results give a first indication of the improvement which can be made in AFM-imaging in general.

The report is structured as follows. Firstly, an introduction to AFM with its physical design and operating modes is given in Chapter 2. Besides, the limitations of current designs are elaborated. Based on the limitations, Chapter 3 continues by describing two pragmatic methods to detect interaction between the AFM-cantilever and the biological sample. While the first (model-based) method was recommended in [6], the second (data-driven) was designed later to operate under less assumptions than the model-based method. Chapter 4 starts with presenting significant contributions concerning system identification of the AFM-cantilever in liquid. Subsequently, the chapter continuous by presenting experimental results of both model-based and data-driven methods. Afterwards, the conclusions are presented in Chapter 5 with recommendations for future research.

Chapter 2

Atomic Force Microscopy

Binnig and Quate invented the Atomic Force Microscope (AFM) in 1986 as a type of Scanning Probe Microscopy [2]. This invention turned out to be very useful, due to its exceptional potential resulotion achievable in versatile conditions [7]. The AFM is not only applicable in vacuum [8], it has also proven itself in ambient conditions in air [2] and in liquids [3]. The Scanning Electron Microscope (SEM) and the Transmission Electron Microscope (TEM) are reasonable competitors of the AFM, considering the achievable resolution. However, the SEM and TEM usually operate in vacuum and require a conductive (or conductive coated) sample [9], which makes those microscopes unfavourable for biologists. Using the AFM, biologists are able to study biomolecules and living cells at (sub-)nanometer resolution [7]. Furthermore, besides scanning and imaging samples at high resolution, the AFM has the potential to manipulate samples at nanometer scale [10].

2-1 Physical Design

Although there are several specific designs, the working principle remains similar for each AFM. Namely, the surface of a sample is scanned by a sharp tip, as is illustrated in Figure 2-1. The tip is attached to a cantilever, from which the displacement is measured. In early designs, the cantilever displacement (or deflection) was being measured using a tip of a Scanning Tunneling Microscope [2]. However, the designs nowadays are based on a laser aimed at the cantilever and photo detector which catches the reflected laser beam [11]. The cantilever deflection is determined from the place of the reflected beam at the photo detector. Based on the measured deflection, the height of the sample can be estimated at that specific point. Since the goal is to obtain a complete height profile of the sample rather than the sample height at one specific point, the scan table has to be actuated in the x- and y-direction. Additionally, a z-actuator is attached to the scan table so that interaction forces between tip and sample can be controlled and samples with coarser relief can be scanned. Note that the cantilever can also be driven by its cantilever-actuator (c-actuator). This is useful in certain operating modes to bring the cantilever in oscillation.

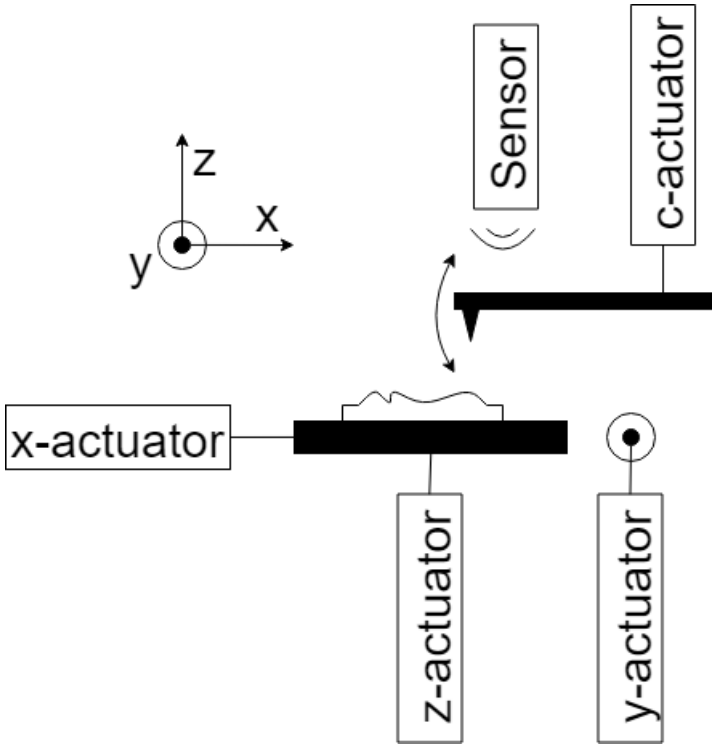


Figure 2-1: Sketch of physical design of an AFM. A cantilever is being driven by the cantilever actuator (c-actuator) above a scanner table with sample. Three actuators are attached to the scanner table to move it in the x-, y- and z-direction. A sensor measures the cantilever's displacement.

Typical scan ranges in x- and y-directions do not exceed 100 microns, although there are exceptions [9]. Typical dimensions of cantilevers are $450 \times 50 \times 2$ microns [9]. Also these dimensions can vary. For example, cantilevers with dimensions $25 \times 10 \times 0.1$ microns have been manufactured to image with high speed [12] and the Cees Dekker Lab [13] uses even smaller cantilevers with dimensions $7 \times 2 \times 0.08$ microns (see Figure 2-2 [14]). Butt, Cappella and Kappl [15] reported the design considerations of AFM cantilevers accurately. The tip of the cantilever should be as small as possible to achieve the highest resolution. However, it is difficult to produce probes with tip radii below 10 nanometers at reasonable costs [9].

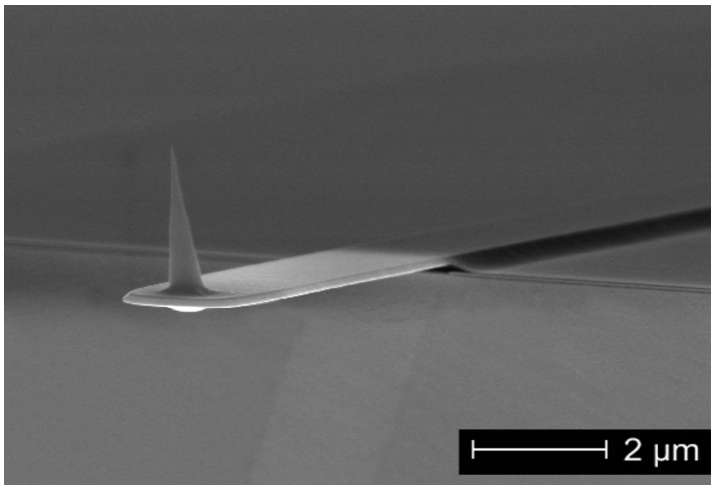


Figure 2-2: Cantilever with dimensions $7 \times 2 \times 0.08$ microns attached at a support chip (upside down).

The cantilever actuator is often a (relatively large) piezo at the location of inclination of the cantilever. As a consequence, most of the force from the piezo is acting on the support chip of circa $3.4 \times 1.6 \times 0.3$ mm [14]. For scanning in liquid, this results in complex behaviour of the cantilever, as the support chip is also immersed [16]. Furthermore, the behaviour can change during scanning due to changing liquid levels or changing viscosity [16]. An alternative to the conventional actuation is the "self-actuating cantilever", in which micro-actuators are being integrated directly on the cantilever. However, although several actuator mechanisms exist [17][18], there are some disadvantages. Firstly, combining the actuator with sensor mechanisms can be problematic for some designs (Lorentz/piezo-electric/magnetic actuation) due to interaction between those mechanisms, especially in water [17]. Secondly, for a mechano-thermal design it may be challenging to realise such a cantilever with small dimensions (in [17] the width was >100 microns). Thirdly, a photo-thermal design [18] is in dimensions often relatively wide [14] (resulting in a higher force constant and therefore higher forces on the sample) compared to a conventional cantilever and the sample might be influenced by the heat, while it is desired to affect the sample as little as possible. Therefore, for the scans of very small biological samples, the conventional piezo is still used as cantilever actuator.

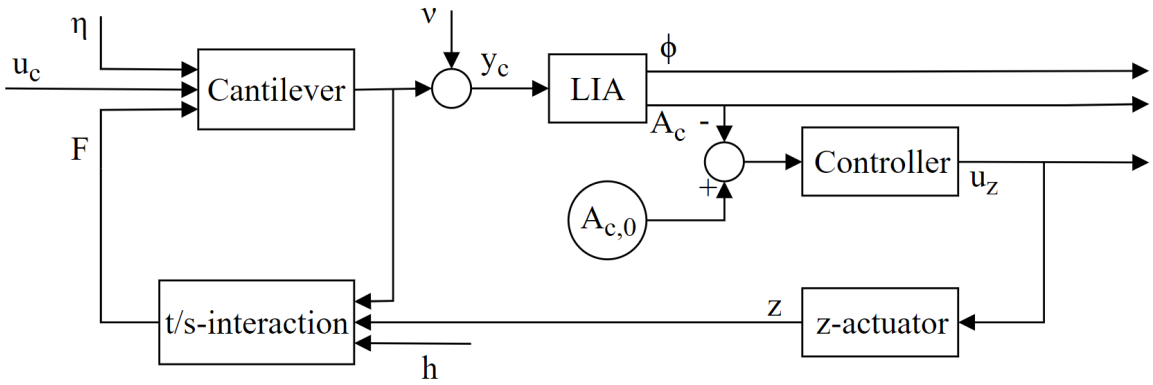


Figure 2-3: Schematic view of mode Amplitude Modulation with Phase Imaging (AM-PI). The cantilever is excited by a sinusoidal signal u_c. The measured cantilever deflection y_c is converted to phase and amplitude, which are used for imaging and control of the z-actuator. An image is generated by plotting one of the three output signals.

2-2 Operating Modes

There exist two generally known modes in which an AFM can be operated, namely "contact mode" and "tapping mode". In the contact mode, the cantilever tip is constantly being held at or near the sample surface. One year after the invention of the AFM, Martin et al. [19] developed a tapping mode, in which the cantilever is brought into oscillation using the cantilever actuator (Figure 2-1). The cantilevers amplitude, frequency and phase will vary due to interaction with the sample. Therefore, in tapping mode the sample height is estimated using one or more of those quantities, generally referred to as AM-PI, PM-AI, FM-AI or AM-FI [20].

Often, the AM-PI-mode is used, from which the schematic is shown in Figure 2-3. The variable u_c stands for the cantilever input (usually a sinusoidal signal); η represents thermal noise; F the tip-sample interaction; ν the measurement noise; y_c the output of the photo detector; A_c, $A_{c,0}$, the amplitude with reference; ϕ the phase; u_z the input to the z-actuator; z the height of the scan table; and h the sample height. The abbreviation LIA stands for Lock-In Amplifier and the block "t/s-interaction" contains the relationship between cantilever deflection, table height z and sample height h on one side and the tip-sample interaction F on the other. Conventionally, only the variables u_c, ϕ, A_c, $A_{c,0}$ and u_z are known.

Figure 2-4 shows the input (u_z) and amplitude plots with feedback (FB) turned on and off (i.e. the input to the z-actuator is either quickly or slowly varying). If FB is turned on, the sample height is being estimated proportionally to the input to the z-actuator:

$$\hat{h} = c_1 u_z. \tag{2-1}$$

Moreover, both amplitude plot and phase plot (the latter not shown) are relatively flat in this case. If, however, FB is turned off[1], then the sample height can be estimated using either

[1]The FB was not turned off completely, but its parameters were set such that the input to the z-actuator was varying very gradually. As a consequence, the cantilever tip was not drifting away from the sample. Furthermore, note that the FB can only be turned off if the variations in sample height are small enough.

the amplitude or phase plot (the latter not shown again). Practical experience suggests that the amplitude plot often contains the most valuable information (unless large differences in hardness of the sample occur), so in this case:

$$\hat{h} = c_2 A_c \tag{2-2}$$

with c_1 and c_2 constants.

For any mode, interaction forces between tip and sample are already present close to the sample. Therefore, it is sometimes unnecessary to hit the sample surface to estimate its height. Nevertheless, the signal-to-noise ratio will be lower, since non-contact forces are smaller than forces caused by actual contact between tip and sample.

For the modes FM-AI and AM-FI, the LIA is replaced with a Phase-Locked Loop (PLL). An extensive explanation including schematics and comparison of operating modes can be found on pages 5 to 8 of [6].

2-3 Limitations

There are multiple factors which can limit the performance of an AFM. Examples are oscillations in the scan table [11], inappropriate resonance characteristics of the cantilever [8][21] and parachuting [22]. For each of those three limitations, improvements have been researched (see pages 8 to 11 of [6]).

However, a literature study [6] and Sahoo, Agarwal and Salapaka [23] have shown that a more fundamental change in signal processing can potentially lead to a larger improvement. Namely, the use of either an LIA or a PLL is accompanied by losing crucial information about the tip-sample interaction F. Moreover, the use of an LIA or a PLL implies that the cantilever output y_c is assumed to be a single-mode oscillation, i.e. the signal consists of just one frequency. In reality, however, this assumption does not hold, since the cantilever is excited by both the sinusoidal cantilever u_c and the tip-sample interaction F. The latter element causes the output signal y_c to be a multi-mode oscillation. Therefore, it is inappropriate to scale the output signal y_c down to just an amplitude and a phase. Instead, the complete envelope of y_c should be measured to reconstruct the tip-sample interaction. As the sample height is simply the cause of the tip-sample interaction, it can be estimated once F is known.

To conclude, all obtainable information about the sample is introduced in the system via F. Hence, the remainder of this report is concerned with estimating the sample height h using reconstruction of tip-sample interaction F from directly measured input and output signals u_c and y_c. In contrast to Sahoo et al. [23], algorithms alternative to the Willsky-Jones Generalised Likelihood Ratio [24] will be investigated (see pages 11, 12 and 15 of [6]). Moreover, simulation experiments have shown better performance for algorithms alternative to Willsky-Jones, while still being convex [25]. Besides, procedures will be studied for system identification of the cantilever immersed *in liquid*. Finally, the obtained images will be assessed both qualitatively and in terms of image resolution.

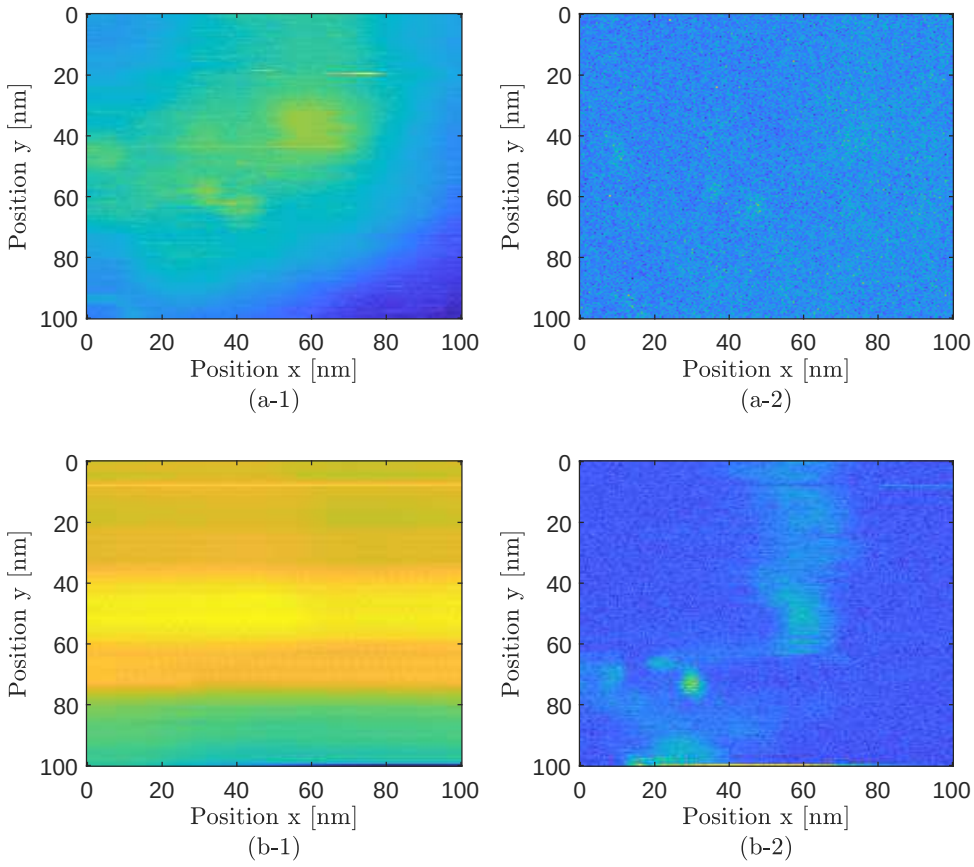


Figure 2-4: Conventional methods to determine sample height, applied at a scan of graphite. The input to the z-actuator (a-1) and the amplitude plot (a-2) if feedback (FB) to the scan table is turned on, and the input to the z-actuator (b-1) and the amplitude plot (b-2) if FB is turned off. Depending on whether the FB is turned on/off, either (a-1) or (b-2) is conventionally used as image, respectively. The sample height was varying between approximately 0 and 1 nm, while the free amplitude of the cantilever was approximately $A_0 = 1.4$ nm.

Chapter 3

Detection of Tip-Sample Interaction

Imaging in AFM is inherent to estimating the height of the sample. The sample height causes interaction with the AFM-cantilever, from which the motion is recorded. Based on the motion, the sample height is estimated. Conventionally, an LIA or a PLL is used for this. However, improvement can be achieved here by recording the complete envelope of the input and the output to reconstruct the tip-sample interaction, from which the sample height can be estimated. Thus, it is needed to study methods for reconstructing the tip-sample interaction using input and output information. This chapter provides explanation of two methods, namely a model-based and a data-driven approach. While the model-based approach relies on Ohlsson et al. [25], the data-driven is based on Yu et al. [26], but is modified so that it fits in the AFM-framework. After a problem formulation in Section 3-1, the model-based approach is presented in Section 3-2 and the data-driven approach in Section 3-3.

3-1 Problem Formulation

Suppose that the cantilever dynamics can be described by the linear time-invariant (LTI) system

$$\begin{aligned} x(t+1) &= Ax(t) + Bu_c(t) + B_2F(t) + \eta(t) \\ y_c(t) &= Cx(t) + \nu(t). \end{aligned} \tag{3-1}$$

with A, B, B_2, C the (known) system matrices and $x(t)$ the state at time step $t \in \mathscr{T} = \{1, 2, ..., N\}$ with N the number of measurements. With $u_c(t)$ and $y_c(t)$ known for $t \in \mathscr{T}$, the task is to recover $F(t)$ for $t \in \mathscr{T} \setminus \{N\}$.

In tapping mode AFM, the cantilever has intermittent contact with the sample. Moreover, for a large part of the oscillation period of the cantilever, the interaction between cantilever tip and sample is negligible, while for a small part, the interaction is relatively large. Therefore, it is assumed that $F(t)$ can be approached by a sparse signal, i.e. many entries in $F(t)$ will equal zero and a few will have a value.

Accordingly, the following minimisation would produce an accurate solution to the problem:

$$\begin{aligned} \min_{\substack{F(k),\eta(k),x(1)\\ k=1,\dots,N-1}} & \sum_{t=1}^{N} \left\| R_\nu^{-1/2}(y_c(t) - Cx(t)) \right\|_2^2 + \sum_{t=1}^{N-1} \left\| R_\eta^{-1/2}\eta(t) \right\|_2^2 \\ \text{s.t. } & x(t+1) = Ax(t) + Bu_c(t) + B_2F(t) + \eta(t); \\ & \|V_F\|_0 = n_0 \end{aligned} \tag{3-2}$$

with

$$V_F = [\|F(1)\|_2, ..., \|F(N-1)\|_2].$$

The variable R_ν is the variance of ν, i.e. the expectation matrice $E[\nu(t)\nu^T(t)]$ and R_η is the variance of η. Variable n_0 is the known number of nonzero entries in $F(t)$. The second constraint $\|V_F\|_0 = n_0$ implies that the zero-norm - i.e. the number of nonzero elements - of V_F equals the known number of jumps. More information and common terminology [27] of norms is presented in Table 3-1.

Unfortunately, there are some drawbacks for direct implementation of optimisation problem 3-2. One of the drawbacks is its non-convexity, which results in strong increase in computational effort for increasing n_0 and N. Besides, the number of jumps n_0 is not known in AFM-measurements. For instance, the intermittent contact between cantilever tip and sample is temporarily interrupted during parachuting, so not every oscillation period of the cantilever contains a pulse in F. Therefore, alternative methods will be studied to find the tip-sample interaction.

The model-based and data-driven approach described in the following two sections are pragmatic methods to find the tip-sample interaction in AFM. Although the model-based approach is convex, the matrix B_2 should be known[1] for execution. On the contrary, the data-driven approach does not require knowledge of B_2, but is only sequentially convex.

Table 3-1: Definitions and common terminology of the norm $\|V_F\|_p$. The notation "card(V_F)" stands for the cardinality of vector V_F.

Range	**Definition**	**Terminology**
$1 \leq p \leq \infty$	$\left(\sum_{j=1}^{N} V_{F,j}^p\right)^{1/p}$	ℓ_p-norm
$0 < p < 1$	$\left(\sum_{j=1}^{N} V_{F,j}^p\right)^{1/p}$	quasi-norm
$p = 0$	card(V_F)	zero-norm

[1] In practice, knowledge of B_2 turns out to be difficult to obtain through system identification, due to $F(t)$ which cannot be measured directly. Therefore, it is beneficial if knowledge of matrix B_2 is not needed to estimate $F(t)$.

3-2 Model-Based Approach

Instead of solving the non-convex problem (3-2), it could be a better option to use State Smoothing by Sum-of-Norms Regularisation (STATESON) [25]. This is a relaxation of optimisation problem (3-2) which makes the problem convex. The computational time will be acceptable, while the results are still good estimates. The optimisation problem will be

$$\begin{aligned} &\min_{\substack{x(1),F(k),\\ 1\leq k\leq N-1}} \sum_{t=1}^{N} \left\| R_\nu^{-1/2}(y_c(t) - Cx(t)) \right\|_2^2 + \sum_{t=1}^{N-1} \lambda \left\| Q^{-1/2} F(t) \right\|_p \\ &\text{s.t. } x(t+1) = Ax(t) + Bu_c(t) + B_2 F(t). \end{aligned} \tag{3-3}$$

Parameter $\lambda \geq 0$ can be tuned to based on a trade-off between the fit to the measurements $y_c(t)$ and the contribution of $F(t)$. Besides, $p \geq 1$ can be tuned to control the sparsity of the outcome $\hat{F}(t)$. Note that the tip-sample interaction $F(t)$ is being handled as a stochastic variable with variance Q:

$$\begin{aligned} F(t) &= \delta_F(t)\gamma_F(t) \\ \delta_F(t) &= \begin{cases} 0 & \text{with probability } 1-\mu \\ 1 & \text{with probability } \mu. \end{cases} \\ \gamma_F(t) &\sim \mathcal{N}(0, Q). \end{aligned} \tag{3-4}$$

To find the optimal values, the optimisation algorithm (3-3) is incorporated in a certain procedure. In this way, an appropriate value for λ will be used and estimates of F will be accurate.

3-2-1 STATESON-algorithm

Below, the procedure of the STATESON-algorithm will be described.

1. Choose or calculate an appropriate value λ:
   (a) Introduce
   $$\varsigma_t \triangleq R_\nu^{-1/2}\Big(y_c(t) - C\Big(\sum_{r=1}^{t-1} A^{t-r-1}Bu_c(r) + A^{t-1}x(1)\Big)\Big) \tag{3-5}$$
   (b) Define $\bar{\varsigma}_t$ as
   $$\bar{\varsigma}_t = \min_{x(1)} \sum_{t=1}^{N} \|\varsigma_t\|_2^2 \tag{3-6}$$
   (c) Calculate the critical value $\lambda^{\max}$:
   $$\lambda^{\max} = \max_{k=1,\ldots,N-1} \left\| 2 \sum_{t=k+1}^{N} (R_\nu^{-1/2} C A^{t-k-1} B_2 Q^{1/2}) \bar{\varsigma}_t \right\|_q \tag{3-7}$$
   with $1/p + 1/q = 1$.

(d) Choose (this value may be tuned somewhat)

$$\lambda = \frac{1}{10}\sqrt{\frac{\|R_\nu\|}{\|Q\|}}\lambda^{\max}. \tag{3-8}$$

2. Set $\alpha^{(0)}(t) = 1$ for $t = 1, ..., N-1$ and $i = 0$.

3. Compute the optimal $F^{(i)}(t)$ using 3-9 with $\alpha(t) = \alpha^{(i)}(t)$:

$$\begin{aligned} &\min_{\substack{x(1),F(k),\\ 1\leq k\leq N-1}} \sum_{t=1}^{N} \left\| R_\nu^{-1/2}(y_c(t) - Cx(t)) \right\|_2^2 + \lambda \sum_{t=1}^{N-1} \alpha(t) \left\| Q^{-1/2}F(t) \right\|_p \\ &\text{s.t. } x(t+1) = Ax(t) + Bu_c(t) + B_2F(t) \end{aligned} \tag{3-9}$$

4. Set

$$\alpha^{(i+1)}(t) = 1/(\epsilon + \left\| Q^{-1/2}F^{(i)}(t) \right\|_p) \tag{3-10}$$

(for which ϵ is a positive tuning parameter) and possibly reduce λ.

5. If there is convergence of the estimates $\hat{F}$, go to the next step, otherwise set $i = i + 1$ and return to step 3.

6. Compute a final estimate of $\hat{F}(t)$ using

$$\begin{aligned} &\min_{\substack{x(1),F(k),\\ 1\leq k\leq N-1}} \sum_{t=1}^{N} \left\| R_\nu^{-1/2}(y_c(t) - Cx(t)) \right\|_2^2 \\ &\text{s.t. } x(t+1) = Ax(t) + Bu_c(t) + B_2F(t) \\ &F(t) = 0 \text{ if } t \notin \mathcal{T} \end{aligned} \tag{3-11}$$

with[2] $\mathcal{T} = \{t|\hat{F}(t) \neq 0\}$.

To summarise, the STATESON-algorithm and specifically the corresponding procedure is a pragmatic and accurate, convex relaxation of optimisation problem 3-2. The contribution of $F(t)$ is penalised with an ℓ_p-norm. The solution will be the sparsest for $p = 1$. An initial estimate of tuning parameter λ is provided in the procedure, as well as a rule for updating the weights $\alpha(t)$ to enhance convergence in subsequent iterations. Due to the use of the ℓ_p-norm, the apparent solution will be biased toward zero. To circumvent this issue, a final estimate $\hat{F}(t)$ is calculated by executing optimisation problem 3-11.

[2] In practice, the constraint $\mathcal{T} = \{t|\hat{F}(t) > \varepsilon\}$ will be used, with ε a small number, so that very small values in $F(t)$ are set to zero.

3-3 Data-Driven Approach

In the previous section, a convex method was proposed to find the sparse tip-sample interaction $F(t)$. However, it was assumed that the matrix B_2 was known. In practice, it can be challenging to find this matrix through system identification. Therefore, a sequentially convex method for which B_2 may be unknown, will be investigated in this section. Moreover, the goal of this data-driven method is to find both B_2 and $F(t)$. The goal is elaborated as follows:

A description of the AFM-cantilever dynamics alternative to Equation 3-1 is:

$$A(q)y_c(t) = B_u(q)u_c(t) + B_f(q)F(t), \tag{3-12}$$

from which $A(q)$, $B_u(q)$ are known from identification without presence of $F(t)$. The quantities $u_c(t)$ and $y_c(t)$, however, are measured for $t \in \mathscr{T}$ in presence of unmeasurable $F(t)$. Although $F(t)$ is unmeasurable, it is known that it is a pulse-like signal. The task is to recover both $B_f(q)$ and $F(t)$.

Sections 3-3-1 to 3-3-3 elaborate the procedure to solve the bilinear optimisation problem. Results from simulations are presented in Section 3-3-4.

3-3-1 Data Equation

A simpler formulation of Equation 3-12 is:

$$Y(k) = B_f(q)F(k) \tag{3-13}$$

with $Y(k)$ known. With a system of order 2, the latter equation is equivalent to

$$Y(k) = (b_{f,1}q^{-1} + b_{f,2}q^{-2})F(k) = b_{f,1}F(k-1) + b_{f,2}F(k-2) \tag{3-14}$$

This system description leads to the data equation

$$\begin{aligned}
&\begin{bmatrix} y_c(k) & y_c(k-1) & y_c(k-2) \\ y_c(k-1) & y_c(k-2) & y_c(k-3) \\ \vdots & \ddots & \vdots \\ y_c(k-N+3) & \dots & y_c(k-N+1) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \end{bmatrix} - \\
&\begin{bmatrix} u_c(k-1) & u_c(k-2) \\ u_c(k-2) & u_c(k-3) \\ \vdots & \vdots \\ u_c(k-N+2) & u_c(k-N+1) \end{bmatrix} \begin{bmatrix} b_{u,1} \\ b_{u,2} \end{bmatrix} = \begin{bmatrix} F(k-1) & F(k-2) \\ F(k-2) & F(k-3) \\ \vdots & \vdots \\ F(k-N+2) & F(k-N+1) \end{bmatrix} \begin{bmatrix} b_{f,1} \\ b_{f,2} \end{bmatrix}.
\end{aligned} \tag{3-15}$$

which can be rewritten in the short form:

$$\mathbf{Y} = \mathbf{F}b_f. \tag{3-16}$$

Note that the data equation can be easily adjusted to describe a system of higher order. The equality 3-16 yields that [28]

$$\mathbf{Y} - \mathbf{F}\mathbf{I}_2^{-1}b_f = 0 \quad \Longleftrightarrow \quad \operatorname{rank}\begin{bmatrix} \mathbf{Y} & \mathbf{F} \\ b_f & \mathbf{I}_2 \end{bmatrix} = \operatorname{rank}(\mathbf{M}) = 2. \tag{3-17}$$

3-3-2 Difference of Convex Programming

The Difference of Convex Programming algorithm, as described in [26], can be used to find both $\mathbf{F}$ and b_f such that $\text{rank}(\mathbf{M}) = 2$. It consists of two steps.

Step 1

Firstly, a minimisation of the nuclear norm will be performed:

$$\begin{aligned} &\min_{F,b_f} \|\mathbf{M}\|_* \\ &\text{s.t. } \mathbf{M} = \begin{bmatrix} \mathbf{Y} & \mathbf{F} \\ b_f & \mathbf{I}_2 \end{bmatrix} \\ &\mathbf{F} = \text{hankel}(F). \end{aligned} \tag{3-18}$$

The found matrix $\mathbf{M}$ will be decomposed using a singular value decomposition:

$$U\Sigma V^\top = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \Sigma \begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix} = \mathbf{M} \tag{3-19}$$

such that U_1 and V_1 are the first two[3] columns of U and V, respectively.

Step 2

The second step consists of iterating Equation 3-20 and 3-21 until convergence.

$$\begin{aligned} &\min_{F,b_f} \Big(\|\mathbf{M}\|_* - \text{tr}(U_1^\top \mathbf{M} V_1) \Big) \\ &\text{s.t. } \mathbf{M} = \begin{bmatrix} \mathbf{Y} & \mathbf{F} \\ b_f & \mathbf{I}_2 \end{bmatrix} \\ &\mathbf{F} = \text{hankel}(F) \end{aligned} \tag{3-20}$$

$$U\Sigma V^\top = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \Sigma \begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix} = \mathbf{M} \tag{3-21}$$

Both steps 1 and 2 are convex minimisations and therefore computationally attractive. The convergence of these iterations should ensure that $\text{rank}(\mathbf{M}) = 2$.

3-3-3 Including prior knowledge of $F(k)$

The outcome of Step 2 is a good first approximation of both F and b_f (up to a scaling factor). To make the estimation more robust against process and measurement noises, the prior knowledge that F is a pulse-like signal can be incorporated. This can be done using the following steps.

[3] Matrices U_1 and V_1 should be the first two columns of U and V, since from Equation 3-17, it is known that the rank of $\mathbf{M}$ should equal 2.

Step 3

Using the most recent updates of U_1 and V_1 of Step 2, the following minimisation should be done:

$$\begin{aligned} \min_{F,b_f} \Big(\|\mathbf{M}\|_* - \text{tr}(U_1^\top \mathbf{M} V_1) \Big) + \lambda_{\text{DCP}} \|\mathbf{F} b_f\|_1 \\ \text{s.t. } \mathbf{M} = \begin{bmatrix} \mathbf{Y} & \mathbf{F} \\ b_f & \mathbf{I}_2 \end{bmatrix} \\ \mathbf{F} = \text{hankel}(F) \end{aligned} \tag{3-22}$$

with λ_{DCP} a tuning parameter. Instead of iterating with new estimates of U_1 and V_1 like in Step 2, this minimisation is needed to perform only once. This is again a convex minimisation.

If the ℓ_1-norm would be included in Equation 3-20, the iterations in Step 2 would not converge, due to inappropriate estimations of U_1 and V_1. Therefore, Step 2 should be completed before executing Step 3.

In Equation 3-22, the ℓ_1-norm on $\mathbf{F}b_f$ is used instead of the norm on F. In this way, tuning parameter λ_{DCP} can be chosen appropriately before starting the execution of the whole algorithm. Moreover, the force F can only be estimated up to a scaling factor, while the quantity $\mathbf{F}b_f$ will be similar to the known order of $\mathbf{Y}$ (see Equation 3-16).

Step 4

Due to the use of the ℓ_1-norm in Step 3, it is likely that the magnitude of the estimated pulses will be too small. Therefore, inspired from [25], a set of time instances at which an estimated pulse occurs can be defined: $\mathcal{T} = \{t | \mathbf{F}(t,:) b_f > \varepsilon\}$ with ε a small value[4]. Using this set, a final optimisation can be done:

$$\begin{aligned} \min_{F,b_f} \Big(\|\mathbf{M}\|_* - \text{tr}(U_1^\top \mathbf{M} V_1) \Big) \\ \text{s.t. } \mathbf{M} = \begin{bmatrix} \mathbf{Y} & \mathbf{F} \\ b_f & \mathbf{I}_2 \end{bmatrix} \\ \mathbf{F} = \text{hankel}(F) \\ F(t) = 0 \text{ if } t \notin \mathcal{T} \end{aligned} \tag{3-23}$$

with $t = \{1, 2, 3, ..., N\}$. In this minimisation, the most recent updates of U_1 and V_1 of Step **2** are used. Similarly to Step 3, this minimisation is convex and needed to perform only once.

3-3-4 Simulations

Simulation experiments were done for the cases in which (a) the steps 1 through 3 are executed successively, and (b) in which the steps 1 through 4 are executed (software in Appendix C).

[4] In [25], the set is defined as $\mathcal{T} = \{t | F(t) \neq 0\}$. However, in practice this will result in $\mathcal{T} = \{t | F(t) > \varepsilon\}$ so that very small values in F are filtered out. A question arises how large to choose ε, especially because F is in Equation 3-22 estimated *up to a scaling factor*. Fortunately, the quantity $\mathbf{F}b_f$ will be of the same order of magnitude as $\mathbf{Y}$ (see Equation 3-16). Therefore, ε can be chosen small compared to the maximum value in $\mathbf{Y}$ and the set $\mathcal{T}$ will be defined as indicated in the text.

The quasi-Newton algorithm is used for minimisations. In each experiment, exactly identical measurement noise with variance 3.05×10^{-4} squared nanometers is included, so that a quantitative comparison can be made. The threshold is chosen to be $\varepsilon = 0.01 \times \|\mathbf{Y}\|_\infty$. In both cases (a) and (b), the tuning parameter λ_{DCP} is varied between 0.02 and 0.78 and the Mean Squared Error (MSE) between the measured $y_c(k)$ and estimated $\hat{y}(k)$ is plotted in Figure 3-1.

For very low values of λ_{DCP}, the MSE is very small. This is due to the ability to fully reconstruct the output signal y_c including the measurement noise. However, it is desired to reconstruct the real cantilever deflection with noise filtered out. Therefore, a solution of $\widehat{\mathbf{F}b_f}$ should be found which is sparse and which has a low MSE.

If tuning parameter λ_{DCP} is increased, the estimated quantity $\widehat{\mathbf{F}b_f}$ becomes sparser. In Figure 3-1 this is confirmed, since it shows increased values of the MSE for case (a), while the MSEs for case (b) remain low. However, if λ_{DCP} is chosen too high, no pulses are detected anymore and the MSE reaches the maximum. Thus, from Figure 3-1 it can be concluded that a sparse solution with low MSE is found in case (b) with $\lambda_{\text{DCP}} = 0.34$.

Figure 3-2 shows solutions of $(\widehat{\mathbf{F}b_f})_a$ and $(\widehat{\mathbf{F}b_f})_b$ with corresponding deflections $\hat{y}_a$ and $\hat{y}_b$ using $\lambda_{\text{DCP}} = 0.34$ for case (a) and case (b), respectively. In case (a), the solution is sparse. However, the estimates have a too small magnitude due to the use of the ℓ_1-norm instead of the zero-norm. In case (b), all entries satisfying $(\widehat{\mathbf{F}b_f})_a < \varepsilon$ are set to zero and Step 4 is applied, so that the solution is sparse, while its ℓ_1-norm is not penalized.

The scaled estimation of b_f is $(\hat{b}_f)_b = \begin{bmatrix} 0.0015 & 0.0145 \end{bmatrix}^\top$, while the actual values were $b_f = \begin{bmatrix} 0.0148 & 0.0145 \end{bmatrix}^\top$. The discrepancy may be caused by putting the ℓ_1-norm on $\mathbf{F}b_f$ instead of F. This may have caused sparsity of $\hat{b}_f$, in addition to F. Nevertheless, outcome still ensures that the estimated interaction is still relatively sparse, while the cantilever deflection (Figure 3-2 (a)) can be reconstructed accurately.

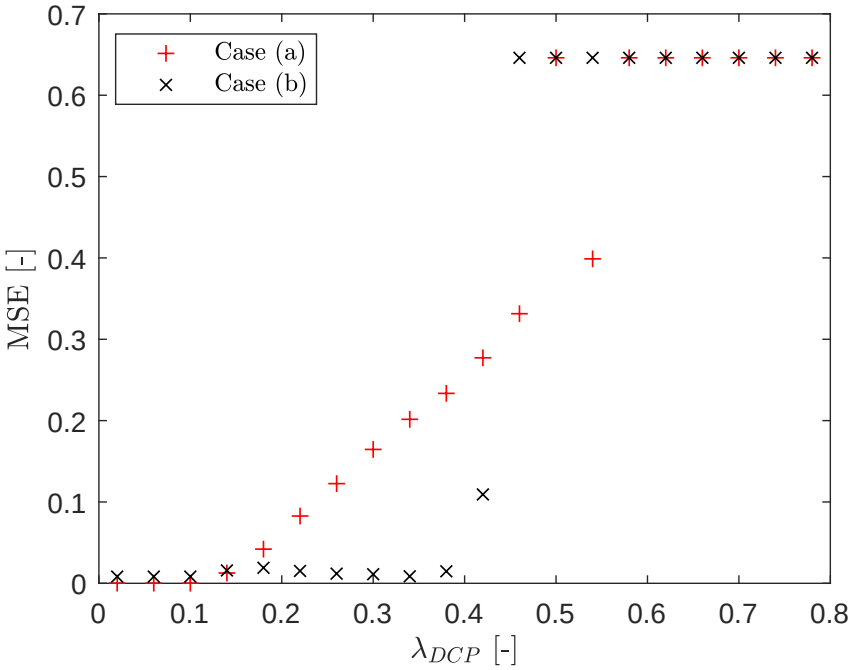


Figure 3-1: MSEs between y_c and $\hat{y}_a$ (red $+$) and $\hat{y}_b$ (black $\times$).

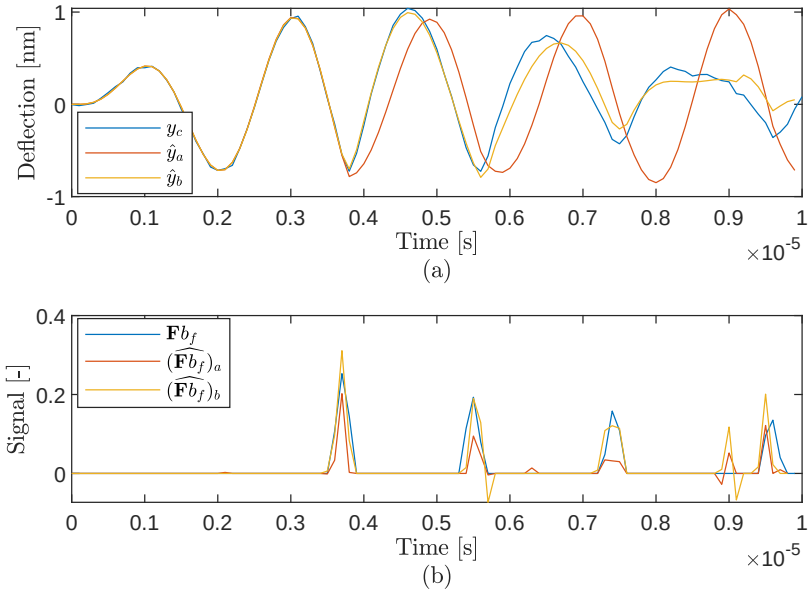


Figure 3-2: (a) Measured cantilever deflection y_c (blue) with estimated deflections $\hat{y}_a$ (red) and $\hat{y}_b$ (yellow) and (b) corresponding signals $\mathbf{F}b_f$ (blue), $(\widehat{\mathbf{F}b_f})_a$ (red) and $(\widehat{\mathbf{F}b_f})_b$ (yellow) with $\lambda_{\text{DCP}} = 0.34$.

Chapter 4

Experimental Results

In this chapter, the experimental results of both model-based and data-driven approach are presented. The results of the model-based approach are more comprehensive including reconstruced AFM-images, whereas the data-driven approach is only tested at a small part of the data. Moreover, the idea is to use the estimation of a parameter ($\hat{b}_f$) from the data-driven approach in the model-based approach.

The chapter is structured as follows. Firstly, the experimental setup is elaborated in Section 4-1. Secondly, two procedures for system identification are presented and tested in Section 4-2, from which the first procedure succeeds in identifying a complex model from the input of the cantilever piezo to the cantilever deflection and the second procedure results in a simple second-order model of the physical cantilever only. The latter model is used for experiments in subsequent sections. After identification, this chapter continues with presenting the reconstructed images of the model-based approach in Section 4-3. Those images are analysed qualitatively in Section 4-3-1 and quantitatively in Section 4-3-2. The latter analysis includes a metric in which the Fourier Ring Correlation is used, while a reference image is unnecessary. Finally, the results of the data-driven approach are presented in Section 4-4. Software can be found in Appendix C.

4-1 Setup

The experiments are performed using an AFM designed by T. Ando [29]. Pictures can be found in Appendix B. A Picoscope 5443D [30] is used to collect the cantilever's I/O-signals with a rate of 1.0417×10^7 Hz.

With a length of 7 μm and a spring constant of 0.15 N/m [14], the cantilever has a resonance frequency in liquid of approximately 500 kHz. The free oscillation amplitude is about 1.4 nm. In the x- and y-dimensions, the maximum scan range is 700 nm. The experiments are done in liquid, since scanning in liquid results usually in less influence on the sample [16] and in addition, biological samples often show their natural behaviour in liquids, which is eventually desired to be observed.

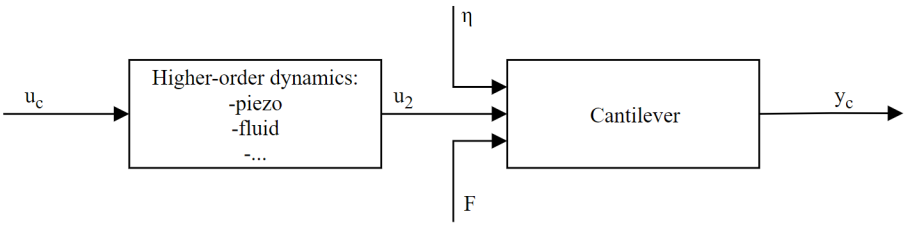


Figure 4-1: The splitting of the whole cantilever system into one block with the higher-order dynamics and one block with easier cantilever dynamics.

The AFM is controlled using the AM-PI-method from Figure 2-3. However, the controller parameters are set such that the input to the z-actuator has minimal variations, i.e. z is approximately constant. As a consequence, the sample height can be estimated by measuring the cantilever's oscillation amplitude, which will be comparison material for the new height estimation algorithms. Yet, the input to the z-actuator cannot be perfectly constant, since the cantilever tip may drift away from the sample.

Graphite is used as test sample, since this brittle material consists of atomic layers. Therefore, it is expected to contain sudden steps in sample height within the order of nanometers. Moreover, studying system dynamics with a step as input is very useful in determination and comparison of the system's performance.

For system identification in Section 4-2, the cantilever is off-sample so that there is no interaction between cantilever tip and sample. However, for both the Model-Based and Data-Driven approach in Sections 4-3 and 4-4, the cantilever does have intermittent contact with sample and the scan table is activated to make scans of the graphite sample.

4-2 System Identification

To perform the new algorithms for estimating the sample height, an accurate model of the cantilever is needed. Such a model can be obtained by system identification. There exist at least two ways to perform system identification of the AFM cantilever. The first manner is to identify the system from the piezo input to the cantilever deflection, i.e. from u_c to y_c in Figure 4-1. However, the result can be a complex model, since higher-order dynamics of for example the actuator (piezo) and fluid will also be part of the model. Elaboration and results regarding this approach for identification can be found in Section 4-2-1.

An alternative is to make use of the thermal motion of the cantilever [31]. In this approach, it is assumed that thermal forces act directly on the cantilever. Accordingly, the higher-order dynamics of (among other things) the piezo and fluid are bypassed, so that a simple model remains. This approach is elaborated in Section 4-2-2.

4-2-1 Using piezo input and cantilever deflection

Although the cantilever seems to be a very trivial element in the AFM, its identification from piezo input to measured cantilever deflection can be a challenging task, even without presence of tip-sample interaction. Moreover, the piezo is attached at an inconvenient location of the cantilever. Furthermore, this piezo itself can have complex dynamics. For those reasons, a black-box identification method is chosen.

A Pseudo-Random Binary Sequence (PRBS) is chosen as input for identification, because of its white noise-like properties while being less harmful for the system than white noise. The input spectrum of the PRBS-signal with duration 2 seconds, is shown in Appendix A-1.

The Past Outputs Multivariable Output-Error State-sPace (PO-MOESP) method [32] is used to identify the cantilever dynamics, while the measurements were done in presence of both process and measurement noise. Moreover, the identification measurements were done in open loop, so that the process and measurement noise are uncorrelated.

Using Hankel matrices of the input and output of size $\mathbb{R}^{s\times(N_b-s)}$ with N_b the number of measurements per segment and s chosen to be 2000, the singular values of R_{32} [32] are shown in Appendix A-1. Now it is important to choose the right order of the model such that the cantilever dynamics can be described satisfactorily. To do so, small sudden steps in magnitude of successive singular values should be found (see Appendix A-1). Figure 4-2 shows the Variance Accounted For (VAF) for each choice of model order n. Those VAFs are determined using a frequency sweep (with duration 1 second) as validation input signal, with spectrum as is presented in Appendix A-1. The VAFs converge to a value near 80% (optimum with $s = 2000$ is 79.2%). Higher percentages cannot be achieved in this case, possibly due to noise contributions and nonlinearities. Besides, the VAF can differ for alternative validation input signals. A better validation signal would have a wider range of frequencies for which the power is constant, since more frequencies are tested with equal weight for the VAF.

Besides, Figure 4-2 shows that the choice of parameter s has influence on the performance of system identification. For higher values of s, larger VAFs can be achieved. It can be expected that the highest achievable VAF will converge for $s \to \infty$, which is gently suggested in Figure 4-2. Namely, the parameter s increases linearly, while the rise in maximum achievable VAF per choice of s declines.

A bode plot of the identified system with the highest VAF is shown in Figure 4-3. The bode plot consists of many peaks due to the high order, which is in case probably due to large contributions in the dynamics of the liquid [16]. Moreover, the large support chip (or: module) of the cantilever (described in Section 2-1) is also immersed in the liquid and accounts for many peaks in the response spectrum [33]. This phenomenon is well known in tapping mode AFM in liquid and also called the “forest of peaks” [33].

Further improvement on system identification can be done by reducing the model order in an appropriate way. Moreover, the process and measurement noise could be coloured instead of white. The dynamics of coloured noise appear in the identified model. Often, those dynamics are included in the states which are the weakest coupled to the input and output. Therefore, the weakest coupled states could be deleted from the model to potentially improve the model.

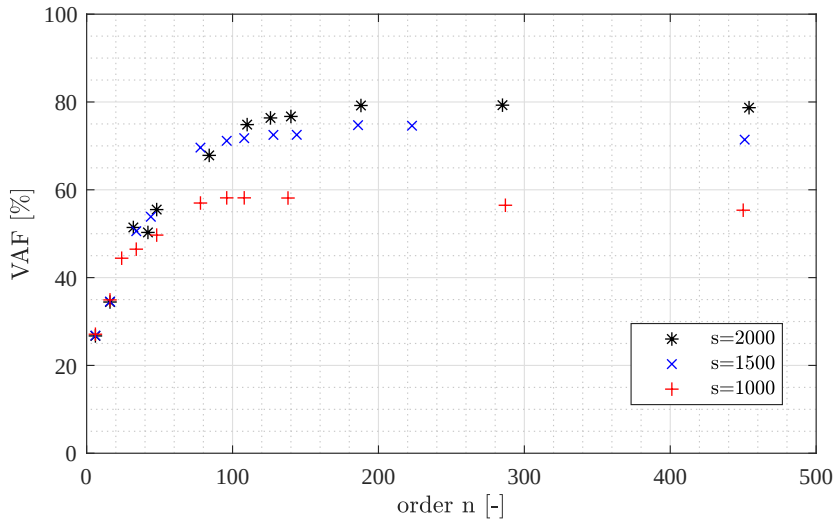


Figure 4-2: VAFs for several choices of model orders for $s = 1000$ (red $+$), $s = 1500$ (blue $\times$) and $s = 2000$ (black $*$).

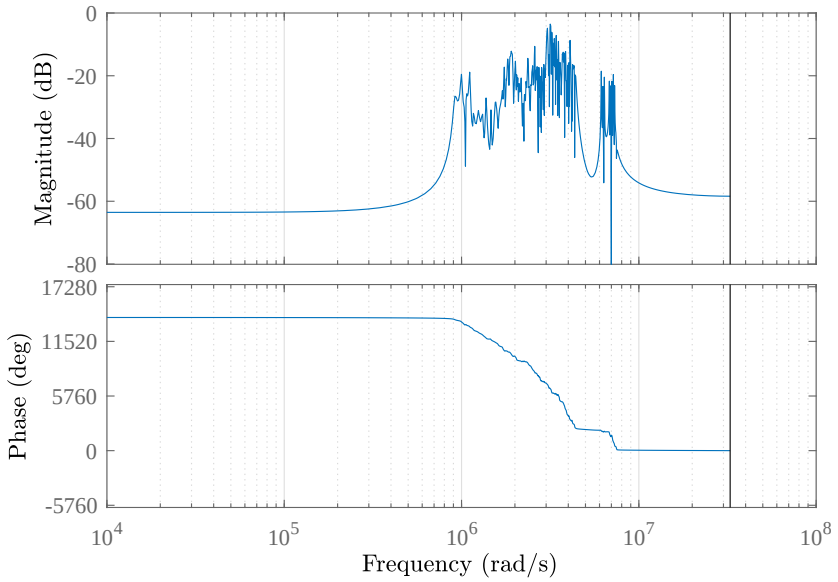


Figure 4-3: Bode plot of model with order 188, obtained using $s = 2000$, identified using piezo input and cantilever deflection.

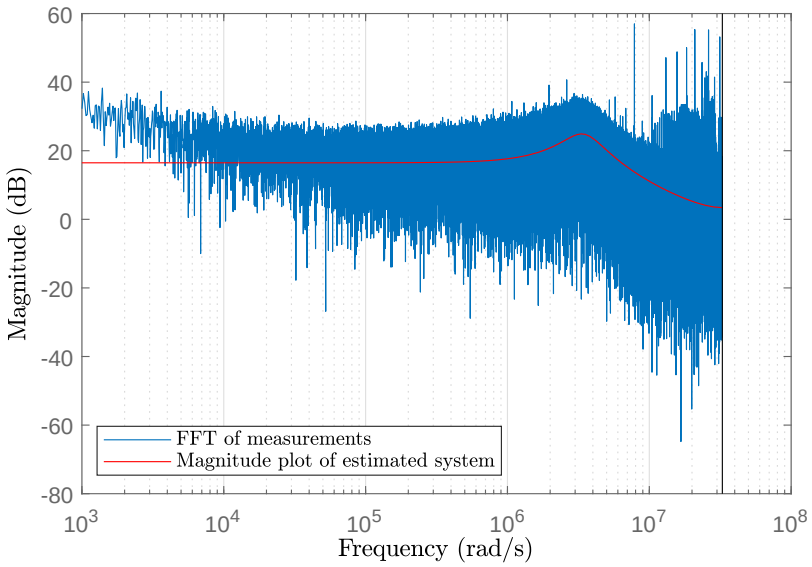


Figure 4-4: Magnitude plot of estimated system (red) and the FFT of the measurements (blue), identified using thermal motion of the cantilever.

4-2-2 Using thermal motion of the cantilever

The system used for identification of the cantilever is

$$\begin{aligned} x(k+1) &= Ax(k) + K\nu(k) \\ y(k) &= Cx(k). \end{aligned} \tag{4-1}$$

After identifying A, C and K using the PO-MOESP-algorithm, the matrices B and D are determined as:

$$\begin{aligned} B &= K \\ D &= 0. \end{aligned} \tag{4-2}$$

After identification from thermal noise to cantilever deflection, a sinusiodal signal is applied to the system, so that the phase difference between u_c and u_2 can be determined, as well as the gain from u_c to y_c (see Figure 4-1). This will be done by fitting the modelled cantilever output with the measured output with the Least Squared Error (LSE), using phase difference τ and gain c alternately as optimisation variables. Afterwards, the input is determined as:

$$u_2(1 : N - \tau) = cu_c(1 + \tau : N). \tag{4-3}$$

Since there is no appropriate control of u_2 and the actual thermal noise η is unknown, the identified model using thermal motion cannot be validated. The model can, however, be assessed by applying the Sum-of-Norms Regularisation.

The magnitude plot of the identified system, together with the Fast Fourier Transform (FFT) of the measured data, is shown in Figure 4-4.

4-3 Model-Based Approach

In this section, the Model-Based Approach (i.e. the STATESON-algorithm) is experimentally tested for one entire frame. Beside comparing the results with the conventional imaging method, the STATESON-algorithm will also be examined for different values of λ. After a detailed explanation of the experiment, the obtained images will be compared qualitatively in Section 4-3-1. Afterwards, the image qualities are quantified in Section 4-3-2.

The STATESON-algorithm is applied as is explained in Section 3-2 with parameters given in Appendix A-2. Each iteration of Equation 3-9 and 3-10, the parameter λ is decreased with a factor 5.

Equation 3-6 is changed to

$$\bar{\varsigma}_t = \sum_{t=1}^{N} \|\varsigma_t\|_2^2 \Big|_{x(1)=0} \tag{4-4}$$

to enhance the computational speed. The second and final minor adjustment to reduce the computational time is the splitting of data in segments of $N_T = 300$ measurements of u_c and y_c. The STATESON-algorithm is applied for each segment with initial state equal to the final state of the former segment.

Effort has been made to implement the "Least Absolute Shrinkage and Selection Operator" (LASSO)-problem [25][34] of Equation 3-9 with $p = 1$ in the "Alternating Direction Method of Multipliers" (ADMM) [35] to further reduce the computational time. However, this implementation - presented in Appendix C-5 - has neither led to improvement nor to degradation.

In Section 4-2, two distinct ways are executed for identifying the cantilever dynamics. While the first method succeeded in identifying the complex dynamics from the piezo input to the cantilever deflection, the second method provided a simpler model of the actual cantilever itself. Even though the model of the second method was not validated and may be less accurate, its simplicity provides the opportunity to reduce the computational effort needed for the STATESON-algorithm. Moreover, the computational time using a second-order model is reduced with a factor of circa 800, compared with using a 188th-order model. Besides, the segments in which the data is split should be larger, since otherwise the estimated initial condition can be used to compensate large contributions of the tip-sample interaction. Finally, the assumption that F influences the model similarly to u_c is less valid than the assumption that F influences the model similarly to u_2.

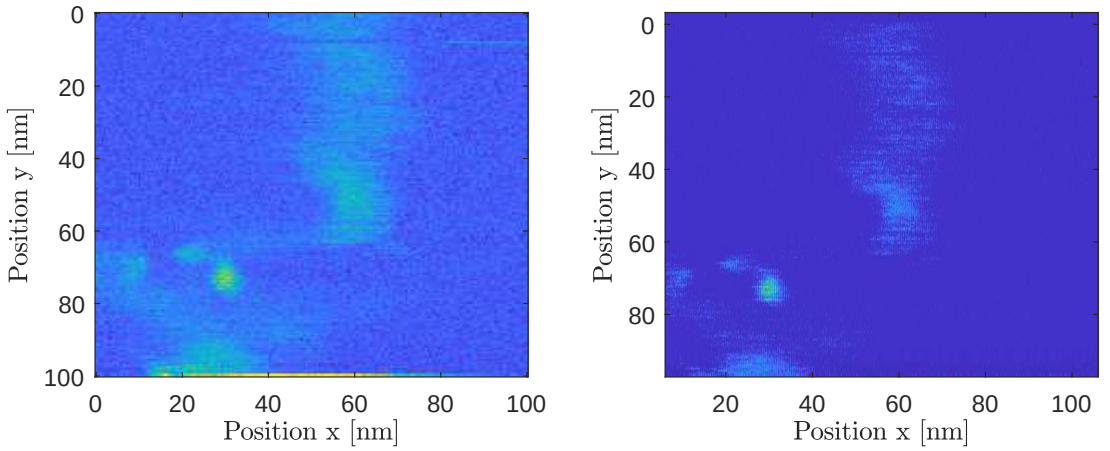
After executing the STATESON-algorithm, the estimates $\hat{F}(t)$ and $\hat{y}_c(t) = C\hat{x}(t)$ are available. Using those outcomes, the sample height will be estimated:

$$c_4 + \frac{1}{c_3}\hat{h}(t = \{1, ..., N_T\} + kN_T) = \min_t \Big(\hat{y}_c(t = \{1, ..., N_T\} + kN_T)\Big) + 0.02 \times \sum_{t=kN_T+1}^{(k+1)N_T} \hat{F}(t) \tag{4-5}$$

with $k = \{0, 1, 2, ...\}$ the segment number and $\{c_3, c_4\}$ constants chosen such that the sample height is estimated between known lower and upper bounds.

4-3-1 Qualitative Image Analysis

The conventional amplitude plot is shown in Figure 4-5 (similar to Figure 2-4 b-2), together with the height reconstruction using the STATESON-algorithm using $\lambda/\lambda^{\max} = 2.683 \times 10^{-3}$.



(a): Conventional image (b): Reconstruction using STATESON

Figure 4-5: Conventional amplitude plot (a) and height reconstruction using STATESON (b) with $\lambda/\lambda^{\text{max}} = 2.683 \times 10^{-3}$. Scan direction was from right to left. The estimated sample height varies between 0 and 1 nm.

Those images are zoomed in in Figure 4-6 (a) and (c) to emphasize the differences.

Considerable improvement in favour of the STATESON-algorithm is visible in Figure 4-6. Using the conventional amplitude plot, no more than a simple dot can be seen. However, the plot using STATESON is more nuanced, as it shows certain vertical lines. It is expected that these lines are the result of oscillations in the scan table due to abrupt changes in direction at the edges of the sample. The vertical lines are oscillations in the scan table for three reasons. Firstly, the vertical lines are *not* induced by the sample structure. Moreover, the distance between adjacent carbon atoms in graphite is 0.142 nm and the distance between the carbon layers is 0.341 nm [36]. On the contrary, the intermittent distance in x-direction between the peaks is circa 0.52 nm. Those distances are not similar and therefore, the vertical lines are not induced by the sample structure.

Secondly, the vertical lines are *not* induced by the cantilever dynamics. The bode plot in Figure 4-3 substantiates this by not showing a peak at 2.6×10^5 rad/s.

Thirdly, the oscillations show up as nice vertical lines in the image. Moreover, similar results with oscillations in the scan table are being obtained in previous studies [37]. In those studies, the oscillations had an amplitude of >5 nm. The STATESON-results in Figure 4-6, however, show oscillations with amplitude in the order of tenths of nanometers (Ångströms). Altogether, the vertical lines in Figure 4-6 (b-d) are induced by oscillations in the scan table.

The effect of choosing a smaller or larger λ is shown in Figure 4-6 (b) and (d), respectively. On one hand, the parameter λ should not be chosen too small, since more noise comes in. But on the other hand, choosing λ too large may result in loss of valuable information.

For a part of Figure 4-6 (c) (the red horizontal line), the underlying data is shown in Figure 4-7. Also here, the oscillations in the scan table can be recognised. Performing the algorithm over the entire 100×100 nm-sized image took circa 50 hours per image.

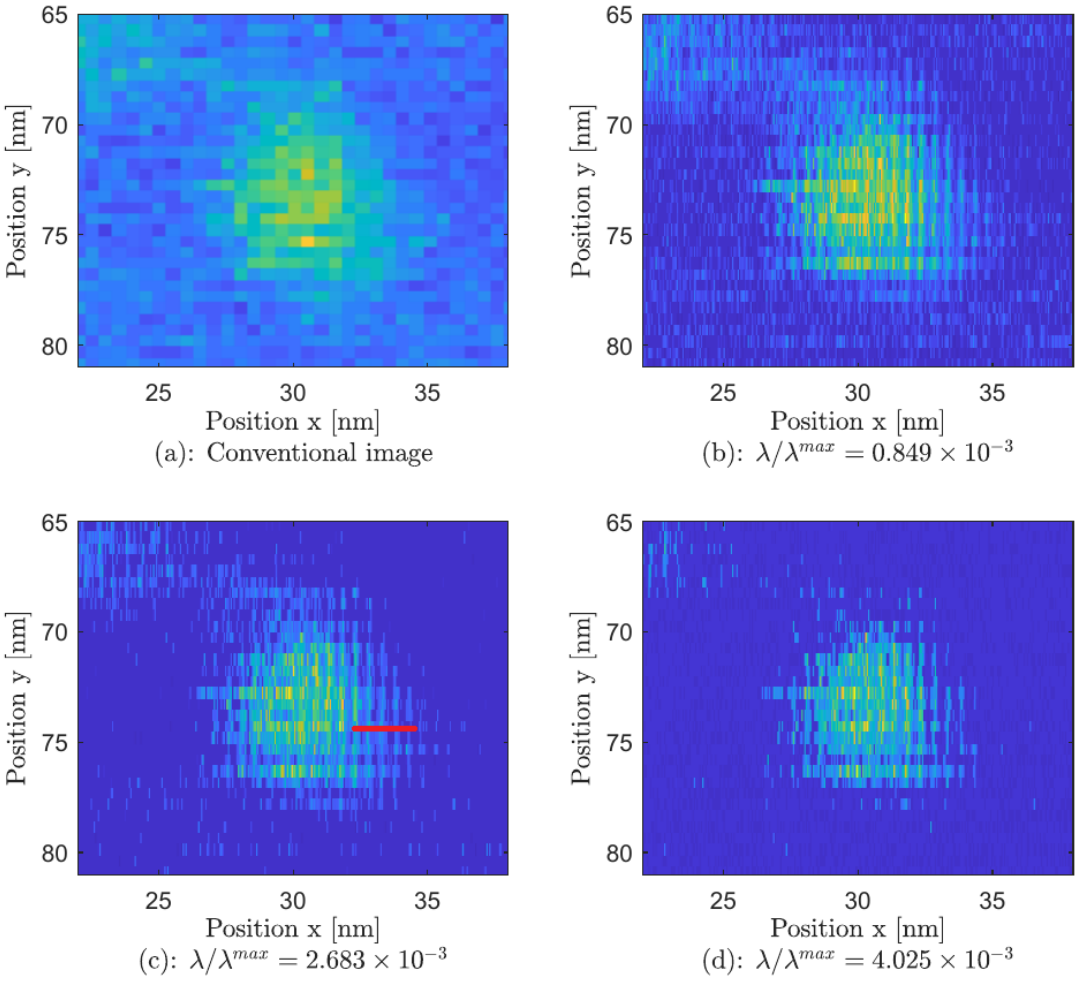


Figure 4-6: Zoom of amplitude plot (a) and height reconstructions using STATESON (b-d) with $\lambda/\lambda^{\text{max}} = \{0.894, 2.683, 4.025\} \times 10^{-3}$, respectively. As λ increases, the image contains less noise. Besides, vertical lines become visible using STATESON.

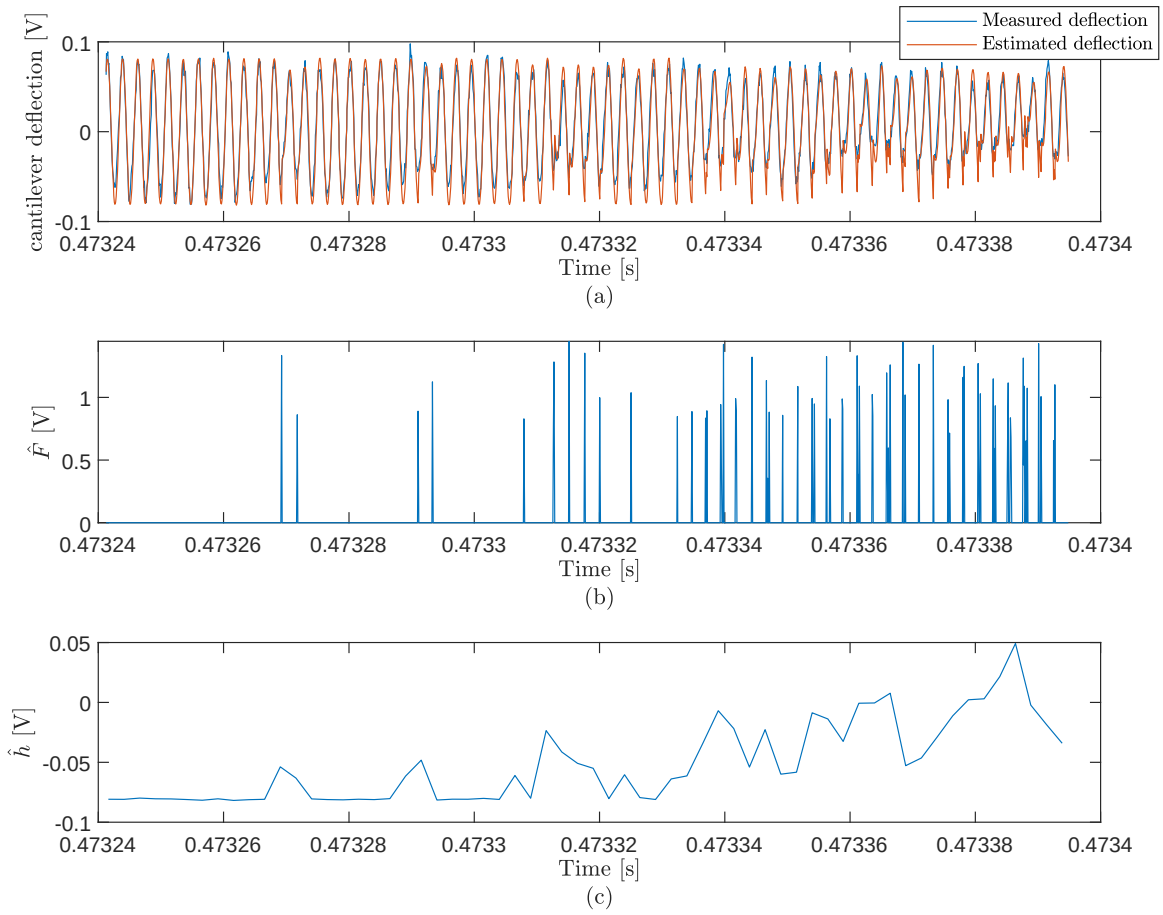


Figure 4-7: Measured (red) and estimated (blue) cantilever deflection (a), estimated tip-sample interaction using STATESON (b) with $\lambda/\lambda^{\max} = 2.683 \times 10^{-3}$ and resulting sample height estimation (c) at the segment corresponding to the red line in Figure 4-6. Note that the scan direction was from right to left.

4-3-2 Quantitative Image Analysis

Although differences in the obtained images and the conventional image are already visible, their quality will be assessed in terms of a value for the resolution. A difficulty is the absence of a reference image[1], which means that the images cannot be compared to an 'ideal' image. As a consequence, a method will be used in which the use of a reference image is avoided. A suitable method splits each image up in two images and compares those using Fourier Ring Correlation (FRC) [38]. Although this method was in [38] used as metric for super-resolution microscopy, the idea is universally applicable for grayscale images, and therefore for AFM. The procedure for determination of the image resolution is as follows.

1. Two images are generated by assigning each pixel of the original image randomly to one of the two new images. So each new image comprises approximately half of zero-valued pixels and half of pixels with values of the original image.

2. The number of pixels of the new images is increased to 3000×3000 pixels and all pixels are filled with a value proportionally to the two images obtained in the previous step. This means that many neighbouring pixels in the resulting images will have the same value. The increase in number of pixels is needed to raise the maximum spatial frequency that can be resolved in the image, according to the Nyquist criterion [38].

3. The images from the previous step, I_1 and I_2 are shifted and Fourier transformed to $\tilde{I}_1$ and $\tilde{I}_2$. Afterwards, the FRC is calculated:

$$\mathrm{FRC}(R) = \frac{\sum_{i \in R} \tilde{I}_1(r_i) \cdot \tilde{I}_2(r_i)^*}{\sqrt{(\sum_{i \in R} |\tilde{I}_1(r_i)|^2) \cdot (\sum_{i \in R} |\tilde{I}_2(r_i)|^2)}} \tag{4-6}$$

   with r_i the pixel in frequency domain at location i, for which i can indicate any pixel (in frequency domain) at the perimeter of the circle with radius R.

4. The σ_3-curve is calculated as [39]

$$\sigma_3(R) = \frac{3}{\sqrt{N_p(R)/2}} \cdot \sqrt{n_{\mathrm{asym}}} \tag{4-7}$$

   with $N_p(R)$ the number of pixels (in frequency domain) at the ring with radius R. Marin van Heel and Michael Schatz [39] reported design considerations of this threshold curve. The parameter n_{asym} determines the symmetry of the image (it equals 1 for assymmetrical images up to 60 for symmetrical images). In this report, it is chosen to be $n_{\mathrm{asym}} = 2$.

5. The values for the radius R are chosen such that it is increasing linearly from 1 to 1500. Those values correspond to the spatial frequencies $f = R/3000$.

[1] Obtaining a reference image entails several complications. For example, if another type of microscope is used for this, the sample should be transferred without being damaged and without being exposed to dust. In addition, exactly the same 100×100 squared nanometers area should be found for both microscopes. Therefore, even when it is actually possible to obtain a reference image, the determination of image quality is more beneficial and will take much less effort without using a reference.

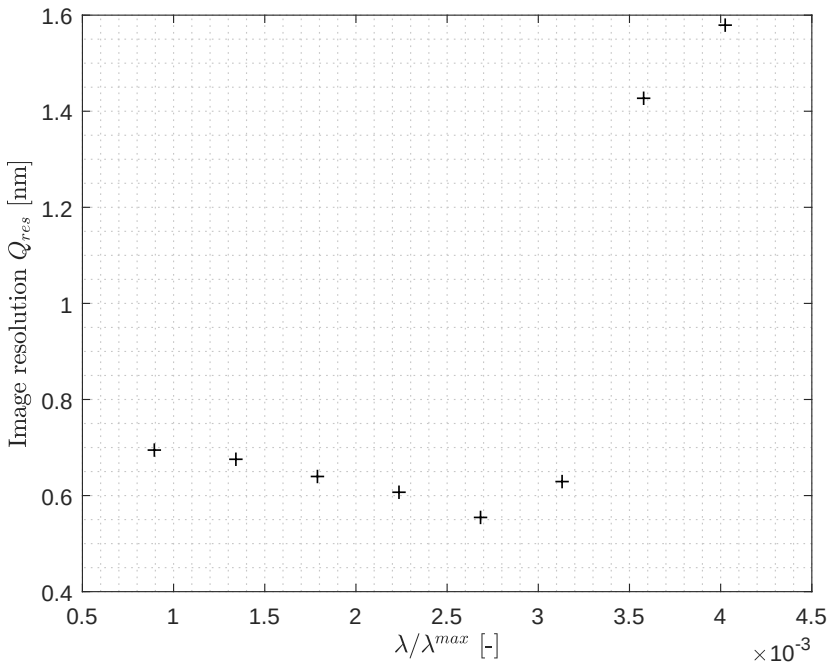


Figure 4-8: Image resolution in nanometers for different choices of λ, according to the FRCs calculated in Steps 1 to 6. Optimum at $\lambda/\lambda^{\text{max}} = 2.683\times10^{-3}$ with image resolution $Q_{\text{res}} = 0.555$ nm.

6. The first point f_0 at which the LOESS smoothed [40] FRC-curve crosses the σ_3-curve with derivative smaller (i.e. more negative) than the derivative of the σ_3-curve, determines the resolution of the original image. Since the original image is 100×100 squared nanometers and the images I_1 and I_2 consist of 3000×3000 pixels, the resolution in nanometers is:

$$Q_{\text{res}} = \frac{100}{3000 \times f_0} \tag{4-8}$$

Steps 1 to 3 are shown schematically in Appendix A-2-1. Steps 3 to 6 are performed with the help of the Fourier Ring Correlation Plugin of Olivier Burri and Alex Herbert [41].

Results Quantitative Image Analysis

While the figures of the FRCs and σ_3-curves are shown in Appendix A-2-1, the outcomes of the obtained resolutions for several values of λ are shown in Figure 4-8. It can be concluded that images in which the STATESON-algorithm is used have a significant better resolution (of optimally 0.555 nm) than the amplitude plot, which has a resolution of 4.136 nm.

Another important observation from Figure 4-8 is the existence of an optimal value for λ. Namely, on one side it is desired to filter out noise ($\lambda \uparrow$), but on the other it is undesirable to lose information of the sample ($\lambda \downarrow$). Figure 4-8 shows that this method for image quality assessment is suitable to find the optimal λ. In this case, $\lambda/\lambda^{\text{max}} = 2.683\times10^{-3}$ is the optimal choice.

The bad resolution of the amplitude plot is partially due to the small number of pixels of the original image. Namely, every oscillation period resulted in a value for a pixel in the STATESON-images, resulting in 200×1907-sized images, while conventionally no more was stored than a 200×200-sized image. If the amplitude was measured every oscillation period as $(\max(y_c) - \min(y_c))/2$ resulting in a 200×1907-sized image, then the resolution would already be increased to 1.685 nm (see Appendix A-2). Yet, a prerequisite for applying this is that the entire envelope of the cantilever deflection is recorded, rather than only its amplitude and phase using the conventional LIA.

Based on the results in this section, it can be concluded that the method from Banterle et al. [38] (with small adaptations) is very suitable to assess the quality of AFM-images. An important advantage is that a perfect reference of the sample is not needed. However, a few side-notes must be made. Namely, the oscillations in the scan table acted in this case as evidence that the STATESON-algorithm performs better than the conventional method to generate an image. Likewise, the assessment method produced better resolutions for the STATESON-algorithm. However, oscillations in the scan table result in less visibility of the sample. Therefore, the actual image resolution with respect to the sample will be worse than the resolutions obtained in this study. In addition, in presence of oscillations in the scan table, care should be taken when comparing AFM-resolutions with resolutions of other types of microscopes.

4-4 Data-Driven Approach

In this section, the Data-Driven Approach is experimentally tested on a small segment of measurements. The quantities a_1, a_2 and $b_{u,1}$, $b_{u,2}$ are known from identification and $\varepsilon = 0.01 \times \|\mathbf{Y}\|_\infty$. Tuning parameter λ_{DCP} is varied between 0 and 0.38 and similarly to the simulations, the MSEs between the measured y_c and estimated $\hat{y}_c$ are plotted in Figure 4-9.

Surprisingly, the MSEs for case (b) are higher than those for case (a), while the estimated $(\widehat{\mathbf{F}b_f})_a$ is penalised with the ℓ_1-norm and $(\widehat{\mathbf{F}b_f})_b$ is not. Probably, this is due to the constraint $F(t) = 0$ if $t \notin \mathcal{T}$ with $\mathcal{T} = \{t|(\widehat{\mathbf{F}(t,:)b_f})_a > \varepsilon\}$. Besides, the actual cantilever is exposed to thermal (process) noise, while this was not contained in simulations. However, a better MSE between y_c and $\hat{y}_c$ does not necessarily imply a better estimate of b_f, since the transfer function from F to y_c is desired to obtain rather than the transfer function between η and y_c. Moreover, to find the transfer function from F to y_c, we look again for a solution which is sparse, while reconstructing the signal y_c appropriately. This solution is found for case (b) with $\lambda_{\text{DCP}} = 0.2$. For this λ_{DCP}, the cantilever deflection is reconstructed for case (a) and case (b) in Figure 4-10.

The corresponding $b_f = \begin{bmatrix} 10.3351 & -7.1368 \end{bmatrix}^\top$ can, together with $\begin{bmatrix} 1 & a_1 & a_2 \end{bmatrix}^\top$ be converted to a state-space system, from which B_2 can be obtained. Using this information, the STATESON-algorithm could be executed and is likely to generate improved results, compared to Section 4-3. The two methods are not being combined yet, since the data-driven method is still in the development phase.

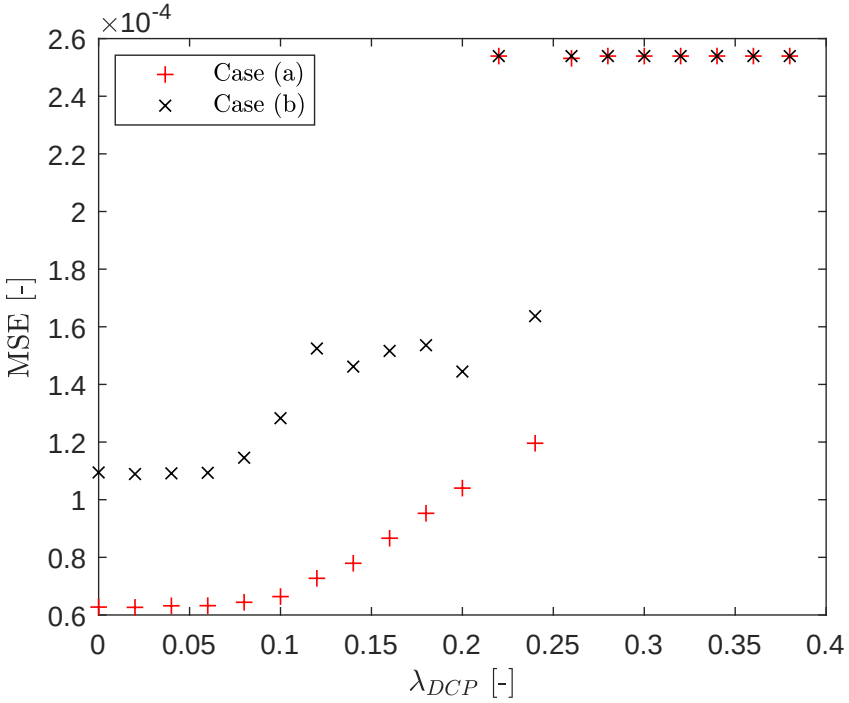


Figure 4-9: MSEs between y_c and $\hat{y}_a$ (red $+$) and $\hat{y}_b$ (black $\times$).

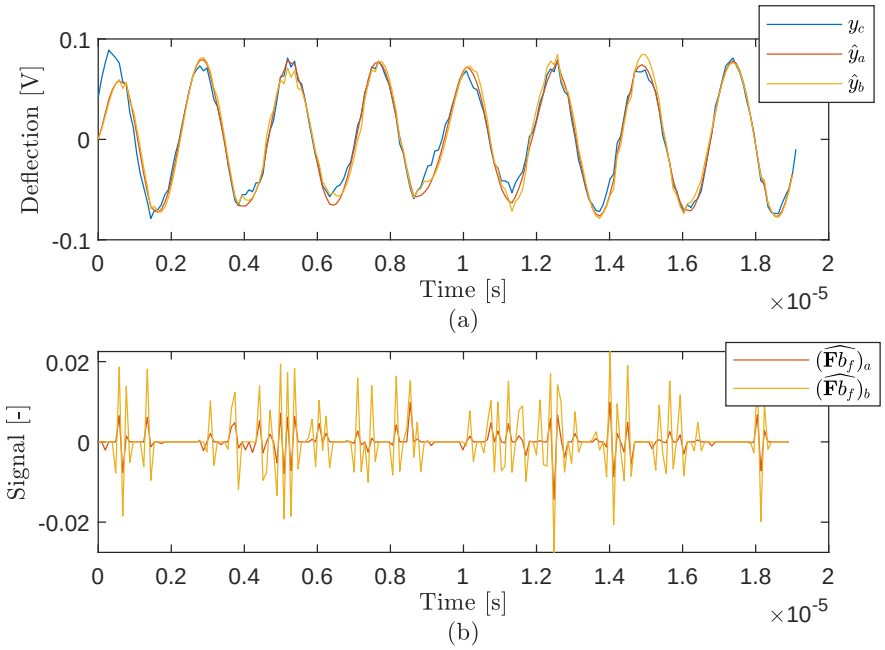


Figure 4-10: (a) Measured cantilever deflection y_c (blue) with estimated deflections $\hat{y}_a$ (red) and $\hat{y}_b$ (yellow) and (b) corresponding signals $(\widehat{\mathbf{F}b_f})_a$ (red) and $(\widehat{\mathbf{F}b_f})_b$ (yellow) with $\lambda_{\text{DCP}} = 0.2$.

Chapter 5

Conclusions

The goal of this research was to improve AFM-imaging in general. After concluding that the use of an LIA or PLL is suboptimal, the research continued studying methods to reconstruct the tip-sample interaction using directly measured input and output of the AFM-cantilever. Moreover, a model-based and a data-driven approach were presented.

Before testing the presented approaches experimentally, a decent system identification was needed. This is being performed in two ways: (1) via measured piezo input and cantilever deflection and (2) via measured cantilever deflection while the piezo was turned off. Case (1) led to interesting results, as up until now, no publications reported a successful identification of an LTI-model to a conventional AFM-cantilever immersed in liquid, from piezo input to cantilever deflection. Probably due to the liquid, the actual system has a very high order. The maximum achievable VAF rises for increased values of parameter s. Using $s = 2000$ and $n = 188$, the identified LTI-model can describe 79.2% of the dynamics. Despite this result, the model-based and data-driven approaches to detect the tip-sample interaction are tested with the second-order model obtained from identification case (2), for the reason of suppressing the computational effort.

The model-based approach to detect tip-sample interaction yielded – in terms of the FRC-metric of Banterle et al. – a resolution of 0.555 nm, implying an increase in resolution with a factor 7.5 compared to the conventional imaging method, which had a resolution of 4.136 nm. Furthermore, the data-driven approach – which is still in the development phase – is expected to improve the resolution even further, since the identification of the cantilever dynamics will be more accurate.

5-1 Recommendations

Based on the results in this report, the following practical recommendations can be made:

- Sample height estimation in AFM is recommended to be based on directly measured input and output signals of the AFM-cantilever. Moreover, the tip-sample interaction

can be detected with those measurements and based on the interaction, the estimated sample height will be more accurate.

- It is recommended to handle the tip-sample interaction as a sparse signal, since it is negligible in magnitude for a large part of the oscillation cycle of the AFM-cantilever, while it is relatively large in magnitude at time instances at which there is physical contact between tip and sample. Furthermore, the obtained images have a significant higher resolution than the original image, while the assumption was made that the tip-sample interaction is pulse-like.
- AFM-images are advised to assess using the FRC-metric of Banterle et al., since a reference image is not required in this pragmatic metric and it is still able to make a trade-off between noise reduction and information conservation. However, care should be taken if significant artefacts occur in the AFM-image, as those artefacts will contribute to a higher resolution according to the metric, while such artefacts are in general detrimental.

In addition to practical recommendations, this report leads to the following recommendations for future research:

- Using the obtained LTI-model from piezo input to cantilever deflection, the cantilever dynamics are even described more accurately and more reliably than the model using thermal motion. Therefore, it is recommended to investigate estimation of tip-sample interaction using this model, while suppressing the computational effort.
- The data-driven and the model-based approach for detecting the tip-sample interaction are not being combined yet. It is encouraged to research the additional improvements which can be made using this combination, since due to the data-driven method to find b_f, it is more likely to find a sparse solution for $F(t)$ in the model-based approach.
- It is recommended to further study the data-driven approach to estimate b_f more precisely. Currently, the estimated b_f supports the model-based approach for finding sparse solutions, while estimating exactly the correct b_f would be even more beneficial.
- Currently, the control to both cantilever input and input to the z-actuator of the scan table, is still determined using the LIA or PLL. Future research should be concerned with implementing real-time estimation of tip-sample interaction, such that the real-time controls can be based on accurate sample height information.
- Due to improvement in sample height estimation through enhanced processing of the measured cantilever deflection, oscillations in the scan table were revealed. Therefore, research should be done to reduce and/or to filter those oscillations to improve the image resolution in AFM further.
- Compressed Sensing is concerned with recovering sparse signals using the least possible measurements and as little as possible computational effort. Therefore, research can be done to replace the relatively slow STATESON-algorithm with algorithms resulting from Compressed Sensing.

The latter recommendation is worked out in the next section with an example of an algorithm from Compressed Sensing.

5-2 Kalman Filtering for Compressed Sensing

Compressed Sensing (CS) is concerned with solving the optimisation problem

$$\min \|\hat{z}\|_0 \qquad \text{s.t.} \quad \sum_{i=1}^{k} \|y_i - H'\hat{z}\|_2^2 \leq \epsilon_{cs} \tag{5-1}$$

with

$$y_k = H'z_k + \zeta_k. \tag{5-2}$$

Unfortunately, this problem is non-convex. However, it has been shown that if H' has the "restricted isometry property" (RIP), then the solution can almost always be obtained by solving the convex relaxation

$$\min \|\hat{z}\|_1 \qquad \text{s.t.} \quad \sum_{i=1}^{k} \|y_i - H'\hat{z}\|_2^2 \leq \epsilon_{cs}. \tag{5-3}$$

Kanevsky, Carmi, Horesh and Gurfil [27], however, invented an algorithm to solve

$$\min_{\hat{z}_k} E_{z_k|\mathcal{Y}_k}[\|z_k - \hat{z}_k\|_2^2] \qquad \text{s.t.} \quad \|\hat{z}_k\|_1 \leq \epsilon'_{cs} \tag{5-4}$$

efficiently. The so-called CS-embedded Kalman Filter (CSKF) works as follows:

1. Set $\hat{z}^1 = \hat{z}_{k|k}$ and $P^1 = P_{k|k}$ (the posterior mean and covariance at time k).
2. **for** $\tau = 1, 2, ..., N_\tau - 1$ iterations **do**
3. $$\begin{aligned} \bar{H}_\tau &= \text{sign}(\hat{z}^\tau) \\ \hat{z}^{\tau+1} &= \left(I - \frac{P^\tau \bar{H}_\tau^\top \bar{H}_\tau}{\bar{H}_\tau P^\tau \bar{H}_\tau^\top + \sigma^2}\right)\hat{z}^\tau \\ P^{\tau+1} &= \left(I - \frac{P^\tau \bar{H}_\tau^\top \bar{H}_\tau}{\bar{H}_\tau P^\tau \bar{H}_\tau^\top + \sigma^2}\right)P^\tau \end{aligned} \tag{5-5}$$
4. **end for**
5. Set $\hat{z}_{k|k} = \hat{z}^{N_\tau}$ and $P_{k|k} = P^{N_\tau}$.

There could be multiple ways to implement this algorithm for finding the tip-sample interaction force in AFM. One way is as follows. Suppose that in Equation 5-2, the vectors are defined as

$$y_k = \begin{bmatrix} y(k) & y(k-1) & y(k-2) \\ y(k-1) & y(k-2) & y(k-3) \\ \vdots & \ddots & \vdots \\ y(k-N+3) & \dots & y(k-N+1) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \end{bmatrix} - \begin{bmatrix} u(k-1) & u(k-2) \\ u(k-2) & u(k-3) \\ \vdots & \vdots \\ u(k-N+2) & u(k-N+1) \end{bmatrix} \begin{bmatrix} b_{u,1} \\ b_{u,2} \end{bmatrix} \tag{5-6}$$

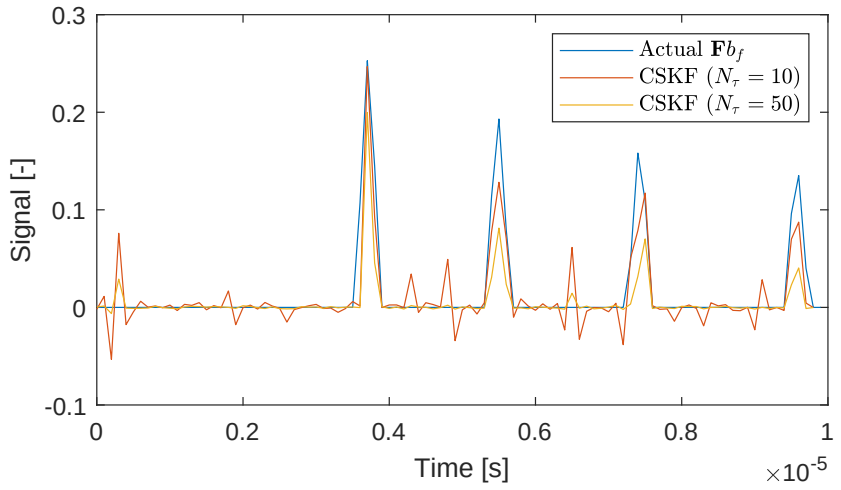


Figure 5-1: Actual $\mathbf{F}b_f = z_k$ (blue) and estimates using the CSKF-algorithm with $N_\tau = 10$ (red) and $N_\tau = 50$ (yellow). Definitions of Equations 5-6 through 5-9 are used.

and

$$z_k = \begin{bmatrix} F(k-1) & F(k-2) \\ F(k-2) & F(k-3) \\ \vdots & \vdots \\ F(k-N+2) & F(k-N+1) \end{bmatrix} \begin{bmatrix} b_{f,1} \\ b_{f,2} \end{bmatrix} \tag{5-7}$$

and

$$H' = I, \tag{5-8}$$

with y_k known and z_k the sparse vector to be reconstructed. In Step 1 of the CSKF-algorithm, $\hat{z}^1 = \hat{z}_{k|k} = y_k$ and the posterior covariance will be

$$P_{k|k} = \begin{bmatrix} * & * & 0 & 0 & \dots & 0 \\ * & * & * & 0 & \dots & 0 \\ 0 & * & * & * & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & * & * & * \\ 0 & 0 & \dots & 0 & * & * \end{bmatrix}. \tag{5-9}$$

Note that besides the main diagonal, also the super- and subdiagonal of $P_{k|k}$ will have values, since each element in z_k is correlated with its neighbour, due to its construction in Equation 5-7.

Simulations in Figure 5-1 show the performance. For simplicity, the posterior covariance is chosen to be $P_{k|k} = I$, $R_\nu = 3.05 \times 10^{-4}$ and $R_\eta = 0$. It can be seen that increasing N_τ has the same effect as increasing λ in the STATESON-algorithm or increasing λ_{DCP} in the data-driven approach. The computational time for 100 samples and $N_\tau = 50$ was 0.0136 seconds, while the STATESON-algorithm would require 1.34 seconds for calculating the tip-sample interaction over 100 samples.

Appendix A

Additional Results

A-1 System Identification

The validation signal (Figure A-2) was applied to the system 10 minutes and 33 seconds after the identification signal (Figure A-1) was applied. The validation signal had a duration of 1 second, while the identification signal had a duration of 2 seconds. A horizontal distribution in both spectra is desirable, since in that case each frequency of the cantilever is excited.

The singular values are plotted in Figure A-3 and zoomed in in Figure A-4. Measurement data were split in segments and R_{32} was iterated by combining the new measurement data with the former R_{32}. For this, the LTI System Identification Toolbox [42] is used. For determination of the right model order such that the cantilever dynamics are described satisfactorily, small sudden steps in the singular values should be found. Examples of promising orders are $n = \{6, 16, 32, 42, 48, ...\}$. Since a clear gap between successive singular values cannot be recognised, even with $s = 2000$ and 100 segments consisting of 20000 measurements each, it can be concluded that a linear system describing the full cantilever dynamics would have an infinite order. The increase of all singular values (except for some near $s = 2000$) per additional batch confirms this conclusion.

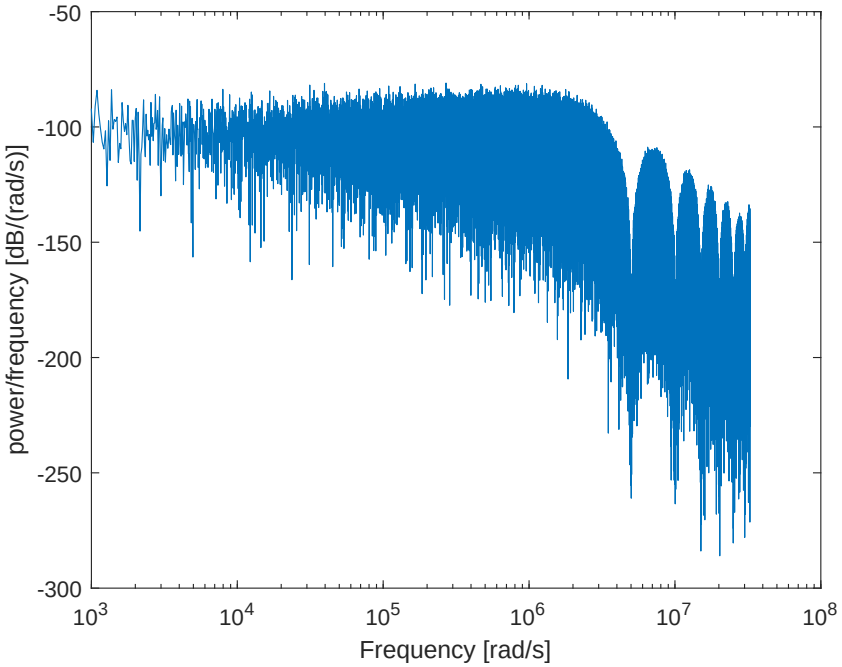


Figure A-1: Spectrum of a PRBS-input for identification.

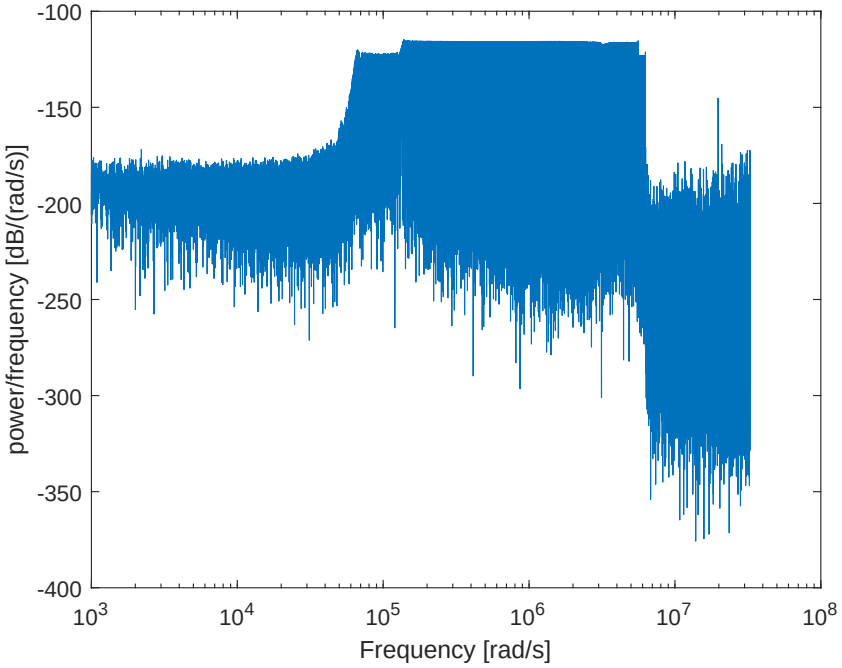


Figure A-2: Spectrum of a frequency sweep for model validation (with 10^7 samples).

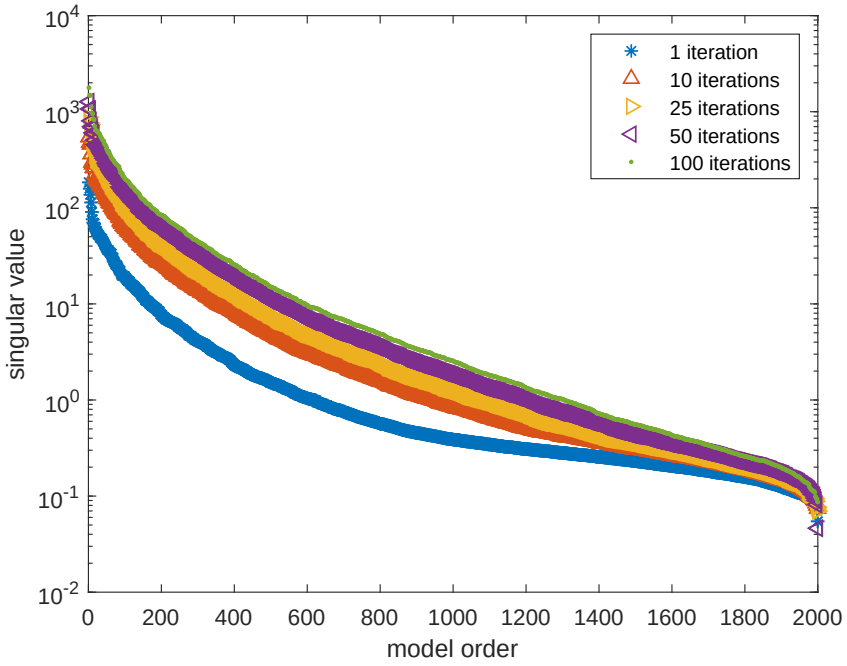


Figure A-3: Singular values of R_{32} after a certain amount of iterations, using $s = 2000$.

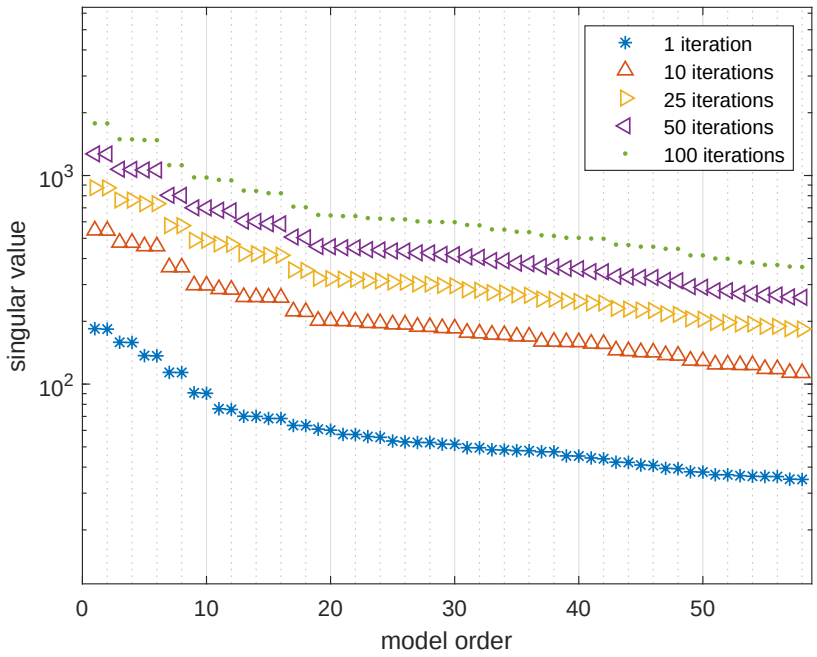


Figure A-4: Singular values of R_{32} after a certain amount of iterations, using $s = 2000$, from $n = 1$ to $n = 59$.

A-2 Model-Based Approach

The parameters used in the STATESON-algorithm are presented in Table A-1.

Table A-1: Parameters used in STATESON-algorithm. Values in A, B, B_2, C and D are rounded and may generate notable different results.

Parameter	Value	Size	Unit
A	$\begin{bmatrix} 0.8002 & 0.3699 \\ -0.2199 & 0.9114 \end{bmatrix}$	2×2	[-]
B	$\begin{bmatrix} 0.0715 & 0.0129 \end{bmatrix}^\top$	2×1	[-]
B_2	$\begin{bmatrix} 0.0715 & 0.0129 \end{bmatrix}^\top$	2×1	[-]
C	$\begin{bmatrix} 0.5577 & -0.1730 \end{bmatrix}$	1×2	[-]
D	0	1	[-]
N_T	300	1	[-]
p	1	1	[-]
q	∞	1	[-]
Q	2	1	V^2
R_ν	0.1	1	V^2
ϵ	$\begin{bmatrix} 0.1 & \dots & 0.1 \end{bmatrix}^\top$	$(N_T - 1) \times 1$	[-]
ε	0.2	1	V
framerate	0.5	1	frames/s
linerate	200	1	lines/frame
#iterations of Eq. 3-9 and 3-10	2	1	[-]

A-2-1 Quantitative Image Analysis

Steps 1 to 3 of Section 4-3-2 are shown schematically in Figure A-5.

The FRC-cures of the conventional amplitude plot and of the improved amplitude plot are shown in Figure A-6. Those of the images resulting from the STATESON-algorithm are presented in Figure A-7.

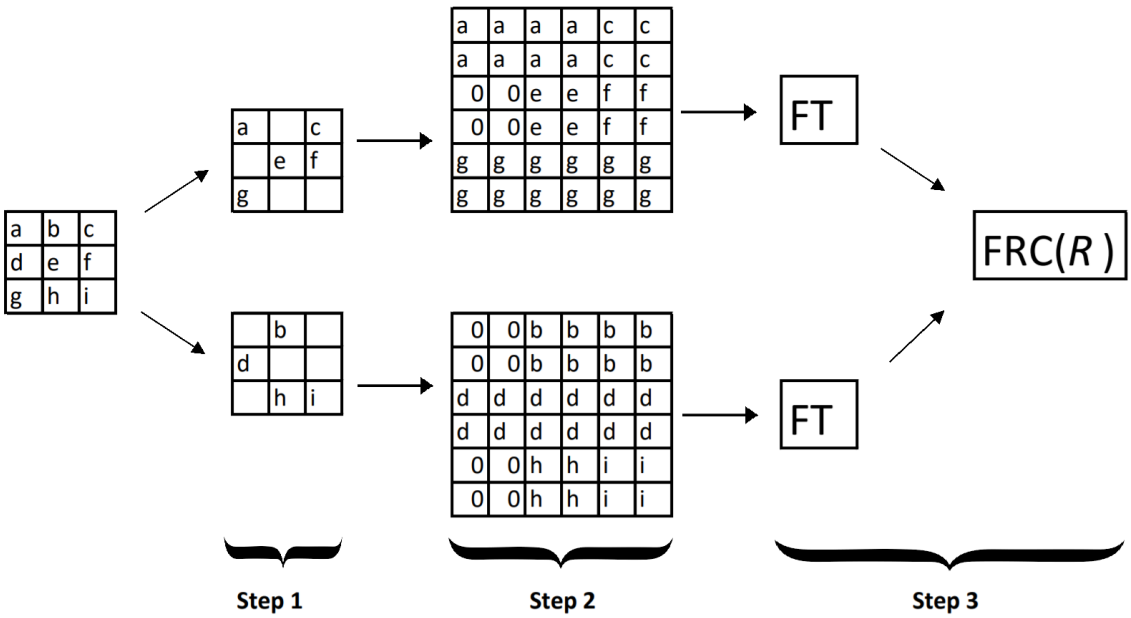


Figure A-5: Schematic presentation of steps 1 to 3. Pixel values a to i are randomly assigned to two new images. Then, the images are enlarged and empty pixels are filled with neighbouring pixel values. Afterwards, both images are Fourier transformed and the FRC(R) can be calculated.

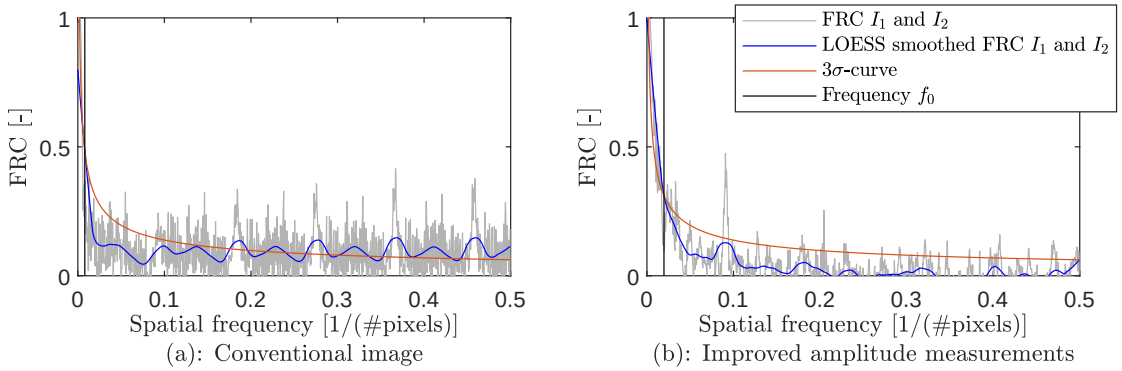


Figure A-6: FRC-curves (gray) as function of spatial frequency (Equation 4-6), the LOESS smoothed FRC (blue) and the σ_3-curves (red) for the conventional amplitude plot (a) ($Q_{\text{res}} = 4.136$ nm) and improved measurement technique for obtaining amplitude (b) ($Q_{\text{res}} = 1.685$ nm) (see results in Section 4-3-2).

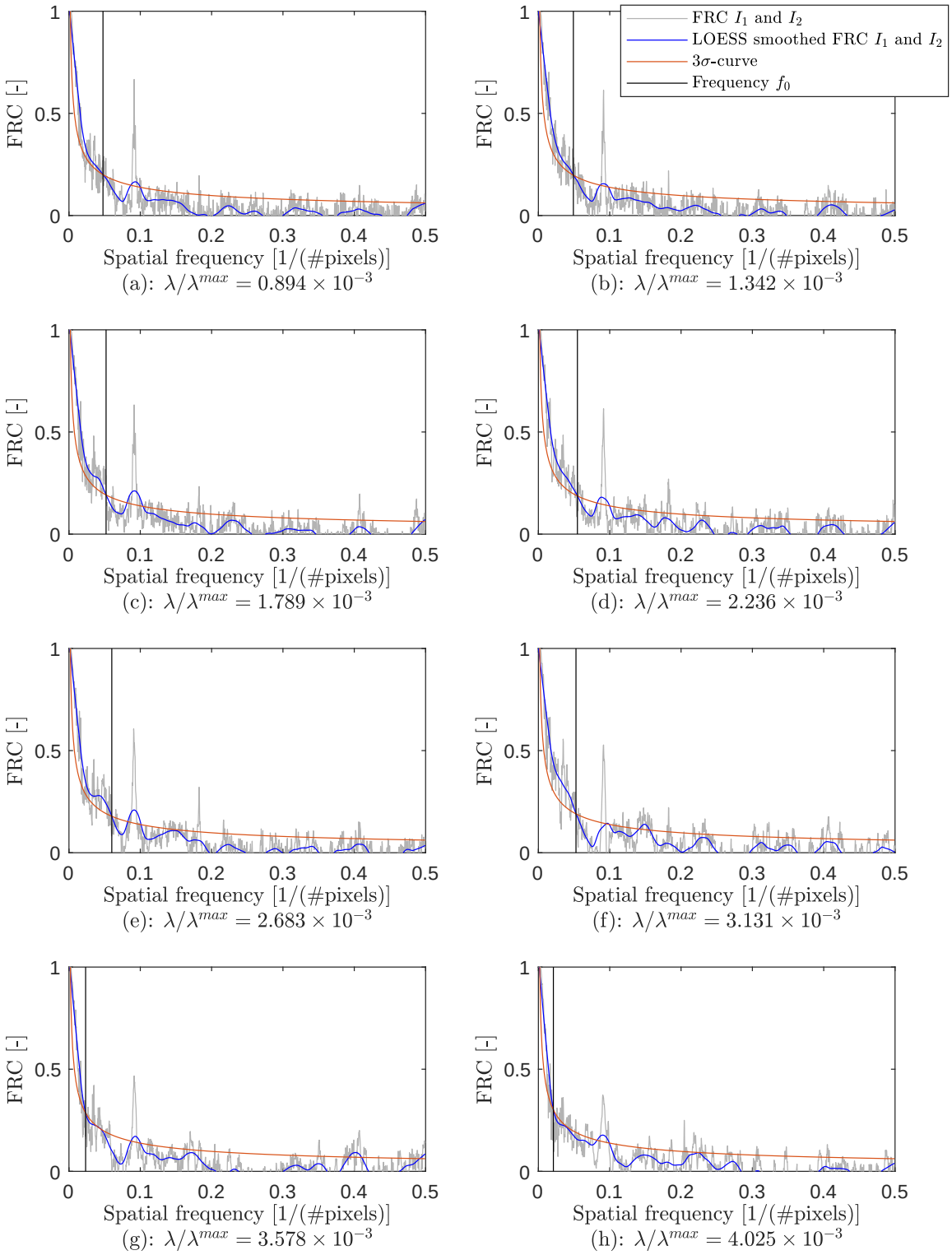


Figure A-7: FRC-curves (gray) as function of spatial frequency (Equation 4-6), the LOESS smoothed FRC (blue) and the σ_3-curves (red) for the STATESON plots with $\lambda/\lambda^{\text{max}} = \{0.894, 1.342, 1.789, 2.236, 2.683, 3.131, 3.578, 4.025\} \times 10^{-3}$ for (a-h) respectively ($Q_{\text{res}} = \{0.695, 0.676, 0.640, 0.607, 0.555, 0.629, 1.427, 1.579\}$ nm).

Appendix B

Pictures of the Setup

The setup is shown in Figures B-1 to B-4.

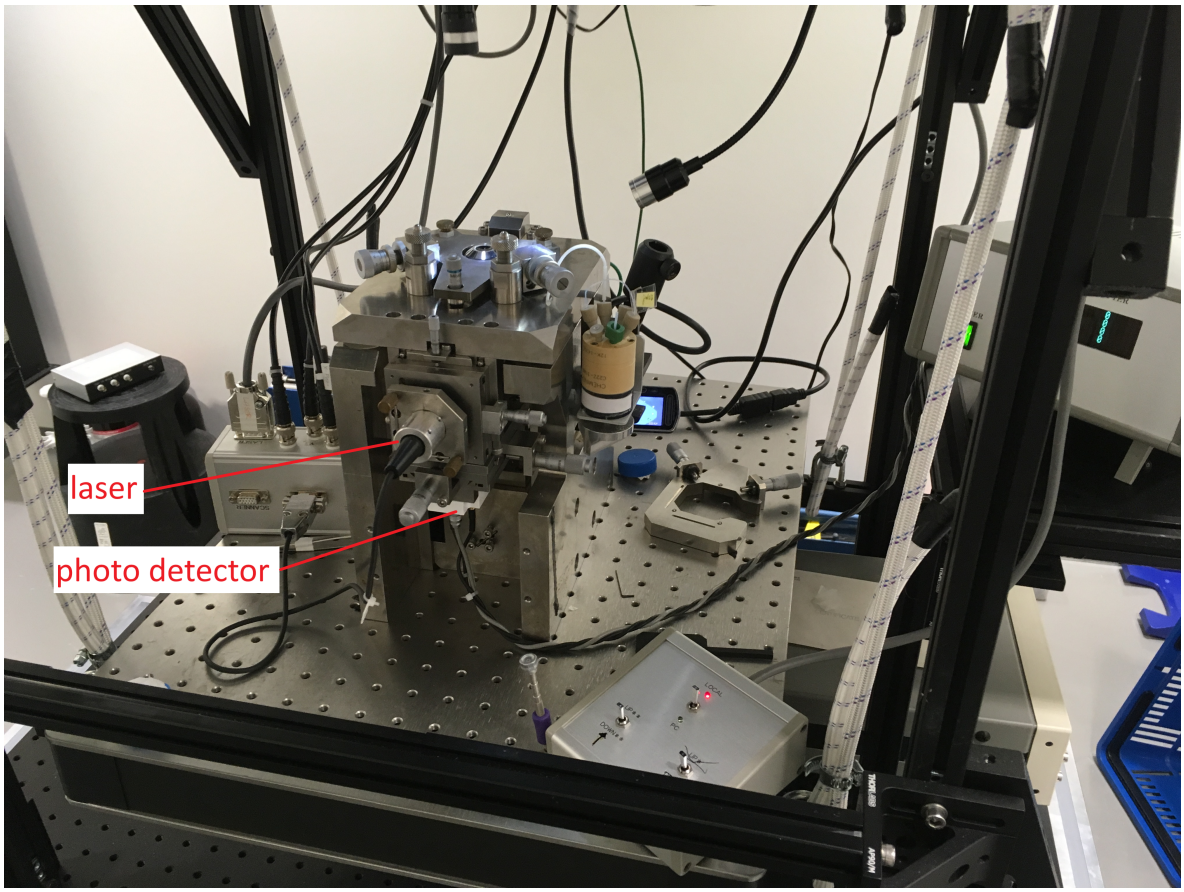


Figure B-1: The atomic force microscope.

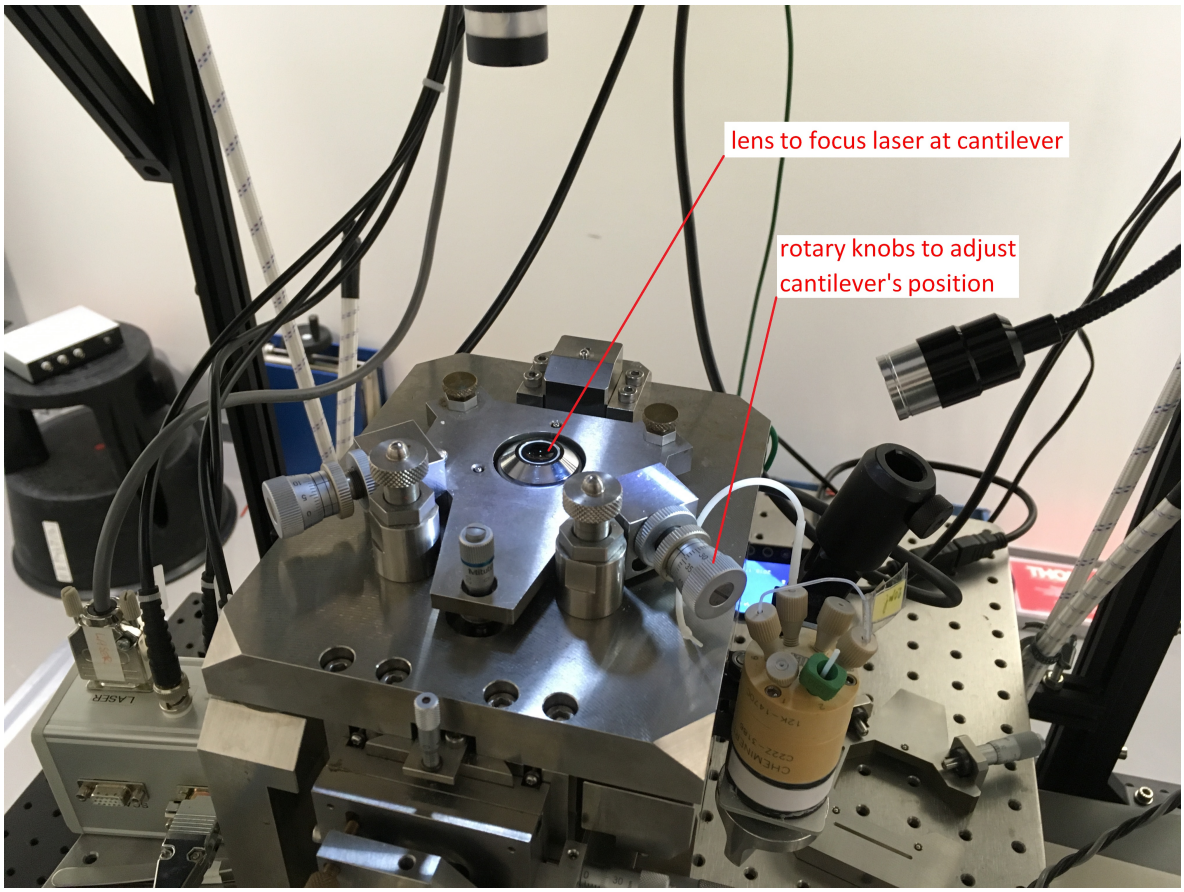


Figure B-2: The atomic force microscope closer.

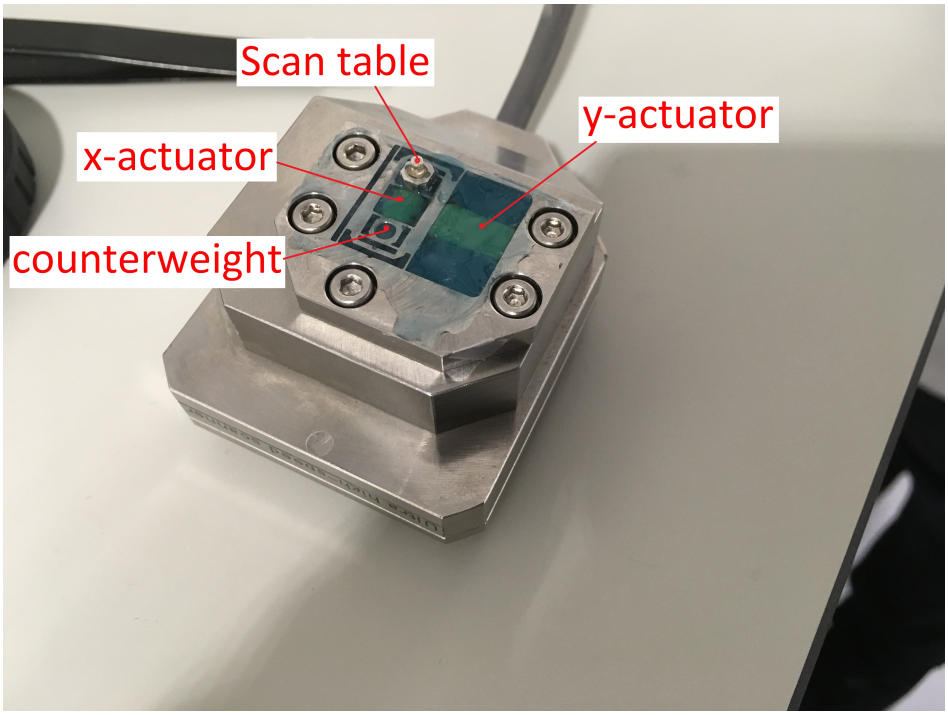


Figure B-3: Scan table with actuators.

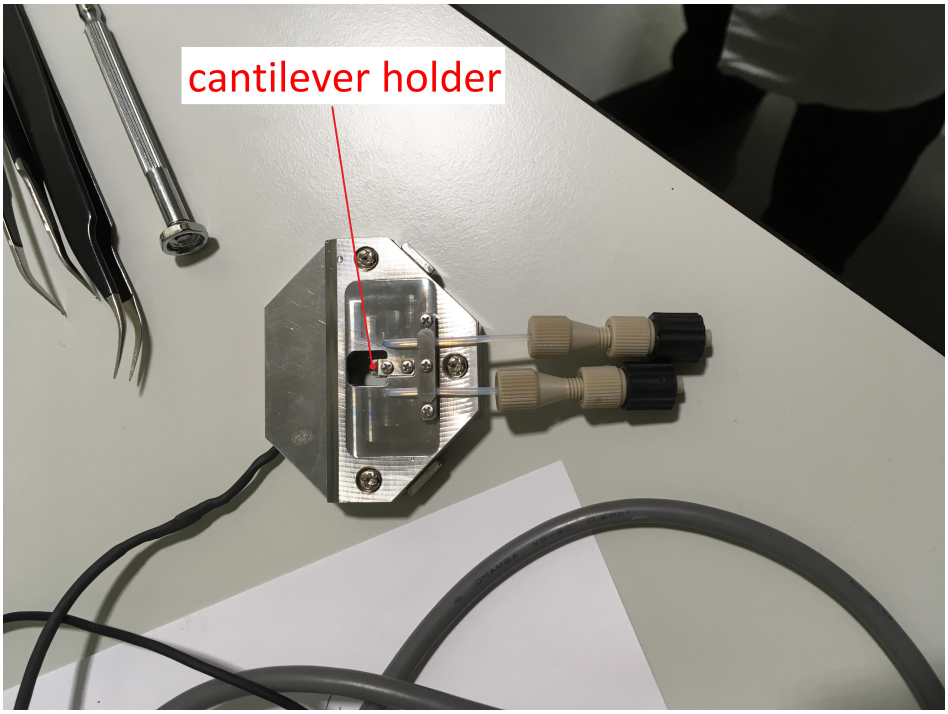


Figure B-4: Cantilever holder.

Appendix C

Software

In this appendix, the Matlab-code is presented. Table C-1 summarises the functionality of each file.

Table C-1: Matlab-files with corresponding applications.

File	Application
AFM_piezo2deflection.m	System identification using piezo input and cantilever deflection
AFM_thermalnoise.m	System identification using thermal motion of the cantilever
int2V.m	Conversion of raw (integer) data to Volts
load_AFM_meas.m	Loads AFM-measurements and scales those to Volts
AFM_STAT.m	Estimation of sample height as function of time, using STATESON
STAT.m	Estimation of t/s-interaction and deflection using STATESON
varepsbar.m	Calculates $\bar{\varsigma}_t$ as indicated in Equation 4-4
lmax.m	Calculates critical value for λ according to Equation 3-7
STATESON.m	Calculates t/s-interaction using Equation 3-9
STATESON2.m	STATESON.m, but using additional data of previous segment
STATESON_final.m	Calculates final estimates of t/s-interaction using Equation 3-11
STATESON_final2.m	Calculates final estimates using additional data of previous segment
im_generation.m	Converts estimated sample height to images
AFM_FRC.m	Generates two images out of one
image_splitf.m	Splits original image and enlarges the new images
AFM_DCP_sim.m	Simulates the data-driven approach of Section 3-3
AFM_DCP_exp.m	Experiment of the data-driven approach of Section 4-4
obj1.m	Objective function of Equation 3-18
obj2_3.m	Objective function of Equations 3-20 and 3-22
obj4.m	Objective function of Equaction 3-23
admmstat.m	Similar to STATESON.m, but using ADMM

C-1 System Identification

C-1-1 AFM_piezo2deflection.m

This program identifies the cantilever dynamics from the input voltage to the output voltage, according to Section 4-2-1. Use is made of LTI-Toolbox 2.4 [42]. Note that "dordpo_m" and "dmodpo_m" are the Matlab-files in the LTI-Toolbox, instead of the mex-files.

```
% Files needed:
% 'stream-20190320-103735.mat'      Identification data
% 'stream-20190320-104808.mat'      Validation data
clear all
close all

%% load data (u, y and ts)
load('stream-20190320-103735.mat') %load raw identification data
ts=single(RunSettings(1))*10^(-9);
u=double(int2V(D(2*10^6:8*10^6),RunSettings,DSettings)); u = u - mean(u);
y=double(int2V(Y(2*10^6:8*10^6),RunSettings,YSettings)); y = y - mean(y);
clear D Y

%% Determine order of the model
k=1;i=1;                        % counters
window = 20000; s=2000;         % #measurements per segment and size of Hankels
[Sn(:,i),Rnew]=dordpo_m(u(k:k+window-1),y(k:k+window-1),s); %Est. w/o. Rnew
k=k+window;
i=i+1;
while i < 101
    [Sn(:,i),Rnew]=dordpo_m(u(k:k+window-1),y(k:k+window-1),s,Rnew); %Est. ...
        w. Rnew
    k=k+window;
    i=i+1;
end
figure;semilogy(Sn(:,1),'*');           % Show singular values
hold on;semilogy(Sn(:,i-1),'.');

%% Start identification
n=188;                                  % Desired order of identified model
[Ae,Ce]=dmodpo_m(Rnew,n,'stable');      % Identify Ae and Ce
[Be,De]=dac2bd(Ae,Ce,u(1:window),y(1:window));  % Identify Be and De
sys=ss(Ae,Be,Ce,De,ts);                 % Compose state-space system

%% Load validation data (u, y)
load('stream-20190320-104808.mat')      % Load raw data
u=double(int2V(D(1:10^7),RunSettings,DSettings)); u = u - mean(u);
y=double(int2V(Y(1:10^7),RunSettings,YSettings)); y = y - mean(y);
clear D Y ye;

%% Validation
if n > 30                               % Prevent exceeding RAM-memory
    j=1;
    ye=[];
    window2 = 10^6/2;
    while j < length(u)-window2+2
        x0e = dinit(Ae,Be,Ce,De,u(j:j+window),y(j:j+window));
```

```
47          ye = [ye;dltisim(Ae,Be,Ce,De,u(j:j+window2-1),x0e)];
48          j=j+window2;
49      end
50  else
51      x0e = dinit(Ae,Be,Ce,De,u(1:window),y(1:window));
52      ye = dltisim(Ae,Be,Ce,De,u,x0e);
53  end
54  VAF=vaf(y(1:length(ye)),ye)
55
56  [cyye,freq]=mscohere(y(1:length(ye)),ye,[],[],[],1/ts); %Coherence
57  figure;semilogx(freq*2*pi,cyye);xlim([5*10^5,10^7]);
```

C-1-2 AFM_thermalnoise.m

This file identifies the cantilever dynamics according to Section 4-2-2. Also here, use is made of LTI-Toolbox 2.4 [42].

```
1  % Files needed:
2  % 'stream-20190115-111428.mat'      Identification data
3  % 'stream-20190320-120206.mat'      Data to find delay and scaling
4  clear all
5  close all
6
7  %% Load data (y and ts)
8  load('stream-20190115-111428.mat')  %load raw identification data
9  ts=single(RunSettings(1))*10^(-9);
10 y=double(int2V(Y(1:3*10^6),RunSettings,YSettings)); y = y - mean(y);
11 kc=0.15;                            %Estimated spring constant
12
13 %% Start identification
14 s=30;                               %size of Hankel matrices
15 [Sn,Rnew]=dordpo([],y,s);           %compute singular values
16
17 n=2;                                %desired order of identified system
18 [Ae,Ce,Ke]=dmodpo(Rnew,n);          %identify Ae, Ce and Ke
19 Be=Ke;
20 De=0;
21
22 %% Compose 2nd-order state-space system with spring constant kc
23 sysid=ss(Ae,Be,Ce,De,ts);
24 systf=tf(d2c(sysid));
25 invm=systf.denominator{1}(3)/kc;
26 systf2=tf([invm/-zero(systf) invm],systf.denominator);
27 sysid2=sysid/dcgain(sysid)/kc;     %identified system
28
29 %% Perform FFT over measurements
30 t=linspace(0,ts*(length(y)-1),length(y));
31 T=max(t)-min(t); N=length(t);
32 k=0:N-1;
33 omega=linspace(-N/T/2,(N/2-1)/T,N)*2*pi;
34 gf=fft(y');
35 gff=[gf(round(length(gf)/2)+1:end),gf(1:round(length(gf)/2))]/sum(y')...
36     *sum(10^7*y'.*double(T)/double(N));
37
```

```
%% Determine delay and scaling factor
[u,y,ts] = load_AFM_meas([1, 2000],'stream-20190320-120206.mat'); % Load ...
    scanning data (u,y and ts)
t=linspace(0,ts*(length(u)-1),length(u))';y=y-mean(y(1:5*10^2));

tau_opt=0;
for i=1:5                    % Alternate fact and tau_opt to find optimum
ysim = lsim(sysid2,u);
tau_opt = phaseopt(y(1:1000),ysim(1:1000));
fact = (ysim(1+tau_opt:10^3+tau_opt)'*ysim(1+tau_opt:10^3+tau_opt))\...
    ysim(1+tau_opt:10^3+tau_opt)'*y(1:10^3);
sysid2=sysid2*fact;
end
sys=sysid2*1.1;              % Apply factor 1.1 as compensation for ...
    presence of tip-sample interaction in scanning data
```

C-1-3 int2V.m

This function provides the measurements in Volts out of raw (integer) data.

```
% Inputs:
% -D            Measured quantity as integer
% -Runsettings  The settings: [sample time in ns; #bits; maximum integer ...
    value]
% -Dsettings    The settings: [Range in V; Offset in V]
%
% Output:
% -V            Measured quantity in Volts
function V = int2V(D, RunSettings, DSettings)
V=(double(D)*DSettings(1))/double(RunSettings(3))+DSettings(2);
end
```

C-1-4 load_AFM_meas.m

This file loads the AFM-measurements and scales the measurements to voltages.

```
% Inputs:
% -index    Range of measurements: [startindex; stopindex]
% -file     The file with raw measurement data
%
% Outputs:
% -u        The measured cantilever input in Volts
% -y        The measured cantilever output in Volts
% -ts       The sampling time
function [u,y,ts] = load_AFM_meas(index,file)
load(file);
ts=single(RunSettings(1))*10^(-9);
u=double(int2V(D(round(index(1)):round(index(2))),RunSettings,DSettings));
u = u - mean(u);
y=double(int2V(Y(round(index(1)):round(index(2))),RunSettings,YSettings));
y = y - mean(y);
```

```
16 end
```

C-2 Model-Based Approach

C-2-1 AFM_STAT.m

This is the main file. Sample height is estimated as function of time. It requires installation of CVX [43] to run appropriately.

```
% Files needed:
% 'stream-20190320-120206.mat'      Scanning data
% 'AFM_STAT3_4_tauopt_sys2.mat'     Identification: tau_opt, sys, n
clear all
close all

%% Load data (u, y, ts, t, tau_opt, sys, n)
[u,y,ts] = load_AFM_meas([1, 2*10^7],'stream-20190320-120206.mat');
t=linspace(0,ts*(length(u)-1),length(u))';
y=y-mean(y(1:5*10^2));
load('AFM_STAT3_4_tauopt_sys2.mat'); %Load identified system

%% Run STATESON
Rnu = 0.1;              % Estimation of the measurement noise
Q = 2;                  % Estimation of magnitude variance of pulses Fh.
T = 300;                % Number of measurements per STATESON evaluation
lambdafactor = 0.012;   % Factor for calculating lambda
epsfactor = 0.1;        % Factor for epsilon.
err = 0.2;              % Minimum magnitude of pulses

tic;
[Fhhat,yhat] = STAT(u(1+tau_opt:end),y(1:end-tau_opt),sys,n,Rnu,Q,...
    T,lambdafactor,epsfactor,err);
Tcomp = toc;

%% Split signal up in oscillation periods
N2=length(yhat);
j=1;
clear k
for i=1:N2-1
    if yhat(i) < 0 && yhat(i+1) ≥ 0
        k(j)=i;
        j=j+1;
    end
    if yhat(i) > 0 && yhat(i+1) ≤ 0
        k(j)=i;
        j=j+1;
    end
end
period=round(2*k(end)/length(k))+1;          % #measurements per oscillation

%% Estimates of sample height
clear hhat5
for i=1:floor(N2/period)
```

```
45      int(i)=sum(Fhhat((i-1)*period+1:i*period));
46      hhat5((i-1)*period+1) = min(yhat((i-1)*period+1:i*period)) +...
47          0.02*int(i); % Estimated sample height
48  end
```

C-2-2 STAT.m

The general STATESON-procedure is set out in this file. For the first data segment, there is no constraint on the initial state. For subsequent segments (i.e. if $i \not< 2$), the constraint on the initial state is added.

```
1   % Inputs:
2   % -u2            Cantilever input (corresponding to Equation 4-3 in report)
3   % -y2            Measured cantilever deflection
4   % -sys           Identified (state-space) system
5   % -n             Order of identified system
6   % -Rnu           Estimated variance of measurement noise
7   % -Q             Estimated variance of pulses in tip-sample interaction
8   % -T             Number of measurements per STATESON evaluation
9   % -lambdafactor Factor for calculating lambda
10  % -epsfactor     Factor for epsilon
11  % -err           Minimum of estimated pulses
12  %
13  % Outputs:
14  % -Fhhat1        Estimated tip-sample interaction
15  % -yhat1         Estimated cantilever deflection
16  function [Fhhat1,yhat1] =...
17      STAT(u2,y2,sys,n,Rnu,Q,T,lambdafactor,epsfactor,err)
18
19  i=1;
20  while i*T < length(u2)
21      u(:,i) = u2((i-1)*T+1:i*T);
22      y(:,i) = y2((i-1)*T+1:i*T);
23      i=i+1;
24  end
25  NT = i-1;         % Number of data segments
26
27  ib=100;           % Data segments per backup file
28  for i=1:NT
29      disp(['Performing algorithm...', num2str(i),'/',num2str(NT)])
30      vepsbar(i,1)=varepsbar(u(:,i),y(:,i),sys.A,sys.B,sys.C,Rnu);
31      lambdamax(i,1)=lmax(sys.A,sys.C,sys.B,Rnu,Q,vepsbar,T);
32      lambda(i,1)=lambdafactor*sqrt(norms(Rnu,1)/norms(Q,1))*lambdamax(i,1);
33
34      alpha = ones(T-1,1);
35      eps=epsfactor*ones(T-1,1);
36      if i<2
37          for j=1:2
38              [X(:,n*(i-1)+1:n*i),Fhhat(:,i)] =...
39                  STATESON(y(:,i),u(:,i),sys.A,sys.B,sys.C,sys.B,Rnu,...
40                  Q,alpha,lambda(i,1),1);
41              alpha = 1./(eps+norms(inv(sqrtm(Q))*Fhhat(:,i),1,2));
42              lambda(i,1) = 0.2*lambda(i,1);
43          end
```

```
        [X(:,n*(i-1)+1:n*i),Fhhat(:,i)] =...
            STATESON_final(y(:,i),u(:,i),sys.A,sys.B,sys.C,sys.B,Rnu,...
            Fhhat(:,i),err);
        yhat(:,i) = sys.C*X(:,n*(i-1)+1:n*i)';
    else
        for j=1:2
            [X(:,n*(i-1)+1:n*i),Fhhat(:,i)] = STATESON2(y(:,i),...
                u(:,i),sys.A,sys.B,sys.C,sys.B,Rnu,Q,alpha,...
                lambda(i,1),1,X(end,n*(i-2)+1:n*(i-1)),u(end,i-1),0);
            alpha = 1./(eps+norms(inv(sqrtm(Q))*Fhhat(:,i),1,2));
            lambda(i,1) = 0.2*lambda(i,1);
        end
        [X(:,n*(i-1)+1:n*i),Fhhat(:,i)] = STATESON_final2(y(:,i),...
            u(:,i),sys.A,sys.B,sys.C,sys.B,Rnu,Fhhat(:,i),err,...
            X(end,n*(i-2)+1:n*(i-1)),u(end,i-1),0);
        yhat(:,i) = sys.C*X(:,n*(i-1)+1:n*i)';
    end
    if i == ib
        save('backup2','yhat','Fhhat')
        ib=ib+100;
    end
end
Fhhat1=vec([Fhhat;zeros(1,NT)]);
yhat1=vec(yhat);
end
```

C-2-3 varepsbar.m

This function calculates $\bar{\varsigma}_t$ as indicated in Equation 4-4.

```
% Inputs:
% -u2           Cantilever input (corresponding to Equation 4-3 in report)
% -y            Measured cantilever deflection
% -A            Identified A-matrix
% -B            Identified B-matrix
% -C            Identified C-matrix
% -Rnu          Estimated variance of measurement noise
%
% Output:
% -varepsbar    Corresponding to Equation 3-5 in repo
function [varepsbar] = varepsbar(u2,y,A,B,C,Rnu)
N=length(y); [nc,n]=size(C);
A1(1:n,1:n)=A^0;
for t=2:N
    clear eps1
    A1(n*(t-1)+1:n*t,1:n)=A^(t-1);
    for r=1:t-1
         eps1(1:n,r)=A1(n*(t-r-1)+1:n*(t-r),1:n)*B*u2(r)+A1(n*(t-1)+...
             1:n*(t),1:n)*zeros(n,1);
    end
    eps(1:nc,t) = inv(sqrtm(Rnu))*(y(t)-C*sum(eps1,2));
end
varepsbar = sum(abs(eps));
end
```

C-2-4 lmax.m

This function calculates the critical value for λ for which no pulses are detected, according to Equation 3-7.

```
% Inputs:
% -A            Identified A-matrix
% -C            Identified C-matrix
% -G            Identified G-matrix (corresponding to B_2 in report)
% -Rnu          Estimated variance of measurement noise
% -varepsbar    Error vector (corresponding to Equation 3-6 in report)
% -T            Number of measurements in data segment
%
% Output:
% -lmax         Critical value for lambda for which no pulses are detected
function [lmax] = lmax(A,C,G,Rnu,Q,varepsbar,T)
for k=1:T-1
    clear lambda1
    lambda1=0;
    for t=k+1:T
        lambda1=lambda1+(inv(sqrtm(Rnu))*C*A^(t-k-1)*G*sqrtm(Q))*varepsbar;
    end
    lambda(k)=norms(2*lambda1,inf);
end
lmax=norms(lambda,inf);
end
```

C-2-5 STATESON.m, STATESON2.m

The following files produce intermediate estimates for the states and tip-sample interactions using Equation 3-9.

```
% Inputs:
% -Y        Measured cantilever deflection
% -u2       Cantilever input (corresponding to Equation 4-3 in report)
% -A        Identified A-matrix
% -B        Identified B-matrix
% -C        Identified C-matrix
% -G        Identified G-matrix (corresponding to B_2 in report)
% -Rnu      Estimated variance of measurement noise
% -Q        Estimated variance of pulses in tip-sample interaction
% -w        Weights (corresponding to alpha in report)
% -lambda   Regularisation term (corresponding to lambda in report)
% -p        Norm indication
%
% Outputs:
% -X        Estimated states of cantilever
% -Fhhat    Estimated tip-sample interaction
function [X Fhhat] = STATESON(Y,u2,A,B,C,G,Rnu,Q,w,lambda,p)
N=length(Y);
n=size(C,2);
cvx_begin quiet
```

```
21    variables X(N,n) Fhhat(N-1,1)
22    minimize( sum((inv(sqrtm(Rnu))*(Y-(C*X')')).^2)+lambda*norms(...
23        inv(sqrtm(Q))*diag(w)*Fhhat,p,1))
24    subject to
25    X(2:end,:) == ((A*[X(1:end-1,:)'])+(B*u2(1:N-1)')+(G*Fhhat'))'
26  cvx_end
27  end
```

```
1  % Inputs (in addition to STATESON.m):
2  % -Xprev        Final estimated state of previous segment
3  % -u2prev       Final input of previous segment
4  % -Fhhatprev    Final estimated tip-sample interaction of previous segment
5  %
6  % Outputs similar to STATESON.m
7  function [X Fhhat] =...
8      STATESON2(Y,u2,A,B,C,G,Rnu,Q,w,lambda,p,Xprev,u2prev,Fhhatprev)
9
10 N=length(Y);
11 n=size(C,2);
12 cvx_begin quiet
13   variables X(N,n) Fhhat(N-1,1)
14   minimize( sum((inv(sqrtm(Rnu))*(Y-(C*X')')).^2)+lambda*norms(...
15       inv(sqrtm(Q))*diag(w)*Fhhat,p,1))
16   subject to
17   X(2:end,:) == ((A*[X(1:end-1,:)'])+(B*u2(1:N-1)')+(G*Fhhat'))'
18   X(1,:) == ((A*Xprev')+(B*u2prev')+(G*Fhhatprev'))'
19 cvx_end
20 end
```

C-2-6 STATESON_final.m, STATESON_final2.m

The following files produce final estimates for the states and tip-sample interactions using Equation 3-11.

```
1  % Inputs:
2  % -Y         Measured cantilever deflection
3  % -u2        Cantilever input (corresponding to Equation 4-3 in report)
4  % -A         Identified A-matrix
5  % -B         Identified B-matrix
6  % -C         Identified C-matrix
7  % -G         Identified G-matrix (corresponding to B_2 in report)
8  % -Rnu       Estimated variance of measurement noise
9  % -Fhhat1    Previously estimated tip-sample interaction over the same segment
10 % -err       Threshold value for Fhhat1
11 %
12 % Outputs:
13 % -X         Estimated states of cantilever
14 % -Fhhat     Estimated tip-sample interaction
15 function [X Fhhat] = STATESON_final(Y,u2,A,B,C,G,Rnu,Fhhat1,err)
16 N=length(Y); n=size(C,2);
17 i=1; k=[];
18 for t=1:N-1
```

```
    if Fhhat1(t) < err
        k(i)=t;
        i=i+1;
    end
end

cvx_begin quiet
  variables X(N,n) Fhhat(N-1,1)
  minimize( sum((inv(sqrtm(Rnu))*(Y-(C*X')')).^2))
  subject to
  X(2:end,:) == ((A*[X(1:end-1,:)']+(B*u2(1:N-1)')+(G*Fhhat'))'
  Fhhat(k) == 0
cvx_end
end
```

```
% Inputs (in addition to STATESON_final.m):
% -Xprev        Final estimated state of previous segment
% -u2prev       Final input of previous segment
% -Fhhatprev    Final estimated tip-sample interaction of previous segment
%
% Outputs similar to STATESON_final.m
function [X Fhhat] =...
    STATESON_final2(Y,u2,A,B,C,G,Rnu,Fhhat1,err,Xprev,u2prev,Fhhatprev)
N=length(Y); n=size(C,2);
i=1; k=[];
for t=1:N-1
    if Fhhat1(t) < err
        k(i)=t;
        i=i+1;
    end
end

cvx_begin quiet
  variables X(N,n) Fhhat(N-1,1)
  minimize( sum((inv(sqrtm(Rnu))*(Y-(C*X')')).^2))
  subject to
  X(2:end,:) == ((A*[X(1:end-1,:)']+(B*u2(1:N-1)')+(G*Fhhat'))'
  X(1,:) == ((A*Xprev')+(B*u2prev')+(G*Fhhatprev'))'
  Fhhat(k) == 0
cvx_end
end
```

C-2-7 im_generation.m

This file converts estimated sample height as function of time to height as function of location. In other words, height profiles are generated.

```
% Files needed:
% 'test%_1.mat' Contains hhat2, hhat3, hhat32, hhat4, hhat5 and period for ...
    first part (``%''-sign between 1 and 8)
% 'test%_2.mat' Contains hhat2, hhat3, hhat32, hhat4, hhat5 and period for ...
    second part (``%''-sign between 1 and 8)
```

```
4
5  linerate=200;                %Lines per frame
6  framerate=0.5;               %Frames per second
7  T=300;                       %Measurements per segment
8  load('test2_1')              %Load estimated sample heights
9  line=double(floor(1/linerate/framerate/period/ts-192)); %Periods per line
10 delay=round(80*T/period);    %Connect data of test%_1 and test%_2 correctly ...
       (=round(55*T/period)) for test1_%
11 start=1350*period;           %Start at the beginning of a scan line.
12
13 clear im5
14 for j=1:90                   %j=1:100 for test1_%
15     im5(j,:)=hhat5((j-1)*period*line+1+start:period:j*period*line+start);
16 end
17
18 load('test2_2')
19 for jj=91:200 %jj=101:200 for test1_%
20     im5(jj,:)=hhat5((jj-j-1)*period*line+1+period*delay+start:period:...
21         (jj-j)*period*line+period*delay+start);
22 end
```

C-3 Quantitative Image Analysis

C-3-1 AFM_FRC.m

This is the main file for generating two images out of one, according to Steps 1 and 2 of Section 4-3-2.

```
1  % Files needed:
2  % 'im5_%.mat'       Contains generated image im5 (``%''-sign between 1 and 8)
3  % 'movie45.mat'     Contains conventional amplitude plot
4
5  %% Import image and normalise
6  load('im5_1');I = im5(end:-1:1,round(length(im5(1,:))/2):-1:1); %Height ...
       reconstruction using STATESON
7  %load('movie45.mat');I = movie45.Ch2.sImage(end:-1:1,end:-1:1,1)'*-1; ...
       %Conventional amplitude plot
8  I=I-min(min(I));        %Set minimum of image to 0.
9  I=I/sum(sum(abs(I)));   %Normalise image.
10
11 %% Set parameters & perform splitting algorithm
12 enlarge=[3000,3000];
13 [I1,I2]=image_splitf(I,enlarge);
14
15 %% imwrite
16 imwrite(I1/max(max(I1)),'I5_1_1.png','PNG');
17 imwrite(I2/max(max(I2)),'I5_1_2.png','PNG');
```

C-3-2 image_splitf.m

This file splits the original image up in two images and enlarges the new images, according to step 1 and 2 of Section 4-3-2.

```
% Inputs:
% -I            Original image
% -enlarge      Pixel size of output images [a, b]
%
% Outputs:
% -I1           Output image 1
% -I2           Output image 2
function [I1,I2] = image_splitf(I,enlarge)
pixels=size(I);
bin1=logical(round(rand(size(I)))); %Create random binary matrices
bin2=logical(1-bin1);
I11=I.*bin1;                %Split the image
I22=I.*bin2;

I1(round(enlarge(1)/pixels(1)*[0:pixels(1)-1])+1,round(...
    enlarge(2)/pixels(2) *[0:pixels(2)-1])+1)=I11(1:pixels(1),1:pixels(2));
I2(round(enlarge(1)/pixels(1)*[0:pixels(1)-1])+1,round(...
    enlarge(2)/pixels(2) *[0:pixels(2)-1])+1)=I22(1:pixels(1),1:pixels(2));

[n1,n2]=size(I1);
for i=n1*n2:-1:1
    ii=1;jj=1;
    if I1(floor((i-1)/n2)+1,i-floor((i-1)/n2)*n2)≠0
        while floor((i-1)/n2)+1+ii≤n1 && I1(...
                floor((i-1)/n2)+1+ii,i-floor((i-1)/n2)*n2)==0

            ii=ii+1;
        end
        while i-floor((i-1)/n2)*n2+jj≤n2 && I1(...
                floor((i-1)/n2)+1,i-floor((i-1)/n2)*n2+jj)==0

            jj=jj+1;
        end
        I1(floor((i-1)/n2)+1:floor((i-1)/n2)+...
            min(ii,round(enlarge(1)/pixels(1))+1),...
            i-floor((i-1)/n2)*n2:i-floor((i-1)/n2)*n2+jj-1)=...
            I1(floor((i-1)/n2)+1,i-floor((i-1)/n2)*n2);
    end
end
for i=n1*n2:-1:1
    ii=1;jj=1;
    if I2(floor((i-1)/n2)+1,i-floor((i-1)/n2)*n2)≠0
        while floor((i-1)/n2)+1+ii≤n1 && I2(...
                floor((i-1)/n2)+1+ii,i-floor((i-1)/n2)*n2)==0

            ii=ii+1;
        end
        while i-floor((i-1)/n2)*n2+jj≤n2 && I2(...
                floor((i-1)/n2)+1,i-floor((i-1)/n2)*n2+jj)==0

            jj=jj+1;
```

```
52          end
53          I2(floor((i-1)/n2)+1:floor((i-1)/n2)+...
54              min(ii,round(enlarge(1)/pixels(1))+1),...
55              i-floor((i-1)/n2)*n2:i-floor((i-1)/n2)*n2+jj-1)=...
56              I2(floor((i-1)/n2)+1,i-floor((i-1)/n2)*n2);
57      end
58  end
59  end
```

C-4 Data-Driven Approach

C-4-1 AFM_DCP_sim.m

This file is used for the simulations in Section 3-3. Lines 1 to 64 are used to simulate the cantilever and lines 64 to 147 are used to apply the data-driven algorithm.

```
1  clearvars
2  close all
3
4  %% Cantilever dynamics
5  scaling = 0.030503;
6  Ts1 = 1/(600*10^6);              %Sampling time (equivalent to 1/Ts Hz)
7  wres = 500000*2*pi;              %Resonance frequency of cantilever [rad/s]
8  zeta = 0.1;                      %Damping factor of cantilever
9
10 a1=1; a2= 2*zeta*wres; a3= wres^2; b1=scaling*10^14;
11 sysc = tf(b1,[a1 a2 a3]);        %Continuous system
12 sys = c2d(ss(sysc),Ts1);         %Discrete system (with high sampling ...
       frequency)
13 A=sys.A; B=sys.B; C=sys.C; D=sys.D;
14 G=B;
15
16 %% Input and output generation of system
17 Rnu=0.01*scaling;                %Variance of measurement noise
18 Q=10*scaling;                    %Estimated variance of pulses in ...
       t/s-interaction
19 c=3/scaling;                     %Factor for tip-sample interaction
20
21 tfinal = 1*10^-5;                %Simulation time
22 t1=[0:Ts1:tfinal]'; N1=length(t1);
23 [n,¬]=size(A);                   %Order of the system
24
25 nu1=wgn(N1,1,Rnu,'linear');      %Simulated measurement noise
26 uc1=sin(wres*t1);                %Simulated input
27
28 x(1:n,1) = zeros(n,1);
29 for i=1:round(N1*1.9/8)          %Construct sample height
30     h(i,1)=-40*scaling;
31 end
32 for i=round(N1*1.9/8)+1:round(N1*3.5/8)
33     h(i,1)=-20*scaling;
34 end
35 for i=round(N1*3.5/8)+1:round(N1*5.5/8)
```

```
36      h(i,1)=-20*scaling;
37  end
38  for i=round(N1*5.5/8)+1:round(N1*7/8)
39      h(i,1)=-10*scaling;
40  end
41  for i=round(N1*7/8)+1:N1
42      h(i,1)=-10*scaling;
43  end
44  for i=1:N1                          %Construct tip-sample interaction
45      x(1:n,i+1)=A*x(1:n,i)+B*uc1(i);
46      y1(i,1)=C*x(1:n,i)+D+nu1(i);
47      if C*x(1:n,i)+D > h(i)
48          Fh1(i,1) = 0;
49      else
50          Fh1(i,1) = c*(h(i)- (C*x(1:2,i)+D));
51      end
52      x(1:n,i+1)=A*x(1:n,i)+B*uc1(i)+G*Fh1(i);
53      y1(i,1)=C*x(1:n,i)+D+nu1(i);
54  end
55
56  %% Use lower sampling rate for STATESON-algorithm
57  clear A a1 a2 a3 B b1 c C D i n sys wres x zeta
58  Ts=1/(10*10^6);
59  t=[0:Ts:tfinal]'; N=length(t);
60
61  for i=1:N
62      Fh(i,1)=Fh1(round((i-1)*Ts/Ts1)+1);
63  end
64
65  %% Cantilever dynamics in state-space and in output-error model.
66  sys = c2d(ss(sysc),Ts);             %Discrete system with lower sampling rate
67  A=sys.A; B=sys.B; C=sys.C; D=sys.D;
68  sysoe=idpoly(sys);                  %Rewrite the system to a transfer function
69  par=getpvec(sysoe); bu=par(1:round(length(par)/2+0.1));
70  a=[1;par(round(length(par)/2+0.1)+1:length(par))]; %Obtain a_1, a_2, ...
        b_{u,1} and b_{u,2} according to Equation 3-15 in the report
71
72  %% Rewrite OE in desired form (Equation 3-16 in the report)
73  y2=lsim(sysoe,uc+Fh,t)+nu;          %Simulate measured cantilever deflection
74  Y=hankel(y2(end:-1:length(a)),y2(length(a):-1:1));
75  Uh=hankel(uc(end-length(a)+length(bu):-1:length(bu)),uc(length(bu):-1:1));
76  Y2=Y*a-Uh*bu;
77
78  Fhh=hankel(Fh(end:-1:length(bu)),Fh(length(bu):-1:1));
79  Fhhbu=Fhh*bu;                       %Validation data (in reality unknown)
80
81  %% Start Optimisation (Step 1)
82  opt=optimset('Display','off','MaxIter',10^6,'MaxFunEvals',10^6,'TolX',...
83      10^-8,'TolFun',10^-8);
84  bf0=randn(length(bu),1); f0=1*randn(N-length(a)+length(bu),1); x0=[f0;bf0];
85
86  x=fminunc(@(x)obj1(Y2,x,bu),x0,opt);
87  f=x(1:end-length(bu)); bf=x(end-length(bu)+1:end);
88  F=hankel(f(1:end-length(bu)+1),f(end-length(bu)+1:end));
89  M=[Y2 , F; bf , 1000*eye(length(bu))];
90  [U,¬,V]=svd(M);U1=U(:,1:length(bu));V1=V(:,1:length(bu));S1=svd(M);
91
```

```
%% DCP (Step 2)
i=1;der=-100;k=1;
while abs(der) > 10^-3
    [x,fval(i,k)]=fminunc(@(x)obj2_3(Y2,x,0,U1,V1,bu),...
        randn(size(x)),opt);
    f=x(1:end-length(bu)); bf=x(end-length(bu)+1:end);
    F=hankel(f(1:end-length(bu)+1),f(end-length(bu)+1:end));
    M=[Y2 , F; bf , 1000*eye(length(bu))];
    [U,¬,V]=svd(M);U1=U(:,1:length(bu));V1=V(:,1:length(bu));S1=svd(M);
    if i > 7
        der = fval(i,k)-fval(i-1,k);
    end
    i=i+1;
end

%% Add 1-norm (Step 3)
for k=2:2:40
    lambda2=0.02*(k-1);          %lambda varied between 0.02 and 0.78
    [x,fval2,exitflag(k)]=fminunc(@(x)obj2_3(Y2,x,lambda2,U1,...
        V1,bu),randn(size(x)),opt);
    f=x(1:end-length(bu)); bf=x(end-length(bu)+1:end);
    F=hankel(f(1:end-length(bu)+1),f(end-length(bu)+1:end));

    % Final step (Step 4)
    err=max(abs(Y2))*10^-2;
    ind2=find(abs(F*bf/1000)>err);
    for j=2:length(bu)
        ind2=[ind2;ind2+1];
    end
    ind=unique(sort(ind2(ind2>0)));

    x2=[zeros(length(ind),1);bf];
    [x2,fval2,exitflag2(k)]=fminunc(@(x2)obj4(Y2,x2,f,ind,U1,...
        V1,bu),randn(size(x2)),opt);
    f=zeros(length(f0),1);
    f(ind)=x2(1:length(ind));
    bf=x2(end-length(bu)+1:end);
    F=hankel(f(1:end-length(bu)+1),f(end-length(bu)+1:end));

     f2(:,k)=f;                  %Final estimate of tip-sample interaction
    bf2(:,k)=bf;                 %Final estimate of bf*1000
    Fbf(:,k)=F*bf./1000;         %Final estimate of F*bf

    % Simulation
    aa=cell(1); aa{1}=a; bub=cell(1); bub{1}=bu; bfb=cell(1);
    bfb{1}=[zeros(length(a)-length(bu),1);bf2(:,k)]./1000;
    FtoY=idpoly(aa,bfb,[],[],[],[],Ts);
    yhat(:,k)=lsim(sysoe,uc(1:length(f)),t(1:length(f)))+...
        lsim(FtoY,f2(end:-1:1,k),t(1:length(f)));

    % Evaluation
    MSE(k)=sum((Fhhbu(end-length(Fbf(:,k))+1:end)-Fbf(:,k)).^2)/...
        length(Fbf(:,k));
    MSE2(k)=sum((y2(1:length(yhat(:,k)))-yhat(:,k)).^2)/length(yhat(:,k));
    clear ind2 ind
end
```

C-4-2 AFM_DCP_exp.m

This file is used for the experiment in Section 4-4.

```
% Files needed:
% 'AFM_STAT_3_4_test1.mat'  Contains scanning data: u, y, ts, sys, tau_opt
clear all
close all

%% Load scanning data
load('AFM_STAT_3_4_test1.mat')
clear yhat

N=200;                  % #measurements to perform algorithm
start=4929797;          % Start point in data y
u2=u(start+tau_opt:start+N-1+tau_opt); % Corresponds to Equation 4-3 in report
y2=y(start:start+N-1);  % Output data used for algorithm
t=[0:length(u2)-1]'*ts;

%% Cantilever dynamics in state-space and in output-error model.
A=sys.A; B=sys.B; C=sys.C; D=sys.D;
sysoe=idpoly(sys);      % Rewrite the system to a transfer function
par=getpvec(sysoe); bu=par(1:round(length(par)/2+0.1));
a=[1;par(round(length(par)/2+0.1)+1:length(par))]; % Obtain a_1, a_2, ...
    b_{u,1} and b_{u,2} according to Equation 3-15 in the report

%% Rewrite OE in desired form
Y=hankel(y2(end:-1:length(a)),y2(length(a):-1:1));
Uh=hankel(u2(end-length(a)+length(bu):-1:length(bu)),u2(length(bu):-1:1));
Y2=Y*a-Uh*bu;

%% Start Optimisation (Step 1)
opt=optimset('Display','off','MaxIter',10^6,'MaxFunEvals',10^6,'TolX',...
    10^-8,'TolFun',10^-8);
bf0=randn(length(bu),1); f0=1*randn(N-length(a)+length(bu),1); x0=[f0;bf0];

x=fminunc(@(x)obj1(Y2,x,bu),x0,opt);
f=x(1:end-length(bu)); bf=x(end-length(bu)+1:end);
F=hankel(f(1:end-length(bu)+1),f(end-length(bu)+1:end));
M=[Y2 , F; bf , 1000*eye(length(bu))];
[U,¬,V]=svd(M);U1=U(:,1:length(bu));V1=V(:,1:length(bu));S1=svd(M);

%% DCP (Step 2)
i=1;der=-100;k=1;
while abs(der) > 10^-3
    [x,fval(i,k)]=fminunc(@(x)obj2_3(Y2,x,0,U1,V1,bu),...
        randn(size(x)),opt);
    f=x(1:end-length(bu)); bf=x(end-length(bu)+1:end);
    F=hankel(f(1:end-length(bu)+1),f(end-length(bu)+1:end));
    M=[Y2 , F; bf , 1000*eye(length(bu))];
    [U,¬,V]=svd(M);U1=U(:,1:length(bu));V1=V(:,1:length(bu));S1=svd(M);
    if i > 7
        der = fval(i,k)-fval(i-1,k);
    end
    i=i+1;
end
```

```
52
53 %% Add 1-norm (Step 3)
54 for k=1:20
55     lambda2=0.02*(k-1);          %lambda varied between 0 and 0.38
56     [x,fval2,exitflag(k)]=fminunc(@(x)obj2_3(Y2,x,lambda2,U1,...
57         V1,bu),randn(size(x)),opt);
58     f=x(1:end-length(bu)); bf=x(end-length(bu)+1:end);
59     F=hankel(f(1:end-length(bu)+1),f(end-length(bu)+1:end));
60
61     % Final step (Step 4)
62     err=max(abs(Y2))*1*10^-2;
63     ind3=find(F*bf/1000>err);
64     for j=2:length(bu)
65         ind3=[ind3;ind3+1];
66     end
67     ind=unique(sort(ind3(ind3>0)));
68
69     x2=[zeros(length(ind),1);bf];
70     [x2,fval2,exitflag2(k)]=fminunc(@(x2)obj4(Y2,x2,f,ind,U1,...
71         V1,bu),randn(size(x2)),opt);
72     f=zeros(length(f0),1);
73     f(ind)=x2(1:length(ind));
74     bf=x2(end-length(bu)+1:end);
75     F=hankel(f(1:end-length(bu)+1),f(end-length(bu)+1:end));
76
77     f2(:,k)=f;                   %Final estimate of tip-sample interaction
78     bf2(:,k)=bf;                 %Final estimate of bf*1000
79     Fbf(:,k)=F*bf./1000;         %Final estimate of F*bf
80
81     % Simulation
82     aa=cell(1); aa{1}=a; bub=cell(1); bub{1}=bu; bfb=cell(1);
83     bfb{1}=[zeros(length(a)-length(bu),1);bf2(:,k)]./1000;
84     FtoY=idpoly(aa,bfb,[],[],[],[],ts);
85     yhat(:,k)=lsim(sysoe,u2(1:length(f)),t(1:length(f)))+lsim(...
86         FtoY,f2(end:-1:1,k),t(1:length(f)));
87
88     % Evaluation
89     MSE(k)=sum((y2(1:length(yhat(:,k)))-yhat(:,k)).^2)/length(yhat(:,k));
90     clear ind2 ind ind3
91 end
```

C-4-3 obj1.m, obj2_3.m, obj4.m

The following files correspond to Steps 1 to 4 in Section 3-3, respectively.

```
1 % Inputs:
2 % -Y2       Cantilever deflection in Hankel matrix (Eq. 3-16 in report)
3 % -x        Vector containing estimation of f and bf
4 % -bu       Vector containing b_{u,1} and b_{u,2} (Eq. 3-15 in report)
5 %
6 % Output:
7 % -obj      Function value (Eq. 3-18 in report)
8 function obj=obj1(Y2,x,bu)
9 f=x(1:end-length(bu)); bf=x(end-length(bu)+1:end);
```

```
F=hankel(f(1:end-length(bu)+1),f(end-length(bu)+1:end));
obj = sum(svd([Y2 F; bf 1000*eye(length(bu))]));
end
```

```
% Inputs (in addition to obj1.m):
% -lambda2  Regularisation parameter (corresponding to lambda_DCP in report)
% -U1       First two columns of U, with [U,¬,V]=svd(M)
% -V1       First two columns of V, with [U,¬,V]=svd(M)
%
% Output:
% -obj      Function value (Eq.3-22 in report)
function obj=obj2_3(Y2,x,lambda2,U1,V1,bu)
f=x(1:end-length(bu)); bf=x(end-length(bu)+1:end);
F=hankel(f(1:end-length(bu)+1),f(end-length(bu)+1:end));
obj = (sum(svd([Y2 F; bf 1000*eye(length(bu))]))-...
    trace(U1'*[Y2 F; bf 1000*eye(length(bu))]*V1))+...
    lambda2*norm(F*bf./1000,1);
end
```

```
% Inputs (in addition to obj1.m):
% -f2       Previous estimation of f
% -ind      Indices for which f ≠ 0 (as indicated in Eq. 3-23 in report)
% -U1       First two columns of U, with [U,¬,V]=svd(M)
% -V1       First two columns of V, with [U,¬,V]=svd(M)
%
% Output:
% -obj      Function value (Eq.3-23 in report)
function obj=obj4(Y2,x,f2,ind,U1,V1,bu)
bf=x(end-length(bu)+1:end);
f=zeros(length(f2),1);
f(ind)=x(1:end-length(bu));
F=hankel(f(1:end-length(bu)+1),f(end-length(bu)+1:end));
obj = (sum(svd([Y2 F; bf 1000*eye(length(bu))]))-...
    trace(U1'*[Y2 F; bf 1000*eye(length(bu))]*V1));
end
```

C-5 Alternating Direction Method of Multipliers

The STATESON-algorithm can be rewritten into the following problem:

$$\begin{aligned} &\min_{x_\mathrm{B}, z_\mathrm{B}} \frac{1}{2} \left\| G x_\mathrm{B} - g \right\|_2^2 + \lambda \left\| z_\mathrm{B} \right\|_1 \\ &\text{s.t.}\, W x_\mathrm{B} - z_\mathrm{B} = 0 \end{aligned} \tag{C-1}$$

with

$$G = \left[\begin{array}{ccccc|ccccc} -R_\nu^{-\frac{1}{2}}C & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & -R_\nu^{-\frac{1}{2}}C & \ddots & \vdots & \vdots & 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -R_\nu^{-\frac{1}{2}}C & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & -R_\nu^{-\frac{1}{2}}C & 0 & \dots & 0 & 0 & 0 \\ \hline -\gamma A & \gamma I & 0 & \dots & 0 & -\gamma B_2 & 0 & \dots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & -\gamma A & \gamma I & 0 & \vdots & \ddots & -\gamma B_2 & 0 & 0 \\ 0 & \dots & 0 & -\gamma A & \gamma I & 0 & \dots & 0 & -\gamma B_2 & 0 \end{array}\right],$$

$$W = \begin{bmatrix} \mathbf{0}_{Nn \times Nn} & \mathbf{0}_{Nn \times N} \\ \mathbf{0}_{N \times Nn} & \mathbf{I}_N \end{bmatrix}$$

$$g = \left[\begin{array}{cccc|cccc} R_\nu y(1) & R_\nu y(2) & \dots & R_\nu y(N) & -\gamma u_c^\top(1) B^\top & -\gamma u_c^\top(2) B^\top & \dots & -\gamma u_c^\top(N-1) B^\top \end{array}\right]^\top$$

$$x_\mathrm{B} = \left[\begin{array}{cccc|cccc} x^\top(1) & x^\top(2) & \dots & x^\top(N) & F(1) & F(2) & \dots & F(N-1) \end{array}\right]^\top$$

$$z_\mathrm{B} = \left[\begin{array}{cccc|cccc} 0 & 0 & \dots & 0 & F(1) & F(2) & \dots & F(N-1) \end{array}\right]^\top$$

(C-2)

and λ and γ tuning parameters. Then, the ADMM-algorithm consists of updating three vectors each iteration [35]:

$$\begin{aligned} x_\mathrm{B}(k+1) &= (G^\top G + \rho W^\top W)^{-1} (G^\top g + \rho W^\top (z_\mathrm{B}(k) - u_\mathrm{B}(k))) \\ z_\mathrm{B}(k+1) &= S_{\lambda/\rho}(W x_\mathrm{B}(k+1) + u_\mathrm{B}(k)) \\ u_\mathrm{B}(k+1) &= u_\mathrm{B}(k) + W x_\mathrm{B}(k+1) - z_\mathrm{B}(k+1) \end{aligned} \tag{C-3}$$

with ρ a penalty parameter and

$$S_\kappa(a) = \begin{cases} a - \kappa & a > \kappa \\ 0 & |a| \leq \kappa \\ a + \kappa & a < -\kappa. \end{cases}$$

It turned out that a varying penalty parameter improved the convergence considerably and made the result less dependent on the initial guess of ρ. Therefore, ρ is adjusted every iteration:

$$\rho(k+1) = \begin{cases} \tau^{\mathrm{incr}} \rho(k) & \text{if } \|r(k)\|_2 > \mu \|s(k)\|_2 \\ \rho(k) / \tau^{\mathrm{decr}} & \text{if } \|s(k)\|_2 > \mu \|r(k)\|_2 \\ \rho(k) & \text{otherwise,} \end{cases} \tag{C-4}$$

with choices for $\tau^{\text{incr}} = \tau^{\text{decr}} = 2$ and $\mu = 10$. The variables r and s are $r(k) = Wx_{\text{B}}(k) - z_{\text{B}}(k)$ and $s(k) = -\rho W^{\top}(z_{\text{B}}(k) - z_{\text{B}}(k-1))$.

Unfortunately, a varying ρ implies that the inverse $(G^{\top}G + \rho W^{\top}W)^{-1}$ should be calculated every iteration. This requires large computational effort. Since the dimensions of G and W are approximately the same and these matrices are (almost) square, a trick with LU-factorisation does not make sense.

C-5-1 admmstat.m

In this file, Equation 3-9 is implemented in the ADMM-algorithm, with $\alpha = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^{\top}$. Its functionality is similar to "STATESON.m". Use is made of the (lasso) script in [44].

```
% Inputs:
% -lambda       Regularisation parameter
% -rho          Initial value for rho (Eq. C-4 in report)
% -relax        Relaxation parameter, typical value between 1.0 and 1.8
% -A1           Identified A-matrix
% -B1           Identified B-matrix
% -C1           Identified C-matrix
% -G2           Identified G-matrix (corresponding to B_2 in report)
% -u2           Cantilever input (corresponding to Eq. 4-3 in report)
% -y            Measured cantilever deflection
% -Rnu          Estimated variance of measurement noise
% -Q            Estiamted variance of pulses in tip-sample interaction
% -gamma        Penalty term on constraint
%
% Output:
% -x            Vector containing [states; t/s-interaction]
function x = admmstat(lambda, rho, relax, A1,B1,C1,G2,u2,y,Rnu,Q,gamma)

%% Global constants and defaults
MAX_ITER = 15;
ABSTOL   = 1e-4;
RELTOL   = 1e-2;
tincr = 2;
tdecr = 2;
mu = 10;

%% Making vectors
M = length(y);
N = (length(B1)+1)*M;
F = [zeros(length(B1)*M), zeros(length(B1)*M,N-length(B1)*M);
    zeros(N-length(B1)*M,length(B1)*M), Q^(-0.5)*eye(N-length(B1)*M)];

for i=1:M-1
    A(i,length(B1)*(i-1)+1:length(B1)*i) = -Rnu^(-0.5)*C1;
end
A(:,(M-1)*length(B1)+1:N) = zeros(M-1,N-(M-1)*length(B1));
b = -Rnu^(-0.5)*y(1:end-1);

for i=1:M-1
    V(length(B1)*(i-1)+1:length(B1)*i,length(B1)*(i-1)+1:length(B1)*(i+1))...
        = [-A1 eye(length(B1))];
```

```
42 end
43 for i=1:M-1
44     V(length(B1)*(i-1)+1:length(B1)*i,length(B1)*(M)+i) = -G2;
45 end
46 V(:,length(B1)*(M)+M)=zeros(length(B1)*(M-1),1);
47 for i=1:M-1
48 g(length(B1)*(i-1)+1:length(B1)*i,1)= -B1*u2(i);
49 end
50 A = [A;gamma*V].*sqrt(2);
51 b = [b;-gamma*g].*sqrt(2);
52
53 %% Data preprocessing
54 [m, n] = size(A);
55
56 % save a matrix-vector multiply
57 Atb = A'*b;
58 AtA = A'*A;
59 FtF = F'*F;
60
61 %% ADMM solver
62 x = zeros(n,1)+0*1i;
63 z = zeros(n,1)+0*1i;
64 u = zeros(n,1)+0*1i;
65
66 % cache the factorization
67 % [L U] = factor(A, rho);
68
69 rnorm=0;snorm=0;
70 for k = 1:MAX_ITER
71     % Calculate inverse with new rho
72     ArhoFinv = inv(AtA + rho*FtF);
73     % x-update
74     q = Atb + rho*F'*(z - u);    % temporary value
75 %     if( m ≥ n )    % if skinny
76 %        x = U \ (L \ q);
77 %     else            % if fat
78 %        x = q/rho - (A'*(U \ ( L \ (A*q) )))/rho^2;
79 %     end
80     x = ArhoFinv*q;
81
82     % z-update with relaxation
83     zold = z;
84     x_hat = relax*x + (1 - relax)*zold;
85     z = shrinkage(F*x_hat + u, lambda/rho);
86
87     % u-update
88     u = u + (F*x_hat - z);
89
90     % Update rho
91     if k > 1
92        if rnorm > mu*snorm
93           rho = tincr*rho;
94           u = u/tincr;
95        end
96        if snorm > mu*rnorm
97           rho = rho/tdecr;
98           u = u*tdecr;
```

```
99          end
100     end
101
102     % calculate norms of r(k+1) and s(k+1)
103     rnorm = norm(F*x-z);
104     snorm = norm(-rho*F'*(z - zold));
105
106     if (rnorm < sqrt(n)*ABSTOL + RELTOL*max(norm(x), norm(-z)) && ...
107        snorm < sqrt(n)*ABSTOL + RELTOL*norm(rho*u))
108          break;
109     end
110
111 end
112 end
113
114 function p = objective(A, b, lambda, x, F, z)
115     p = ( 1/2*sum((A*x - b).^2) + lambda*norm(z,1) );
116 end
117
118 function z = shrinkage(x, kappa)
119     z = max( 0, x - kappa ) - max( 0, -x - kappa );
120 end
121
122 function [L U] = factor(A, rho)
123     [m, n] = size(A);
124     if ( m ≥ n )     % if skinny
125        L = chol( A'*A + rho*speye(n), 'lower' );
126     else             % if fat
127        L = chol( speye(m) + 1/rho*(A*A'), 'lower' );
128     end
129
130     % force matlab to recognize the upper / lower triangular structure
131     L = sparse(L);
132     U = sparse(L');
133 end
```

Bibliography

[1] J. M. Eeftens, A. J. Katan, M. Kschonsak, M. Hassler, L. de Wilde, E. M. Dief, C. H. Haering, and C. Dekker, "Condensin Smc2-Smc4 Dimers Are Flexible and Dynamic," *Cell Reports*, vol. 14, no. 8, pp. 1813–1818, 2016.

[2] G. Binnig and C. F. Quate, "Atomic Force Microscope," *Physical Review Letters*, vol. 56, no. 9, pp. 930–933, 1986.

[3] T. Fukuma, M. Kimura, K. Kobayashi, K. Matsushige, and H. Yamada, "Development of low noise cantilever deflection sensor for multienvironment frequency-modulation atomic force microscopy," *Review of Scientific Instruments*, vol. 76, no. 5, 2005.

[4] M. Ganji, I. A. Shaltiel, S. Bisht, E. Kim, A. Kalichava, C. H. Haering, and C. Dekker, "Real-time imaging of DNA loop extrusion by condensin," *Science*, vol. 360, pp. 102–105, 2018.

[5] T. Terakawa, S. Bisht, J. M. Eeftens, C. Dekker, and C. H. Haering, "The condensin complex is a mechanochemical motor that translocates along DNA," *Science*, vol. 358, pp. 672–676, 2017.

[6] J. Noom, "Control Enhancement in Atomic Force Microscopy: Improving the Image Resolution," tech. rep., Delft University of Technology, Delft, 2018.

[7] Y. F. Dufrêne, T. Ando, R. Garcia, D. Alsteens, D. Martinez-Martin, A. Engel, C. Gerber, and D. J. Müller, "Imaging modes of atomic force microscopy for application in molecular and cell biology," *Nature Nanotechnology*, vol. 12, no. 4, pp. 295–307, 2017.

[8] T. R. Albrecht, P. Grütter, D. Horne, and D. Rugar, "Frequency modulation detection using high-Q cantilevers for enhanced force microscope sensitivity," *Journal of Applied Physics*, vol. 69, no. 2, pp. 668–673, 1991.

[9] P. Eaton and P. West, *Atomic Force Microscopy*. New York: Oxford University Press Inc., 2010.

[10] D. Fotiadis, S. Scheuring, S. A. Müller, A. Engel, and D. J. Müller, "Imaging and manipulation of biological structures with the AFM," *Micron*, vol. 33, no. 4, pp. 385–397, 2002.

[11] D. Y. Abramovitch, S. B. Andersson, L. Y. Pao, and G. Schitter, "A tutorial on the mechanisms, dynamics, and control of atomic force microscopes," *Proceedings of the American Control Conference*, pp. 3488–3502, 2007.

[12] G. E. Fantner, G. Schitter, J. H. Kindt, T. Ivanov, K. Ivanova, R. Patel, N. Holten-Andersen, J. Adams, P. J. Thurner, I. W. Rangelow, and P. K. Hansma, "Components for high speed atomic force microscopy," *Ultramicroscopy*, vol. 106, no. 8-9, pp. 881–887, 2006.

[13] C. Dekker, "Cees Dekker Lab." https://ceesdekkerlab.nl/. Accessed: 19-06-2019.

[14] M. Detterbeck, "High Speed Scanning - NanoWorld Ultra-Short Cantilevers." https://www.highspeedscanning.com/ultra-short-cantilevers.html. Accessed: 19-06-2019.

[15] H. J. Butt, B. Cappella, and M. Kappl, "Force measurements with the atomic force microscope: Technique, interpretation and applications," *Surface Science Reports*, vol. 59, no. 1-6, pp. 1–152, 2005.

[16] C. A. J. Putman, K. O. Van Der Werf, B. G. De Grooth, N. F. Van Hulst, and J. Greve, "Tapping mode atomic force microscopy in liquid," *Applied Physics Letters*, vol. 64, no. 18, pp. 2454–2456, 1994.

[17] G. Fantner, W. Schumann, R. J. Barbero, A. Deutschinger, V. Todorov, D. S. Gray, A. M. Belcher, I. Rangelow, and K. Youcef-Toumi, "Use of self-actuating and self-sensing cantilevers for imaging biological samples in fluid," *Nanotechnology*, vol. 20, no. 43, 2009.

[18] G. C. Ratcliff, D. A. Erie, and R. Superfine, "Photothermal modulation for oscillating mode atomic force microscopy in solution," *Applied Physics Letters*, vol. 72, no. 15, pp. 1911–1913, 1998.

[19] Y. Martin, C. C. Williams, and H. K. Wickramasinghe, "Atomic force microscope-force mapping and profiling on a sub 100-Å scale," *Journal of Applied Physics*, vol. 61, no. 10, pp. 4723–4729, 1987.

[20] S. Belikov, J. Alexander, M. Surtchev, and S. Magonov, "Implementation of atomic force microscopy resonance modes based on asymptotic dynamics using Costas Loop," *Proceedings of the American Control Conference*, pp. 6201–6208, 2016.

[21] A. D. L. Humphris, J. Tamayo, and M. J. Miles, "Active quality factor control in liquids for force spectroscopy," *Langmuir*, vol. 16, no. 21, pp. 7891–7894, 2000.

[22] N. Kodera, M. Sakashita, and T. Ando, "Dynamic proportional-integral-differential controller for high-speed atomic force microscopy," *Review of Scientific Instruments*, vol. 77, no. 8, p. 83704, 2006.

[23] D. R. Sahoo, P. Agarwal, and M. V. Salapaka, "Transient force atomic force microscopy: A new nano-interrogation method," *Proceedings of the American Control Conference*, pp. 3135–3140, 2007.

[24] A. S. Willsky and H. L. Jones, "A Generalized Likelihood Ratio Approach to the Detection and Estimation of Jumps in Linear Systems," *IEEE Transactions on Automatic Control*, vol. 21, no. 1, pp. 108–112, 1976.

[25] H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd, "Smoothed state estimates under abrupt changes using sum-of-norms regularization," *Automatica*, vol. 48, no. 4, pp. 595–605, 2012.

[26] C. Yu, L. Ljung, and M. Verhaegen, "Gray Box Identification Using Difference of Convex Programming," *International Journal of Automatic Control*, vol. 50, no. 1, pp. 9462–9467, 2017.

[27] D. Kanevsky, A. Carmi, L. Horesh, P. Gurfil, B. Ramabhadran, and T. N. Sainath, "Kalman Filtering for Compressed Sensing," in *13th International Conference on Information Fusion*, (Edinburgh, UK), IEEE, 2010.

[28] R. Doelman and M. Verhaegen, "Sequential convex relaxation for convex optimization with bilinear matrix equalities," in *2016 European Control Conference (ECC)*, (Aalborg, Denmark), pp. 1946–1951, IEEE, 2016.

[29] T. Ando, "SS-NEX - Ando model." https://www.ribm.co.jp/en/product/sub_ssnex_en. Accessed: 19-06-2019.

[30] "Flexible resolution oscillocscopes." https://www.picotech.com/oscilloscope/5000/flexible-resolution-oscilloscope. Accessed: 19-06-2019.

[31] A. J. Katan, M. H. Van Es, and T. H. Oosterkamp, "Quantitative force versus distance measurements in amplitude modulation AFM: A novel force inversion technique," *Nanotechnology*, vol. 20, no. 16, 2009.

[32] M. Verhaegen and V. Verdult, *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, 2007.

[33] T. E. Schäffer, J. P. Cleveland, F. Ohnesorge, D. A. Walters, and P. K. Hansma, "Studies of vibrating atomic force microscope cantilevers in liquid," *Journal of Applied Physics*, vol. 80, pp. 3622–3626, 1996.

[34] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[35] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[36] C. Desch, "The Chemistry Of Solids," *Cornell University Press*, vol. 180, 1934.

[37] Y. K. Yong, S. O. R. Moheimani, and I. R. Petersen, "High-speed cycloid-scan atomic force microscopy," *Nanotechnology*, vol. 21, no. 36, 2010.

[38] N. Banterle, K. H. Bui, E. A. Lemke, and M. Beck, "Fourier ring correlation as a resolution criterion for super-resolution microscopy," *Journal of Structural Biology*, vol. 183, pp. 363–367, 2013.

[39] M. van Heel and M. Schatz, "Fourier shell correlation threshold criteria," *Journal of Structural Biology*, vol. 151, pp. 250–262, 2005.

[40] W. S. Cleveland and S. J. Devlin, "Locally Weighted Regression : An Approach to Regression Analysis by Local Fitting," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 596–610, 1988.

[41] O. Burri and A. Herbert, "Fourier Ring Correlation Plugin - ImageJ." https://imagej.net/Fourier_Ring_Correlation_Plugin. Accessed: 19-06-2019.

[42] I. Houtzager, "LTI System Identification Toolbox." https://www.dcsc.tudelft.nl/~jwvanwingerden/lti/ltitoolbox_product_page.html. Accessed: 19-06-2019.

[43] M. C. Grant and S. P. Boyd, "CVX Research, Inc. | Software for Disciplined Convex Programming." http://cvxr.com/. Accessed: 19-06-2019.

[44] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "MATLAB scripts for alternating direction method of multipliers." https://web.stanford.edu/~boyd/papers/admm/. Accessed: 19-06-2019.

Glossary

List of Acronyms

ADMM	Alternating Direction Method of Multipliers
AFM	Atomic Force Microscopy
AM-FI	Amplitude Modulation with Frequency Imaging
AM-PI	Amplitude Modulation with Phase Imaging
CS	Compressed Sensing
CSKF	CS-embedded Kalman Filter
DCP	Difference of Convex Programming
DNA	DeoxyriboNucleic Acid
FB	feedback
FFT	Fast Fourier Transform
FM-AI	Frequency Modulation with Amplitude Imaging
FRC	Fourier Ring Correlation
I/O	Input and Output
LASSO	Least Absolute Shrinkage and Selection Operator
LIA	Lock-In Amplifier
LOESS	Locally weighted regression
LSE	Least Squared Error
LTI	Linear Time-Invariant
MSE	Mean Squared Error

PLL Phase-Locked Loop

PM-AI Phase Modulation with Amplitude Imaging

PO-MOESP Past Outputs Multivariable Output-Error State-sPace

PRBS Pseudo-Random Binary Sequence

RIP Restricted Isometry Property

SEM Scanning Electron Microscope

STATESON State Smoothing by Sum-of-Norms Regularisation

TEM Transmission Electron Microscope

VAF Variance Accounted For

List of Symbols

α	Weighting vector to enhance convergence of model-based approach
$\delta_F(t)$	Binary digit
ϵ	Positive tuning parameter to modify α in the next iteration
ϵ_{cs}	Upper bound in CSKF
η	Thermal noise
$\gamma_F(t)$	Zero-mean Gaussian distributed variable with variance Q
λ	Tuning parameter for model-based approach
$\lambda^{\max}$	Critical value for λ for which the estimation would be $\hat{F} = 0$.
λ_{DCP}	Tuning parameter for data-driven approach
μ	Probability whether $F(t)$ is nonzero
ν	Measurement noise
ϕ	Phase of oscillating cantilever
$\sigma_3(R)$	Threshold curve as function of R, used in relation with FRC(R)
τ	Phase difference between u_c and u_2
ε	Threshold value for which estimates $\hat{F}(t)$ are set to zero or not
ς_t	Scaled error between measured and modelled signal if $F(t) = 0$
ζ_k	Noise in CSKF
A, B, B_2, C	State-space matrices of the cantilever dynamics
$A(q)$	Denominator of discrete-time transfer function $B_u(q)/A(q)$
a_1, a_2	Factors of $A(q)$
A_c	Amplitude of oscillating cantilever
$A_{c,0}$	Reference amplitude of oscillating cantilever
b_f	Vector containing factors in B_f: i.e. $b_f = \begin{bmatrix} b_{f,1} & b_{f,2} \end{bmatrix}^\top$

$B_f(q)$	Numerator of discrete-time transfer function $B_f(q)/A(q)$
$B_u(q)$	Numerator of discrete-time transfer function $B_u(q)/A(q)$
$b_{u,1}$, $b_{u,2}$	Factors of $B_u(q)$
c	Constant to relate u_2 with u_c
c_1	Constant
c_2	Constant
c_3	Constant
c_4	Constant
F	Tip-sample interaction
f	Spatial frequency
f_0	Spatial frequency determining the resolution of the original image
h	Sample height
H'	Matrix having the "restricted isometry property", used in CSKF
i	Indication of pixel i, used in FRC(R)
I_1	First image resulting from original image
I_2	Second image resulting from original image
K	Kalman gain
k	Segment number in model-based approach
N	Number of measurements
n	Order of Linear Time-Invariant model of the cantilever
n_0	Number of nonzero entries in $F(t)$
N_τ	Number of iterations in CSKF
n_{asym}	Degree of symmetry
N_b	Number of measurements per segment in System Identification
$N_p(R)$	Number of pixels in frequency domain at the ring with radius R
N_T	Number of measurements per segment in model-based approach
p	Tuning parameter for model-based approach
Q	Variance of $F(t)$
q	Tuning parameter for model-based approach, dependent on choice of p
Q_{res}	Image resolution in nanometers
R	Radius from center to ring in Fourier transformed images, used in FRC(R)
R_η	Variance of η
R_ν	Variance of ν
r_i	The pixel in frequency domain at location i, used in FRC(R)
s	Tuning parameter determining size of Hankel matrices in Subspace Identification
U	Matrix resulting from singular value decomposition of $\mathbf{M}$
U_1	Matrix consisting of the first two columns of U
u_2	Input to the cantilever at the location of the thermal noise
u_c	Input to cantilever actuator
u_z	Input to the z-actuator of the scan table

V	Matrix resulting from singular value decomposition of $\mathbf{M}$
V_1	Matrix consisting of the first two columns of V
V_F	Vector containing the ℓ_2-norms of $F(t)$
$x(t)$	State at time step t
$Y(k)$	Quantity resulting from the error $A(q)y_c(t) - B_u(q)u_c(t)$
y_c	Measured cantilever deflection
y_k	Known vector in CSKF
z	Height of the scan table
z_k	Sparse vector to be reconstructed in CSKF
$\hat{y}_a$	Estimated cantilever deflection for case (a) of data-driven approach
$\hat{y}_b$	Estimated cantilever deflection for case (b) of data-driven approach
$\mathbf{F}$	Hankel matrix of tip-sample interaction
$\mathbf{M}$	Matrix composed of $\mathbf{Y}$, $\mathbf{F}$, b_f and I_2
$\mathbf{Y}$	Vector resulting from error $Y(k)$
$\mathscr{T}$	The set $\{1, 2, ..., N\}$
$\tilde{I}_1$	Fourier Transform of I_1
$\tilde{I}_2$	Fourier Transform of I_2
FRC(R)	Fourier Ring Correlation as function of radius R