

# A Dedicated Endstation for Waveguide-based X-Ray Imaging



Dissertation  
zur Erlangung des mathematisch-naturwissenschaftlichen  
Doktorgrades  
“Doctor rerum naturalium”  
der Georg-August-Universität Göttingen

vorgelegt von  
Sebastian Kalbfleisch  
aus  
Schwalmstadt-Ziegenhain

Göttingen, 2012

Referent: Prof. Dr. Tim Salditt  
Koreferent: Prof. Dr. Hans Hofsäss  
Tag der mündlichen Prüfung: 17.4.2012

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Interferometry and Holographic Imaging with X-Ray Waveguides</b>	<b>5</b>
2.1	X-Rays as Electromagnetic Waves . . . . .	5
2.2	X-Ray Interaction with Matter . . . . .	6
2.3	X-Ray Waveguides . . . . .	8
2.4	Reference Beam Holography for Interferometric Applications	12
2.5	Inline Holography and Iterative Reconstruction Approaches	15
<b>3</b>	<b>Conceptual Experiments</b>	<b>21</b>
3.1	X-Ray Interferometry with Waveguides . . . . .	21
3.2	X-Ray Inline Holography . . . . .	28
<b>4</b>	<b>The Coherent Imaging Endstation of Beamline P10 at PETRA III of DESY</b>	<b>39</b>
4.1	Requirements and Design Considerations for a Dedicated Waveguide-based Imaging Instrument . . . . .	39
4.2	Instrument Overview . . . . .	40
4.3	The Layout of Beamline P10 . . . . .	42
4.4	Focusing Optics . . . . .	45
4.5	Waveguide Positioning . . . . .	52
4.6	Sample Positioning . . . . .	54
4.7	Instrument Table . . . . .	55
4.8	Detectors . . . . .	58
4.9	Instrument Control System and Interfaces to P10 . . . . .	64
<b>5</b>	<b>Results of Instrument Commissioning</b>	<b>73</b>
5.1	Alignment and Characterisation of the Focusing Optics . . .	73
5.2	First Imaging Experiments . . . . .	91

<b>6</b>	<b>Summary &amp; Outlook</b>	<b>103</b>
<b>A</b>	<b>Characterisation of the Waveguide Positioning System</b>	<b>107</b>
A.1	Interferometer Setup . . . . .	107
A.2	Linearity Measurements . . . . .	109
A.3	Repeatability Measurements . . . . .	114
<b>B</b>	<b>Source Code</b>	<b>121</b>
B.1	spec Macros . . . . .	121
B.2	Temperature Logger Source Code . . . . .	184
	<b>Bibliography</b>	<b>198</b>

# Chapter 1

## Introduction

X-rays have been used for imaging since their discovery in 1895 by Wilhelm Conrad Röntgen [72]. The ability of X-rays to penetrate thick and optically opaque matter has made radiographic absorption imaging a very important and powerful method in medicine and many other fields of science. The short wavelength of X-rays has also opened up many different applications in microscopy with high spatial resolution. To this end, a major challenge is the fabrication of suitable lenses for X-rays due to their weak interaction with matter. With the development of Fresnel zone plates and the availability of synchrotron radiation sources, X-ray microscopy became feasible [40] and is nowadays a well-established microscopy method at many synchrotrons. Current state of the art X-ray microscopes based on Fresnel zone plates achieve a resolution of 15 nm [12] in the soft X-ray regime between the carbon and oxygen K absorption edges at 283 eV and 543 eV. In this spectral range, known as water window, biological samples show a natural absorption contrast. In the hard X-ray regime of photon energies above 1 keV the spatial resolution is typically in the range of 40 nm – 50 nm [29]. The resolution of Fresnel zone plates depends mainly on the outer most zone width [2], and is thus coupled to advances in nanofabrication techniques.

Besides absorption a sample imprints also a phase shift onto the X-ray beam, depending on the inner variation of the optical density. In combination with a coherent or partially coherent illumination this can be used for propagation based phase contrast imaging. Modern synchrotron radiation sources deliver X-ray beams of sufficient partial coherence, that enable phase contrast imaging of macroscopic samples [13, 81]. As no further magnification optics are involved, the spatial resolution is mainly

limited by the pixel size of the detector.

An approach to overcome the technological limitation of Fresnel zone plate fabrication or detector resolution is coherent diffractive X-ray imaging (CDXI). CDXI relies on the coherent illumination of the sample, that can be easily realised by the selection of a coherent wavefront out of a synchrotron beam, e.g., by the use of a pinhole [49]. The sample information is encoded in the recorded diffraction signal and is reconstructed by a numerical inversion of the signal. The absence of any optical element for image formation limits the resolution of CDXI only to the highest diffraction angle at which an analysable signal is detected. The inversion suffers from the lost phase information of the wavefield, as only the intensity of the diffracted X-rays is recorded. The phase information can be retrieved by algorithms propagating the field iteratively between the detector plane (Fourier space) and the sample plane (real space) and applying a priori information of the sample, e.g., size, thickness, maximum expected phase shift, and other constraints in the Fourier space. As the undiffracted part of the beam is much more intense than the diffracted signal, it has to be blocked to avoid any saturation or overexposure of the detector. This missing information of the diffraction pattern complicates the phase retrieval and influences its uniqueness. CDXI has also been successfully combined with focussing optics [86]. The higher flux in the focus leads to an increased scattered signal of the sample, but still requires a beamstop. The resolution is not limited to the spot size, as in conventional scanning X-ray microscopy, as long as the coherence condition is fulfilled.

An alternative approach to lensless X-ray imaging is offered by X-ray waveguides. They act as a coherence filter and decouple the sample illumination from the coherence properties of the upstream optics and source [61]. X-ray waveguides emit a highly divergent and coherent beam from a rather small cross section. The divergence provides a geometric magnification of the sample [22]. This way, the resolution of the sample is not limited to the size of a detector pixel. Also, the intensity is distributed over a larger detector area, giving a more uniform exposure. There is no need in any beamstop leading to information loss in the diffraction pattern. The coherent illumination enables holographic imaging of the sample. As holography encodes the full wavefield information including the phase [24], the reconstruction is a deterministic single step inversion of the hologram. Holographic imaging with waveguides has been demonstrated earlier in reference beam geometry [22], but is limited to rather small samples not disturbing the reference beam. The generic inline ge-

ometry can be extended to scanning mode [27]. The intrinsic twin image problem has been successfully treated by a combination of holographic reconstruction with iterative algorithms [26].

Motivated by these advantages the aim of this thesis is the design, construction and commissioning of a dedicated synchrotron endstation for waveguide-based imaging. It is installed at the beamline P10 at the synchrotron source PETRA III of the Deutsches Elektronen-Synchrotron (DESY). The source parameters and beamline layout are optimised for coherence applications. Whilst this thesis highlights mainly methods and instrumentation for waveguide-based imaging, the endstation is designed to benefit from the remarkable coherence properties of the beamline. Due to their outstanding surface quality, the focussing mirrors are capable of coherent focussing. With the flexibility of the instrumentation this allows for an easy application of other lensless imaging methods, such as propagation-based or coherent diffractive imaging.

Chapter 2 begins with a brief introduction to the description of X-rays as electromagnetic waves and their interaction with matter. Based on these fundamentals, waveguides for X-rays are explained along with the types of waveguides used within this thesis. The use of Y-shaped double channel waveguides as an interferometer is developed. As a last part of this chapter the basics of inline holography with waveguides and its combination with iterative reconstruction algorithms are elucidated.

Chapter 3 reports on conceptual experiments of waveguide-based interferometry and holographic imaging with iterative object reconstruction. The use of waveguide-based interferometry as a probe for local dynamics is exemplified on Au colloids as a model system. The procedure of data analysis is described and an evaluation of the method is given. After a description of the experimental setup at the ID22NI beamline of the European Synchrotron Radiation Facility (ESRF) used for the inline holography, the results of different reconstruction methods are shown. Besides the single-step holographic reconstruction the extension to ptychography [27] on rectangular and spiralled scan grids is shown. Finally the first successful iterative reconstruction of an inline hologram of a freeze-dried biological cell is shown [26], demonstrating the capabilities of waveguide-based holographic imaging.

The concept of the endstation and its instrumental layout is described in detail in Chapter 4. After considerations on the instrumental requirements, a general overview of the endstation design is given. Based on the layout and source characteristics of the beamline P10, the outstanding properties

of the focussing optics are discussed. The degrees of freedom and their realisation for a proper waveguide and sample alignment are elaborated in the following. The available detectors used for the measurements are described, before the instrument control system and its interfaces to beamline components are explained.

The commissioning of the endstation is presented in Chapter 5. Special emphasis is placed on the alignment strategy and characterisation of the focussing optics. Finally, first results in holographic imaging of test patterns and freeze-dried biological cells are shown.

The Appendix describes the experimental setup and results on the repeatability characterisation of the waveguide positioners and lists all relevant macros, that were developed and programmed to control the instrument components.



## Chapter 2

# Fundamentals of Interferometry and Holographic Imaging with X-Ray Waveguides

### 2.1 X-Rays as Electromagnetic Waves

The regime of electromagnetic waves with photon energies of about 1 keV up to 100 keV is commonly denoted as *hard X-rays*. As an electromagnetic wave, X-rays propagating in vacuum are described by MAXWELL'S equations

$$\nabla \cdot \vec{E} = 0 \quad (2.1)$$

$$\nabla \cdot \vec{B} = 0 \quad (2.2)$$

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (2.3)$$

$$\nabla \times \vec{B} = \epsilon_0 \mu_0 \frac{\partial \vec{E}}{\partial t} \quad (2.4)$$

with  $\vec{E}$  for the electric field,  $\vec{B}$  for the magnetic field,  $\epsilon_0$  for the electric field constant, and  $\mu_0$  for the magnetic field constant. From equations (2.1) – (2.4) the wave equation for the electrical field  $\vec{E}$  of electromagnetic waves propagating in vacuum can be derived [45]

$$\Delta \vec{E} - \epsilon_0 \mu_0 \frac{\partial^2 \vec{E}}{\partial t^2} = 0. \quad (2.5)$$

Among the many solutions of the wave equation (2.5) there are two solutions that are of essential interest for the following descriptions: the plane wave and the spherical wave.

The plane wave can be expressed in complex notation as an exponential function of the location  $\vec{x}$  and time  $t$  as

$$\vec{E}(\vec{x}, t) = \vec{E}_0 \exp\left(i(\vec{k} \cdot \vec{x} - \omega t)\right), \quad (2.6)$$

where  $\vec{E}$  denotes the complex amplitude of the wave,  $\vec{k}$  is called the wave vector and denotes the propagation direction of the wave, and  $\omega = 2\pi\nu$  denotes the circular frequency with  $\nu$  being the frequency of the wave. At a fixed time  $t_0$  the phase of a plane wave is constant in every plane perpendicular to its direction of propagation  $\vec{k}$ . The wave number rely on the wavelength  $\lambda$  by  $|\vec{k}| = k = \frac{2\pi}{\lambda}$ . The propagation velocity  $c$  of the wave is given by the dispersion relation  $c = \frac{\omega}{k}$ .

A wave that is emitted by a point source propagates equally in each spatial direction and is thus rotation-symmetric, which allows a scalar notation of the distance  $r$ . Due to energy conservation, the amplitude  $E_0$  of this spherical wave is decaying with increasing distance  $r$  from the source by  $\frac{1}{r}$ . In complex notation a spherical wave reads then as

$$E(r, t) = \frac{E_0}{r} \exp(i(kr - \omega t)). \quad (2.7)$$

The phase of a spherical wave is constant on spheres of equal distance  $r$  from the source for a fixed time  $t_0$ . For very large distances  $r$  from the source, the spherical wave can be locally treated as a plane wave.

Operations, properties, and formalisms known from wave optics of visible light are applicable to waves in the X-ray regime in many cases.

## 2.2 X-Ray Interaction with Matter

A fundamental property of X-rays is their weak interaction with matter, and thus their ability to penetrate it. This allows the non-destructive analysis of optical opaque objects. This is commonly used in X-ray radiography, where absorption distinctions of bulk specimens are visualised. Besides the absorption, X-rays undergo also a phase shift when propagating through matter compared to free space propagation. The absorption

and phase shift of a homogeneous material are described by the refractive index  $n$  in complex notation

$$n = 1 - \delta - i\beta. \quad (2.8)$$

The imaginary part  $\beta$  describes the attenuation of X-rays in the medium. In the vicinity of absorption edges, it is related to the linear attenuation coefficient  $\mu$ , which is a property of the medium, by

$$\beta = \frac{\mu\lambda}{4\pi}. \quad (2.9)$$

Also in the vicinity of absorption edges, the real part  $\delta$  is given by

$$\delta = \frac{\lambda^2 \rho_e r_e}{2\pi}, \quad (2.10)$$

where  $\rho_e$  denotes the electron density and  $r_e$  is the the classical electron radius [1]. The values for  $\delta$  and  $\beta$  depend on the material and the photon energy and are tabulated for example in [34]. In contrast to the values of the refractive index of visible light, the refractive index of X-rays is smaller than unity. As the phase velocity of light in a medium is given by  $c_m = \frac{c_0}{n}$ , the speed of X-rays in matter is faster than the speed of light in vacuum  $c_0$ . The resulting wavenumber in a medium is then  $n \cdot k$ . A plane wave that is propagating through a medium in  $x$ -direction reads then as

$$\begin{aligned} E(x, t) &= E_0 \exp(i(nkx - \omega t)) \\ &= \underbrace{E_0 \exp(i(kx - \omega t))}_{\text{propagation in vacuum}} \cdot \underbrace{\exp(-i\delta kx)}_{\text{phase shift}} \cdot \underbrace{\exp(\beta kx)}_{\text{absorption}}. \end{aligned} \quad (2.11)$$

The phase shift is related to  $\delta$  and the absorption is related to  $\beta$ .

When X-rays propagating in a medium with the refractive index  $n_1$  enter a medium with refractive index  $n_2$  the propagation direction is altered. This phenomenon is known as refraction and is described by SNELL'S law

$$\frac{\cos \theta_1}{\cos \theta_2} = \frac{n_2}{n_1}. \quad (2.12)$$

The angles in the X-ray regime are usually measured between the propagation direction and the surface hit by the beam, in contrast to visible light optics. If  $\theta_2 = 0$ , X-rays impinging the surface under the corresponding

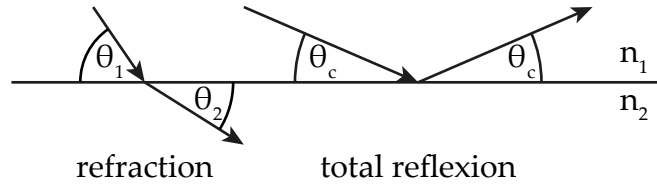


Figure 2.1: Refraction (left) and total reflexion (right) of an X-Ray beam propagating in a homogeneous medium with refractive index  $n_1$  hitting a homogeneous medium with refractive index  $n_2$  with  $\delta_1 < \delta_2$ .

angle  $\theta_c$  are totally reflected. Assuming medium 1 to be vacuum with  $n = 1$ , SNELL'S law yields

$$\cos \theta_c = n_2. \quad (2.13)$$

With an approximation of the cosine this leads to the formula of the critical angle

$$\theta_c = \sqrt{2\delta}. \quad (2.14)$$

The incoming X-ray beam is totally reflected, if it impinges on a surface under an angle  $\theta \leq \theta_c$ .

### 2.3 X-Ray Waveguides

In the previous section the propagation of X-rays in bulk material was summarised. In this section the propagation in thin layers and channels will be elucidated.

A thin layer of thickness  $d$  is composed of a material with the refractive index  $n_1 = 1 - \delta_1 - i\beta_1$ . The layer is surrounded by a cladding material with the refractive index  $n_2 = 1 - \delta_2 - i\beta_2$ , see Figure 2.2. The thickness of the cladding is assumed to be much larger than the layer thickness  $d$ . The width of the layer and the cladding in  $y$  direction is also assumed to be large compared to their length  $l$ , so that the problem can be treated in two dimensions. A plane wave  $\Psi$  propagating in  $x$  direction impinges perpendicularly to the front side and enters the layer and the cladding. Due to the differences in  $\delta$  of the two materials the wave is guided inside the layer, whilst the evanescent tails of the wave propagating in the cladding are stronger damped because of its larger absorption coefficient  $\mu$ . This arrangement of cladding and guiding layer forms a *planar waveguide*.

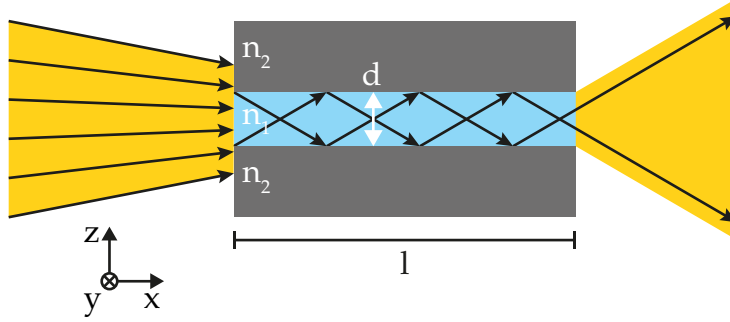


Figure 2.2: Schematic of a planar waveguide. The guiding core of the thickness  $d$  with refractive index  $n_1$  is surrounded with the cladding material of refractive index  $n_2$ .

The distribution of the electrical field inside a planar waveguide can be computed using the HELMHOLTZ equation [21, 57]

$$\Delta\Psi + k_0^2 n^2(z) \Psi = 0 \quad (2.15)$$

where  $\Delta$  denotes the LAPLACE operator and  $k_0 = \frac{2\pi}{\lambda}$  the vacuum wave-number.  $n(z)$  describes the location dependent refractive index, encoding the layered structure of the planar waveguide as

$$n(z) = \begin{cases} n_2 & \text{if } z < 0 \\ n_1 & \text{if } 0 \leq z \leq d \\ n_2 & \text{if } z > d \end{cases} \quad (2.16)$$

according to Figure 2.2. With the ansatz  $\Psi(x, y, z) = \psi(z) \exp(i\zeta x)$  this leads to the differential equation [27]

$$\psi''(z) + \left(k_0^2 n^2(z) - \zeta^2\right) \psi(z) = 0 \quad (2.17)$$

with  $\zeta$  being a propagation constant. The absorption coefficient  $\beta$  is typically two orders of magnitude smaller than  $\delta$  and is thus neglected in the following calculation, along with the  $\delta^2$  of  $n^2$  terms. With these assumptions (2.17) leads to the eigenvalue problem

$$\psi''(z) - 2\delta(z)k_0^2 \psi(z) = \left(\zeta^2 - k_0^2\right) \psi(z). \quad (2.18)$$

In the case of a planar waveguide, as described above, a solution of (2.18)

can be written as [21]

$$\Psi(z) = \begin{cases} A \exp(\gamma x) & \text{for } -\infty < z < 0 \\ A \cos(\kappa z) + B \sin(\kappa z) & \text{for } 0 \leq z \leq d \\ C \exp(-\gamma(z-d)) & \text{for } d < z < +\infty \end{cases} \quad (2.19)$$

with  $\kappa = \sqrt{k_0^2 n_1^2 - \zeta^2}$  and  $\gamma = \sqrt{(n_1^2 - n_2^2)k_0^2 - \kappa^2}$ . The latter condition along with the transcendental equation  $\tan(\kappa d) = 2\kappa\gamma/(\kappa^2 - \gamma^2)$  derived from the continuity conditions at the interfaces determines the modes. Solving the transcendental equation, one obtains a fixed number  $N$  of resonant modes at parameters  $\kappa_N$ , where  $N$  only depends on  $\delta_1$ ,  $\delta_2$ , and  $d$ . An important limiting case is the critical guiding layer width  $W_c = \pi/k_0\sqrt{2\delta}$  of a waveguide with ideal interfaces and vacuum guiding layer, at which the waveguide supports only a single mode, forming a fundamental length scale of X-ray waveguide optics. It fixes the minimum waveguide width  $d_{\min}$  of the highest possible wave confinement, e.g., for rectangular waveguides  $d_{\min} = W_c/\pi$  [8, 65]. The intensity distribution of the wave broadens both for larger and smaller  $d_{\min}$ , since the evanescent waves in the cladding become more pronounced, if  $d$  is reduced below  $d_{\min}$ .

The damping of the evanescent wave can be reduced, if an additional layer with  $\delta_1 < \delta_2 < \delta_{\text{inter}}$  and a smaller attenuation than the cladding material is inserted between the guiding layer and the cladding [78]. As the evanescent wave is less attenuated in the interlayers, the total intensity of the exiting wave is increased.

The exiting wave of planar waveguides described above is confined in one direction, i.e., it can be approximated as a cylindrical wave. To get an exiting wave that is confined in two directions the cladding has to delimit in two directions forming a guiding core of a *channel waveguide*. For this geometry the HELMHOLTZ equation has to be extended to three dimensions. Its solutions depend on the shape of the guiding core. There is no general analytical solution for arbitrary shapes [21]. However, the field distribution can be gained from numerical simulations, as demonstrated in [21, 43].

The propagation through a channel waveguide is tolerant to curvatures as long as the incidence angle of the propagating wave onto the interface between the guiding core and the cladding is smaller than the critical angle  $\theta_c = \sqrt{2\delta}$  [21]. This allows the realisation of a pair of curved waveguides as sketched in Figure 2.3. At the entrance, the two waveguide channels are separated by the distance  $d_{\text{in}}$ . Along the propagation their separation is increased to the distance  $d_{\text{out}}$  at the exit of the waveguides.

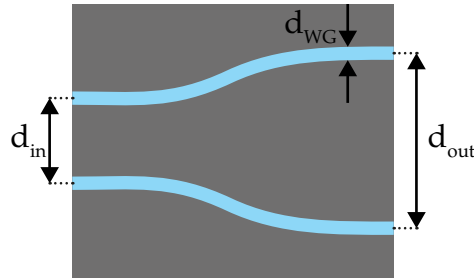


Figure 2.3: The distance  $d_{in}$  at the entrance of two waveguides can be increased with a slight curvature of the channels to an distance  $d_{out}$  at the waveguide exits.

When  $d_{in}$  is within the coherence length of the coupling beam, the coherence is preserved, such that the two exiting waves interfere and form a YOUNG pattern. Its fringe spacing is related to the distance  $d_{out}$ . According to their layout, they are called *Y-waveguides*. Their properties and applications will be discussed in the following section 2.4.

Channel waveguides have been successfully fabricated with electron beam lithography in two variants [21, 41]. In the first variant several waveguides with different channel width were exposed by electron beam lithography into a layer of resist. After the development, free standing structures of the resist polymers remained and were evaporated with a layer of the same material of the substrate, e.g., Si or Ge. The evaporated substrate was then cleaved to the desired length  $l$  of the waveguides. In a last step the front and exit faces were cleaned from debris and polished with a focussed Ga ion beam (FIB).

The second variant exposes the positive structure of the waveguides into the resist, such that the areas for the waveguide channels are free from resist after the development step. The remaining resist acts as an etching mask for reactive ion etching. In the areas without the resist, the reactive ions hit the substrate and etch anisotropically the channels into it. After the removal of the resist, the structured substrate is attached onto a clear substrate by wafer bonding. The waveguides are then cut to the desired length  $l$  by wafer dicing. Finally the channel faces are also cleaned by the FIB.

The advantage of the channel waveguides fabricated by reactive ion etching and wafer bonding is the air-filled guiding core. This decreases the attenuation of the guided wave compared to polymer-filled channels. The fabrication of bonded waveguides is more delicate because the step

of wafer bonding is very sensitive to the quality, waviness, roughness and cleanness of the used substrates, resulting in blocked channels on the waveguide chip.

A beam confinement in two directions can also be realised by the crossing of two planar waveguides [43]. The fabrication process of planar guiding layers allows the insertion of interlayers and a better control of their thickness and quality compared to the cross section of channel waveguides fabricated by electron beam lithography. This gives then the possibility to a beam confinement down to the theoretical limit of monomodal propagation through each guiding layer [43]. If a sample is placed closely to such a *crossed waveguide*, it has significantly different distances  $x_1$  to the first and the second planar waveguide. As the magnification  $M = \frac{x_1+x_2}{x_1}$  depends strongly on  $x_1$ , the sample hologram is magnified differently in the horizontal and the vertical direction. This astigmatism has then to be taken into account for the holographic reconstruction.

## 2.4 Reference Beam Holography for Interferometric Applications

The intensity distribution in the far field of a coherently illuminated double slit is known as YOUNG pattern. An equivalent intensity distribution can be observed, when Y-waveguides are illuminated coherently. This YOUNG pattern is the result of the interference of the two waves  $R$  and  $S$  exiting from the waveguides with a distance  $d_{\text{out}}$ , see Figure 2.4. As the exit diameter of the waveguide guiding core  $d_{\text{WG}}$  is small compared to their distance  $d_{\text{out}}$  the exiting waves can be assumed to be spherical waves in the detection plane. The finite size of the waveguide  $d_{\text{WG}}$  results in an enveloping function of the YOUNG pattern corresponding to the diffraction pattern of a slit with the diameter  $d_{\text{WG}}$ . The electrical field  $E(0, y, z, t)$  in a point  $P(0, y, z)$  in the detection plane is then the superposition of the two waves  $R(r, t) = R_0 \exp(i(kr - \omega t + \varphi_1))$  and  $S(r, t) = S_0 \exp(i(kr - \omega t + \varphi_2))$

$$E(0, y, z, t) = R(s_1, t) + S(s_2, t). \quad (2.20)$$

For measured intensity it follows

$$\begin{aligned} I &= |E^2| = \langle EE^* \rangle = \langle (R + S)(R + S)^* \rangle \\ &= R_0^2 + S_0^2 + 2RS \cos(k(s_2 - s_1) + \varphi_2 - \varphi_1). \end{aligned} \quad (2.21)$$



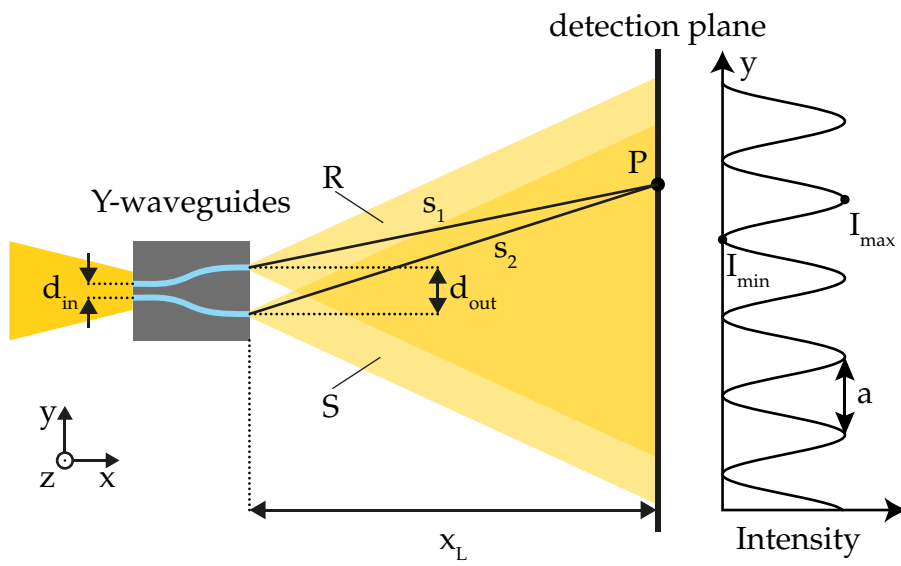


Figure 2.4: Schematic setup for a two beam X-ray interferometer based on Y-waveguides. The interference of the exiting waves  $R$  and  $S$  form a YOUNG pattern in the detection plane. The recorded intensity shows fringes with the spacing  $a$ . From the intensity values  $I_{max}$  and  $I_{min}$ , the contrast  $K$  can be calculated, indicating the degree of coherence of  $R$  and  $S$ .

The intensity thus depends on the path difference  $s_2 - s_1$ . The intensity in an arbitrary point depends on its coordinates in the detection plane. From the geometry shown in Figure 2.4,  $s_{1,2}$  are given by

$$s_1 = \sqrt{x_L^2 + \left(y - \frac{d_{\text{out}}}{2}\right)^2 + z^2}$$

$$s_2 = \sqrt{x_L^2 + \left(y + \frac{d_{\text{out}}}{2}\right)^2 + z^2}.$$

If the distance  $x_L$  between the waveguide and the detection plane is much larger than the distance between the waveguide exits  $d_{\text{out}}$ , i.e.,  $\frac{d_{\text{out}}}{x_L} \ll 1$  and the size of the detector is small compared to  $x_L$ , i.e.,  $\frac{y}{x_L} \ll 1$  and  $\frac{z}{x_L} \ll 1$ ,  $s_{1,2}$  can be approximated as

$$s_{1,2} = x_L \sqrt{1 + \frac{z^2}{x_L^2} + \left(\frac{y \mp \frac{d_{\text{out}}}{2}}{x_L}\right)^2}$$

$$\approx x_L \left(1 + \frac{z^2}{2x_L^2} + \frac{\left(y \mp \frac{d_{\text{out}}}{2}\right)^2}{2x_L^2}\right).$$

The difference  $s_2 - s_1$  then reads as

$$s_2 - s_1 = \frac{y d_{\text{out}}}{2x_L} + \frac{y d_{\text{out}}}{2x_L} \quad (2.22)$$

$$= \frac{d_{\text{out}}}{x_L} y. \quad (2.23)$$

The intensity in the detector plane depends also on the amplitudes  $R_0$  and  $S_0$ . As  $s_1 \approx s_2 \approx x_L$  the amplitude fall-off in the detector plane is nearly equal, leading to

$$A = R_0(0, y, z) = S_0(0, y, z). \quad (2.24)$$

Equation (2.21) can then be written as

$$I(0, y, z) = 2A^2 \left(1 + \cos\left(\frac{2\pi d_{\text{out}}}{\lambda x_L} y + \varphi_2 - \varphi_1\right)\right)$$

$$= 4A^2 \cos^2\left(\frac{\pi d_{\text{out}}}{\lambda x_L} y + \frac{1}{2}(\varphi_2 - \varphi_1)\right). \quad (2.25)$$

The fringes are separated by

$$a = \frac{\lambda x_L}{d_{\text{out}}}. \quad (2.26)$$

The pattern has a periodicity of  $\pi$ . The location of the fringes in the detection plane depends on the relative phase difference  $\Delta\varphi = \varphi_2 - \varphi_1$  between the two exiting waves. A change in the relative phase difference results in a shift of the YOUNG pattern. The change can be induced by a modified illumination of the waveguide entrances, or a sample passing one of the exiting waves  $R$  or  $S$ . If the illumination is constant the observation of a pattern shift gives information on the phase shift of a sample. This is a realisation for the hard X-ray regime of the MACH-ZEHNDER interferometer for visible light [33].

The contrast

$$K = \frac{I_{\text{max}} - I_{\text{min}}}{I_{\text{max}} + I_{\text{min}}} \quad (2.27)$$

of the maximum intensity  $I_{\text{max}}$  and the minimum intensity  $I_{\text{min}}$  of a fringe indicates the degree of the coherence  $|\gamma|$  of the exiting waves  $R$  and  $S$  with equal amplitudes  $R_0 = S_0$ , see Figure 2.4 [45]. From the contrast  $K$  of the YOUNG pattern of Y-waveguides the degree of coherence at the waveguide entrance can be deduced, assuming that the degree of coherence is not altered by the curved waveguides [60, 61]. The design scheme corresponds to an *optical stethoscope* and enables the quantification of the degree of coherence  $|\gamma|$  of the coupling wave at the two entrance positions of the waveguides from the contrast  $K$ .

## 2.5 Inline Holography and Iterative Reconstruction Approaches

The principle of inline holography is based on the interference of a reference wave  $R$  and a wave  $S$  scattered by a sample when illuminated with the reference wave  $R$  [24], see Figure 2.5a for a schematic drawing. An area detector placed at the distance  $x_{\text{eff}}$  from the sample measures the intensity

$$\begin{aligned} I &= |R + S|^2 \\ &= RR^* + R^*S + RS^* + SS^* \end{aligned}$$

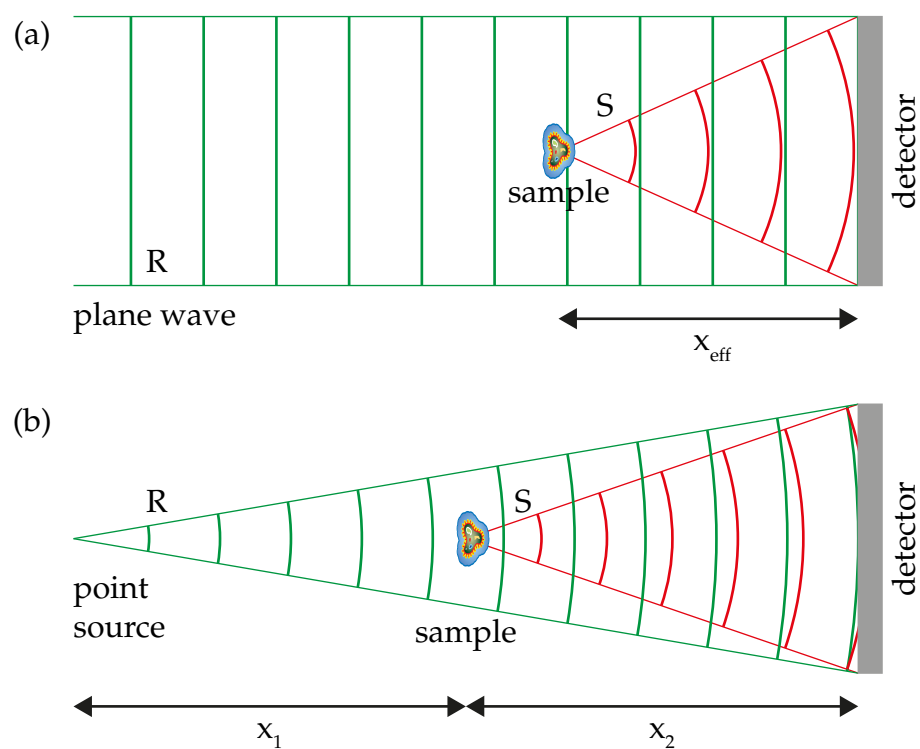


Figure 2.5: (a) Schematic setup for inline holography with a plane wave and a sample at a distance  $x_{\text{eff}}$  before the detector. (b) Schematic setup for inline holography with a divergent beam emitted by a point-like source. The sample is placed at distances  $x_1$  behind the source and  $x_2$  before the detector.

formed by the coherent superposition of the reference wave  $R$  and the scattered wave  $S$ . For inline holography using visible light the intensity is often recorded with photographic film, whose positive is re-illuminated with the reference wave for image reconstruction [45]. In the X-ray regime the re-illumination with the reference wave is rather difficult and unpractical. Therefore,  $I$  is recorded with a pixel-based area detector, e.g., a CCD, and the wavefield in the sample plane is reconstructed by numeric back-propagation. In the plane wave geometry illustrated in Figure 2.5a the resolution of the reconstructed wavefield is limited by the pixel size of the detector, as the reconstructed image is not magnified. A magnification of the sample can be achieved when it is illuminated with the divergent beam emitted by a point-like source, e.g., a channel waveguide, see Figure 2.5b. The recorded hologram is then magnified by

$$M = \frac{x_1 + x_2}{x_1},$$

where  $x_1$  denotes the distance between the source and the sample and  $x_2$  the distance between the sample and the detector [14, 21]. The geometry of a divergent beam can be transformed into an equivalent effective geometry [14, 21], as long as the Fresnel approximation is valid [25]. In the effective geometry the sample is illuminated with a plane wave and it is placed at

$$x_{\text{eff}} = \frac{x_1 x_2}{x_1 + x_2}$$

in front of the detector. The effective pixel size  $d_{\text{eff}}$  is thus the magnified pixel size of the detector  $d_{\text{eff}} = \frac{d_{\text{real}}}{M}$ . A hologram recorded in the divergent beam geometry is first transformed into the effective geometry and then reconstructed.

The wavefield at the sample plane  $E_s$  is proportional to

$$E_s \propto RRR^* + RR^*S + RRS^* + RSS^*$$

when re-illuminated with the reference wave  $R$  for reconstruction. The first term gives a constant background corresponding to the directly transmitted beam. The last term is very small for thin and weak scattering samples, and is thus negligible for most samples examined with hard X-rays. The information of the sample wave  $S$  is encoded in the second term. The third term represents a wave conjugated to the sample wave,

forming a conjugated or a so called *twin image*, that blurs the actual image of the sample. To compute numerically the wavefield in the sample plane the recorded intensity  $I$  is backpropagated from the detector plane to the sample plane with the nearfield propagator [43]

$$\mathcal{D}_x = \mathcal{F}^{-1} \exp \left( i x \sqrt{k^2 - k_y^2 - k_z^2} \right) \mathcal{F},$$

where  $\mathcal{F}$  denotes the Fourier transformation and  $\mathcal{F}^{-1}$  its inverse transformation,  $k$  is the wave number and  $k_y, k_z$  are the components of the wave vector  $\vec{k}$  in the corresponding direction.

The twin image problem can be successfully treated by iterative sample reconstruction algorithms. Iterative algorithms are widely used in the field of coherent X-ray diffractive imaging (CXDI) for recovering the phase information of the sample that is lost at the measurement of the diffracted intensity [55]. In the recent years several algorithms with different approaches and requirements have been proposed and successfully applied [64, 55]. They are based mainly on the same idea: The initially unknown wavefield with its phase information is refined in each iteration step by applying constraints and a priori information, e.g., sample size and thickness or the known amplitude in the detector plane, to the iterated wavefield until the calculated diffraction pattern is in close congruence with the measured diffraction pattern [64]. In contrast to CXDI the phase information is preserved and encoded in the measured intensity in inline holography, but spoiled by the twin image. A significant improvement of the reconstruction can then be achieved by application of a modified version [26] of the classical Hybrid-Input-Output (HIO) algorithm [20] for pure phase objects, which can be regarded as an extension of the scheme proposed in [30], where an update of the current amplitude  $|\chi_n|$  is suggested according to

$$|\chi_{n+1}| = |\chi_n| - \beta \cdot (|\chi'_n| - 1),$$

slowly pushing  $|\chi_n|$  towards 1.  $|\chi'_n|$  denotes the amplitude of the  $n$ -th iterate after application of the detection plane constraint, i.e.,  $\chi'_n := P_M(\chi_n)$  with  $P_M(\chi_n) = \mathcal{D}_{-x_{\text{eff}}}[\sqrt{I} \cdot \varphi(\tilde{\chi}_n)]$  denoting the modulus replacement operation in the detection plane and  $\varphi(z) := \arg(z)$  for any  $z \in \mathbb{C}$ . For the modified HIO, a phase constraint is added to the update of the amplitude, namely

$$\varphi(\chi_{n+1}(y, z)) = \begin{cases} \varphi(\chi_n(y, z)) - \gamma \cdot \varphi(\chi'_n(y, z)) \quad \forall (y, z) \notin S \\ \min\{\varphi(\chi'_n(y, z)), 0\} \quad \forall (y, z) \in S. \end{cases}$$

The support area  $S$  can be determined very accurately from a single-step holographic reconstruction. Similar to the amplitude constraint the phase constraint causes a gentle decrease of the phase to a constant  $C$  in the area, where no object is located. The phase inside the support area, however, is left untouched, as long as it is not larger than  $C$ , allowing for phase changes  $\Delta\varphi(y, z)$  in one direction only, as expected for objects with  $|\Delta\varphi(y, z)| < \pi$ . The speed of convergence is determined by the feedback parameters  $\gamma \in [0, 1]$  and  $\beta \in [0, 1]$ .





# Chapter 3

## Conceptual Experiments

### 3.1 X-Ray Interferometry with Waveguides<sup>1</sup>

X-ray interferometry can detect phase shifts corresponding to optical path length differences down to the picometre scale [64]. However, the phase shift between reference and object wave is averaged over large lateral distances. In the case of BONSE-HART interferometry, the two beams with typical cross sections in the range of  $(0.1 \text{ mm} - 10 \text{ mm})^2$  are created by BRAGG reflection, separated macroscopically, and are finally brought into superposition again by further reflections [10], a scheme which can hardly be scaled down. In grating interferometry, it is not the beam size but rather the grating period and the pixel size of the CCD detector, that defines the lateral size, over which phase information is averaged, with  $10 \mu\text{m}$  as a typical length [67, 66]. The possibilities of X-ray interferometry with highly collimated beams emitted from a pair of channel waveguides with lateral cross-sections of about  $60 \text{ nm}$  are explored here. The lateral distance  $d$  between the channels can be varied along the optical axis. This design scheme corresponds to a *phase stethoscope*, and enables the quantification of the mutual coherence correlation function  $\gamma_{12}$  of the fields at the two input positions, from the phase shift, and the contrast  $K$  of the YOUNG interference pattern, recorded in the far field of the channel output positions. The quantification of lateral coherence and of temporal wavefront drifts in X-ray optical systems are important issues in X-ray focusing [79] and lensless X-ray imaging [50, 82] applications.

An application of this novel type of X-ray interferometre is in the field

---

<sup>1</sup>This section is based on an unpublished manuscript by S. Kalbfleisch and T. Salditt.

of X-ray photon correlation spectroscopy (XPCS). Slow dynamics such as Brownian motion and diffusion of small colloidal particles on length scales inaccessible to light scattering can be probed by XPCS [15], as well as unlabelled and opaque systems. However, for many applications the decay of scattering intensity with momentum transfer  $\mathbf{q}$  is so strong that the flux does not allow one to extract the temporal correlation function  $g(t)$  from the intensity fluctuations. Heterodyne mixing of the primary and a scattering wave [31] can boost the signal of a weakly scattering sample far above the background level and may render short length scale dynamics observable. Instead of full image reconstruction, the analysis of shifts in the far field pattern yields spatially resolved dynamical information. Translations of nanometre-sized test objects within the cone of a coherent waveguide beam were resolved with a translational accuracy of 10 nm [57], based on the analysis of shifts in the inline hologram. Here, the idea of phase-sensitive detection in a projection geometry to a full two-beam interferometry setup is generalised, where the focussed X-ray beam is coupled into a lithographic double waveguide structure.

This device is used to study the temporal phase drift on time scales of seconds and longer, and to compare the phase trajectories of far field patterns measured with and without additional propagation of the beam through a colloidal suspension.

The basic scheme of the nanoscopic X-ray interferometer is sketched in Figure 3.1: The sample is illuminated by either (i) one, (ii) one out of two, (iii) or two beams emitted by a single [37] or a pair of waveguide channels [21]. By diffraction the beams diverge and interfere in the far field, forming the classical YOUNG double slit pattern with an envelope function given by the squared Fourier transform of the waveguide mode structure at the exit of the waveguides. After passing through the sample located at an adjustable distance  $x_1$  downstream from the source, the wave propagates over a flight path  $x_2$  and is recorded at a distance  $x_L = x_1 + x_2$  by a CCD camera. Importantly, the two beams should not overlap in the sample plane located at propagation distance  $x_1$  downstream from the waveguide. This requirement can be easily met by control of the waveguide separation  $d_{\text{out}}$ . This distance can be controlled precisely by lithography and several geometries are realisable on the same chip. Simultaneously, the constraints imposed by the lateral coherence length in the focal plane of the pre-focusing optics are fulfilled by curving the waveguides channels. As in any other two-beam interferometer, the phase information encoded in the wave front is then converted to intensity contrast.

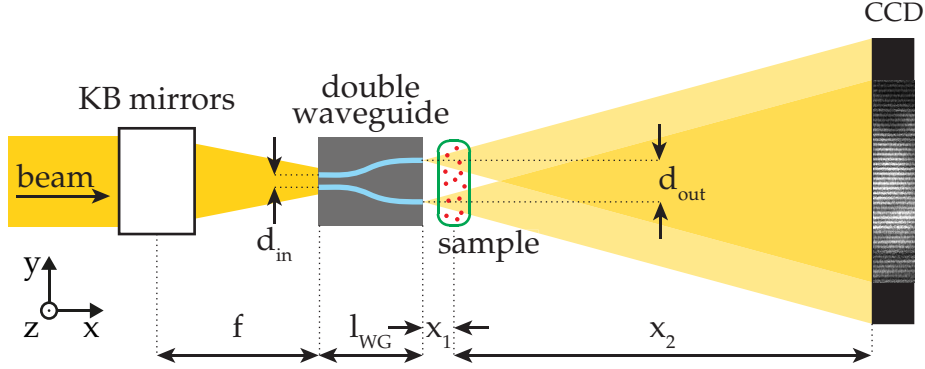


Figure 3.1: Sketch of the experimental setup. The entrance of the Y-shaped double waveguide is placed in the focal spot of the Kirkpatrick-Baez mirrors. The two channels are located in close proximity  $d_{in}$  on the front side, adapted to the smaller lateral coherence length of the focussed beam. The channels are widened to a spacing of  $d_{out}$  at the exit of the beams. The YOUNG interference pattern is recorded by a CCD at distance  $x_1 + x_2$  downstream. In addition, a sample can be placed into the beams at a variable defocus distance  $x_1$ .

Assuming two identical waveguide channels illuminated by a fully coherent beam and the same intensity in each path, the far field pattern can be written as

$$I(q) = 2 I_{WG}(q) (1 + K \cdot \cos(\theta + \Delta\varphi)), \quad (3.1)$$

where  $\theta$  denotes the relative phase shift due to the path length differences between the two waveguides,  $\Delta\varphi$  the phase shift induced by a sample, and  $I(q) \propto |\mathcal{F}[\psi]|^2$  is the squared Fourier transform of the near field distribution in the waveguide channels. The spacing between the interference fringes on the CCD is then given by  $d_Y = \lambda x_L / d_{out}$ . In the case of finite mutual coherence of the two beams the experimental contrast between minima and maxima  $K = (I_{max} - I_{min}) / (I_{max} + I_{min})$  is reduced with respect to the ideal case  $K = 1$ . A reduced  $K$  can also result from a temporal average of different phase shifts.

The waveguides used in the experiment were fabricated with electron beam lithography into a polymer (calixarene, Allresist XAR-N7600/2) as guiding core in Si [37, 38]. The wafer was cleaved to a length of  $l_{WG} = 4.1$  mm. The diameter of the guiding core in the vertical direction was given by the thickness of the photoresist, which was about 90 nm before development, and has probably shrunk during further processing.

In the horizontal direction the diameter of each channel was about 60 nm. Given these values and materials, the channels are thus not yet monomodal. However, field calculations [23] indicate that the field at the end faces is dominated by the fundamental mode of approximately Gaussian shape.

The holography experiment was carried out at the ID22 undulator beamline of the European Synchrotron Radiation Facility in Grenoble (France). The photon energy was 12 keV, and the beam was focussed by two perpendicularly-crossed Kirkpatrick-Baez mirrors. The front faces of the waveguides were placed in the focal spot with cross sectional dimensions of about  $3.7 \mu\text{m} \times 8 \mu\text{m}$ .

Out of the several double waveguide channels fabricated on the same chip, data are shown taken with a double-channel waveguide with  $d_{\text{in}} = 100 \text{ nm}$  and an exit spacing of  $d_{\text{out}} = 3 \mu\text{m}$ . Even in the limiting case of considering the focusing mirrors as a completely incoherent source, the value of  $d_{\text{in}} = 100 \text{ nm}$  is chosen to accept two beams out of a laterally coherent wave front. The distance in the focal plane roughly matches to the corresponding lateral coherence length  $d_{\text{in}} \simeq \lambda f / (2R_s)$ , with  $f = 0.19 \text{ m}$  the focal length of the KB mirror and  $R_s \simeq 0.3 \text{ mm} - 0.5 \text{ mm}$  the geometric acceptance, depending on the slit settings in front of the KB mirror. The holograms were recorded using a back-illuminated CCD camera (Princeton Instruments LCX,  $20 \mu\text{m} \times 20 \mu\text{m}$  pixel size) placed  $x_L = 2.95 \text{ m}$  downstream of the waveguides.

In the next step a set of sample and empty beams was measured, with a data accumulation protocol as follows: for each sample as well as for two empty beams a series of 150 images with a single exposure time  $t = 10 \text{ s}$  was recorded. Figure 3.2 shows a typical raw image of a far field pattern with gold colloid sample in the beam and the data reduction scheme.

The data was treated as follows: Each raw image was corrected by a dark image of same exposure time. Afterwards a region of interest (ROI) was cut out. The intensity values inside the ROI were summed up in vertical direction to an intensity profile (column sum). From a spline interpolation of the intensity profile the maxima positions were derived. The mean contrast of an intensity profile was calculated from the averaged contrast  $K$  for each pair of maximum and minimum of the column sum. The integration of all pixel intensities gave the overall intensity of the ROI.

Colloidal suspensions of varied number concentration  $\rho_{\text{col}}$ , radii  $R$ , and water/glycerol (Fluka) ratio were prepared from Gold colloids (British

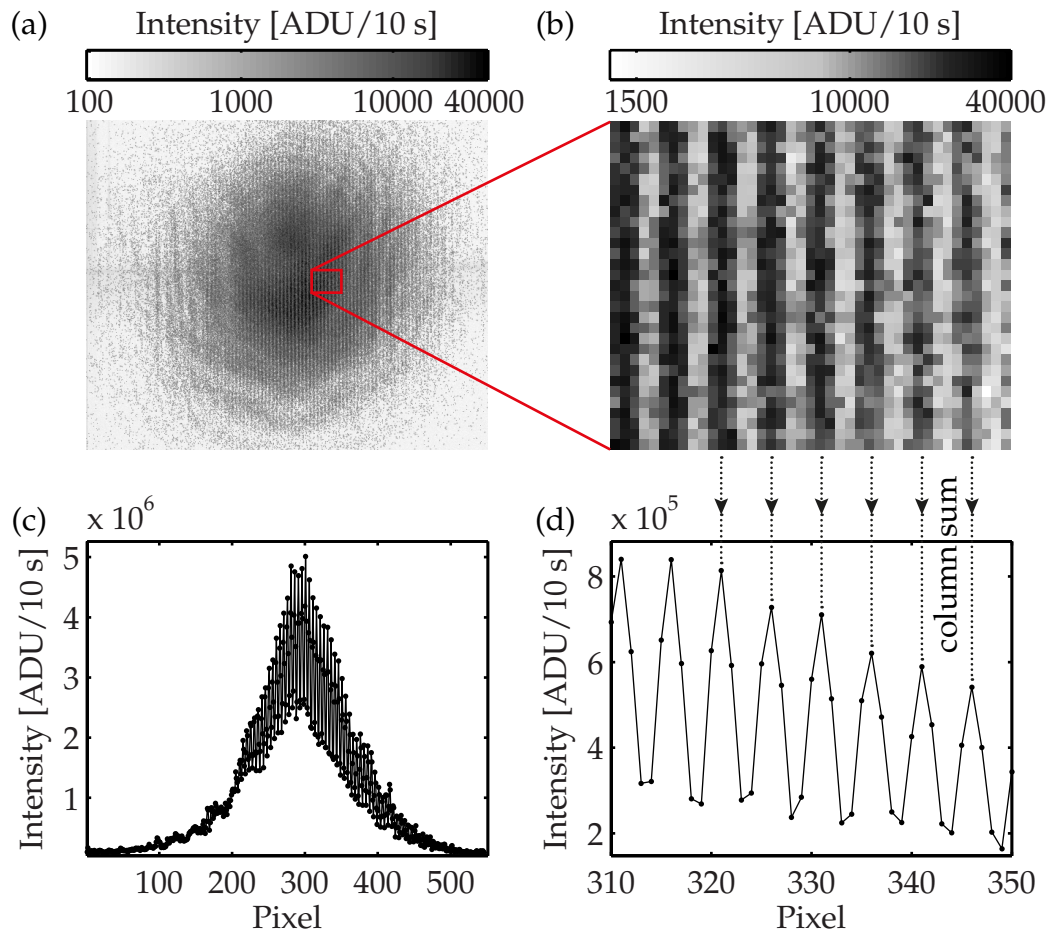


Figure 3.2: Typical data reduction. **(a)** Far field pattern of the Y-waveguide with gold colloid sample of radius  $R = 100$  nm in the beam. **(b)** Overall beam shape of the Y-waveguide derived from vertical summation of the pixel values of (a), given in analog-to-digital units (ADU). **(c)** Same as in (a) but zoomed into a ROI in the central part of the far field pattern, used for further data analysis. **(d)** The data from (c) after vertical summation.

Table 3.1: List of measured samples with their colloid radius  $R$ , viscosity  $\eta$ , particles per volume  $\rho_{\text{col}}$ , the average particle number  $N_c$ , and the typical residence time  $\tau$  in the waveguide beam.

Sample	$R$ [nm]	$\eta$ [mPa s]	$\rho_{\text{col}}$ [1/ml]	$N_c$	$\tau$ [s]
Au100-6	50	310	$1.23 \times 10^{10}$	10	18.0
Au100-8	50	98	$3.04 \times 10^9$	4.5	5.7
Au200-6	100	310	$1.54 \times 10^9$	2	36.1

Biocell International), covering a wide range in concentration  $\rho_{\text{col}}$ , viscosity  $\eta$ , size  $R$ , and corresponding diffusion time scales  $\tau$ . The colloid suspensions were sealed in pulled borosilicate capillaries with filament (Hilgenberg) and positioned at a distance  $x_1 \approx 1$  mm behind the waveguide, where the beam size of the two waveguide beams was broadened to about  $\Delta d = 1$   $\mu\text{m}$ , due to the intrinsic divergence. The average number  $N_c$  of colloids in the beam was then calculated from the concentration  $\rho_{\text{col}}$ , the lateral beam size  $\Delta s$ , and the diameters of the capillaries, which were in the range of 250  $\mu\text{m}$  – 460  $\mu\text{m}$ , as determined by scanning each capillary through the beam. The typical residence time  $\tau = 3\pi\eta R\Delta s^2 / (2k_b T)$  of a single colloid in the beam results from its Brownian motion at room temperature. The sample parameters corresponding to the curves shown in Figure 3.3 are tabulated in Table 3.1.

The empty beam setup can be used to characterise the optical system itself, and in particular the stability of the focusing scheme. Strong drifts are observed over the data accumulation run. Such a behaviour is unfortunately not unusual for highly focussed synchrotron beams and presents a tremendous challenge to hard X-ray nanoprobe experiments. Small drifts, both rotational and translational, of the incident beam relative to the waveguide entrance, or equivalently in the waveguide positioning, led to significant intensity fluctuations, which were higher than the fluctuations of the integrated KB beam, typically by a factor of 100.

In addition to the intensity trace, a phase trajectory, and a contrast trajectory are measured interferometrically, as shown in the top row of Figure 3.3. The phase trajectories  $\Delta\varphi(t)$ , plotted in pixel units with  $\pi = 5.078$  pixel, show total variations of up to about  $3\pi$ . The corresponding path length differences in the two waveguide beams of up to  $1.5\lambda$  could result from the drifts of the focussed beam relative to the waveguide. The dominating contribution of the phase shift must be due to drifts in the

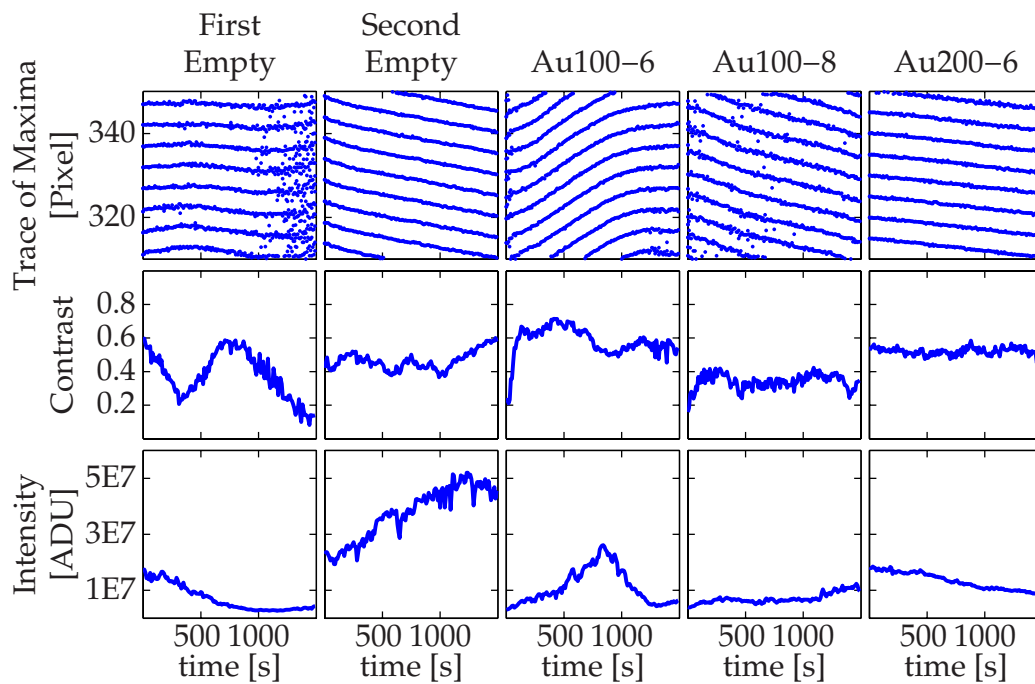


Figure 3.3: Typical results for colloidal samples and empty beams: the trace indicates the position of the maxima on the CCD over time (top row). The centre row shows the mean contrast  $K$  along with the intensities integrated over the ROI plotted in the bottom row.

optical system, since the phase and contrast curves look quite similar for the runs with and without sample. However, for all five runs the time scales of the phase fluctuations are much slower than those of the intensity and contrast functions. The rather smooth behaviour of  $\Delta\varphi(t)$  should enable the detection of short time scale phase fluctuations due to path length differences induced by sample dynamics.

The highest sensitivity to small phase shifts  $\Delta\varphi$  is given at the position of maximum slope (positive or negative) of the YOUNG pattern. The smallest observable phaseshift  $\Delta\varphi_{\min}$  can be estimated from equation (3.1), regarding the Poissonian noise by  $I_0 + \sqrt{I_0} \leq I_{\Delta\varphi} - \sqrt{I_{\Delta\varphi}}$ , where  $I_0$  denotes the intensity of an empty beam at the slope of the cos-function, and  $I_{\Delta\varphi}$  the intensity of a phase shifted beam by  $\Delta\varphi_{\min}$ , respectively. This leads to

$$\Delta\varphi_{\min} \simeq \frac{2}{K\sqrt{2I_0}}. \quad (3.2)$$

A typical contrast of  $K \approx 0.6$  and an integral intensity  $I_{\text{WG}} \approx 10^5$  photons/s (in all exploitable detector pixels located in the slopes), yield to  $\Delta\varphi_{\min} \approx 7$  mrad, which matches roughly the effective phase shift of a single gold colloid with  $R = 100$  nm in a waveguide beam with a diameter of  $\approx 1$   $\mu\text{m}$ . The time resolution is mainly limited by the detector readout of about 1 s. Studies of very fast dynamics would require an additional absorption grating of the same period as the interference pattern adjusted in front of a fast avalanche photo diode [67]. Given future technical improvements, notably vibration isolation, and the fast pixel detectors which are presently becoming available [42, 68], the waveguide based X-ray interferometer presented here can be used to study dynamical properties in sample volumes, which are smaller by several orders of magnitude with respect to present capabilities, e.g., in X-ray photon correlation spectroscopy (XPCS) [15, 31], down to single particle heterodyne photon correlation spectroscopy.

## 3.2 X-Ray Inline Holography

The concept of inline holography with X-ray waveguides has already been demonstrated [21, 57], but was dominated by the twin image problem. An improvement by the suppression of the twin image can be expected from combining inline holography with iterative methods [25]. Compared to reference beam holography, which requires a free area beside the sample



for an undisturbed reference wave, inline holography can be easily applied to extended samples without the restriction of a free area, giving also the possibility of scanning microscopy techniques, such as ptychography [71].

The demonstrating experiments of iterative inline holography with X-ray waveguides were carried out at the ID22NI beamline of the European Synchrotron Radiation Facility (ESRF) at a photon energy of 17.5 keV [9, 18]. The multilayer mirrors of the Kirkpatrick-Baez system gave a focus size of  $129 \text{ nm} \times 166 \text{ nm}$  ( $h \times v$ ) monochromatising the pink beam of the undulator to  $\frac{\Delta E}{E} = 0.02$ . The limitation of the focal spot size due to the undulator source size is overcome by the generation of a secondary source [9, 18]. In the standard configuration the endstation is optimised for X-ray fluorescence mapping, phase contrast imaging, and scanning transmission X-ray microscopy [9, 18]. Therefore, the sample is mounted on a high-precision piezo stage behind the KB mirrors. As this standard setup did not allow the mounting and alignment of an X-ray waveguide and a sample at the same time, it had to be modified to accommodate the required motorised degrees of freedom. The used setup is shown in Figure 3.4a. The waveguide is mounted on a set of nanopositioners by *attocube systems AG*, providing xyz translations for the positioning of the centre of rotation in the focal spot. For the angular alignment two rotation stages (ANR50/RES, ANGt101/RES) are used. An additional translation stage (ANPx51/RES) on top of the rotations allows the placement of individual channels of a waveguide chip in the centre of rotation and thus in the focus. This set of nanopositioners is the predecessor of the waveguide stage for the instrument at the P10 beamline of PETRA III, presented in Chapter 4. The sample was placed on a set of yz piezo stages (Physik Instrumente P-625.1CD and P-622.ZCD). The piezo stages were placed on a set of xyz translations (Huber 5103.1, two MiCos MTS-65) for coarse positioning. Both the waveguide stage and the sample stage were mounted upright. To allow small distances between the waveguide and the sample a gibbet construction was used, resulting in an unforeseen space limitation for crossed waveguides, which were very short, see red circle in Figure 3.4b. The waveguide and sample pre-alignment was supported by two optical on-axis microscopes. The holograms were recorded with a MAXIPIX detector (pixel size  $55 \mu\text{m}$ ) [68] and a directly illuminated CCD (Roper Scientific LCX, pixel size  $20 \mu\text{m}$ ), both placed on a motorised detector stage at a distance of  $x_2 = 3.09 \text{ m}$  from the focus. Based on the experiences gained with the setup used at the ID22NI several improvements were implemented for the waveguide imaging instrument at the P10 beamline

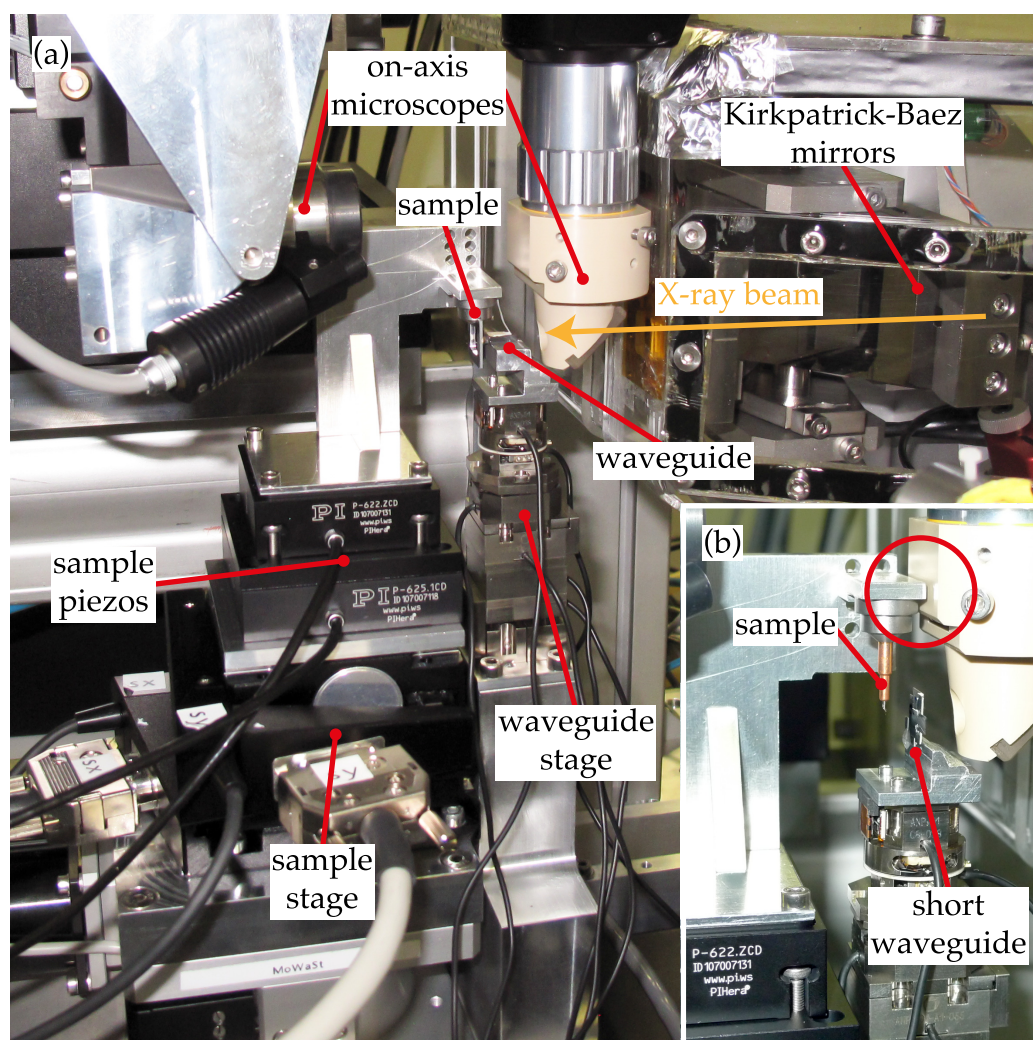


Figure 3.4: (a) Experimental setup at the ID22NI beamline of the ESRF used for waveguide-based inline holography experiments. The waveguide and sample stage were designed to allow the waveguide and sample alignment within the available space behind the Kirkpatrick-Baez mirrors. The pre-alignment of waveguides and samples was supported by two optical on-axis microscopes. (b) Resulting from the upright installation of the waveguide and sample stages, the minimal distance in between was limited for short waveguides, as indicated by the red circle.

of PETRA III. Due to the deficient accuracy of the waveguide positions with resistive encoders (ANR50/RES, ANGt101/RES, and ANPx51/RES) these positioners were replaced with new models with more precise optical encoders. The upright arrangement of both the waveguide and the sample stage led to space restrictions in the waveguide to sample distance. As a consequence the waveguide stage for the P10 setup was mounted upside-down. The gibbet construction would also block the X-ray beam when used for tomography measurements.

For waveguide-based inline holography a Siemens star on a high resolution chart (NTT-AT, model # ATN/XRESO-50HC, 500 nm Ta on SiC membrane) was imaged in the beam of a bonded Si channel waveguide with a length of  $l_{WG} = 13$  mm and an exit cross section of  $140 \text{ nm} \times 24 \text{ nm}$  ( $h \times v$ ). It was placed at a distance  $x_1 = 0.926$  mm behind the exit of the waveguide, as determined with the motorised optical on-axis microscope, giving a geometric magnification of  $M = \frac{x_1+x_2}{x_1} = 3338$ . The Siemens star was scanned on a grid of  $21 \times 21$  evenly spaced points with a step size of 250 nm giving a total scan area of  $5 \mu\text{m} \times 5 \mu\text{m}$ . The holograms were recorded at each scan point with the MAXIPIX detector. An example of a hologram recorded at a scan point is shown in Figure 3.5b. The data set was processed in two different approaches of holographic reconstruction and iterative ptychographic reconstruction. Intensity fluctuations and drifts were removed by the normalisation of the recorded holograms to their average intensity. The holographic reconstruction was then a single step back-propagation of each hologram to the sample plane, that were transformed into the effective geometry before, as described in [27]. The individual back-propagated reconstructions were then stitched corresponding to their scan position. The reconstructed phase of the holographic reconstruction is shown in Figure 3.5a, along with a zoomed area of the inner part of the Siemens star (Figure 3.5c) and a line profile across the zoomed area (Figure 3.5d). In the second approach the data set was reconstructed with the ptychographic method described in [27]. The retrieved phase information after 40 iterations is shown in Figure 3.6a, along with a magnified inner part of Siemens star (Figure 3.6c), a line profile across it (Figure 3.6d) and the amplitude of the probe in the sample plane retrieved from the ptychographic reconstruction (Figure 3.6b).

While the overall structure of the test pattern is recovered up to very fine details in the centre region with both reconstruction methods, the retrieved phase values are only in coarse agreement with the expected phase shift of 0.4 rad between the void areas and the Ta structure. In

the ptychographic reconstruction the retrieved phase difference between filled and void areas is generally smaller than the expected value, the situation is reversed in the holographic reconstruction. The holographic reconstruction also shows stronger artefacts and non-reconstructed spatial frequencies due to the phase contrast transfer function [77]. As visible in both approaches of reconstructions the spatial resolution is higher in vertical than in horizontal direction. There is a direct correspondence between this anisotropy and the lateral dimensions of the cross section of the waveguide guiding core, which confines the guided field in the vertical direction stronger than in the horizontal direction. The obtained resolution was estimated by the edge analysis line profile of the reconstructed phase across the inner structures of the Siemens star. For the holographic reconstruction an edge width of 36 nm (FWHM) was determined, and for the ptychographic reconstruction an edge width of 34 nm (FWHM).

The ptychographic reconstruction on an evenly distributed grid of scan points may introduce artefacts in the obtained sample image [25]. Therefore, a scan procedure was developed which scan points are located on an ARCHIMEDEAN spiral. The spiral is specified by the probe diameter  $p$ , the distance  $d$  between two subsequent scan points, and their overlapping ratio, as illustrated in Figure 3.7.

Using such a pattern of spiralled scan points the Siemens star was imaged at a distance  $x_1 = 1$  mm with the MAXIPIX detector and the same waveguide used for the grid measurements discussed above. The probe diameter was assumed to be  $p = 1$   $\mu\text{m}$  and the overlapping ratio was set to 0.5, resulting in a distance of  $d = 500$  nm between two subsequent scan points. On 313 points a circular area with a radius of  $R = 5$   $\mu\text{m}$  was scanned from the interior to the exterior. The recorded spiral dataset was reconstructed with the ptychographic iterative engine described in [25]. The phase information obtained after 200 iterations is plotted in Figure 3.8a. The result is of the same quality as the grid scan. In the inner part of the Siemens star the 50 nm structures are clearly resolved in the vertical direction (Figure 3.8b), whilst the resolution in the horizontal direction is smeared out as expected from the asymmetric dimensions of the waveguide. The fit to an edge of a line profile through the inner vertical structures indicate a resolution of 32 nm (Figure 3.8c). The overall phase information appears to be more noisy than the phase information obtained from the rectangular grid data in Figure 3.6, which is mainly related to the larger step size of 500 nm between two subsequent scan points. Thus, the total intensity per area is smaller, leading to the

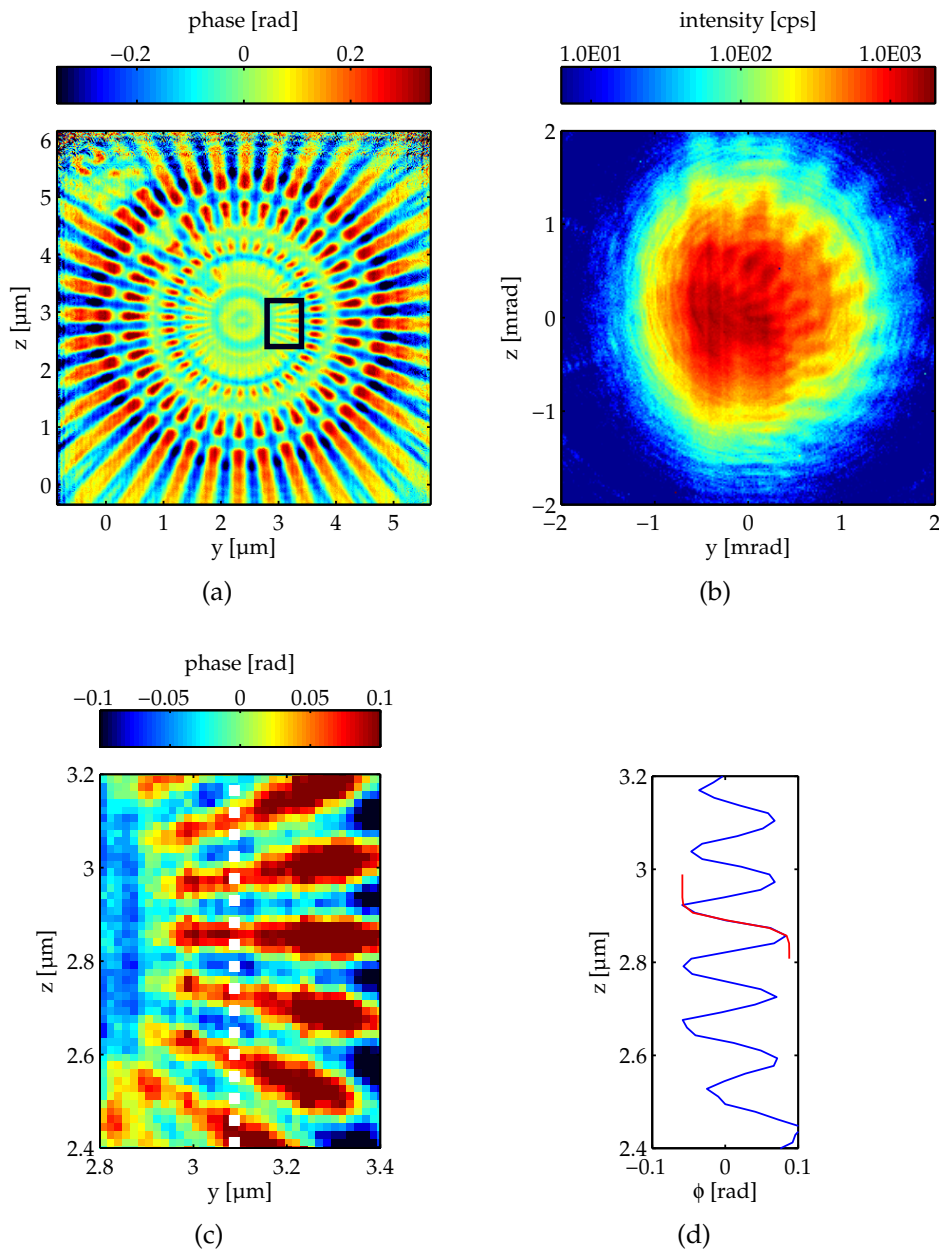


Figure 3.5: (a) Phase shift retrieved from the holographic reconstruction. (b) Example of a recorded inline hologram of the inner part of the Siemens star. (c) Zoomed area of the rectangle in (a). (d) Line profile across the dotted line of (c) [27].

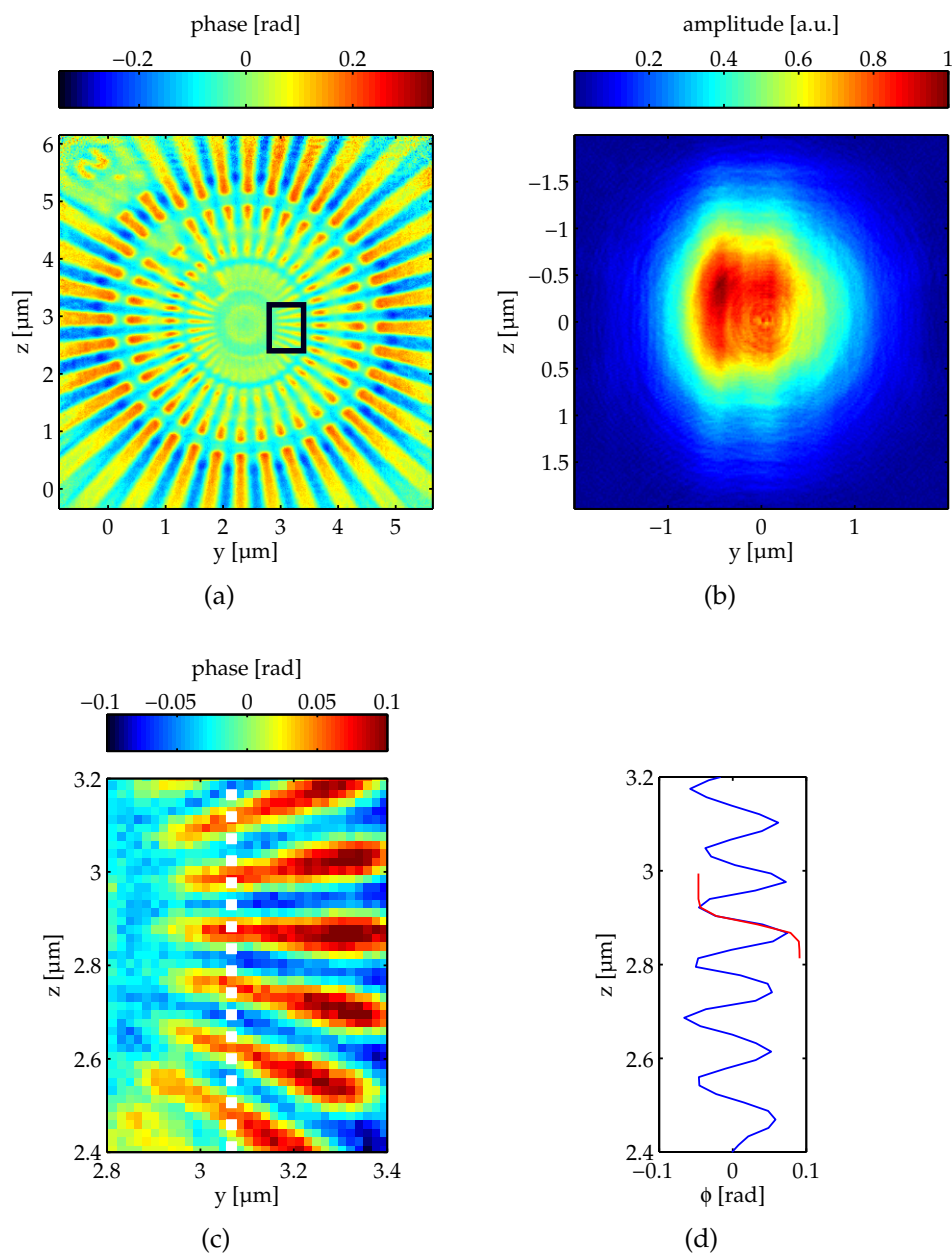


Figure 3.6: (a) Phase shift retrieved from the ptychographic reconstruction. (b) Amplitude of the reconstructed probe in the sample plane. (c) Zoomed area of the rectangle in (a). (d) Line profile across the dotted line of (c) [27].

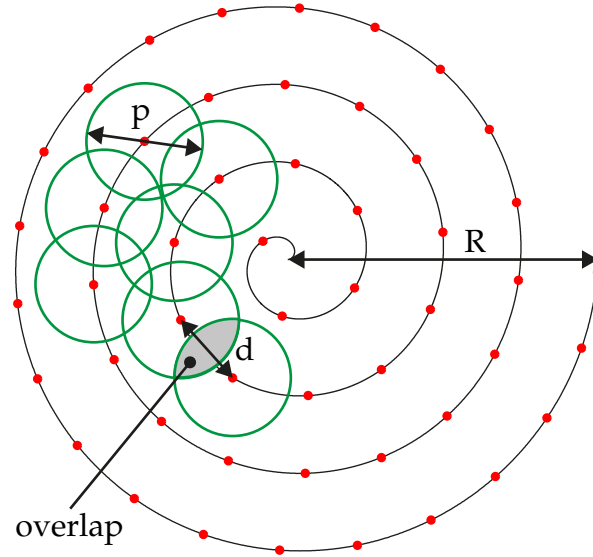
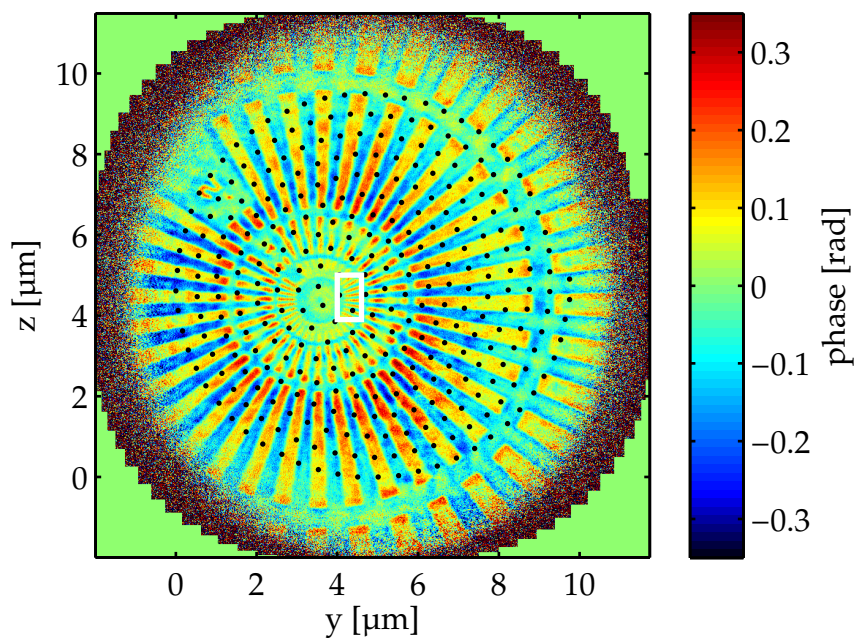


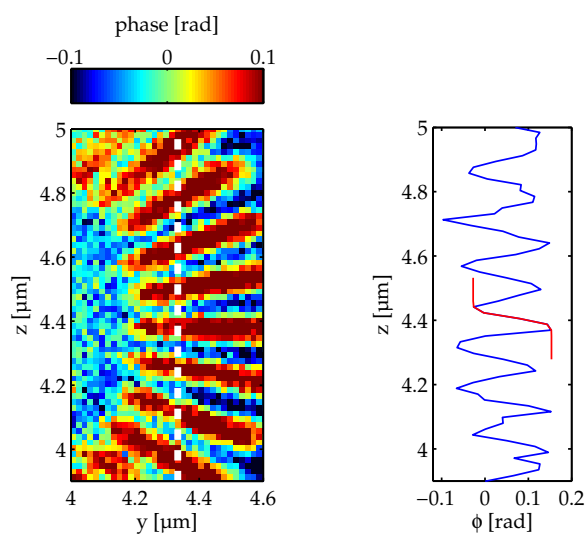
Figure 3.7: Scan on an ARCHIMEDEAN spiral with total diameter  $R$ : The scan points (red dots) are separated by the distance  $d$ . The overlap is given by the probe with a diameter  $p$  (green circles) and the scan point distance  $d$ . The distance between successive turnings is a function of the overlap ratio and  $d$ .

increased noise.

Waveguide-based inline holography is also applicable for fullfield imaging, where the X-ray beam illuminates the whole sample, not only small fractions of it [26]. Using the setup shown in Figure 3.4 freeze-dried *Dictyostelium discoideum* cells were placed at a distance of  $x_1 = 8.83$  mm behind the waveguide. The divergent beam of the crossed high-transmission waveguide [78, 44] made of Mo[30 nm]/C[35 nm]/Mo[30 nm] layers in Ge cladding was widened up to  $\approx 30 \mu\text{m}$  at the sample position. The data was recorded in an alternating sequence of holograms  $I_i^h$  with the sample in the beam and the empty waveguide beam  $I_i^e$ . With an exposure time of  $t = 0.2$  s per image and a total number of 451 sample holograms and 451 empty beam images the overall exposure time was  $t_{\text{total}} = 90.2$  s. In this mode of data acquisition intensity fluctuations and drift effects could be corrected successfully, taking advantage of the fast piezo stages and the very fast read out of the MAXIPIX detector. For the further analysis the sum of the sample holograms  $I^h = \sum_i I_i^h$  and the sum of the empty beams  $I^e = \sum_i I_i^e$  was used. The normalised hologram  $\frac{I^h}{I^e}$  of the cells



(a)



(b)

(c)

Figure 3.8: (a) Reconstructed phase shift of the Siemens star obtained from a ptychographic spiral scan. The black dots indicate the scan points. (b) Zoomed area of the rectangle in (a). (c) Line profile across the dotted line of (b).



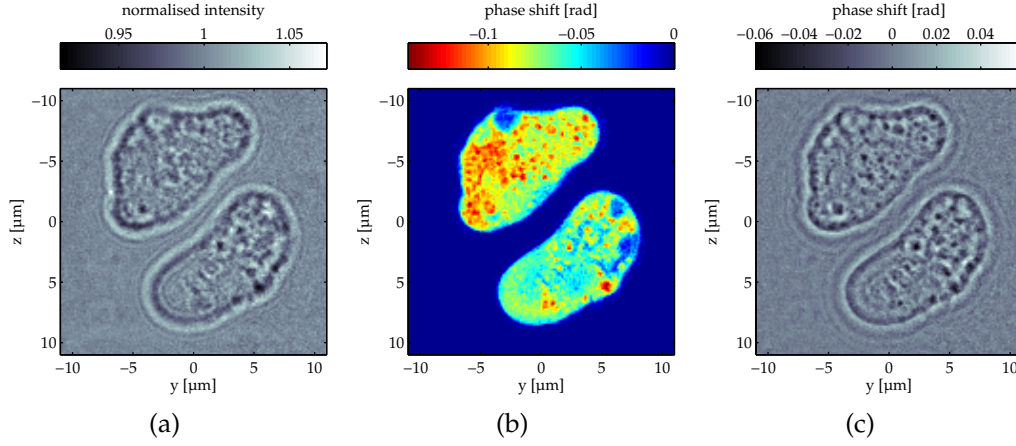


Figure 3.9: (a) Normalised hologram of freeze-dried cells of *Dictyostelium discoideum*, recorded at a distance  $x_1 = 8.83$  mm with the MAXIPIX detector. (b) Reconstructed phase as obtained by a modified HIO reconstruction scheme. (c) Reconstructed phase as obtained by holographic reconstruction. [26]

is depicted in Figure 3.9a. As the cells are unstained, thin ( $\lesssim 5 \mu\text{m}$ ), and composed of low-Z elements, they can be treated as pure phase shifting objects with negligible absorption. The phase reconstruction shown in Figure 3.9b was obtained by averaging the complex reconstructions from 25 independent runs of the modified HIO algorithm, which stopped at an average iteration number of  $N_{\text{it}} = 2347$  and showed a very small distribution of the resulting phase values with a standard deviation below  $3.5 \times 10^{-4}$  rad in each pixel.

As an initial guess a numerically generated realisation of the random function

$$R = \begin{cases} 1 + a_0 Z_j \exp(i\phi_0 Z_j) & \text{if } (y, z) \notin S, \\ 1 + a_0 Z_j \exp(i(\phi_0 Z_j - 0.1)) & \text{else} \end{cases}$$

assigned to each pixel at position  $(y, z)$  was used. The parameters  $a_0$  and  $\phi_0$  were chosen as  $a_0 = \phi_0 = 0.2$  and equally distributed pseudorandom numbers on the interval  $J = [-0.5, 0.5]$  were used for  $Z_j$ . The algorithmic feedback parameters were set to  $\beta = \gamma = 0.2$  in the present example. The maximum experimentally determined phase change of ca. 0.14 rad due to the freeze-dried cells is in agreement with expected values. In the reconstruction shown in Figure 3.9b several subcellular features are

visible. While it cannot be excluded that some features such as the rather large void areas could be due to possible ruptures in the freeze-drying process, characteristic and abundant globular features of several hundred nanometers in size are attributed to mitochondria, providing chemical energy for the highly active and mobile state in which the cells were fixated.

# Chapter 4

## The Coherent Imaging Endstation of Beamline P10 at PETRA III of DESY

### 4.1 Requirements and Design Considerations for a Dedicated Waveguide-based Imaging Instrument

For waveguide based imaging experiments both the waveguide and the sample have to be aligned in the X-ray beam.

Due to the small geometric acceptance of the waveguide cross section the total flux through the waveguide can be increased when it is placed in the spot of a highly focussed X-ray beam. As both the waveguide cross section and the spot size are typically smaller than 1  $\mu\text{m}$ , the motorisation used for waveguide alignment must provide adequate accuracy in linear and angular movements.

The imaging experiments are carried out in different regimes. For full-field imaging the whole sample is illuminated by the waveguide beam, i.e., the sample is placed at a distance where the divergent waveguide beam has widened up to a size larger than the sample. For samples on the  $\mu\text{m}$  scale, such as cells, the distance is in the range of mm to cm. The scanning applications take advantage of the magnification  $M = \frac{x_1+x_2}{x_1}$  gained by small distances  $x_1$  between the waveguide exit and the sample. The sample is then scanned through the waveguide beam with step sizes smaller than the beam diameter at the sample position. Piezo-based

positioners with encoders can achieve the desired accuracy necessary for such experiments.

Most synchrotron endstations with high gain focussing optics provide motorisation for the sample only. For waveguide experiments the additionally needed motorisation has to fit into the existing setup. This implies various construction efforts under space limitations, as the endstations are mainly optimised for applications with just the sample in the focussed beam. For experiments at different beamlines the setup has also to be modified each time, because of differences of the beamlines. However, this approach led to excellent results [37, 58, 57, 22, 21, 44, 27].

With the dedicated instrument for waveguide based imaging, as presented in this chapter, the properties of the focussing optics (focal length, spot size, energy range) and the motorisation of the sample and waveguide can be chosen to get flexible experimental conditions, also in terms of the capability to study samples at various conditions, e.g. cryo streams or humidity chambers. Tomographic measurements can be carried out as well with the current design of the sample environment. Sample inspection and pre-alignment are supported by two optical on-axis microscopes.

The achievable resolution of the sample image depends also on the pixel size of the detector and its distance to the sample. Therefore a long detector distance with a large active area is preferable, to acquire the full far field of the waveguide with high sampling.

Experiments aiming at a resolution of a few 10 nm require high thermal and vibrational stability. Drifts related to thermal expansion of material can be minimised with a stable ambient temperature by air conditioning of the experimental hutches. Vibrations are immanent and have to be damped to an amplitude smaller than the resolution aimed at, e.g., by a heavy solid foundation and a rigid setup for the instrument components.

## 4.2 Instrument Overview

With the construction of the new synchrotron PETRA III at the Deutsches Elektronen-Synchrotron (DESY) the opportunity came up to build a dedicated instrument for waveguide-based lensless imaging at the coherence beamline P10. This *Göttingen Instrument for Nano-Imaging with X-Rays (GINI-X)* is one of three experimental endstations of the P10 beamline and is located in the second experimental hutch, together with a general purpose diffractometer. Both endstations are interchangeable. Thus, the

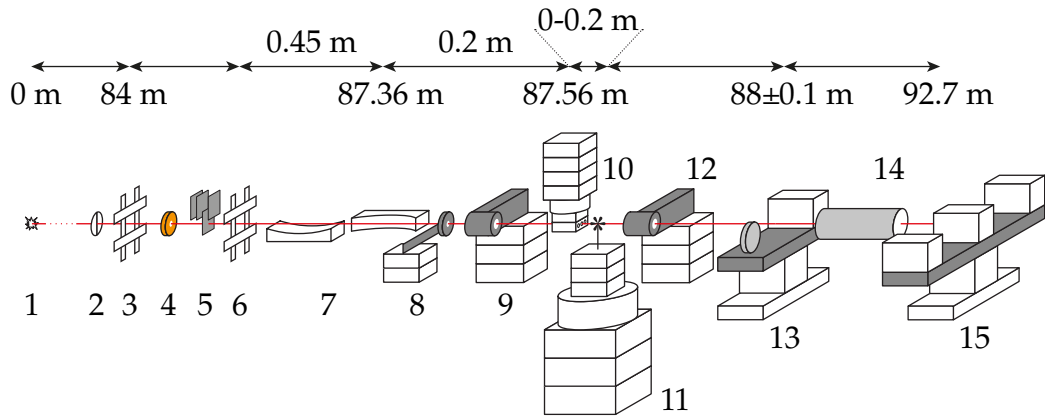


Figure 4.1: Schematic layout and distances of the instrument components and motors: 1 source, 2 fast shutter, 3 slits, 4 beam monitor, 5 attenuators, 6 scatterless slits, 7 focussing optics (Kirkpatrick-Baez mirrors), 8 pinhole, 9 optical on-axis microscope, 10 waveguide positioning, 11 sample positioning, 12 optical on-axis microscope, 13 front detector bench, 14 flight tube, 15 rear detector bench.

main parts of GINI-X are built on a movable instrument table as described in section 4.7. Figure 4.1 shows the schematic arrangement of the relevant components and their distances to the source (1 in Figure 4.1). The central parts are the focussing optics (7), the waveguide positioning system (10) and the sample positioning system (11). The waveguide positioning system and the sample positioning with its high precision tomography rotation are built to fit within the focal length of the Kirkpatrick-Baez mirrors used as focussing optics. A pinhole (8) between the focussing optics and the waveguide positioning system cleans up the X-ray beam. Additional optical on-axis microscopes (9 and 12) support the inspection and alignment of waveguides and samples in the X-ray beam. A first detector bench (13) is located on the instrument table. At the end of the experimental hutch a second detector bench (15) is installed. Several types of point detectors and area detectors are available, each giving different resolution, sensitivity and frame rates, see section 4.8. The exposure times of the detectors are realised with a fast shutter (2). Between the two detector benches a flight tube (14) reduces absorption and scattering in air. The size of the beam hitting the focussing optics is defined by two sets of slits (3 and 6), one of them (6) with hybrid-metal blades for the reduction of scattering [47]. A beam monitor (4) observes the intensity of the incoming beam from

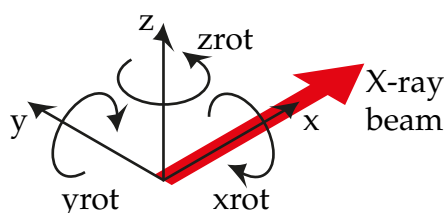


Figure 4.2: The instrument coordinate system.

the source (1). A set of attenuators (5) can reduce the beam intensity for alignment purposes. All relevant components are remote controlled and integrated in the main control software *spec* [11].

For the following description of the instrument the coordinate system is used as shown in Figure 4.2. The direction of the X-ray beam denotes the  $x$  axis. The horizontal axis indicates the  $y$  axis whilst the vertical axis indicates the  $z$  axis. The signs are chosen to build a right-handed coordinate system. Mathematical positive rotations around the  $x$  axis are describes as  $xrot$ , and  $yrot$  around the  $y$  axis,  $zrot$  around the  $z$  axis, respectively.

### 4.3 The Layout of Beamline P10

The beamline P10 is located at the new high-brilliance storage ring *PE-TRA III* of DESY in Hamburg, Germany. It is dedicated to coherence based applications such as X-ray Photon Correlation Spectroscopy (XPCS) and lensless X-ray imaging techniques.

The schematic layout of the P10 beamline is shown in Figure 4.3. The X-rays are generated in a 5 m long undulator (A in Figure 4.3) with a period of  $\lambda_u = 29$  mm and a maximum magnetic field of  $B_0 = 0.81$  T. The undulator is installed in a low- $\beta$  section of the storage ring, providing a source size of  $36 \mu\text{m} \times 6 \mu\text{m}$  ( $h \times v$ ,  $1\sigma$ ) with a divergence of  $28 \mu\text{rad} \times 3.7 \mu\text{rad}$  ( $h \times v$ ,  $1\sigma$ )[46, 3]. The brilliance

$$B = \frac{\text{photons/second}}{\text{mrad}^2 \text{ mm}^2 \text{ 0.1 \% bandwidth}}$$

delivered by this undulator is shown in Figure 4.4. The source covers an energy range from 3.5 keV to 40 keV with a decrease of brilliance of less than one order of magnitude in this range.

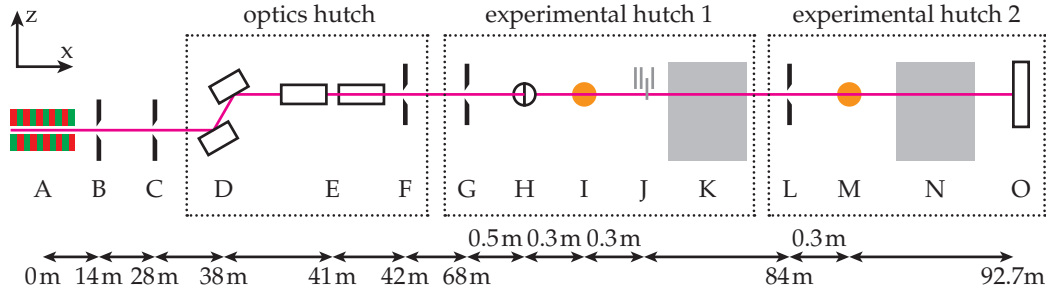


Figure 4.3: Schematic layout of beamline P10: *A* undulator, *B* primary slits, *C* secondary slits, *D* double crystal monochromator, *E* horizontal mirrors, *F* girder system with slits, *G* slits, *H* fast shutter, *I* monitor, *J* attenuators, *K* experimental setup in EH1 (in preparation), *L* slits, *M* monitor, *N* experimental setups in EH2 (interchangeable), *O* rear detector bench.

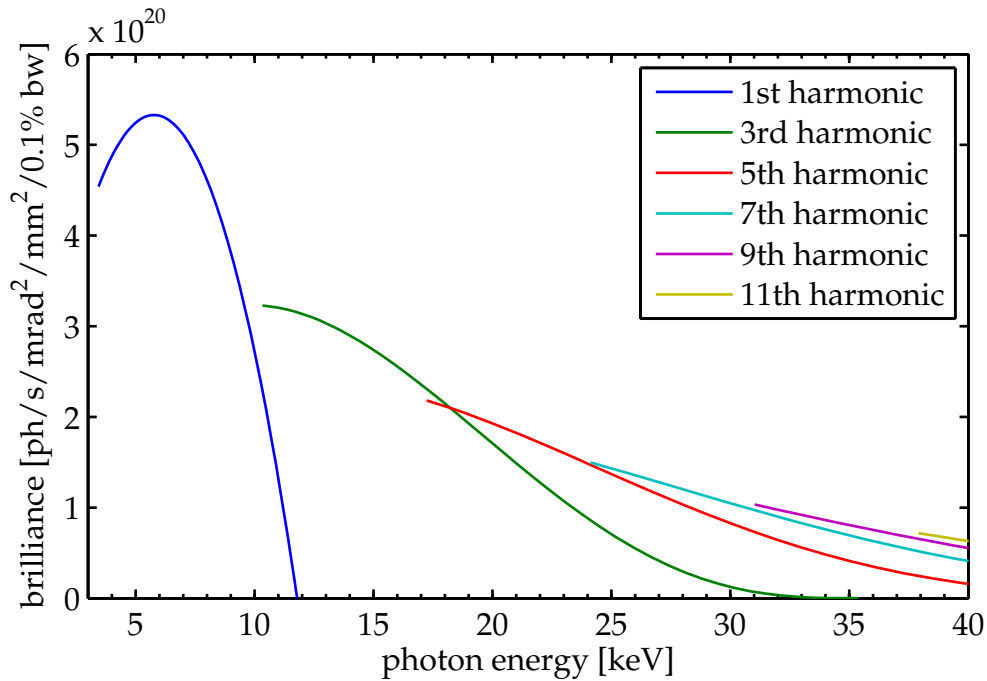


Figure 4.4: Simulated brilliance of the 5 m long undulator of the P10 beamline in a low- $\beta$  section [3].

	$\Delta E/E$	$\xi_l$ at 8 keV
Si(111)	$1 \cdot 10^{-4}$	1.5 $\mu\text{m}$
Si(311)	$3 \cdot 10^{-5}$	5 $\mu\text{m}$
pink beam	$6 \cdot 10^{-3}$	0.025 $\mu\text{m}$

Table 4.1: Monochromaticity of the different monochromator crystal types and the corresponding longitudinal coherence length  $\xi_l = \lambda(\lambda/\Delta\lambda)$  [46].

The primary and secondary slits (B and C in Figure 4.3) are installed at distances of 14 m and 28 m, respectively. The first optical element inside the optics hutch is the double crystal monochromator (D) at a distance of 38 m from the source. It is built in a vertical deflecting, fixed exit geometry, i.e. independently of the operating energy the X-ray beam exits always at the same height. The monochromator is equipped with different interchangeable sets of crystals: Si(111) and Si(311), each providing different levels of monochromaticity and thus different values for the longitudinal coherence length  $\xi_l = \lambda(\lambda/\Delta\lambda)$ . The calculated values are listed in Table 4.1 [46]. Due to the high heat load of the undulator radiation onto the monochromator crystals, they have to be cooled with liquid nitrogen. However, the pumping of the liquid nitrogen excites vibrations on the pitch angle of the crystals. The vibrational amplitude increases with the frequency of the liquid nitrogen pump. Smaller undulator gaps produce more intense radiation and therefore a higher cooling power with higher pumping frequencies is required. Due to the vibrations of the crystals the source size effectively appears enlarged in the vertical direction. Pitch vibrations in the range of 150 nrad were observed [83], resulting in an increased vertical source size of a factor of 1.95. This issue is under close investigation by the responsible PETRA III scientists and engineers.

For the suppression of the higher harmonics of the undulator radiation a pair of horizontal deflecting mirrors (E) can be inserted optionally into the beam path. Each of the mirrors has a length of 600 mm and is made of Si with additional stripes of Rh and Pt coatings for increased reflectivity at higher photon energies. With a distance of 900 mm in beam direction between the two mirrors and a typical incidence angle of  $0.16^\circ$  the beam is shifted horizontally by 5 mm when the mirrors are in use. Behind the mirrors a slit system (F) collimates the beam.

The user experiments are carried out in two experimental hutches (EH1 and EH2). For EH1 a rheometer setup (K) is planned [46], but not yet installed. EH2 accommodates two interchangeable setups (N): a



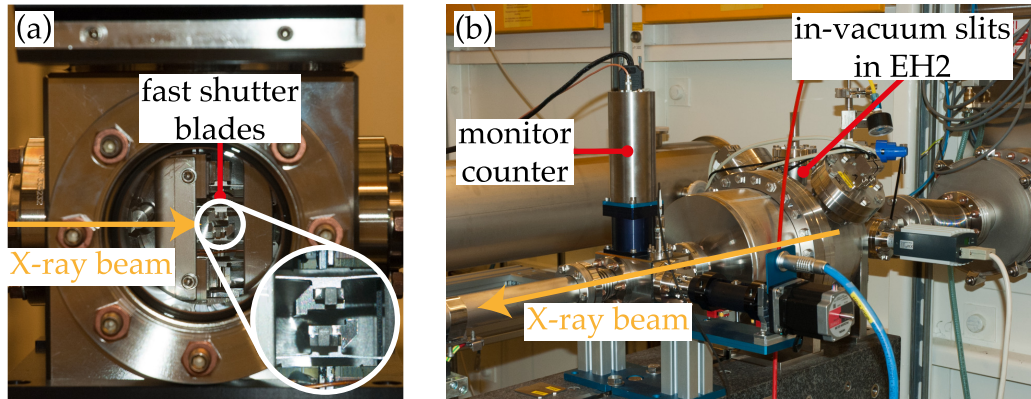


Figure 4.5: (a) Photograph of the fast shutter (*H* in Figure 4.3). (b) Photograph of the entrance slits (*L*) and the monitor (*M*).

multipurpose 6-circle diffractometer for XPCS and the waveguide-based imaging setup, which is described in detail in section 4.2. At the entrance of each experimental hutch a slit (*G*, resp. *L*) is used for beam definition, each followed by a monitor counter (*J*, resp. *M*). The fast shutter (*H*) is used for the exposure time control of the detectors (*O*). This piezo-based shutter (CEDRAT FPS400M) with a response time of 1.01 ms enables exposure times down to a few milliseconds.

The optics and experimental hutches are conditioned to a maximum temperature variation of 0.1 K. All beamlines of PETRA III are built on the same monolithic concrete foundation (dimensions ca. 300 m × 60 m × 1 m). This gives excellent thermal and vibrational stability conditions for imaging applications with sub-micron resolution.

## 4.4 Focusing Optics

The total flux through an 2D X-ray waveguide in an unfocused synchrotron beam is small and mostly not suitable for imaging applications [21, 57, 36]. Due to its cross section the waveguide accepts only a small fraction of the beam, which has widened up to diameters in the range of millimetres at the experiment position. The flux through a waveguide can be increased by positioning it in the spot of an X-ray focusing optics device [37], whose spot diameter is in the range of the waveguide diameter. In the last years several methods have been established being capable of focusing X-ray beams down to diameters below 1  $\mu\text{m}$ .

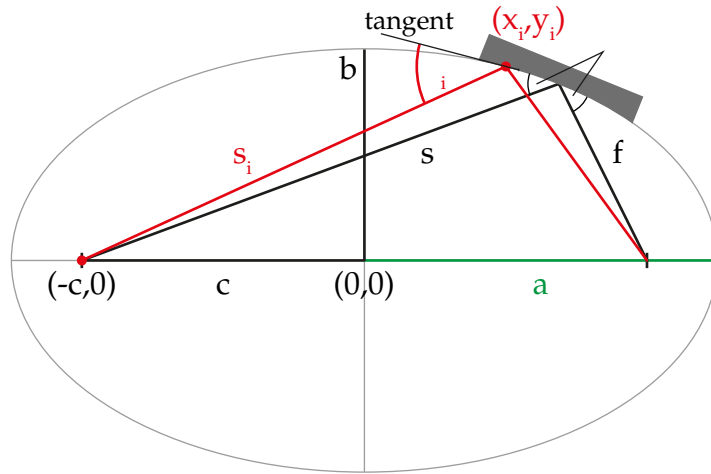


Figure 4.6: Geometrical relations of an ellipse.

For the instrument described here a Kirkpatrick-Baez mirror system (KB) [39] was chosen. KB mirror systems are non-dispersive focussing devices with a high efficiency. They are composed of two curved mirrors in orthogonal arrangement. The first mirror (v-mirror) focuses the X-ray beam in the vertical direction, the second mirror (h-mirror) in the horizontal direction. The curvature follows an ellipse whose foci are defined by the distance  $s$  from the source to the mirror centre and the planned position of the waveguide entrance, which represents the focal length  $f$ , see Figure 4.6. The excentricity  $e$  is then given by

$$e = 2c = \sqrt{s^2 + f^2 - 2sf \cos(\pi - 2\theta)}$$

where  $\theta$  is the incidence angle at the mirror centre. With the well known relations of an ellipse the major axis  $a$  and the minor axis  $b$  are

$$a = \frac{s + f}{2}$$

$$b = \sqrt{a^2 - c^2}.$$

This leads to the parametric equation of the ellipse

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1$$

$$\Leftrightarrow \frac{4x^2}{(s + f)^2} + \frac{2y^2}{sf(1 - \cos(2\theta))} = 1$$

or written as a function  $y = y(x)$

$$\Rightarrow y = \pm \sqrt{\frac{sf \left( (s+f)^2 - 4x^2 \right) \sin^2 \theta}{(s+f)^2}}.$$

The incidence angle  $\theta_i$  on a point  $(x_i, y_i)$  of the ellipse is the angle between the tangent to the ellipse in  $(x_i, y_i)$  and the line  $s_i$  through the source  $(-c, 0)$  and the point  $(x_i, y_i)$ .  $\theta_i$  can be calculated from the slope  $m_s$  of  $s_i$  and the slope  $m_T$  of the tangent by

$$\theta_i = \text{atan } m_s - \text{atan } m_T$$

with

$$m_s = \frac{y_i - 0}{x_i + c}$$

and

$$m_T = \frac{-b}{a^2 \sqrt{1 - \frac{x_i^2}{a^2}}} x_i.$$

With these relations the elliptic curvature of the mirrors can be calculated for given values of the focal length  $f$ , the source distance  $s$  and the incidence angle  $\theta$ . Due to small deviations during the manufacturing process the final values vary slightly from the given design parameters. The final values are determined by fitting the ellipse equation to the measured figure of the mirrors.

The mirrors were fabricated by two different companies. The v-mirror was produced by *WinlightX*. It has a final focal length of  $f = 302$  mm with a source distance of  $s = 85$  m and an incidence angle of  $\theta = 3.954$  mrad at the centre of the mirror. The elliptical curvature was polished into an 100 mm long Si substrate by Magnetorheological Finishing (MRF) [32]. The reflectivity at higher photon energies was increased with a 55 nm thick Pd layer. The final height profile was measured by the manufacturer with a Zygo GPI interferometer, see Figure 4.7a. The obtained figure error from the ideal ellipse is shown in Figure 4.7c. The maximum deviation from the ellipse is 13.9 nm (peak-valley) whilst the rms deviation is 3.9 nm.

The h-mirror was polished by *JTEC*. Its elliptical curvature corresponds to a focal length of  $f = 200$  mm, a source distance of  $s = 85.5$  m and an incidence angle of  $\theta = 4.05$  mrad at the centre of the mirror. The h-mirror is made of an 100 mm long silica substrate. The figure was polished by Elastic Emission Machining (EEM) [52]. A 35 nm thick Pd layer was

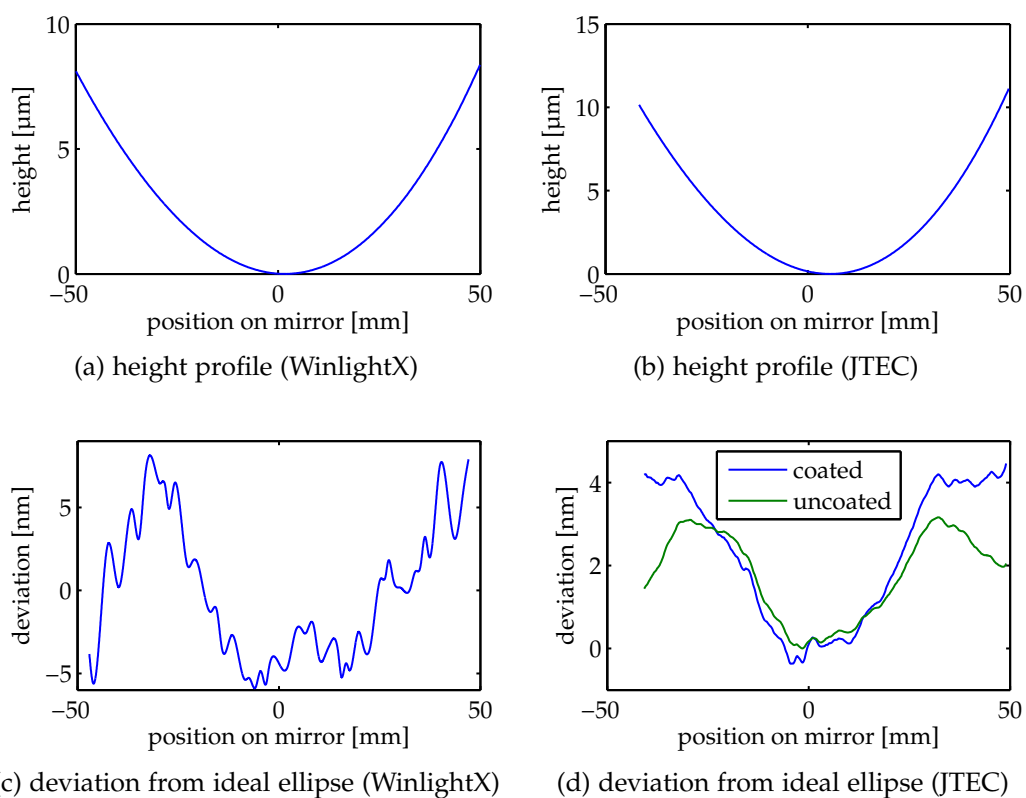


Figure 4.7: Results of optical measurements of the focussing mirrors: (a) height profile of coated v-mirror, (b) height profile of coated h-mirror, (c) deviation from ideal ellipse of coated v-mirror, (d) deviation from ideal ellipse of h-mirror before and after the coating.

	vertical mirror	horizontal mirror
$s$	85 m	85.5 m
$f$	302 mm	200 mm
$\theta$	3.954 mrad	4.05 mrad
$e$	85.301 992 m	85.699 993 m
substrate	Si	SiO <sub>2</sub>
length	100 mm	100 mm
polished area	94 mm × 2 mm	93.8 mm × 5.4 mm
coating	55 nm Pd	35 nm Pd
polishing	MRF	EEM
manufacturer	WinlightX	JTEC

Table 4.2: List of the key parameters of vertical and horizontal Kirkpatrick-Baez mirrors.

coated onto the polished substrate by *AXO-DRESDEN*. The figure of the h-mirror was measured before and after the coating at the BESSY-II Optics Laboratory of the Helmholtz Zentrum Berlin with the Nanometer Optical Component Measuring Machine (NOM) [80]. Figure 4.7b shows the obtained height profile of the coated h-mirror. For the uncoated h-mirror the maximum deviation from the ellipse is 3.1 nm (peak-valley). The final figure has a maximum deviation of 4.8 nm.

The parameters of both mirrors are summarised in Table 4.2. With the fitted values for  $s$ ,  $f$  and  $\theta$  the incidence angles  $\theta_i$  along the mirror axis can be calculated, see Figure 4.8a. The steepest incidence angle for the h-mirror is  $\theta_{h,\max} = 4.68$  mrad and for the v-mirror it is  $\theta_{v,\max} = 4.33$  mrad. Assuming the largest possible incidence angle  $\theta_{\max}$  as critical angle  $\theta_c$  for Pd, the maximum useable energy is 13.94 keV. Consequently, when the mirrors are operated at higher photon energies, the reflectivity decreases. Effectively, the mirrors appear to be shorter in this case, which affects the numerical aperture and, thus, the size of the focal spot.

The incidence angle for each mirror is adjusted with a tilt mechanism designed and created by *WinlightX*. The axis of rotation is centred on the mirror at the level of its reflecting surface. The bearing is realised with a flexural pivot, which operates free from backlash and without any lubrication. The rotational movement is generated with a 170 mm long lever arm whose end is driven by a Picomotor 8301-UHV (*New Focus*). The full tilt range is  $\pm 1^\circ$  ( $\approx \pm 17.5$  mrad). The minimum incremental motion of the motor is specified to be  $< 30$  nm [53], corresponding to an angular

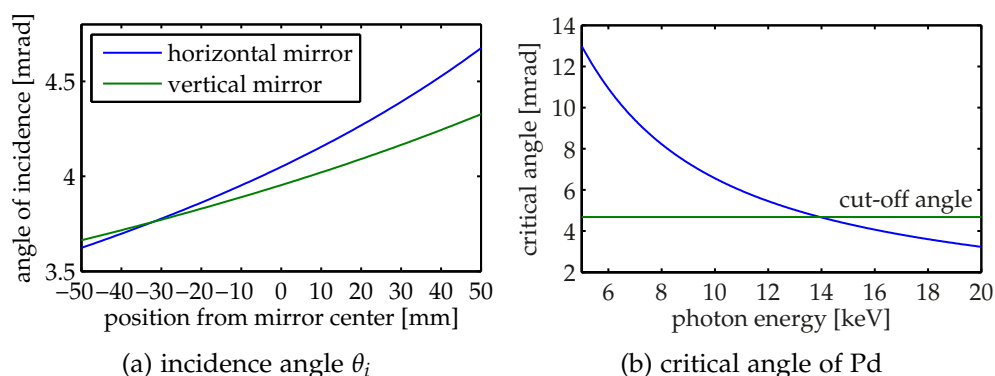


Figure 4.8: (a) Incidence angle  $\theta_i$  along the h- and v-mirror. (b) Critical angle of Pd for different photon energies. The cut-off angle indicates the steepest incidence angle of the KB mirrors.

resolution of  $0.2 \mu\text{rad}$ . Due to the actuation principle of this motor type the step size of the forward and backward motions may vary by about 20% [53]. The repeatability can be improved by the retrofitting of optical encoders. The encoder system has to be vacuum compatible and must have only a minimal heat input to reduce the effects of thermal drift and to minimise the time for equilibration. Additionally, when the encoder system is compatible with the closed-loop controller New Focus 8751-CL, the currently installed motors need not to be exchanged. The encoder systems of *MicroE Systems* fulfil these requirements. The mechanical integration of this solution is under investigation by WinlightX and the Institut für Röntgenphysik. Apart from the tilt for the incidence angle the mirror mechanics are not equipped with any additional degree of freedom. Each mirror is always hit by the beam at the same lateral stripe. The beam deviation of the v-mirror has to be compensated by a tilt of the optical table (see section 4.7) so that the beam impinges parallel to the h-mirror axis. The whole vacuum vessel with the tilt mechanics inside can be translated with a motorised yz-table. Thus, the mirrors can be positioned in the X-ray beam.

The mirrors are operated under ultra-high vacuum conditions to reduce the beam damage of the mirror coatings. Therefore the mirror system is installed in a vacuum vessel. The windows in the beam path are made of  $1 \mu\text{m}$  thick  $\text{Si}_3\text{N}_4$  membranes. As they are very homogeneous in their thickness and exhibit a small roughness, the effect of the  $\text{Si}_3\text{N}_4$  membranes on the wavefronts of the X-ray beam is expected to be very

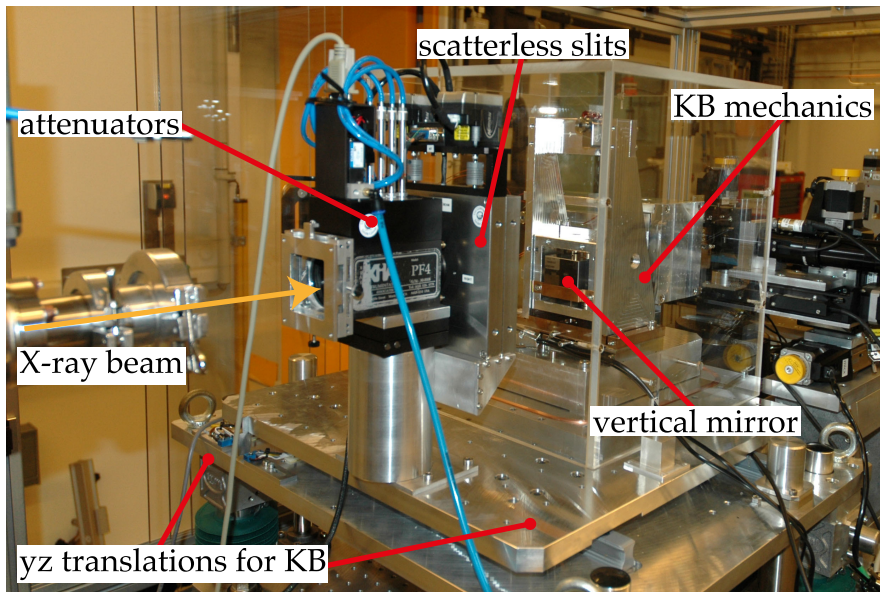


Figure 4.9: Photograph of the KB mechanics during the commissioning phase. The KB mechanics is mounted on top of the yz translation and operated in air under a dust cover box. The attenuator box and the scatterless slits are installed in front of the KB.

small. Optionally, a set of 127  $\mu\text{m}$  thick, both side polished Be windows with a roughness better than 0.1  $\mu\text{m}$  is available. During the experiment the vessel is kept under vacuum with an ion getter pump (Vacom 45s), which operates without any moving parts and thus does not induce any vibrations.

In front of the vacuum vessel the beam size is set to the geometrical acceptance of the mirrors with slits. The scattering caused by the illumination of an X-ray beam with high spatial coherence is suppressed with hybrid metal-single-crystal blades [47].

Between the mirrors and the focus a pinhole suppresses the primary beam passing the mirrors and the single deflected beams. It is aligned in the double deflected, focussed beam with yz-motors, which are mounted on a vertical breadboard attached to the vacuum vessel.

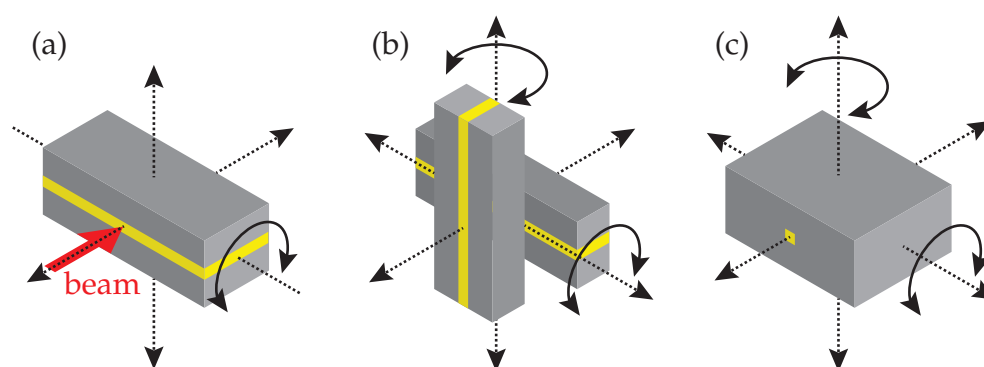


Figure 4.10: Various types of waveguides and their degrees of freedom for alignment: (a) planar (1D) waveguide, (b) crossed waveguide, (c) channel (2D) waveguide. Yellow indicates the guiding material.

## 4.5 Waveguide Positioning

X-ray waveguides require an alignment in several degrees of freedom, depending on the type of waveguide used. All types of waveguides require an axial degree of freedom when placed in a focussed X-ray beam, so that the entrance of the guiding layer can be placed in the focal spot. A planar waveguide has to be positioned in the X-ray beam along one lateral degree of freedom perpendicular to its guiding layer. The incidence angle of the beam coupling into the waveguide is adjusted with one rotational degree of freedom (Figure 4.10 a). Since a crossed waveguide is a compound of two planar waveguides the quantity of each degree of freedom for alignment has to be doubled. So two lateral and two rotational degrees of freedom are required (Figure 4.10 b). The same holds for the alignment of a channel waveguide (Figure 4.10 c).

The required degrees of freedom are realised with a set of compact axes (by attocube systems AG), as shown in Figure 4.11. The whole set is mounted upside down on a vibration reduced extension arm. This allows a condensed arrangement of the waveguide positioning axes and the sample positioning axes with its high precision rotation within the focal length of the KB mirrors. This way the waveguide to sample distance can also be minimised.

For the rotation around the z-axis an ANR101/NUM nanopositioner is used. An ANGp101/NUM nanopositioner performs the tilt around the y-axis. The arrangement of both nanopositioners gives a centre of rotation at the crossing of their rotation axes. A set of two ANPx101/NUM nanoposi-



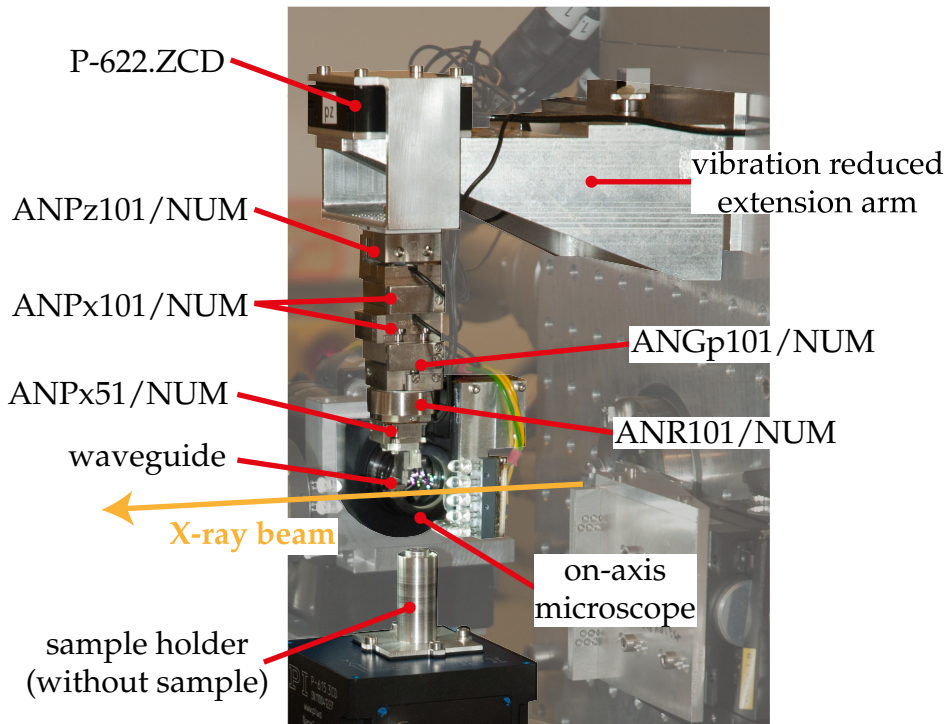


Figure 4.11: Arrangement of the positioners for the waveguide alignment as installed in GINI-X.

tioners for x- and y-translations and an ANPz101/NUM nanopositioner for z-translation is mounted above the rotations. With these xyz-translations the centre of rotation can be aligned relative to the focal spot. An object in the centre of rotation does not move laterally when it is tilted. For the waveguide alignment it is important to keep the waveguide entrance within the focal spot so that the beam can couple into the waveguide. An additional y-translation (ANPx51/NUM) between the centre of rotation and the rotations is used to bring individual channels of a channel waveguide chip or the guiding layer of a crossed waveguide respectively into the centre of rotation. The nanopositioners are all equipped with optical encoders, whose position measurements are used for closed loop operation. The positioners are driven by a step based slip stick actuator. The amplitude of the saw-tooth voltage gives the typical step size. Positions between two steps are accessible in closed loop operation by applying a DC voltage to the slip stick actuator, which then elongates but does not perform a full step.

The performance of the nanopositioners has been characterised with a

Michelson interferometer (SIOS SP-S 120). The instrument, the measurements, and results are described in Appendix A. During the measurements the ANPz101/NUM showed significant side moves in the range of  $1\ \mu\text{m}$  perpendicular to its moving direction, even for movements below  $1\ \mu\text{m}$ . Thus, when a waveguide shall be aligned in the nanofocus with the ANPz101/NUM, these side moves will probably kick the waveguide out of the focus. The behaviour demands the installation of an additional z piezo stage with a flexure (Physik Instrumente P-622.ZCD) between the waveguide positioning tower and the vibration reduced extension arm, which is then used for the fine positioning in z.

The alignment of waveguides is supported by an optical on-axis microscope (by Optique Peter). It uses a long-distance objective with a working distance of 50 mm and a motorised zoom. A tilted mirror with a drilled hole (diameter 1.5 mm) for the X-Ray beam deflects the viewing direction, so that the microscope can be mounted on a set of xyz motors (MiCos PLS-85) beside the X-ray beam.

## 4.6 Sample Positioning

The sample positioning tower is conceived as a combined setup for scanning microscopy and tomography measurements. For scanning microscopy a xyz-piezo stage (Physik Instrumente P-615 NanoCube) with a travel range of  $350\ \mu\text{m} \times 350\ \mu\text{m} \times 250\ \mu\text{m}$  is used. It operates with capacitive sensors in closed loop with a specified repeatability of 7.5 nm at a resolution of 1 nm. The piezo stage is mounted on top of a rotation stage (MiCos UPR-160 AIR). Thus, the piezo stage can also be used to align the samples in the rotation axis for tomography measurements. The axis of rod-like tomography samples can be aligned to the rotation axis with a manual tip-tilt stage ( $\pm 3^\circ$  in each direction) on top of the piezo stage. When samples in their holder are further off from the rotation axis than the travel range of the piezo stage, a manual xy table can be used for a coarse prealignment. This xy table is installed between the rotation stage and the piezo stage. With its air bearings the rotation stage is specified for an eccentricity of  $\pm 100\ \text{nm}$ , a flatness of  $\pm 50\ \text{nm}$  and a wobble of  $\pm 1.25\ \mu\text{rad}$  [51]. This precision allows to keep the sample in an X-ray beam with a diameter below  $0.5\ \mu\text{m}$ .

The combined tomography and scanning stage is mounted on top of a motorised xyz stage (two MiCos HPS-170 for x and y, MiCos UPL-160

for  $z$ ), which is used to position the rotational axis and thus the sample with respect to the beam. The sample can be placed up to 3 mm upstream of the focal spot of the KB mirrors and downstream to a defocus distance of 200 mm.

The sample alignment and inspection is facilitated by an on-axis video microscope (OAV, *Bruker ASC*), whose viewing direction is antiparallel to the X-ray beam. The drilled objective (inner diameter 1.2 mm) allows the observation of the sample at a fixed working distance of 32 mm simultaneously to X-ray measurements. The field of view covers an area of  $2.1 \text{ mm} \times 1.6 \text{ mm}$  to  $0.18 \text{ mm} \times 0.12 \text{ mm}$ , depending on the zoom setting. The images are recorded by a CCD ( $576 \times 768$  pixels) connected to a PC based frame grabber. The OAV is mounted on a motorised xyz stage (MiCos PLS-85), so it can follow the sample when it is moved with the sample stage.

The OAV is also suitable for the alignment of a tomographic sample in the rotation axis within the optical resolution and its depth of focus, respectively.

The sample mounting is kept to high flexibility for different sample types and special sample environments such as wet chambers. With a maximum distance of 39 mm between the piezo stage and the X-ray beam even large environments can be used. The main limitation is given by the maximum load of the piezo stage, which can carry up to 100 g. A higher load will reduce the stability of the closed-loop operation dramatically. For common sample systems ( $\text{Si}_3\text{N}_4$  membranes, Kapton spoons by MiTeGen, sample plates by HUBER Diffraktionstechnik) holders and adapters have already been developed and are ready to use.

## 4.7 Instrument Table

The instrument components are installed on a motorised optical table (DES-104, IDT) with a mass of 1.3 t. The breadboard (size  $1500 \text{ mm} \times 1000 \text{ mm}$ ) is mounted on three height adjustable jacks, which can also be moved laterally, see Figure 4.13. This allows a translation of the breadboard in height and laterally as well as the rotation around three axes. The movements are then combined movements of the jacks. For the height movement all three  $z$  translations  $z_1$ ,  $z_2$ ,  $z_3$  are moved by the same distance

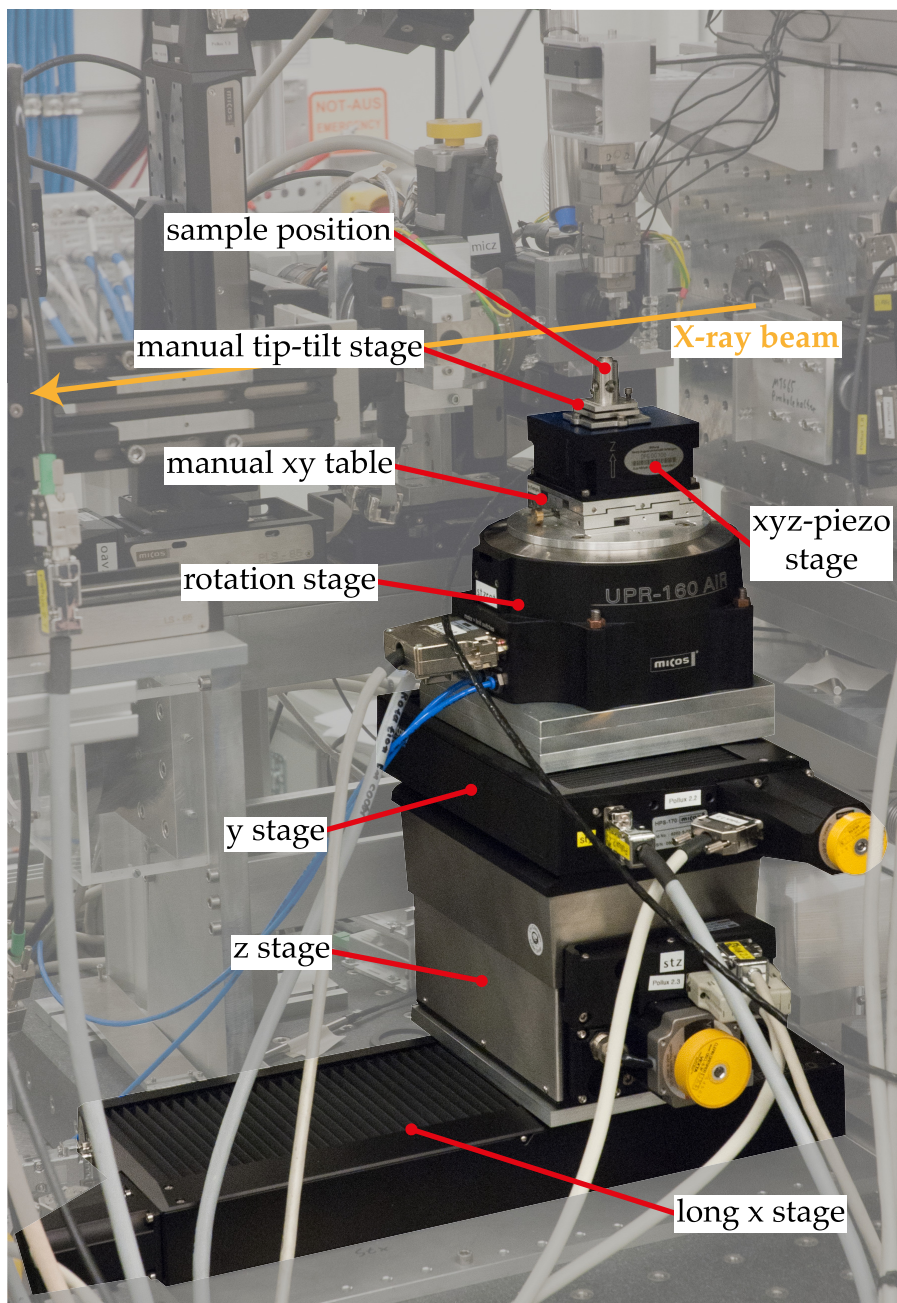


Figure 4.12: Photograph of sample stage with the stacked translations, rotation stage for tomography, manual xy table for coarse positioning, xyz-piezo stage, and the manual tip-tilt stage for the sample mounting.

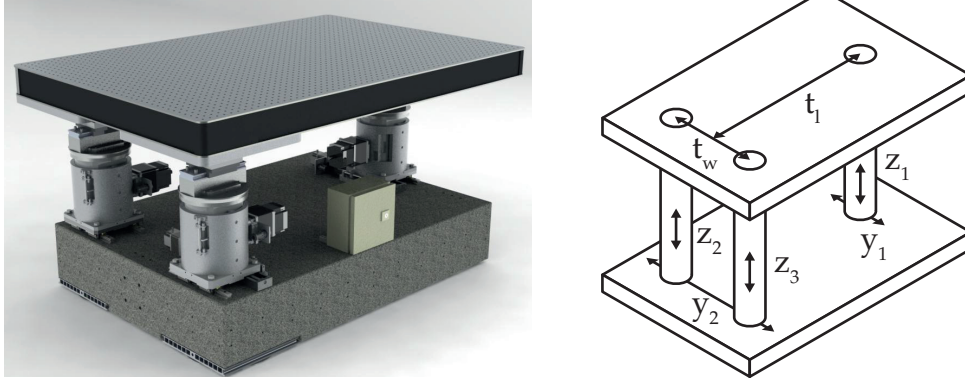


Figure 4.13: Left: photograph of the optical table (adapted from [85]). Right: schematic layout of the degrees of freedom.

$\Delta z$ . The table height  $z$  is then calculated from the positions of  $z_1$ ,  $z_2$ ,  $z_3$  by

$$z = \frac{z_1 + z_2 + z_3}{3}.$$

The lateral translation  $y$  is composed of the parallel movement of  $y_1$  and  $y_2$  with

$$y = \frac{y_1 + y_2}{2}.$$

The third jack is not motorised in  $y$  but coupled to the  $y$  movement of the second jack. The tilt angles are given by

$$\begin{aligned} xrot &= \text{atan} \left( \frac{z_3 - z_2}{t_w} \right) \\ yrot &= \text{atan} \left( \frac{\frac{z_2 + z_3}{2} - z_1}{t_l} \right) \\ zrot &= \text{atan} \left( \frac{y_2 - y_1}{t_l} \right) \end{aligned}$$

where  $t_l = 1100$  mm and  $t_w = 600$  mm are the distances between the supporting points, see Figure 4.13. The combined movements are integrated as macro motors in the instrument control system, see 4.9 and B.1.

The optical table as well as the P10 diffractometer setup are equipped with air pads, which allows a simplified move of them in the hutch when the setups are exchanged. The optical table can be placed within a range of a few millimetres to its reference position. Consequently, the rotation of the

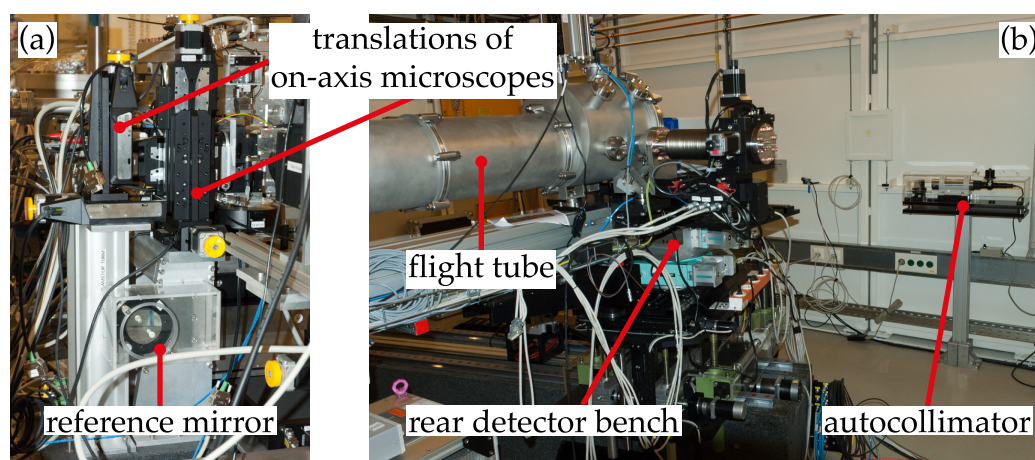


Figure 4.14: (a) Reference mirror attached to the microscope support frame. (b) Autocollimator installed in EH2 behind the rear detector bench.

breadboard relative to the X-ray beam is not aligned properly after a move on the air pads. To reduce the realignment efforts of the KB mirrors at the beginning of an experiment session, an autocollimator (Möller-Wedel AKR 300/65/14,7) is permanently installed in EH2 (Figure 4.14). It has an angular resolution of  $\Delta\alpha = \frac{\Delta s}{2f} \approx 17 \mu\text{rad}$ , where  $\Delta s = 0.01 \text{ mm}$  describes the line width of the cross line reticle and  $f = 300 \text{ mm}$  the focal length of the autocollimator. On top of the optical table a reference mirror is attached to the microscope support frame. After the initial alignment of the instrument the reference mirror was adjusted in the beam path of the autocollimator, which is then the angular reference for the breadboard and the KB mirrors on top of it. At the beginning of a new experiment run, the optical table is placed on its reference marks on the ground floor. Then the breadboard is tilted until the reference mirror is again aligned in the autocollimator. This procedure brings back the angular relation of the instrument relative to the X-ray beam. The lateral position can be regained with the slits in front of the KB mirrors: The PIN diode (see Section 4.8) is placed between the slits and the vacuum vessel. Then the y and z positions of the breadboard are scanned through the X-ray beam.

## 4.8 Detectors

For the waveguide and sample alignment as well as the recording of far fields and holograms various detectors are implemented into the

instrument. The intensity dependent alignment is usually performed with point detectors such as scintillation based detectors or silicon PIN diodes.

The detection principle of silicon PIN diodes is based on charge separation when a photon is absorbed by the silicon crystal. The generation of a so called electron-hole pair requires an average energy of  $3.66 \text{ eV} \pm 0.03 \text{ eV}$  [63]. The absorption of a single multi-keV photon will generate a number of electron-hole pairs equivalent to the energy of the photon. The resulting charge separation is measured as a current. The occurring current depends mainly on the number of photons absorbed per time, which depends on the thickness of the diode and the photon energy. For typical X-Ray beams of third generation synchrotrons with intensities of  $\approx 10^{12}$  photons/s and a photon energy  $E = 12.4 \text{ keV}$  the expected current is  $\approx 0.5 \text{ mA}$  for a  $500 \mu\text{m}$  thick silicon PIN diode. As the generated current is directly proportional to the intensity of the X-ray beam, the minimum detectable intensity depends on the sensitivity of the used amperemeter. This also gives a high dynamical range. In the setup presented here a silicon PIN diode (Canberra PD300-500CB, thickness  $500 \mu\text{m}$ , diameter of active area  $19 \text{ mm}$ ) is used with a picoammeter (Keithley 6485), which is read by the instrument control software, see Section 4.9. Each PIN diode can be calibrated for flux determination by comparing the measured current with the signal of a single photon counting detector, e.g. a scintillation based detector or a hybrid pixel detector.

For the monitoring of the beam intensity variations during the experiment an ionisation chamber (JJ X-Ray Ion Chamber) can be optionally installed at the entrance or the exit of the KB mirror vessel. The absorption of the X-ray photons causes the ionisation of the gas inside the chamber. The generated charge is proportional to the absorbed photons/time and collected in the electrical field of a high voltage capacitor. The resulting current is again measured with a picoammeter (Keithley 6485). The absorption of the gas inside the chamber and the Kapton windows of the chamber is negligible. A further beam monitor is permanently installed behind the entrance slits of EH2, see Figure 4.3 on page 43. This device counts the photons scattered by a Kapton foil hit under  $45^\circ$  by the X-ray beam. The photons are detected by a scintillation based detector (FMB Oxford Cyberstar with NaI(Tl) crystal). Absorbed X-ray photons excite the emission of optical photons, which are then converted to photoelectrons and amplified by a photomultiplier. The signal of the photomultiplier is then filtered by a single channel analyser (SCA), which generates a TTL pulse for each detected X-ray photon, when the gain of the photomultiplier

and the filter window settings of the SCA are set appropriately. Compared to a PIN diode a scintillation based detector is capable of single photon counting. The TTL pulses are recorded by a counting board (National Instruments 6602) connected to a computer. The signals of the monitors are used for the normalisation of measurements, when the total flux varies over time.

The far fields and holograms of the KB-focussed beam, waveguides, and samples are recorded with different 2D detectors each providing different spatial resolution, active area, sensitivity, and maximum count rates. Two types of CCDs (Charge-Coupled Devices) are available for experiments at the P10 beamline. In the direct-illuminated CCD (Roper Scientific LCX) the X-ray photons are absorbed in a deep-depletion Si chip, where free charges are created in a comparable process as in PIN diodes. The Si chip is cooled to reduce the charge creation by thermal effects, leading to a dark current. The charges are stored in a potential in the pixel and after the exposure transferred to the analog-to-digital converter (ADC), which measures the charges and returns the ADU value (Analog to Digital Units). The ADU value is proportional to the energy and number of photons, that hit a pixel, but does not represent an absolute number of photons. A pixel can only collect a maximum charge per exposure. Overexposures will lead to saturation and permanent degeneration due to defects in the Si crystal structure caused by the X-ray photons. Direct illuminated CCDs are very sensitive, with the expense of long readout times, as the charges have to be shifted strip wise from pixel to pixel before they are measured in the ADC.

The second type of CCDs is based on a scintillation foil with an optical taper (Roper Scientific SCX). In the scintillation foil ( $\text{Gd}_2\text{O}_2\text{S:Tb}$ ) the absorbed X-ray photons are converted to visible photons with a wavelength of  $\approx 550$  nm. These photons are then collected by the 1:1 optical taper and guided to the back-illuminated CCD chip, where they are detected. The point spread function of the SCX is about 2–3 pixel [74]. The key parameters of the SCX and LCX are listed in Table 4.3, along with the parameters of both hybrid pixel detectors (DECTRIS PILATUS and ESRF MAXIPIX).

Both the PILATUS and the MAXIPIX are based on the same detector concept. In a simplified model of operation each pixel of a sensor chip is directly connected by a bump bond to its distinct counting circuit on a separate electronics chip. The pixels on the sensor chip are built by electronic structures that apply an electrical field in each pixel. The charges



	SCX	LCX	PILATUS	MAXIPIX
pixel size	20 $\mu\text{m}$	20 $\mu\text{m}$	172 $\mu\text{m}$	55 $\mu\text{m}$
width of point spread function	2 – 3 px	1 px	1 px	1 px
dimensions	1300 $\times$ 1340	1300 $\times$ 1340	487 $\times$ 619	516 $\times$ 516
active area [mm]	26 $\times$ 26.8	26 $\times$ 26.8	83.8 $\times$ 106.5	28.4 $\times$ 28.4
modules	1	1	3 $\times$ 1	2 $\times$ 2
intermodule gap	–	–	17 px	4 px
read-out time	1.8 s	1.8 s	3.6 ms	2.8 ms
maximum frame rate	0.56 Hz	0.56 Hz	200 Hz	350 Hz
dynamic range	16 bit	16 bit	20 bit	13.5 bit
sensor material and thickness	Gd <sub>2</sub> O <sub>2</sub> S:Tb	Si (bulk)	Si 320 $\mu\text{m}$	Si 500 $\mu\text{m}$

Table 4.3: Key parameters of the implemented detectors.

created in a pixel by the absorption of X-ray photons are transported by this electrical field through the bump bond to the counting electronics. In the counting circuit the charge signal is amplified and then discriminated by an SCA. Charge signals that originate from noise and not from an X-ray photon are filtered by shaping and threshold levels. This gives the possibility to count single photons as the number of created charges is proportional to the photon energy [75]. The events output by the SCA are stored in a digital register. The charge signals are processed in real time and simultaneously for all pixels of the sensor. So the detector can be read by requesting the values of the digital register, which can be done very fast, giving frame rates of a few 100 Hz. The PILATUS and MAXIPIX differ in the fabrication process and the design of the sensor chip and the counting circuits. Details can be found in [75]. Furthermore, the key parameters of the detectors installed at the P10 beamline are listed in Table 4.3. Both the PILATUS and the MAXIPIX are assembled from multiple sensor modules, resulting in gaps with no pixels in between. Pixel detectors suffer from radiation damage in the electronics chip. The counter circuits are hit by X-ray photons not absorbed in the sensor chip. There they cause defects in the electronics, that increase the noise level and the dark current in the transistors of the circuits. This leads to reduced sensitivity up to a short-circuit fault.

For experiments that require very high resolution with a pixel size in the range of  $\mu\text{m}$ , a custom-built detector (*PCO camera*) can be integrated in the instrument [6, 56, 4]. It is based on thin scintillation foils that are imaged by an optical microscope with a high sensitivity CCD for visible light.

The PILATUS detector is permanently installed at a distance<sup>1</sup> of  $\approx 5.29$  m on a motorised detector bench. It is equipped with yz-translations and an additional rotation around the x-axis. The latter is used to align the pixel columns parallel to the rotational axis of the tomography stage. The rear detector bench is part of the P10 standard infrastructure. Beside the PILATUS two additional detectors can be installed on the rear detector bench, as required by the experiment. On the optical table a second, front detector bench with yz-motorisation is located. It can carry up to two detector units and has a position to let the X-ray beam pass to the rear detector bench. One of the usually installed units is the combined assembly of PIN diode and FDI camera. Both are mounted side by side on a xz-stage, which is used to position the FDI camera at different defocus distances in beam direction. It is also possible to bring the FDI camera in the focal plane, where the FDI is utilised for the prealignment of the KB mirrors. The detector units on the front detector bench can be moved manually in x-direction so that the distance from the focus can be varied. At short focus distances the travel range in y is limited by the optical microscopes, which may be hit.

The beam path between the front and the rear detector bench is bridged by a flight tube, which reduces the absorption in air. The entrance window with a diameter of 100 mm and the exit window with a diameter of 180 mm are covered with Kapton foils. A set of motorised beamstops is integrated inside the flight tube. For common waveguide imaging experiments these beamstops are not necessary, but may be used for diffraction experiments. The flight tube and its beamstop system are also part of the P10 infrastructure.

As not all detectors are capable for measurements at the full intensity of the X-ray beam, a set of attenuators is installed in front of the entrance slits of the KB mirrors. This attenuator box (XIA PF4 with XIA PFCU) holds four independent filters which can be inserted in the beam via the instrument control software. The actual material and thickness of the filters determines their absorption and thus they have to be customised for the photon energy of each experiment.

---

<sup>1</sup>The distance varies with the current position of the optical table and the current position of the rear detector bench.

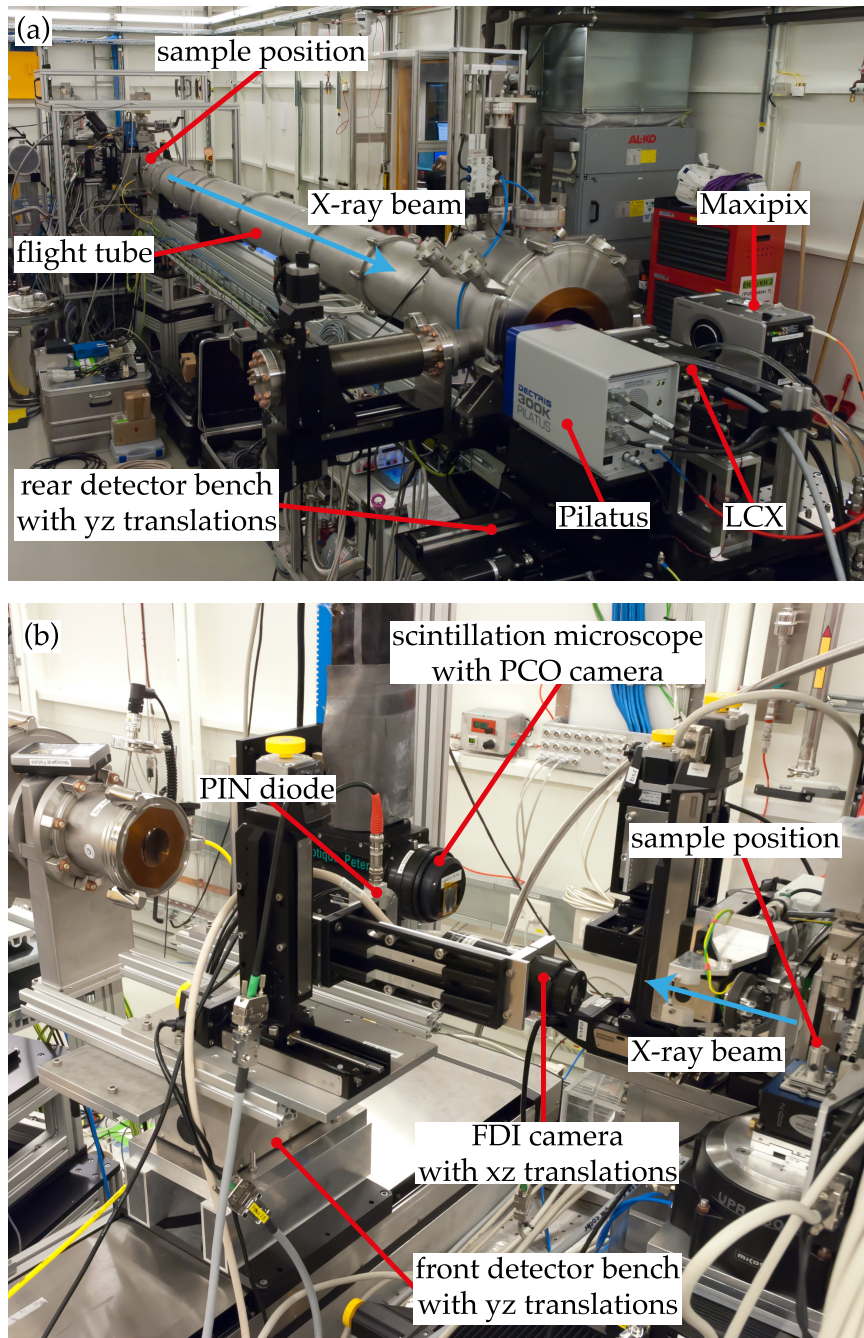


Figure 4.15: (a) Rear detector bench equipped with PILATUS, LCX, and MAXIPIX detectors. The absorption and scattering by air between the sample position and the rear detectors is reduced by a flight tube. (b) Front detector bench with the FDI alignment camera, the PIN diode and the scintillation based microscope with a PCO camera.

## 4.9 Instrument Control System and Interfaces to P10

Most components of the instrument are controlled by software. The central control and data acquisition software *spec* is intended to be the interface between the experimenter and the instrument with its components. *spec* is a Unix-based software with a powerful command line interface and an extendable macro language [11]. It provides easy access to different types of interfaces and protocols (serial ports, GPIB, Ethernet, Tango, etc.) and comes with built-in support for many devices such as various motor controllers and counter/timer cards. Devices, that are not natively supported by *spec*, are accessible with macro motors and macro counters by programming the communication with these devices via the supported interfaces. The macro motors can also be used as calculational motors, such as slit gaps or offsets, that are calculated from the position of the slit blades. The capability of these extensions to motors, counters and user-commands allows a flexible customisation to meet the requirements of imaging experiments.

*spec* is installed on a rack-mounted PC running Linux (CentOS 5.5). This PC with the name “haspp10wg”<sup>2</sup> houses the interface cards for ethernet, serial connections (EXSYS EX-41098 with 8 ports), ethernet (built-in with 2 ports) and GPIB (National Instruments PCI-GPIB) and the counter/timer card (National Instruments PCI-6602 with BNC-2121). The data is stored on a RAID 5 storage system with size of 808 GB. The detailed configuration is listed in Table 4.4.

The connection scheme of all relevant instrument related devices is shown in Figure 4.16. For clarity, some connections, such as serial lines or TTL pulses, are concatenated, although technically the devices require separate cabling.

haspp10wg is connected to the DESY network with one of its ethernet interfaces. The second ethernet interface is connected to an instrument internal network. Computers in the internal network can access the internet using haspp10wg as NAT<sup>3</sup> router. haspp10wg acts also as a fileserver (Samba) to provide computers in the internal network access to the experimental data for storing and online analysis. Beside this, haspp10wg hosts several network services, such as DHCP, SSH, dynDNS, web server for

---

<sup>2</sup>haspp10wg: hasylab petra p10\_waveguide

<sup>3</sup>Network Address Translation

main board	Kontron <i>xBP-13E5P7_2</i> with 7 PCI slots 1 PCIe x16 slots 1 PCIe x4 slots 3 PCIe x1 slots 2 LAN interfaces
processor	Intel Core2Quad Q8400 2.66 GHz
RAM	4 GB DDR2
disk space	808 GB RAID 5
additional cards	National Instruments PCI-GPIB counter/timer card National Instruments PCI-6602 Exsys EX-41098 with 8 serial ports

Table 4.4: Configuration of the main control PC *haspp10wg*.

instrument wiki, and X-server. The rack-mounted *haspp10wg* is remote-controlled by the PC “otto” via an X-server session over internal network (not shown in Figure 4.16).

The components of the P10 standard equipment are controlled by TANGO servers [84]. TANGO is an object oriented distributed control system which is used at PETRA III for hardware control of most beamline components. TANGO servers represent an abstraction layer between the low-level hardware access and the high-level user’s control software. *spec* supports the communication with TANGO servers when it is linked to TANGO libraries at installation.

Via this TANGO support the MAXIPIX and PILATUS detectors can be used with *spec*. The corresponding macros with the user commands and the TANGO communication are listed in Listing B.1 and Listing B.2. Also beamline motors, such as the rear detector bench, connected to VME OMS58 controllers are moveable with *spec*, see Listing B.3. Additional interfaces to TANGO servers are implemented for the monitoring of the storage ring current (Listing B.4) and the switching of I/O registers (Listing B.5). With these I/O registers the fast shutter can be triggered manually from the users level in *spec*. Or the fast shutter control is handed over by the logic box to one of the CCD detectors. Note that the PILATUS and MAXIPIX detectors do not need a fast shutter for operation, but it is triggered automatically by *spec* commands before and after an

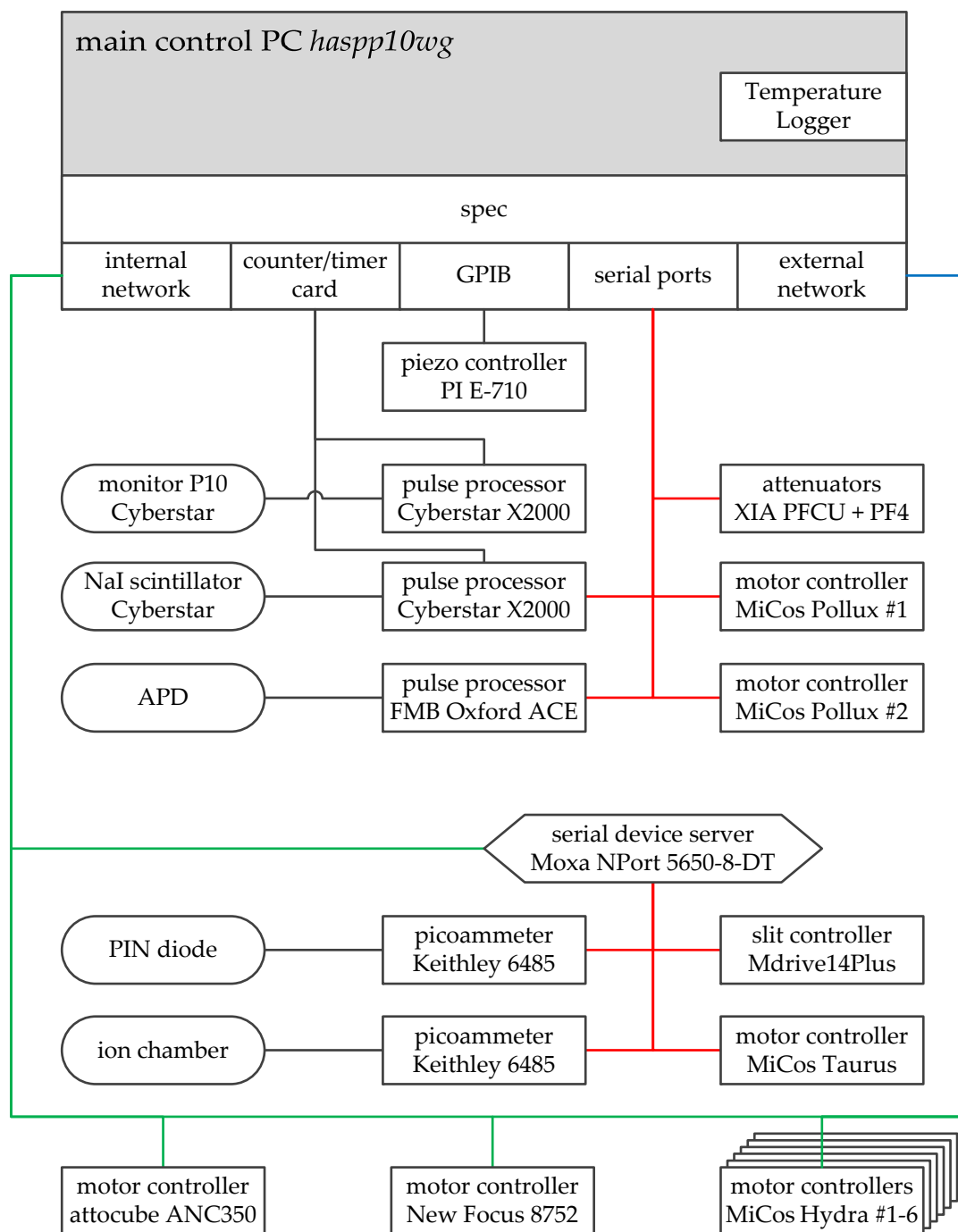


Figure 4.16: Connection schema of all relevant instrument devices. The schema is continued on next page.

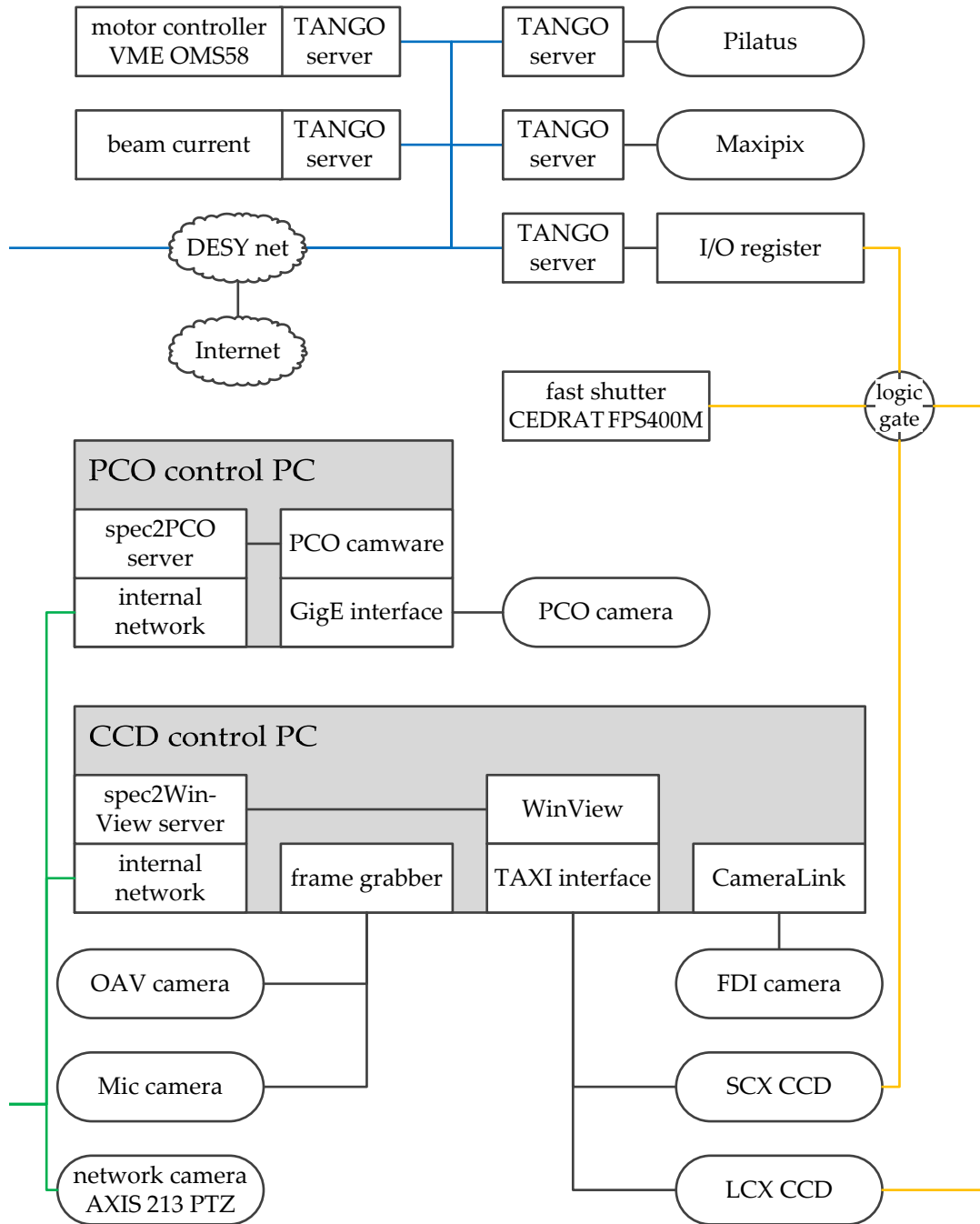


Figure 4.16 (continued)

acquisition to reduce the applied radiation dose on the detectors and the sample. The CCD detectors LCX and SCX are connected via a proprietary TAXI interface to the CCD control PC. This Windows XP based computer runs the acquisition software “WinView”. spec sends commands for CCD control to WinView through the “spec2WinView” server [57]. The server has been extended for support of pixel binning and region of interest settings within spec. spec2WinView receives commands from spec over an UDP [69] connection in the internal network and accesses WinView through its built-in Visual Basic API, see Listing B.6 for the spec implementation. The CCD control PC grabs also the images of the alignment X-ray camera (Photonic Science FDI) via a CameraLink connection and the video signals of the two optical on-axis microscopes via a frame grabber card (The Imaging Source DFG/MC4/PCIE). The optional high resolution detector (called “PCO camera”), based on a scintillation foil imaged by an optical microscope with a *pco.2000* camera, is controlled similar as the CCDs. The PCO camera is connected over a GigE interface [28] to a dedicated PCO control PC. The spec2PCO server passes the commands sent by spec over a UDP connection to the “PCO camware” program.

The pulse processing units X2000 of the Cyberstar scintillation detector and ACE of the avalanche photo diode (APD) are configured over serial lines. The parameters of the pulse processing units, such as the high voltage of the Cyberstar photomultiplier, the bias voltage of the APD or the window settings of the single channel analysers (SCA), are implemented as macro motors in spec, see Listing B.7 and Listing B.8. This way, the parameters can be easily changed from the spec command line or can be scanned, which is useful especially for the SCA window settings. The TTL pulses generated by the pulse processing units are then counted by the counter/timer board (National Instruments 6602). The ACE unit for the APD has also a faster internal counter, which can count up to a pulse rate of 180 MHz. The value of this internal counter is available as a macro counter, which requests the value over the serial line from the ACE unit.

The signals of the PIN diode and the ionisation chamber are measured with picoammeters (Keithley 6428). The picoammeters are implemented as macro counters, see Listing B.10 and Listing B.11 for the control of the high voltage supply for the ion chamber (not shown in Figure 4.16). The picoammeters are addressed over serial lines, which are connected to a serial device server inside the experimental hutch. This avoids long cable paths from the PIN diode/ion chamber to the picoammeters or too long serial lines to the haspp10wg computer. The serial device server



(Moxa NPort 5650-8-DT) operates transparently and sends the command strings received from spec over the internal network to the connected serial devices. The serial device server is also used to access two motor controllers (Mdrive14Plus and MiCos Taurus) inside the experimental hutch. The MiCos Hydra and Pollux controllers are located outside the experimental hutch to reduce the heat load for better thermal stability.

For a detailed list of all motors and controllers see Table 4.5. The unit of the travel range is equal to the unit used in spec for motor movements.

Table 4.5 contains also calculational motors for the horizontal and vertical gaps and offsets of the KB slits (Listing B.12), the translations and rotations of the optical table (Listing B.13 – B.17) and beam-following motors (Listing B.18). These beam-following motors are implemented for the translations along the beam direction of OAV microscope, the sample tower and the sample fine positioning with the piezos on top of the tomography rotation. As the X-ray beam is deflected by the KB mirrors, the direction of movement of the motors is not parallel to the beam direction. The beam-following macro motors compensate this by additional transversal and horizontal movements with the corresponding physical motors of the OAV translations, the sample tower, or the sample piezos, respectively. The magnitude of the correcting movements is calculated by the beam-following angles between the X-ray beam and the direction of the x-motor. The angle has to be determined for each x-motor separately. Therefore, a sample or alignment tip is placed in the X-ray beam at two different x-positions. From the displacement in y and z between these two positions the beam-following angles are calculated.

As thermal stability is essential for positioning at the nanometre scale, a system for temperature monitoring (Driesen+Kern DK8010-T) is installed in the instrument. Six temperature sensors with an absolute accuracy of 0.12 K and a resolution better than 0.01 K are distributed to significant places of the instrument: close to the sample position, between the optical microscopes, KB table, at the foot of the sample tower and the ambient of the experimental hutch. The temperature data is recorded with a Python script (Listing B.21 – B.23). As the temperature changes rapidly, when the experimental hutch is entered, the temperature monitoring helps to ensure that the instrument and the hutch is back in thermal stability for measurements.

Table 4.5: List of all instrument motors and controllers.

spec name	type	encoder	travel range	application
<i>Pollux #1 with RS232 interface</i>				
oavx	PLS-85	optical	52 mm	microscope translation
oavy	PLS-85	optical	52 mm	
oavz	PLS-85	optical	155 mm	
micx	PLS-85	optical	52 mm	microscope translation
micy	PLS-85	optical	52 mm	
micz	PLS-85	optical	52 mm	
piny	MTS-65	optical	13 mm	pinhole translation
pinz	MTS-65	optical	13 mm	
<i>Pollux #2 with RS232 interface</i>				
stx	HPS-170	optical	205 mm	sample translation
sty	HPS-170	optical	52 mm	
stz	UPL-160	optical	26 mm	
fdix	LS-65	–	102 mm	diode and FDI translation
fdiz	LS-65	–	102 mm	
<i>Taurus with RS232 interface</i>				
stzrot	UPR160-Air	optical	360°	tomography rotation
<i>Hydra #1 – 6 with ethernet interfaces</i>				
mdety	LS-270	–	508 mm	front detector bench
mdetz	UPL-160	–	26 mm	
oavzoom	custom	–	35 a.u.	microscope zoom
oavpol	custom	–	360 a.u.	polarisation filter
miczoom	custom	–	34 a.u.	microscope zoom
kby	custom	–	45 mm	KB translation
kbz	custom	–	45 mm	
otz1	custom	–	±25 mm	optical table
otz2	custom	–	±25 mm	
otz3	custom	–	±25 mm	
oty1	custom	–	±25 mm	
oty2	custom	–	±25 mm	

Table 4.5: (continued)

spec name	type	encoder	travel range	application
<i>macro motors for optical table</i>				
oty	translation	–	±25 mm	optical table
otz	height	–	±25 mm	
otxrot	roll	–	±1°	
otyrot	pitch	–	±1°	
otzrot	yaw	–	±1°	
<i>New Focus 8753 with ethernet interface</i>				
kbyrot	8301-UHV	–	±17 mrad	KB mirror tilt
kbzrot	8301-UHV	–	±17 mrad	
<i>Mdrive slit controller with RS422 interface</i>				
skbb	Mdrive14plus	–	±12 mm	KB slits
skbt	Mdrive14plus	–	±12 mm	
skbr	Mdrive14plus	–	±12 mm	
skbl	Mdrive14plus	–	±12 mm	
<i>macro motors for KB slits</i>				
skbhg	horiz. gap	–		KB slits
skbho	horiz. offset	–		
skbvg	vertical gap	–		
skbvo	vertical offset	–		
<i>E-710 with GPIB interface</i>				
px	P-615.3CD	capacitive	350 µm	sample fine positioning
py	P-615.3CD	capacitive	350 µm	
pz	P-615.3CD	capacitive	250 µm	
wgz	P-622.ZCD	capacitive	250 µm	wg positioning
<i>ANC350 with ethernet interface</i>				
ax	ANPx101/NUM	optical	5000 µm	waveguide positioning
ay	ANPx101/NUM	optical	5000 µm	
az	ANPz101/NUM	optical	5000 µm	
ayrot	ANGp101/NUM	optical	5400 m°	
azrot	ANR101/NUM	optical	360 000 m°	
wgy	ANPx51/NUM	optical	3000 µm	

Table 4.5: (continued)

<b>spec name</b>	<b>type</b>	<b>encoder</b>	<b>travel range</b>	<b>application</b>
<i>calculational macro motors for beam following</i>				
stxb		–		beam-following
pxb		–		
oavxb		–		
<i>macro motors for X2000 pulse processor with RS232 interface</i>				
cshv	high voltage	–	1250 V	scintillator Cyberstar
cslow	lower SCA level	–	4 V	
csup	upper SCA level	–	4 V	
csgain	gain	–	100 a.u.	
<i>macro motors for ACE pulse processor with RS232 interface</i>				
cspeak	peaking time	–	1000 ns	avalanche photo diode
apdhv	bias voltage	–	600 V	
apdlow	lower SCA level	–	–0.2 V to 5 V	
apdwin	SCA window	–	5 V	
<i>macro motors for P10 VME OMS58 TANGO server</i>				
dety	TANGO	–		rear detector bench
detz	TANGO	–		
detrx	TANGO	–		

# Chapter 5

## Results of Instrument Commissioning

### 5.1 Alignment and Characterisation of the Focusing Optics

The initial commissioning of the GINI-X instrument was performed at a photon energy of  $E = 7.9$  keV. Figure 5.1a shows the schematic setup used for the KB alignment. The incidence angle of the vertical KB mirror was adjusted with the tilt motor *kbyrot*, and *kbzrot* for the horizontal KB mirror, respectively. As the tilt motors do not have any encoders or reference marks for a zero position, the FDI alignment camera was used for the preliminary adjustment of the incidence angle on both mirrors. Therefore the active area of the FDI camera was placed in the focal plane. Figure 5.1b shows an image taken with wide opened slits. The lower left spot represents the undeflected primary beam resulting from the over-illumination of the mirrors. The upper left spot and the lower right spot originate from the single deflected beams of the vertical mirror, and the horizontal mirror respectively. The upper right spot is the focussed beam deflected by both the vertical and the horizontal mirror. The required pixel distance between the spots can be calculated from the focal length and the desired incidence angle. With the pixel size of  $6.5 \mu\text{m}$  of the FDI camera the gap between the spots has to be 188 pixels for a focal length of 302 mm and an incidence angle of 4 mrad, and 123 pixels for a focal length of 200 mm and incidence angle 4 mrad. The *kbyrot* and *kbzrot* motors were tilted until this gap was achieved.

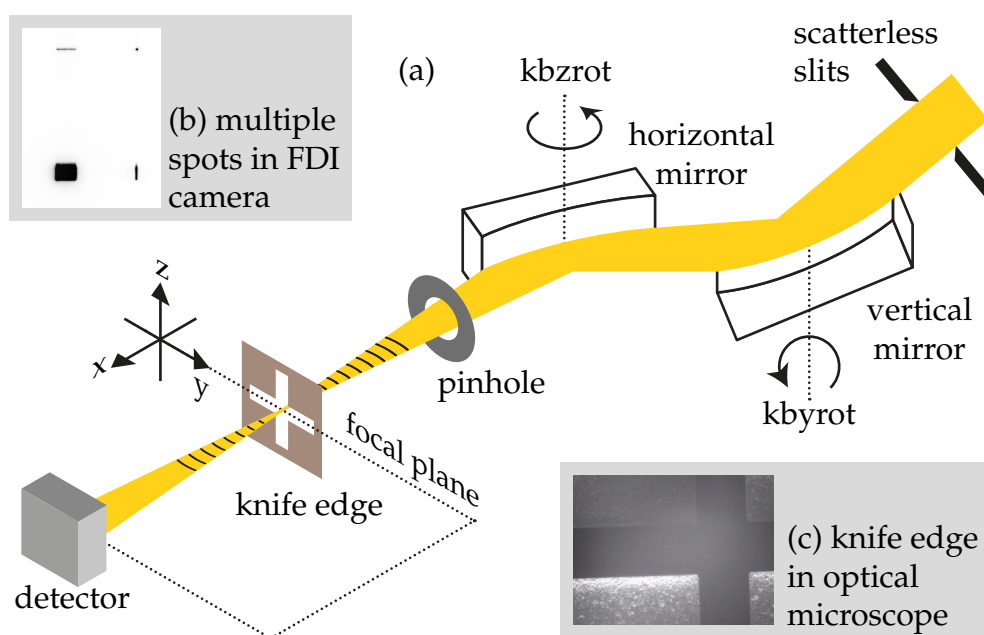
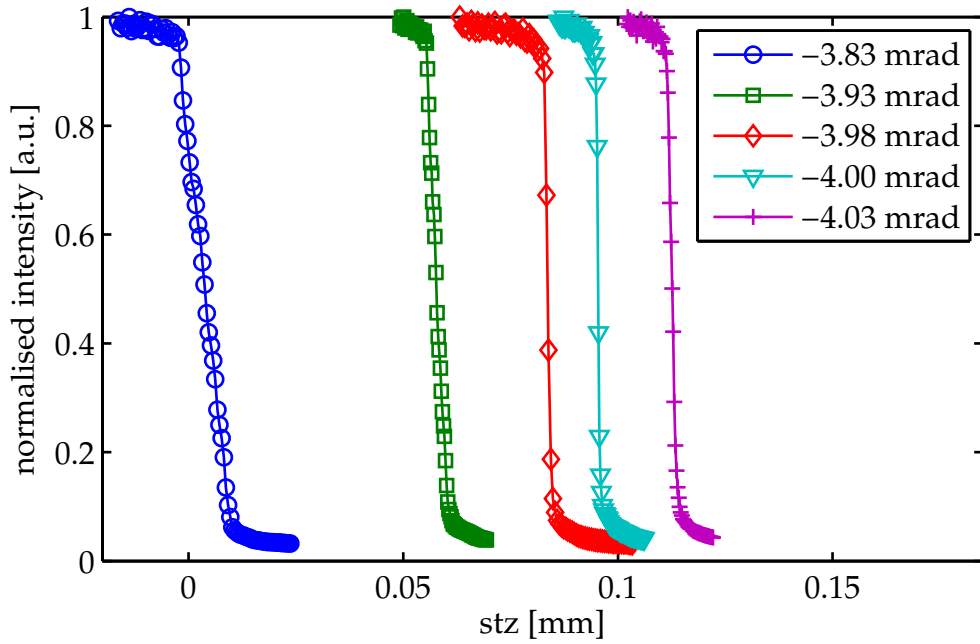


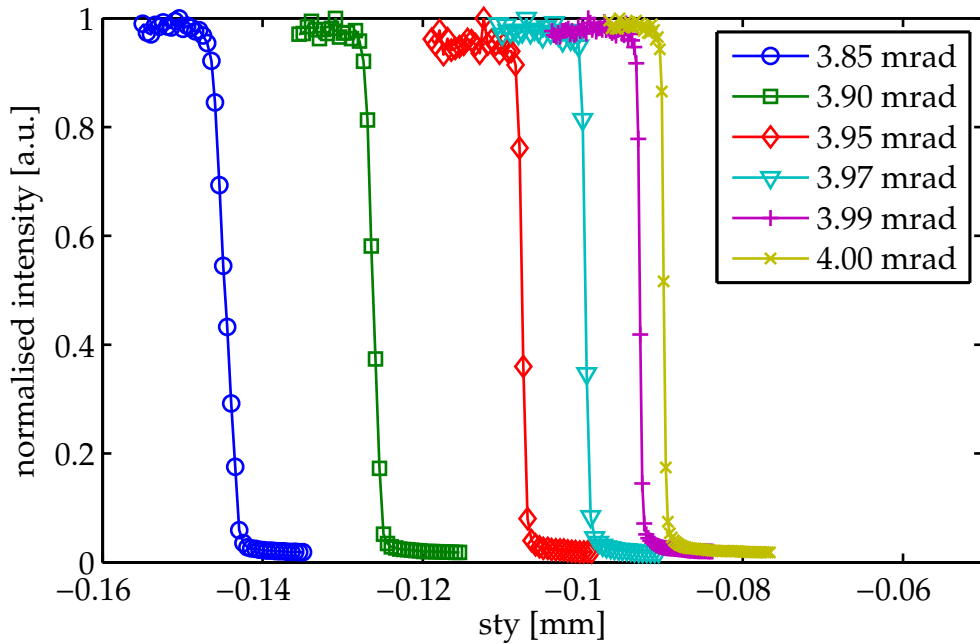
Figure 5.1: (a) Schematic setup for the KB alignment (b) spots of the KB as seen with the FDI (c) LIGA knife edge in the Mic.

In the next step the diameter of the focal spot was measured with a Au knife edge placed on the sample translations. The slits in front of the KB mirrors were opened to  $400\ \mu\text{m} \times 400\ \mu\text{m}$ , which corresponds to the geometrical acceptance of the tilted mirrors. To block undeflected or single deflected parasitic spots a  $450\ \mu\text{m}$  thick W pinhole with a diameter of  $1\ \text{mm}$  was inserted in the beam path between the mirrors and the focal spot. The knife edge was fabricated in a LIGA<sup>1</sup> [76] process by Arndt Last at the *Karlsruher Institut für Technologie*. It provides horizontal and vertical edges of a  $10\ \mu\text{m}$  thick Au layer, see Figure 5.1c for an image taken with the on-axis microscope. The knife edge was scanned through the focal spot with the *stz* motor in the vertical direction and with the *sty* motor in the horizontal direction. The resulting intensity profiles were measured for different incidence angles with the PIN diode mounted on the front detector bench, see Figure 5.2 for the profiles. The slope of the profiles is the steepest for the incidence angle the mirror is designed for. From the steepness the FWHM of the focus can be calculated by fitting an error

<sup>1</sup>German: *Lithographie, Galvanik und Abformung* (Lithography, electroplating, and moulding)



(a)  $kbyrot$  variation, vertical direction



(b)  $kbzrot$  variation, horizontal direction

Figure 5.2: Horizontal and vertical knife edge measurements with Au knife edge for different angles of  $kbyrot$  and  $kbzrot$ .

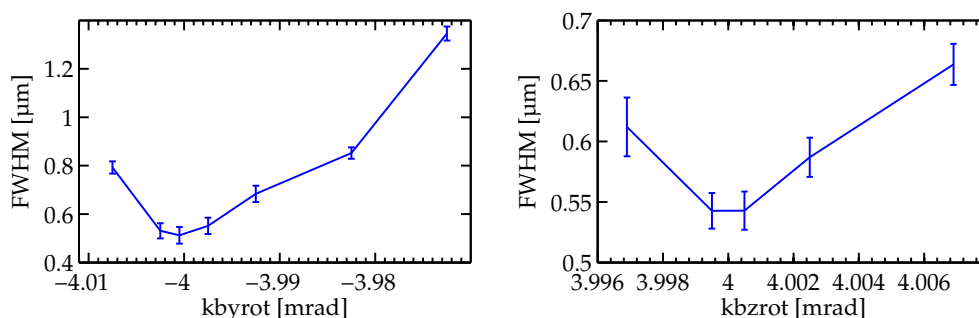


Figure 5.3: Focus diameter in focal plane for small variations of  $kbyrot$  and  $kbzrot$ . The FWHMs were retrieved from erfc-fits to the knife edge measurements.

function

$$f(x) = \frac{a}{2} \operatorname{erfc}\left(\frac{x - x_0}{\sigma}\right) + b x + c$$

to the measured profile. The FWHM is then

$$\text{FWHM} = 2\sqrt{\ln 2} \sigma.$$

The smallest FWHMs observed in the scans shown on Figure 5.2 were  $0.716 \mu\text{m}$  for the vertical direction and  $0.535 \mu\text{m}$  for the horizontal direction. At these positions of  $kbyrot$  and  $kbzrot$  the motor positions were set to be 4 mrad. These are the design values for both mirrors, which differ slightly from the incidence angles retrieved from the inspection of the mirrors as listed in Table 4.2 on page 49. With the change of the incidence angle on the mirror the X-ray beam is more deflected, resulting in shift of the edge position of the profiles.

The focus diameter is very sensitive to small variations on the incidence angle. Figure 5.3 shows the FWHMs retrieved from erfc-fits to profiles measured in the focal plane for small variations of  $kbyrot$  and  $kbzrot$ , respectively. Changes in the range of  $\mu\text{rad}$  cause a measurable increase of the focus diameter. However, the achieved FWHMs could not be brought below  $0.53 \mu\text{m}$  in both directions. The edge quality of the knife edge was probably limiting the measurements of smaller FWHMs.

To prove the correct position of the focal plane, the focus diameter was probed at different  $stx$  positions along the beam for various incidence angles. The retrieved FWHMs are shown in Figure 5.4. Even for positions 6 mm behind the nominal focal plane focus diameters below  $1 \mu\text{m}$  could



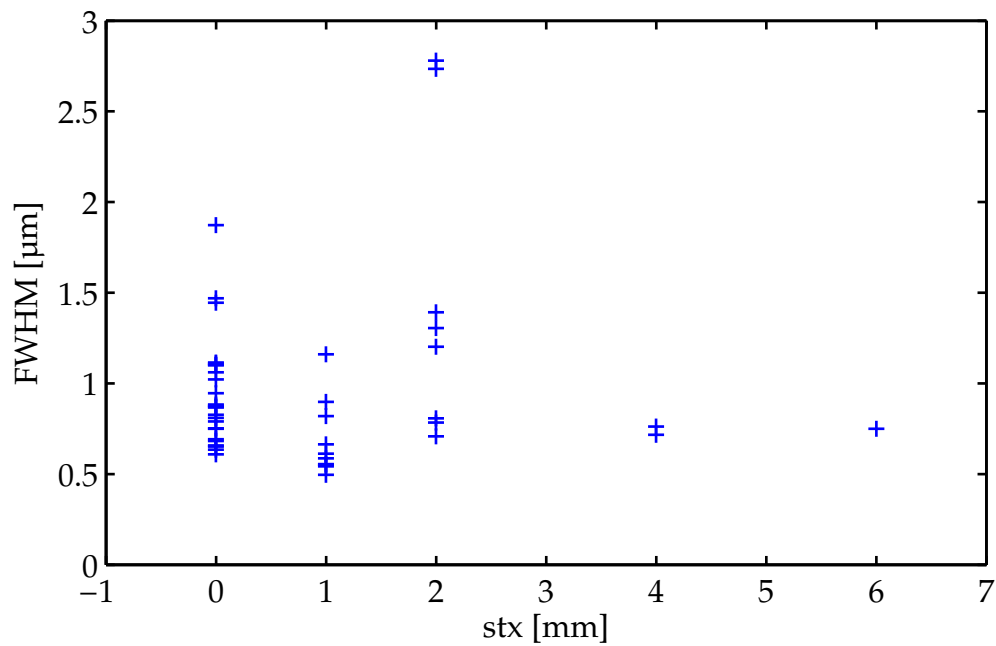


Figure 5.4: Horizontal focus diameters for different *kbzrot* angles at various *stx* positions, measured with Au knife edge. Due to the strong hysteresis effects of the *kbzrot* motor, the nominal values of *kbzrot* are not comparable between different knife edge scans and are thus not stated, see Figure 5.10.

be achieved when the incidence angle on the horizontal mirror deviated significantly from its design value.

In the last alignment step the focus diameters were measured with X-ray waveguides. To bring the waveguide into the focal plane, the focal plane of the on-axis microscope was first matched with the KB focal plane by focusing on the knife edge before it was removed from the sample position. Then the waveguide was brought in the focal plane of the on-axis microscope with an accuracy of approximately  $5\ \mu\text{m}$  which is the depth of focus of the microscope.

For the fine alignment and characterisation of the vertical KB mirror a one-dimensional waveguide on a Si waveguide chip was used. The waveguide was fabricated by electron beam lithography and reactive ion etching with a subsequent bonding step for the cap wafer [27]. The guiding layer of the 1D waveguide had a thickness of 29 nm and a length of 1.5 mm. Scanning a waveguide through the focus gives directly the beam profile in the scanning direction, when the width of the guiding layer is much smaller than the width of the beam. After the angular alignment of the waveguide the incidence angle of the vertical mirror was varied with the *kbyrot* motor until the width of the focus profile was minimised. The variations of the incidence angle on the mirrors were in the range of  $\mu\text{rad}$  and thus much smaller than the angular acceptance of the waveguide. The measured curve is shown in Figure 5.5. The fit of a Gaussian gave a FWHM of 221 nm for the vertical beam size.

The horizontal KB mirror was fine-tuned and characterised with the 1D section of a crossed waveguide. Its 35 nm thick C guiding layer was separated by 30 nm thick Mo interlayers from the Ge cladding [44]. The Mo interlayer has a higher optical contrast to the guiding layer than the Ge cladding, whilst the absorbance in the interlayer of the evanescent wave is reduced compared to the cladding [78]. The crossed waveguide was again first aligned in the focus. Then the incidence angle of the horizontal mirror was varied with the *kbzrot* motor until the width of the focus profile was minimised. The measured curve is shown in Figure 5.5. The fit of a Gaussian gave a FWHM of 203 nm for the horizontal beam size. So the achieved focus sizes at 7.9 keV was  $203\ \text{nm} \times 221\ \text{nm}$  (horiz.  $\times$  vert.).

With the optimised incidence angles on both KB mirrors the depth of focus was determined by measuring beam profiles at different *ax* positions along the beam. The vertical direction was scanned with the 1D waveguide on a bonded Si waveguide chip whilst the horizontal direction was scanned with the 1D section of the crossed waveguide, according

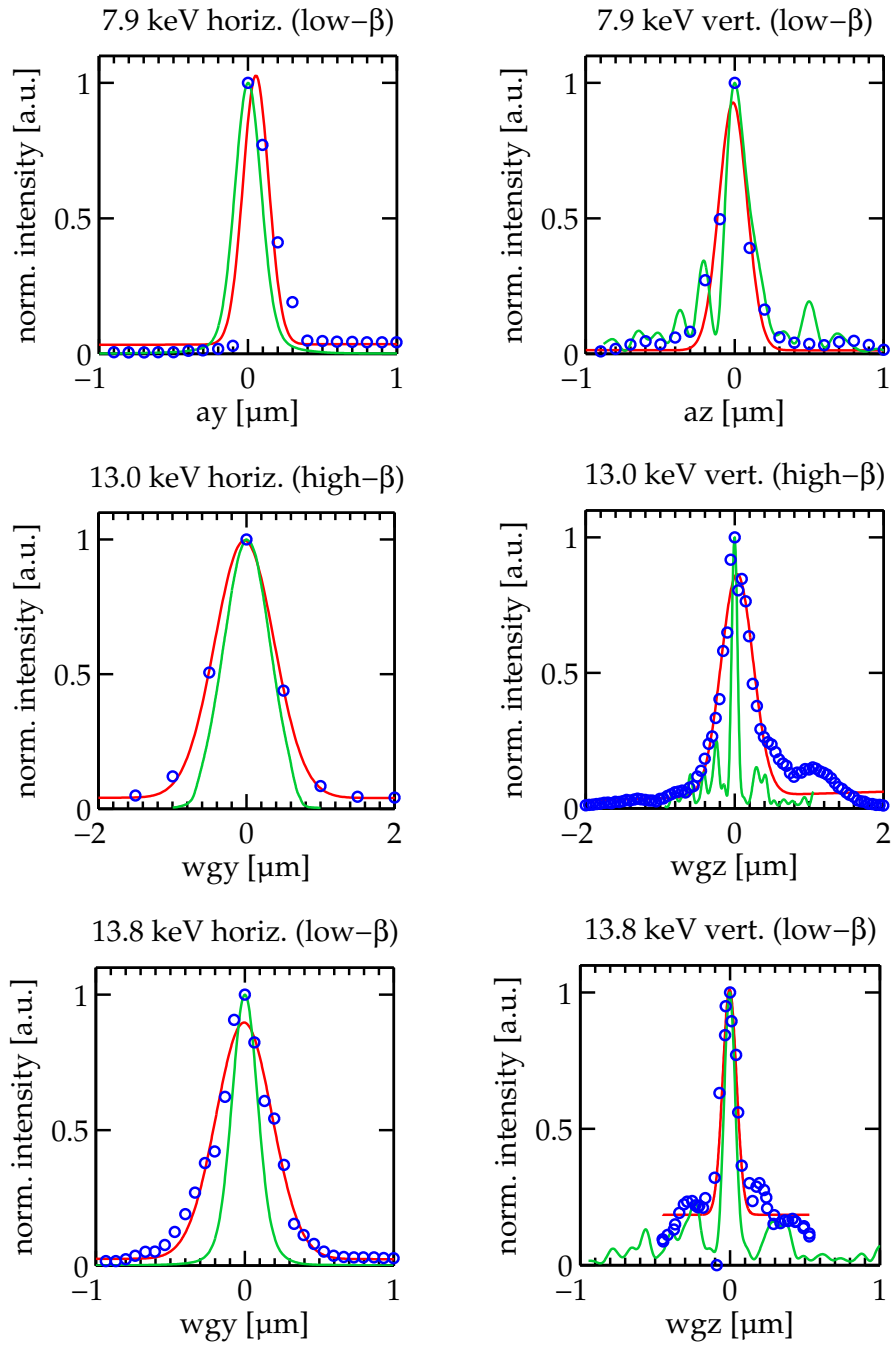
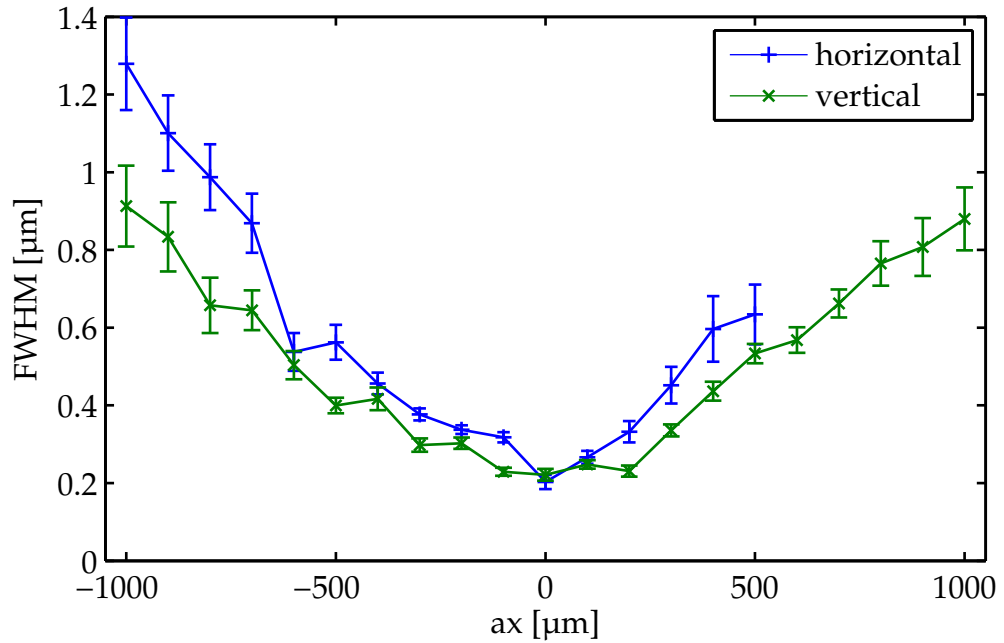
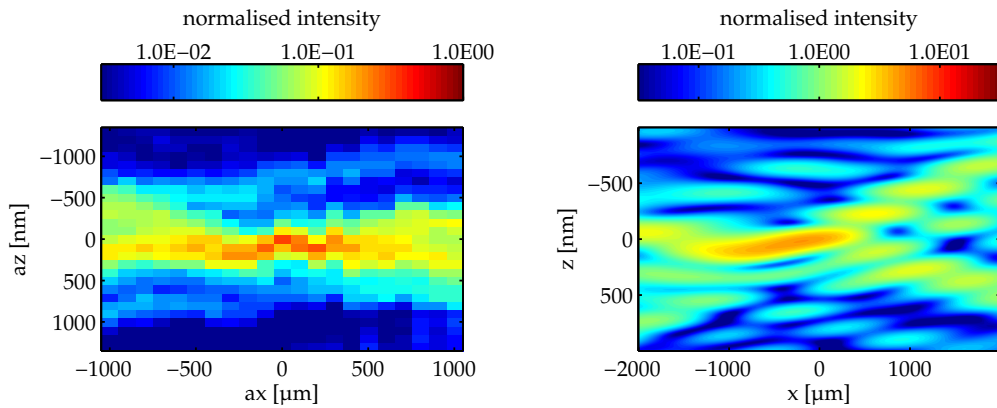


Figure 5.5: Measured horizontal (left row) and vertical (right row) beam profiles for 7.9 keV, 13.0 keV, and 13.8 keV. The Gaussian fits to the measurements are shown as the red curves. The green curves represent the beam profile as expected from Fresnel-Kirchhoff simulations. The resulting widths are summarised in Table 5.1.



(a) Horizontal and vertical depth of focus retrieved from Gaussian fits to beam profiles measured at various  $ax$  positions. The error bars indicate the uncertainty of the fit.



(b) Intensity map concatenated from the vertical beam profiles

(c) Simulated intensity map for the vertical direction.

Figure 5.6: (a) Depth of focus for 7.9 keV, (b) measured vertical near field intensity map, (c) simulated vertical near field intensity map.

to the procedure used to measure the focus size. Figure 5.6a shows the horizontal and vertical FWHMs of the beam as retrieved from Gaussian fits to the profiles at different  $ax$  positions. The measured vertical beam profiles can also be concatenated to a near field and plotted as an intensity map, see Figure 5.6b for the logarithmic 2D plot. Each measured profile was normalised to the integral intensity of the profile, assuming that each scan must contain the same overall intensity due to the conservation of energy. A simulation of the vertical near field at 7.9 keV is shown in Figure 5.6c. The data was generated by a numerical calculation of the Fresnel-Kirchhoff diffraction integral. It is developed by Markus Osterhoff as a part of his dissertation at the Georg-August-Universität Göttingen [59]. In the simulation an extended source is modelled as a partially coherent superposition of many point sources [61, 62]. The electrical fields of each point source are then propagated to the data points on the mirror surface, that is given either the ideal height profile of the elliptical KB mirrors or the real, measured height profile, as shown in Figure 4.7 on page 48. The simulation takes into account the local reflectivity of the coating in each data point on the curved mirror surface, before the electrical field is propagated from the mirror surface to the near field points. The near field is the averaged intensity of many repeated simulation runs, each starting with different random phases in the source points.

With this simulation also the expected focus size was calculated for photon energies of 7.9 keV and 13.8 keV of a low- $\beta$  source ( $36 \mu\text{m} \times 6 \mu\text{m}$  ( $h \times v, 1\sigma$ )) and for 13.0 keV of a high- $\beta$  source ( $141 \mu\text{m} \times 5.5 \mu\text{m}$  ( $h \times v, 1\sigma$ )), shown as green curves in Figure 5.5.

Later experiments with the instrument have been performed at these energies. The KB mirrors were again aligned after the procedure described above. In a short summary, the main alignment steps are:

1. Placement of the optical table on the reference marks in the experimental hutch.
2. Angular alignment of the breadboard to the reference of the autocollimator.
3. Lateral alignment of the breadboard in the X-ray beam with the KB slits in front of the KB mirrors.
4. Check of incidence angle on KB mirrors with FDI camera placed in the focal plane of the KB.

E [keV]	simulated FWHM (h × v) [nm]	measured FWHM (h × v) [nm]	photon flux [photons/s]
7.9	214 × 183	203 × 221	$3.38 \cdot 10^{11}$
13.0	758 × 85	946 × 507	$2.89 \cdot 10^{12}$
13.8	211 × 82	443 × 109	$5.56 \cdot 10^{11}$

Table 5.1: Simulated and measured FWHMs of the focus size for various photon energies E. The values for 13.0 keV apply for a high- $\beta$  source. The photon flux is determined from the current of a calibrated PIN diode and are upscaled to a storage ring current of 100 mA for comparison.

5. Pre-adjustment of the incidence angles with the Au knife edge (placed in the focal plane) by maximising the edge slope.
6. Fine-adjustment of the incidence angles with the 1D sections of a crossed waveguide.
7. Characterising of the horizontal and vertical focus size with the 1D sections of the crossed waveguide.

The later experiments used already the modified waveguide translations with positioners of improved accuracy stability for the y and z direction, see Section A.1 on page 107. The measured focus profiles for 13.0 keV and 13.8 keV are shown in Figure 5.5. The beam width was again determined by a Gaussian fit (red curve). The vertical focus size at 13.0 keV with a high- $\beta$  source differs considerably from the simulation. The discrepancy may be a result of the vibrations induced by the strong cooling of the monochromator crystals. For 13.0 keV the undulator gap was closed to 10.77 mm to generate the photons in its third harmonic, resulting in a high heat load on the monochromator crystals. The deviation of the horizontal focus size from the simulation at 13.8 keV is unclear. The horizontal mirror was aligned with the same procedure as the other mirrors and energies. The results for the measured and simulated FWHMs of the beam for the three used energies are listed in Table 5.1 along with the measured photon flux (scaled to a storage ring current of 100 mA) in the focus. The total photon flux has been derived from the measured current of the PIN diode placed close to the focal plane. As described in section 4.8 the current of a PIN diode generated by X-ray photons is proportional to the photon flux. As the calibration factor is energy dependent, the factor has to be calculated for each of the three used photon energies. The conversion

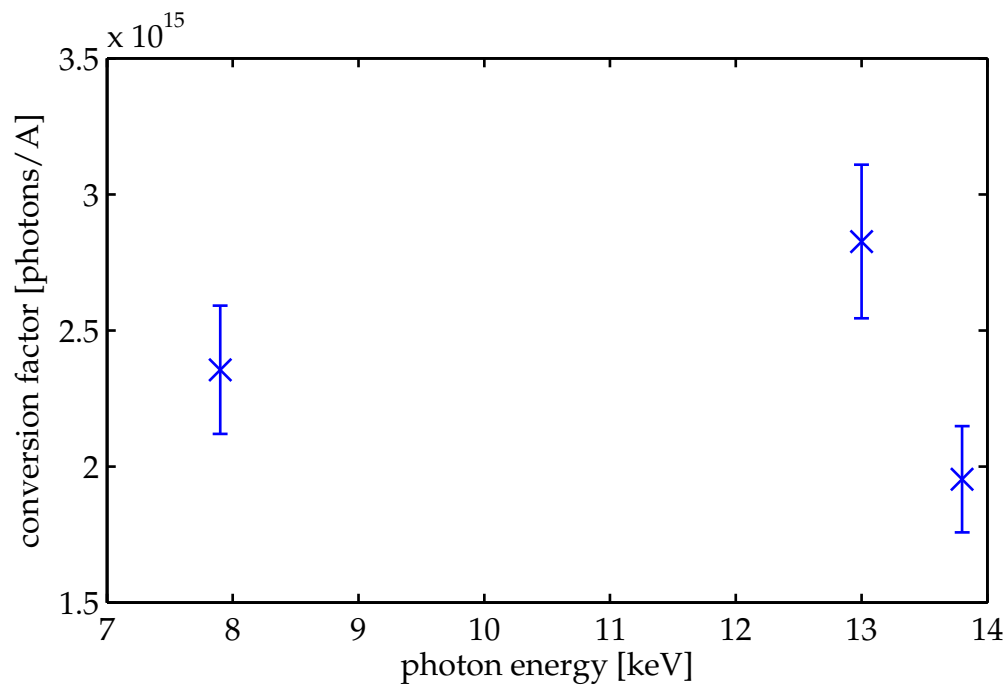


Figure 5.7: Calibration factors of the PIN diode. The error bars indicate estimated imponderabilities of the individual calibration processes.

factors are retrieved after the same principle, but with slight differences in the individual realisation. The calculation is based on the linear relation of the integral photon counts measured with a single photon counting device (PILATUS) and the current generated in a PIN diode and measured with a picoammeter (Keithley 6485). The retrieved conversion factors for 7.9 keV, 13.0 keV and 13.8 keV are shown in Figure 5.7. Ideally, the PILATUS and the PIN diode are both placed at the same distance from the source for the measurements. Then the absorption losses are equal and the beam intensity is the same for both detectors. This was the case for the the 13.8 keV measurements. For the conversion factor measurements at 13.0 keV the PILATUS was placed on the rear detector bench whilst the PIN diode remained at its standard position on the front detector bench. So the calculation of the conversion factor for this energy includes the transmission of the flight tube between the two detector benches resulting in a too small value. The PILATUS measurements and diode current measurements for the conversion factor at 7.9 keV were taken on different days. Drift effects of the monochromator crystals, that were particularly observed during the first commissioning experiments, had an impact to the intensity of the incoming beam. The monitor signals could not be used for normalisation as the monitor counts are not recorded for PILATUS images. These effects are estimated to give an error of 10 %, as indicated by the error bars in Figure 5.7. Although the theory of operation does not indicate a change over time of the conversion factors, it is strongly recommended to calibrate the PIN diode at the beginning of each experiment run and for each photon energy to reflect changes in the total accuracy of the used devices.

During the commissioning experiments the far fields of the KB mirrors were measured with the LCX, SCX and PILATUS detector, see Figure 5.8 for the images plotted in logarithmic scale. The far fields form a rectangular beam on the detector as expected from the perpendicular arrangement of the vertical and horizontal mirror of the KB system. The far fields show vertical fringes in all measurements. Variations on the size of the illuminating beam did not change the fringes, indicating that the fringes do not originate from the unpolished mirror border area of the mirror but from the real height profile created through the polishing of the vertical mirror. The central beam of the LCX image (Figure 5.8a) has a rather homogeneous intensity distribution with no large artefacts visible but small varying horizontal and vertical lines. Figure 5.8b was taken with the SCX detector. The features clearly visible in the LCX detector appear



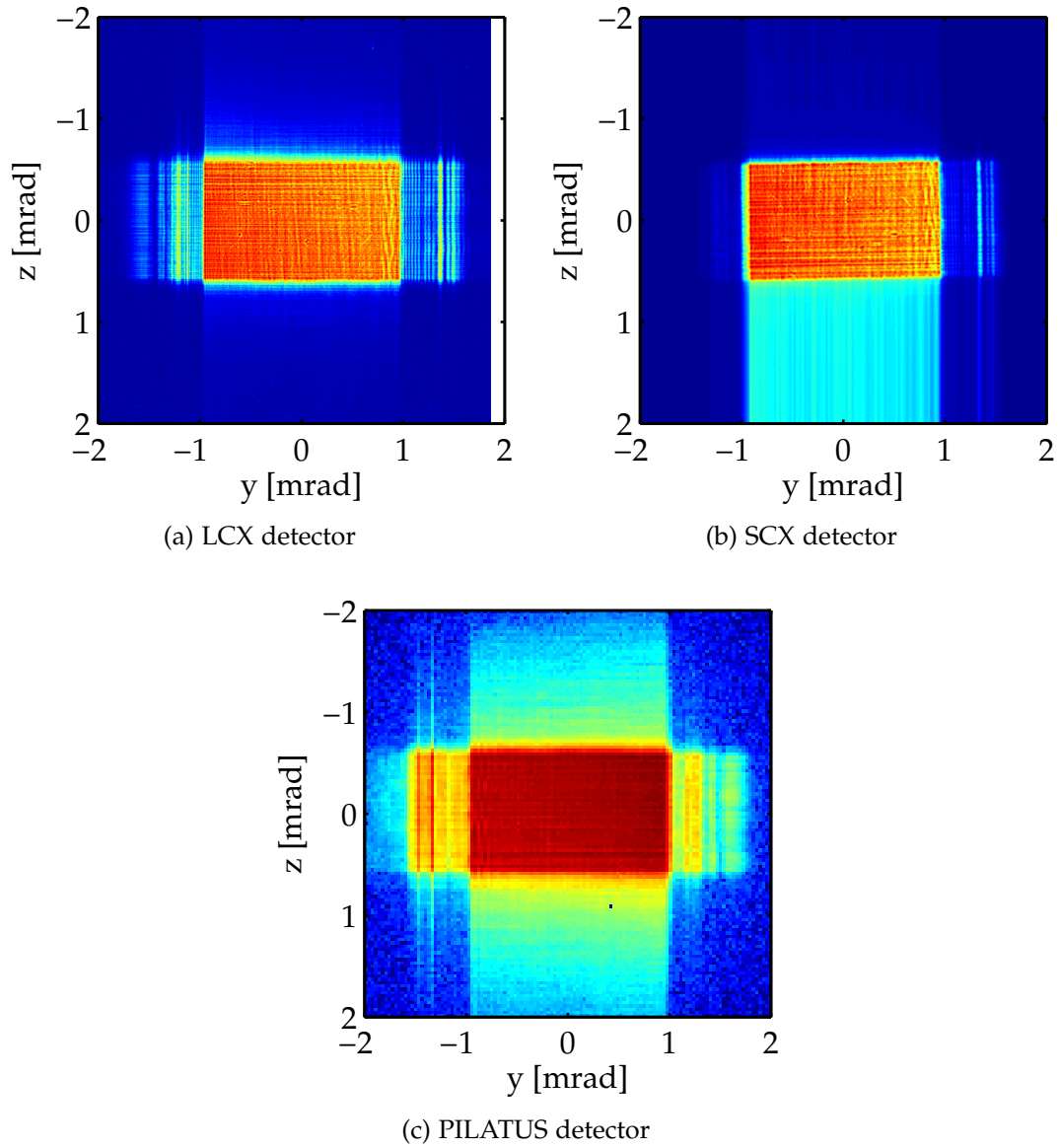


Figure 5.8: KB far fields at 7.9 keV measured with different detectors.

to be washed-out, mainly due the larger point-spread-function of the SCX detector. Also the vertical fringes are less distinguishable and less intense than in the LCX detector whilst the central beam has comparable intensities. The area below the central beam with the increased intensity results from the readout of the CCD chip of the SCX detector. The image was taken with an Uniblitz XRS6 fast shutter<sup>2</sup> which has a closing time of 20 ms. Due to technical problems the readout of the CCD chip started before the shutter was fully closed generating the vertical readout halo. The central beam is also slightly rotated caused by an inclined mounting of the SCX detector. As the problem of the readout synchronisation could not be solved and the SCX has also an reduced resolution at reduced sensitivity the SCX detector was not used anymore in the following experiments with GINI-X. The image taken with the PILATUS detector (Figure 5.8c) shows even less details in the central beam as its pixels are 8.6 times larger than the pixels of the LCX detector. In the vertical direction the PILATUS image shows a homogeneous scattering at the width of the central beam. The same effect is also visible in the LCX image although the effect is considerably weaker here. With the single photon counting pixel and the higher dynamical range weak signals get visible before the other detector pixels in areas with much higher intensity saturate.

The beam divergence  $\delta$  of the focussed X-ray beam defines the beam size  $s_b$  of the far field on the detector at a distance  $d$  behind the focus. In small angle approximation  $\delta$  is then  $\delta = \frac{s_b}{d}$ . The expected X-ray beam divergence can be obtained from the geometrical dimensions of the KB mirrors, as sketched in Figure 5.9. From the geometrical relations of a triangle one obtains for small incidence angles  $\theta$

$$\delta = \delta_1 + \delta_2$$

with

$$\delta_1 = \frac{l\theta}{2\sqrt{f^2 + \frac{l^2}{4} - fl\theta}}$$

$$\delta_2 = \frac{l\theta}{2\sqrt{f^2 + \frac{l^2}{4} + fl\theta}}.$$

The expected divergence for the vertical mirror with its polished length of  $l = 94$  mm is then  $\delta_{\text{vert.}} = 1.25$  mrad and for the horizontal mirror with its

<sup>2</sup>The CEDRAT FPS400M fast shutter described in section 4.3 was not yet installed for these measurements.

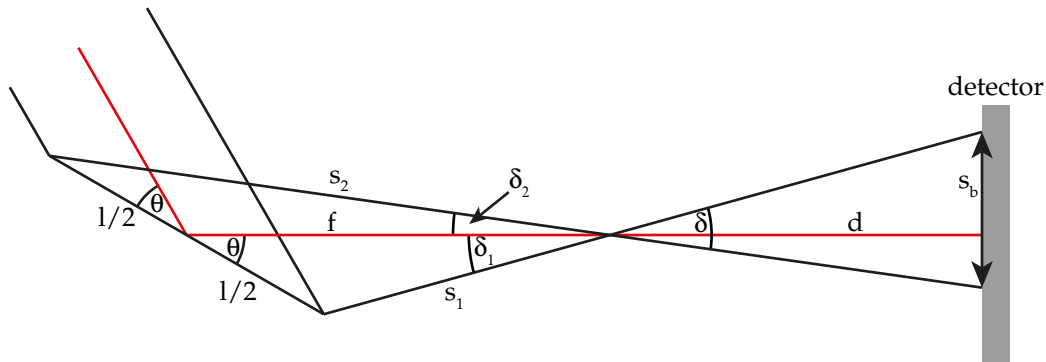


Figure 5.9: Sketch of KB dimensions and angles used for the determination of the beam divergence  $\delta$ .

polished length of  $l = 93.8$  mm is  $\delta_{\text{horiz.}} = 2.01$  mrad. The measured far fields shown in Figure 5.8 are in very good agreement with the expected values of the beam divergence.

During the commissioning the *kbyrot* and *kbzrot* motors showed strong hysteresis effects in their movement. Figure 5.10 shows seven subsequent scans of the *kbzrot* motor over the edge of the Au LIGA structure. For motors with good linearity and no hysteresis all scans should overlay. The first scan #66 was scanned over a range from 4.6 mrad to 4.8 mrad in 20 steps. The scans #66 – #69 were scanned over the same range from 4.6 mrad to 4.8 mrad in 40 steps. The scans #70 – #72 were scanned over a smaller range from 4.7 mrad to 4.8 mrad, but also in 40 steps. Between the scans the *kbyrot* motor was only moved from the last point of the previous scan to the first point of the next scan. No other motors were moved. The intensity profiles show for the larger scans #66 – #69 a drift of approximately 0.015 mrad between the scans, and a drift of approximately 0.0075 mrad for the shorter scans #70 – #72. This is a non-linearity of 7.5 % for both the longer and the shorter range. Similar behaviour was observed for the *kbyrot* motor.

The  $\text{Si}_3\text{N}_4$  membranes used as vacuum windows of the KB vessel showed degeneration effects after one week of beam exposure, see Figure 5.11. The entrance window was hit by the X-ray beam in the lower right quadrant. The exit window was hit close to the center of the membrane due to the beam deflection by the KB mirrors. While the areas of the plain beams show no visible modifications the surrounding areas have suffered from the beam exposure. As the membranes are fragile when touched it is quite difficult to examine the visible defects and figure out

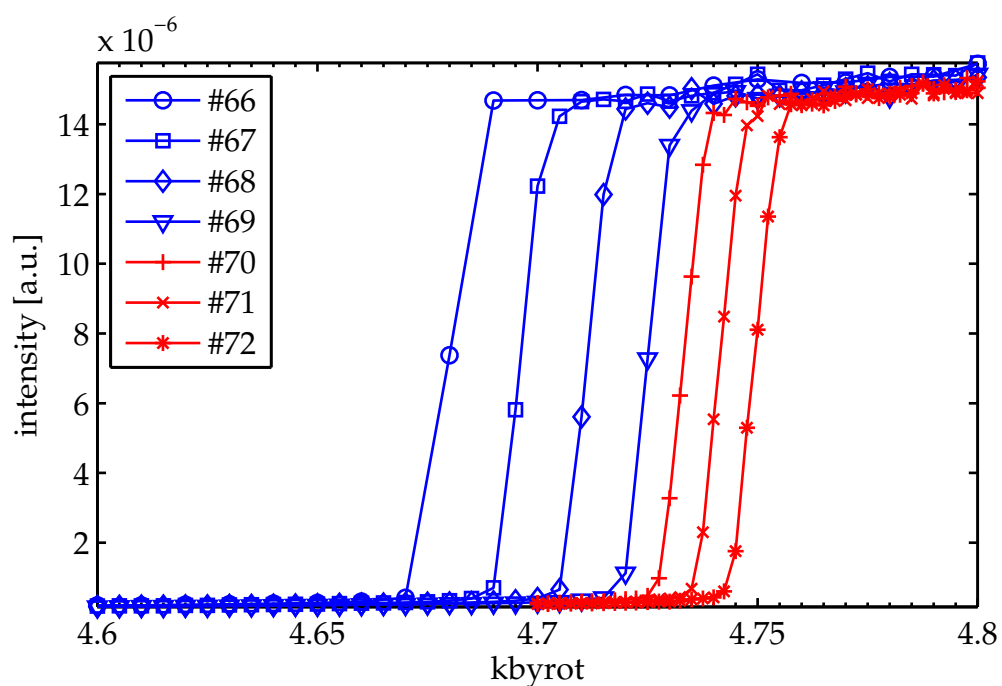


Figure 5.10: Repeatability of the *kbyrot* motor: The Au knife edge was scanned in seven subsequent scans, #66 – #69 having a step size of 0.005 mrad and #70 – #72 having a step size of 0.0025 mrad. The shift of the edge from scan to scan indicates an inhomogeneity for the movements in forward and backward direction.

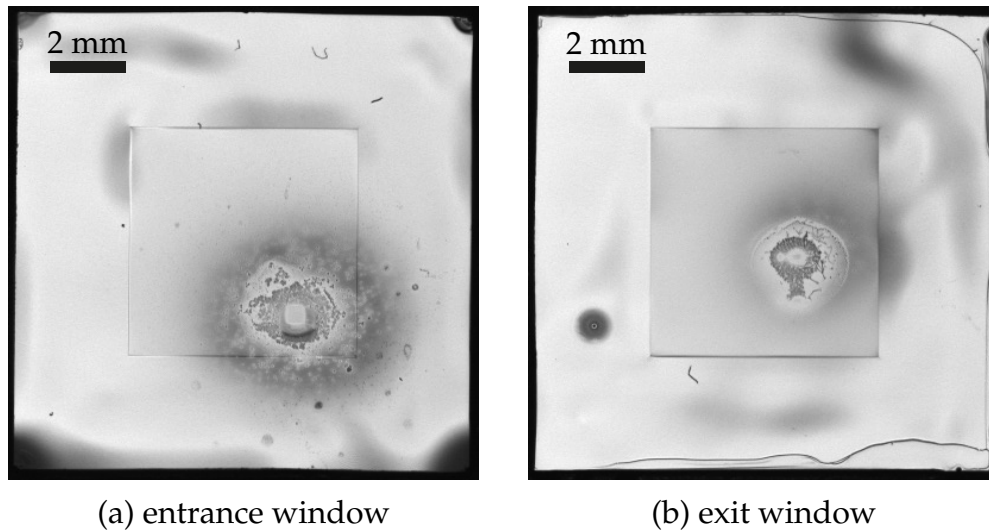


Figure 5.11: Images  $\text{Si}_3\text{N}_4$  membranes: The areas exposed by the X-ray beam and surrounding areas show a strong contamination with debris affecting the stability of the  $\text{Si}_3\text{N}_4$  membranes.

whether the defects are debris on the surface or defects in the membrane structure itself. The high intensity of the X-ray beam creates ionized gas and ozone in the air, that will probably react with the membranes. To avoid any risk of a burst of the  $\text{Si}_3\text{N}_4$  membranes during an experiment run, the windows were exchanged before each start of an experiment run. One of the replacement membranes burst due to a fabrication defect during the evacuation of the KB vessel. After this accident the far field images of the KB got more features, as already shown above. Although the features may be caused by the degeneration of the mirror coatings, it is more likely that burst fragments of the membrane deposit on the mirror surface. The seen far field features are then the phase contrast images of the fragments. A cleaning is rather difficult as the mirrors are built into the mechanics and thus the surfaces are not easily accessible. Also the cleaning procedure may cause additional damage on the mirror coating. If the features increase to a level, where the KB beam gets unusable, the coatings of mirrors have to be renewed.

Figure 5.12 shows typical temperature profiles measured at different instrument positions over a period of 12 hours. The temperature increases after the access to the experimental hutch, as indicated by the vertical dotted lines. After a short time access the temperature goes back to its

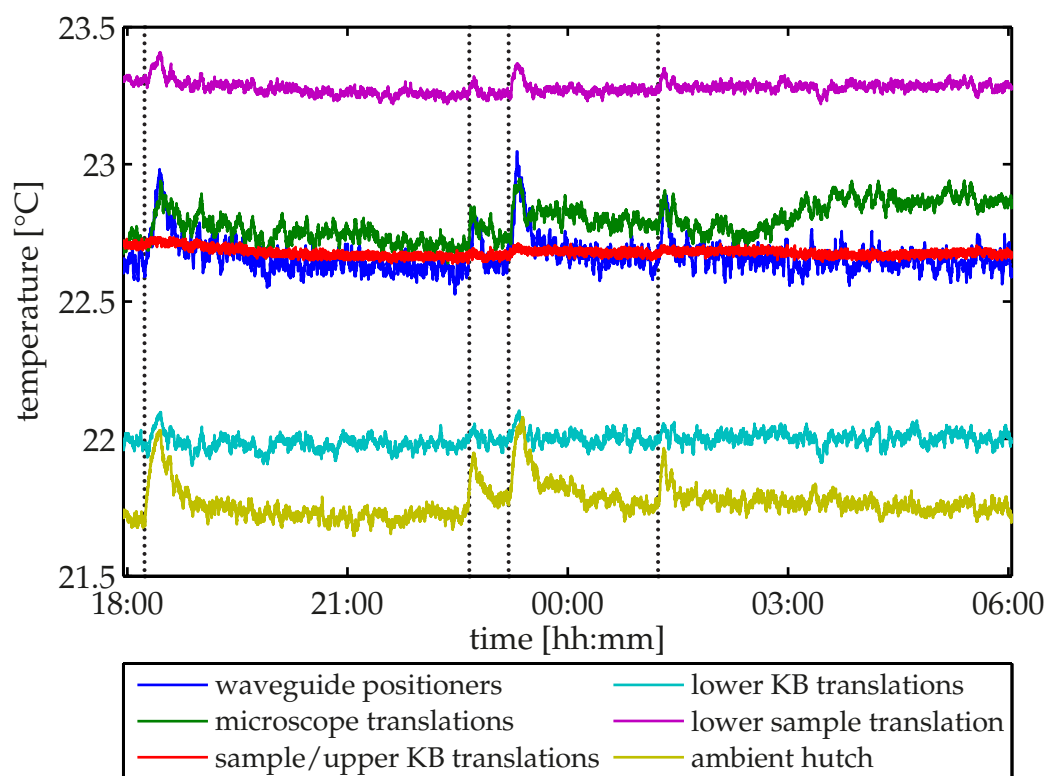


Figure 5.12: Measured temperature profiles at different instrument positions over 12 hours. The dotted vertical lines indicate the times of access to the experimental hutch.

normal level in less than 30 minutes. When back at its normal level the long-term temperature stability is better than 0.1 °C, giving optimal conditions for extended measurements with minimal thermal drift.

## 5.2 First Imaging Experiments

The imaging capabilities of the setup presented in Chapter 4 were demonstrated on resolution objects and biological samples studied in waveguide beams and in the plain beam of the KB mirrors.

For a fullfield hologram the Siemens star of the NTT-AT ATN/XRESO-50HC test pattern was placed 20 mm behind the exit of a bonded Si waveguide chip. The used channel had a cross section of 87 nm × 20 nm and a length of 1.5 mm. The hologram was recorded at a photon energy of 7.9 keV with the LCX CCD in 10 accumulations a 5 s, resulting in a total exposure time of 50 s (Figure 5.13a). The waveguide far field without sample but the same exposure parameters as the hologram is shown in Figure 5.13b. Both images are corrected by a dark image, that was recorded with the same exposure parameters but without X-rays. The dark image subtraction is used to correct the electronic read out noise. After the correction some pixel values were negative, caused by the statistical fluctuations of the read out noise. As negative intensities do not have a physical sense, these negative pixel values were set to one. A value of zero seems to be more reasonable but causes numeric problems when images are divided by the corrected images. The introduced error is negligible as the mean count rate over all pixels of the detector is about 3 – 4 orders of magnitude larger than one. The division of the corrected hologram by the corrected waveguide far field yields the normalised hologram of Figure 5.13c. Whilst the waveguide far field and the hologram of the Siemens star both are dominated by the fringe pattern of the multimodal waveguide, the division of the hologram by the waveguide far field can nearly fully remove these fringes, yielding to the normalised hologram in Figure 5.13c. A holographic reconstruction as described in section 2.5 of the corrected hologram contains these fringes as artefacts (Figure 5.13d). However, the holographic reconstructed phase of the normalised hologram (Figure 5.13e) does not show any fringes related to the illumination. The whole Siemens star is resolved, even in areas of the illumination with low intensities. A magnification of the inner part (Figure 5.13f, indicated by the white rectangle in Figure 5.13e) shows clearly distinguishable structures

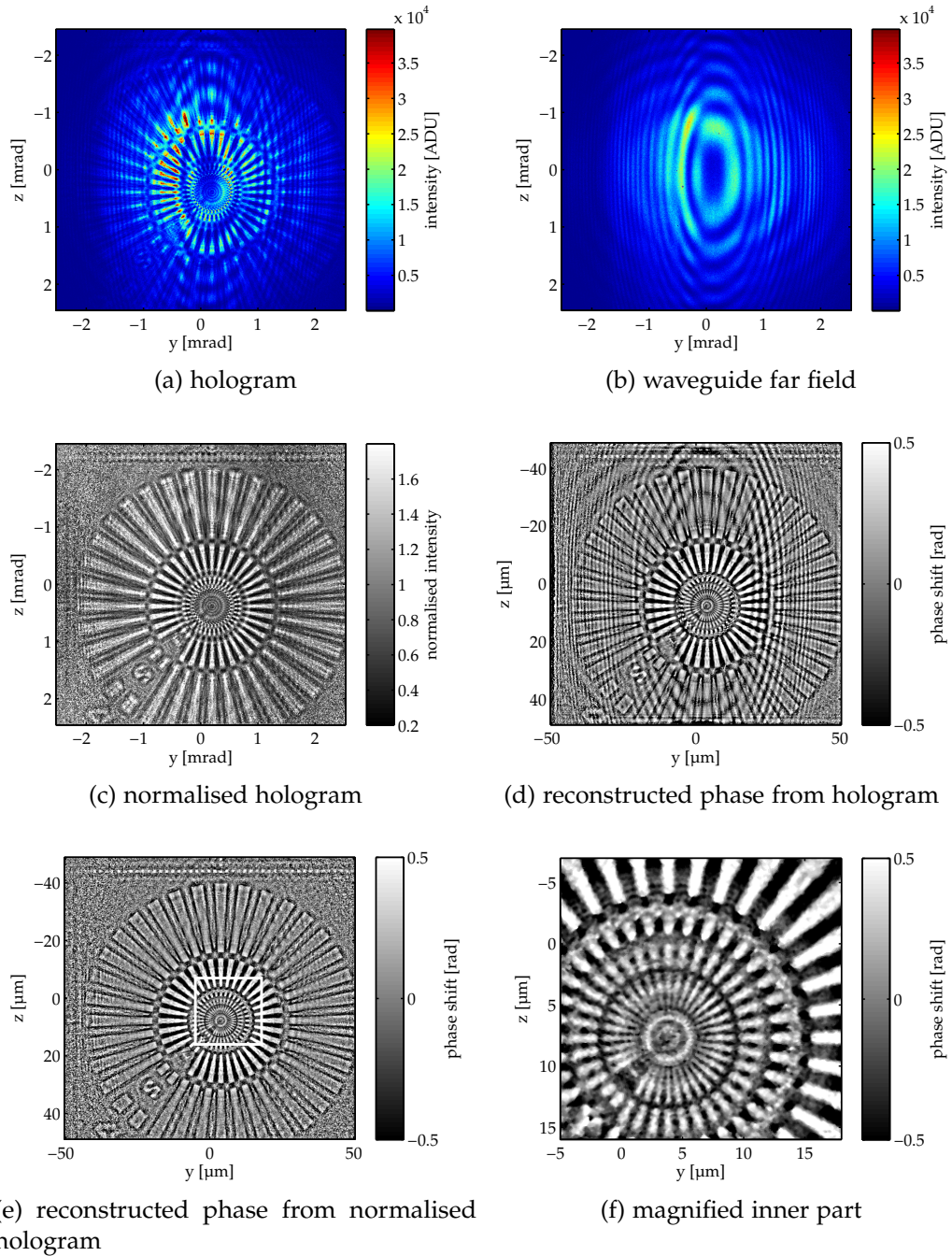


Figure 5.13: Holography results of the Siemens star placed 20 mm behind a bonded Si waveguide.



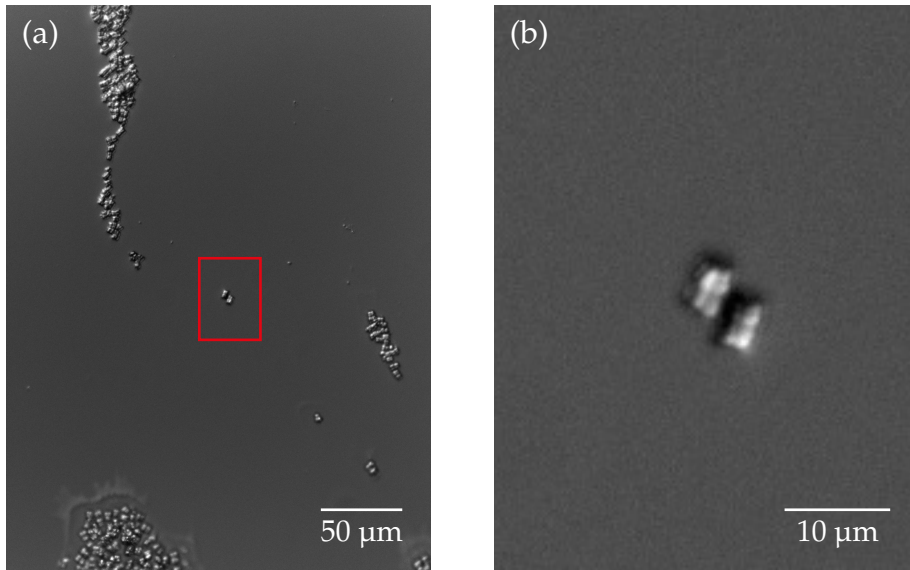


Figure 5.14: (a) Optical light microscopy image of the *D. radiodurans* sample in differential interference contrast. (b) Magnification of the red rectangle in (a). This isolated group of cells was measured.

of 200 nm line width in both the horizontal and the vertical direction, although the intrinsic twin image blurs the image.

One of the first biological specimens successfully measured with GINI-X was a sample of freeze-dried *Deinococcus radiodurans* cells. The *D. radiodurans* bacteria were prepared on a  $\text{Si}_3\text{N}_4$  membrane, see Figure 5.14 for an optical light microscopy image in differential interference contrast, and placed  $x_1 = 5.10$  mm behind the exit of a crossed X-ray waveguide, as determined by the optical on-axis microscope. The LCX CCD detector<sup>3</sup> was placed  $x_2 = 5.17$  m away from the waveguide. This configuration gave a geometric magnification of  $M = 1015$  and an effective pixel size of 20 nm in the sample plane. The *D. radiodurans* were illuminated by the beam of a photon energy of 13 keV exiting from a crossed waveguide with Mo[30 nm]/C[35 nm]/Mo[30 nm] layers in a Ge cladding [43]. The waveguide entrance was placed in the focal spot of the KB mirrors. As the undulator source was operated in high- $\beta$  mode during this experiment, i.e., the source size was increased to  $141 \mu\text{m} \times 5.5 \mu\text{m}$  ( $h \times v$ ,  $1\sigma$ ), the diameter of the KB focus was  $946 \text{ nm} \times 507 \text{ nm}$  ( $h \times v$ , FWHM), cf. Table 5.1. The hologram data of the *D. radiodurans* was recorded in an

---

<sup>3</sup>The MAXIPIX detector was not yet available for this experiment.

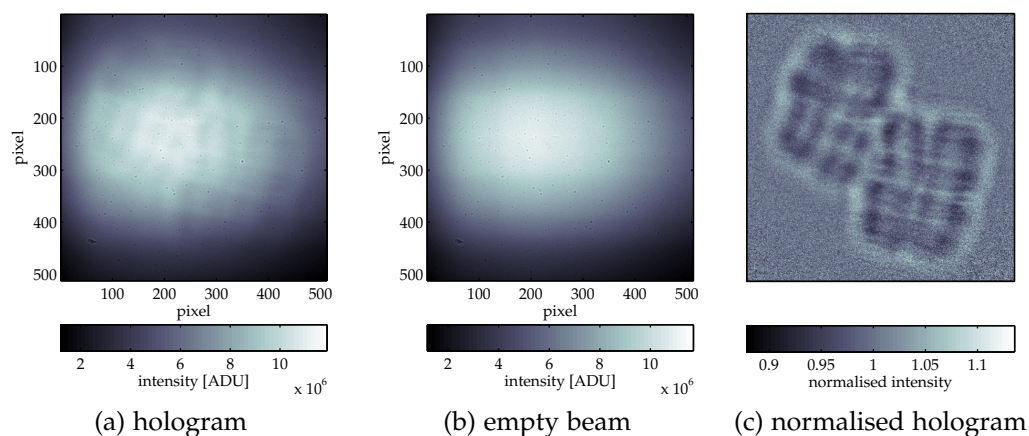


Figure 5.15: (a): Sum of all 750 dark-field subtracted acquisitions with the *D. radiodurans* cells in the waveguide beam. (b): Sum of all 750 dark-field subtracted acquisitions of the empty waveguide beam. (c): Normalised hologram.

alternating sequence of acquisitions with the sample in the beam  $I_{S,i}$  and without the sample in the beam  $I_{E,i}$  (empty beam). A total of 750  $I_{S,i}$  and 750  $I_{E,i}$  acquisitions were taken, each at an exposure time of 4 s. For the analysis a full dataset of all acquisitions and a reduced dataset of the first 250  $I_{S,i}$  and 250  $I_{E,i}$  acquisitions was prepared. As the measurement of the full dataset took 159 minutes, including the exposure time, detector read out, motor movements and communications overhead, the reduced dataset is expected to be less affected by drift effects. The sum of all dark-field subtracted sample holograms  $I_S$  and empty beam acquisitions  $I_E$  are shown in Figure 5.15a and (b). Both images show a pattern of tiny spots having the same positions in both images. They originate from dust particles on the windows of the flight tube. The pattern is eliminated in the normalized hologram  $I_S/I_E$ , see Figure 5.15c, that is used for further analysis. A comparison of the single-step holographic reconstruction of the full and the reduced dataset in Figure 5.16 shows sharper edges between the cell compartments in the reduced dataset of 250 sample acquisitions than in the full dataset of all 750 sample acquisitions, i.e., the resolution of the full dataset is worse due to drift effects during the measurement. Therefore, the reduced dataset is used for the analysis with the modified hybrid-input-output algorithm (mHIO), which is capable of an effective suppression of the intrinsic twin image of single-step

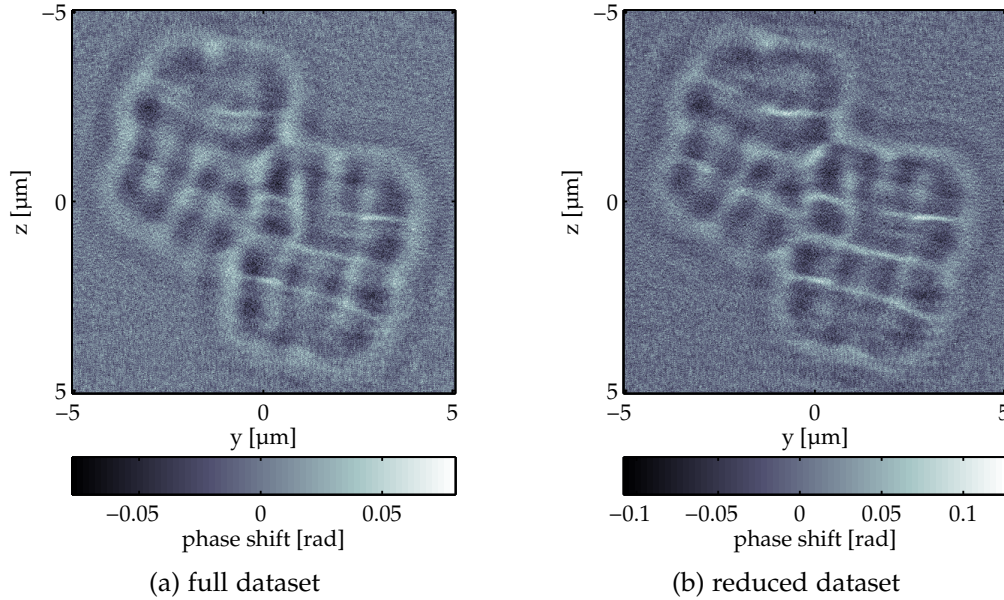


Figure 5.16: Single-step holographic reconstruction of the full dataset (a) and the reduced dataset (b). The separation of the cell compartments is smeared out in the full dataset compared to the reduced dataset.

holographic reconstruction [25, 26]. The algorithmic feedback parameters  $\beta$  and  $\gamma$  were found to give the best results for  $\beta = \gamma = 0.2$ . The smallest threshold parameter converging within 5000 iterations was  $\tau = 0.65$ . As the threshold parameter reflects the noise characteristics of the data [26], a value of  $\tau \geq 1$  is expected for Poisson-distributed data, that also include detector-specific sources of noise. The present value  $\tau \leq 1$  results from the applied median filter to the normalised hologram, reducing the noise characteristics of the data. The mHIO algorithm was run 25 times with these parameters, each run having a different random initial guess. The final result was then obtained by the complex averaging of the results. The retrieved phase shift is shown in Figure 5.17. The outer contour of the group of cells clearly define and is not blurred by the twin image, as shown in Figure 5.16 for the single-step holographic reconstruction. Inside the sample cell compartments are distinguishable. Further details are hard to see due to the limited resolution. The resolution in future experiments could be increased by several improvements. A faster and more sensitive detector with single photon counting capabilities, e.g. the MAXIPIX detector, would reduce the total measurement time, and thus

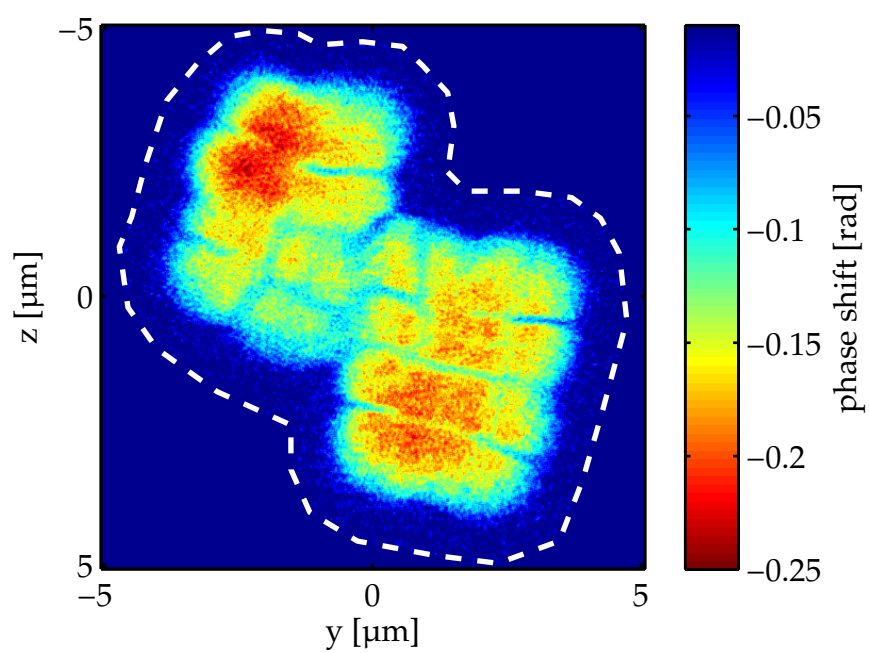


Figure 5.17: Reconstructed phase shift of *D. radiodurans* retrieved with the mHIO algorithm. The dashed white line indicates the support used for the mHIO reconstruction.

the drift. Shorter exposure times would also allow scanning microscopy in smaller waveguide beams at a shorter defocus distance  $x_1$ , giving also a larger geometric magnification  $M$ .

Although the LCX CCD detector does not provide single photon counting directly, the photon flux, and thus the applied dose on the sample, can be estimated by an analysis of the POISSON noise of the acquired images. For measurements of POISSON distributed photons the width of the distribution  $\sigma$  depends on the photon counts  $N$  by  $\sigma = \sqrt{N}$ . The returned pixel values of the CCD are given in analog-to-digital units (ADU). These values are measured by the ADC and are proportional to the charge accumulated in the pixels. The charge in a pixel again is proportional to the number of photons and their energy detected by the pixel. The ADU value  $n_{\text{ADU}}$  is then proportional to  $N$  by  $N = a n_{\text{ADU}}$ . The width of distribution of the ADU values scales also with  $a$ , leading to

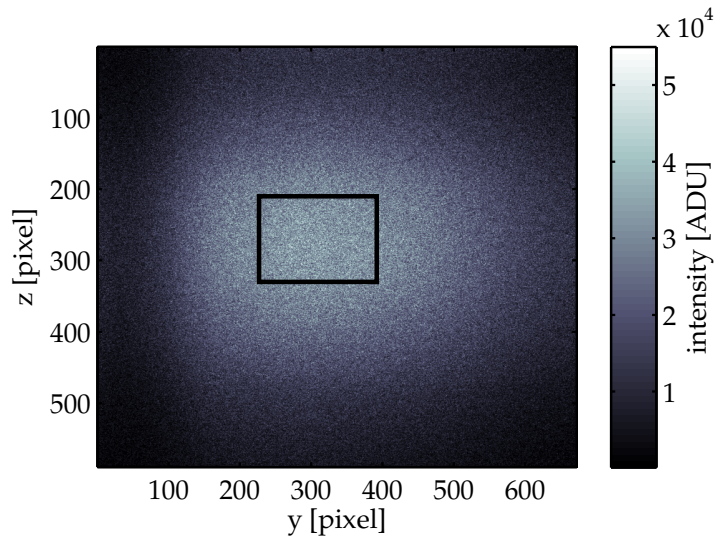
$$\begin{aligned}\sigma &= \sqrt{N} \\ \Rightarrow a \sigma &= \sqrt{a n_{\text{ADU}}} \\ \Rightarrow a &= \frac{n_{\text{ADU}}}{\sigma^2}.\end{aligned}$$

The conversion factor  $a$  from ADUs to photons can thus be obtained from the peak position  $n_{\text{ADU}}$  and the width of a histogram of a monochromatically and nearly homogeneously illuminated region on the CCD.

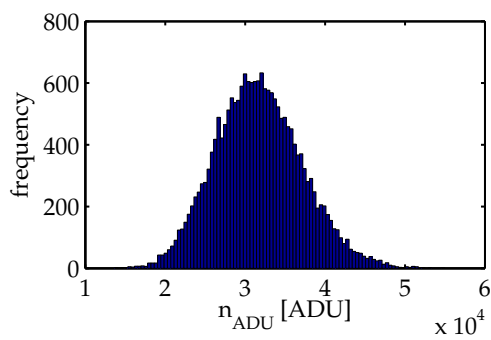
Both the charge generation in a pixel and the charge measurement in the ADC generate additional noise to the photon statistics. The noise of the charge generation depends on the Fano factor of the sensor material [75, 19]. It is usually smaller than the Poisson noise of photon counting [70]. The resulting broadening of the distribution is thus assumed to be negligible. An additional source of noise is the electronics, especially the ADC of the CCD chip. Its noise characteristic depends strongly on its quality. The manufacturer specifies for the LCX detector a CCD readout noise of two electrons (rms) and a system read noise of eight electrons (rms) for 1 MHz digitisation [73]. This is much smaller than the expected charge of approx. 3600 electrons generated in Si by a single photon of 13 keV. The signal offset due to dark current in the CCD pixel can be corrected by the subtraction of a dark image of same exposure time. Any additional source of noise in the system would cause a broadening of the distribution. This would give an increased width and thus a smaller count number, resulting in an underestimation of the applied dose to the sample.

For the flux estimation of the data set presented above, only the empty beam images were analysed. After the dark image subtraction the central part of the waveguide far field of approximately same intensity was chosen as ROI, see Figure 5.18a. For each of the 750 accumulations the histogram of the ADU values inside the ROI was calculated, see Figure 5.18b for an example. As the Poisson distribution can be approximated for large  $N$  with a Gaussian, the width of the distribution  $s$  and the position of the maximum  $n_{\text{ADU}}$  was obtained by a fit of  $f(n) = A_0 \exp\left(-\frac{(n-n_{\text{ADU}})^2}{2s^2}\right)$  to the histogram data. The retrieved data for  $s = a\sigma$  are plotted against  $n_{\text{ADU}}$  in Figure 5.18c. The data points follow a parabola, as expected. The average of  $a_i = \frac{n_{\text{ADU},i}}{s_i^2}$  for all acquisitions gives the conversion factor  $a = 1.06(2) \times 10^{-3}$  for ADUs to photons at 13 keV. The sum over all 750 acquisitions with sample inside the support area used for the iterative reconstruction gives a total of  $1.19 \times 10^{12}$  ADUs, which is equivalent to  $1.26 \times 10^9$  photons. Taking the exposure time of 4 s per acquisition and the detector efficiency of approx. 20 % at 13 keV [73] into account, the flux is  $2.1 \times 10^6$  photons/s. This corresponds to an applied dose of  $48.8 \times 10^3$  Gy over the total exposure time of 3000 s. The dose calculation followed the description in [25, 35].

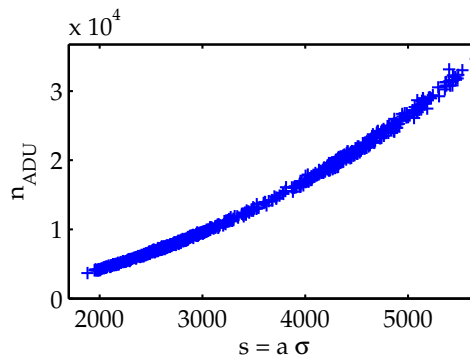
GINI-X was also used for measurements on free-standing black lipid membranes (BLM) [54]. BLMs as a model system for lipid bilayers are of general interest for the understanding of membranes of pro- and eukaryotic cells. The preparation as a free-standing membrane in an aqueous ambient is very similar to their native occurrence and allows investigations without any disturbing influence of a supporting substrate [6]. The local changes of the electron density in the aqueous ambient by the BLM are visible in phase contrast when illuminated with a coherent X-ray beam [7]. For the measurements at GINI-X the BLM was prepared in a wet chamber, as shown in Figure 5.19a. The membrane was then placed 8.5 mm behind a crossed waveguide with Mo[30 nm]/C[35 nm]/Mo[30 nm] layers in a Ge cladding [43, 44]. Figure 5.19b shows the resulting phase contrast image of a BLM, normalised with an empty image without the membrane in the beam. The images were taken with the MAXIPIX detector at a photon energy of 13.8 keV and distance of 5.29 m behind the waveguide. For the analysis the diffraction pattern of a model function of the membrane's electron density composed of a sum of Fresnel sine and cosine functions is calculated analytically in the detection plane. The parameters of the



(a) example of empty single acquisition



(b) histogram of ROI indicated in (a)

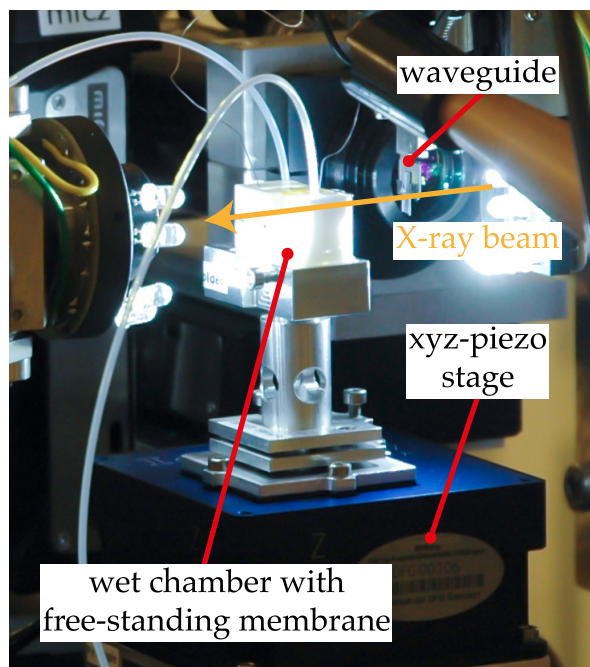


(c)

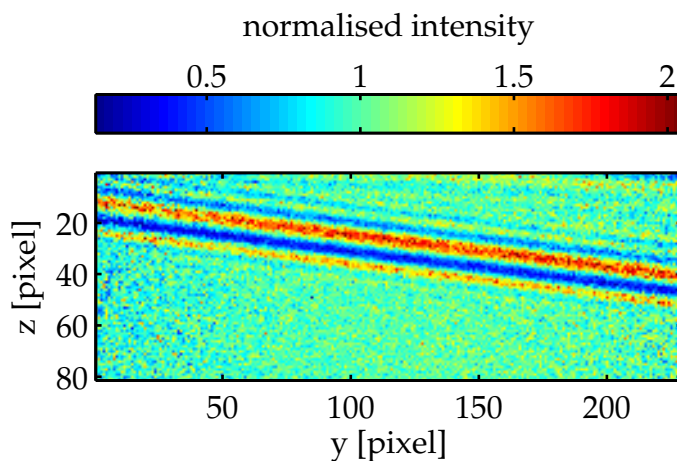
Figure 5.18: (a) Example of a single acquisition of the empty waveguide beam. The rectangle indicates the ROI used for further analysis. (b) Histogram of the pixel values inside the ROI. (c) Plot of the retrieved maximum positions against  $s = a\sigma$ , with  $a$  being the scaling factor.

model function are then determined by fitting the propagated model to the measured intensity. More details on model functions and obtained results can be found in [6, 7, 48, 54].





(a) experimental setup



(b) normalised diffraction pattern

Figure 5.19: (a) Photograph of the experimental setup with wet chamber and waveguide for measurements on free standing membranes. (b) Diffraction pattern of a black lipid membrane, taken with the MAXIPIX detector and normalised with an empty beam image. The analysis of the fringe pattern reveals information about the electron density of the membrane.



# Chapter 6

## Summary & Outlook

The aim of this work was the design, construction and commissioning of the dedicated instrument for waveguide-based X-ray imaging installed at the coherence beamline P10 of the PETRA III storage ring of the Deutsches Elektronen-Synchrotron (DESY) in Hamburg, Germany, accompanied by conceptual proof-of-principle experiments at the European Synchrotron Radiation Facility (ESRF) in Grenoble, France, for methodological and instrumental developments.

In Chapter 2 the wave-optical formalism of X-rays and the interaction with matter of X-rays were described. Based on this, the concept of waveguides for X-rays was described in conjunction with the various types of X-ray waveguides used within this thesis, e.g., planar waveguides, crossed planar waveguides, channel waveguides or Y-shaped double channel waveguides. Based on the successfully demonstrated reference-beam holography with Y-shaped double waveguides [22], the concept of reference-beam holography for interferometric applications was developed. Finally, the fundamentals of inline holography and iterative object reconstruction were summarised.

In Chapter 3 first conceptual experiments for waveguide-based reference beam interferometry and inline holography with iterative object reconstruction were presented. The experiments were carried out at the ESRF. The results of the waveguide interferometry experiment showed the capability of this method as a probe for the phase relation and stability of the focussed beam coupling into the Y-shaped waveguide. The intended application as a probe for local particle dynamics could not fully be demonstrated due to limitations in flux and detector read-out time. The principle of inline holography with iterative object reconstruction was

demonstrated successfully with channel waveguides as well as crossed planar waveguides on resolution test patterns and biological samples. Besides the well-known single-step holographic reconstruction of a Siemens star, the object reconstruction was extended to ptychography [27] on regularly spaced grids as well as spiralled grids of scan points. The first successful image of a biological cell (freeze-dried *Dictyostelium discoideum*) taken with waveguide-based inline holography was obtained by iterative object reconstruction with a modified hybrid-input-output algorithm [26]. For these conceptual experiments the standard setups at the beamlines of the ESRF were extended by required instrumentation for waveguide alignment and sample inspection.

The very useful insights gained from these experiments were incorporated in the design and construction of the dedicated waveguide-based imaging endstation GINI-X presented in Chapter 4. In this chapter at first the particular requirements of imaging experiments with waveguides were summarised and an overview of the general beamline layout and the instrument concept was given. The undulator installed at the P10 beamline provides a wide accessible energy range from 3.5 keV to 40 keV with a source size of  $36 \mu\text{m} \times 6 \mu\text{m}$  ( $h \times v$ ,  $1\sigma$ ) and a peak brilliance of  $5.3 \times 10^{20}$  photons/s/mrad<sup>2</sup>/mm<sup>2</sup>/0.1% bandwidth. The X-ray beam is focussed into the waveguides by a Kirkpatrick-Baez mirror system installed at a distance of 87.36 m behind the source. The elliptical curvature was polished into the substrates, giving a peak-valley deviation from the ideal ellipse of 13.9 nm for the vertical mirror and 4.8 nm for the horizontal mirror. This extremely high shape accuracy is essential to preserve the coherence of the focussed beam. The waveguide is aligned in the focus by a set of compact and precise positioners, allowing a minimal distance between the waveguide and the sample. Piezo-based positioners with encoders give the required accuracy for scanning microscopy. The sample positioning contains also an air-bearred rotational stage for tomographic measurements. Two optical on-axis microscopes support the prealignment of waveguides and samples and allow their inspection during X-ray measurements. The GINI-X instrument is built on an optical table with motorised jacks, that are used to realign it to the reference of a permanently installed autocollimator. This reduces the alignment efforts of the focussing mirrors after an exchange of GINI-X with the second diffractometer setup for X-ray photon correlation spectroscopy and coherent diffractive imaging, that is located in the same experimental hutch. Various detectors with different field of views and pixel sizes for different

imaging requirements are available, cf. Table 4.3. The user can run all relevant components of the endstation within the main control software *spec*. The hardware controllers are either directly integrated into *spec* or accessed via TANGO, that is mainly used for components shared between the two setups. The control schema is designed for an easy extension with new components.

The results of the commissioning and the first imaging experiments were reported in Chapter 5. The main part of the commissioning experiments was the initial alignment the Kirkpatrick-Baez mirror system and its performance characterisation. The procedure developed for the alignment of the focussing mirrors, starting from a prealignment with an X-ray camera and refinement with a knife edge to the final optimisation of the incidence angle with waveguide measurements of the focal spot, was described. The best measurements indicate a smallest focus size of  $203 \text{ nm} \times 221 \text{ nm}$  ( $h \times v$ , FWHM) with a photon flux of  $3.38 \times 10^{11}$  photons/s at 7.9 keV and a storage ring current of 100 mA. The flux calibration and measurement with a PIN diode was also explained. The mechanics of the mirror tilts fabricated by *WinlightX* suffered from a lack of repeatability due to hysteresis effects of the piezo-driven tilt motors, making the KB mirror alignment difficult. The proposed solution is currently under evaluation. The imaging capabilities of GINI-X were successfully demonstrated on test patterns and biological samples. Whilst this thesis was written up, the endstation was in regular use and the first tomogram of freeze-dried *Deinococcus radiodurans* cells was reconstructed from iteratively reconstructed inline holograms [5]. Free standing lipid membranes were also examined in wet chambers or various cell types were investigated under cryogenic conditions. The very high quality of the focussing mirrors allowed also the implementation of propagation-based imaging in the focussed beam without any coherence filtering by X-ray waveguides. Due to the high flexibility of GINI-X, an additional sample stage in front of the rear detector bench could be integrated easily, giving the ability of phase contrast tomography of large samples in the widened beam of the focussing mirrors. In a currently ongoing upgrade, energy dispersive detectors are integrated to gather fluorescence radiation of samples imaged in scanning mode, giving spatially resolved information of the element composition.

In summary, the built GINI-X endstation has proven its versatile capabilities for lensless X-ray imaging techniques for a wide range of samples under various conditions.



# Appendix A

## Characterisation of the Waveguide Positioning System

### A.1 Interferometer Setup

The waveguide positioning system was characterised for linearity and repeatability of the linear axes with a Michelson interferometer (SP-S 120, SIOS GmbH [16]). For selected axes the lateral stability behaviour was analysed as well. The testing setup is shown in Figure A.1. The attocube nanopositioners were mounted upside down on the vibration reduced extension arm, as used in the instrument. The interferometer consists of two independent sensor heads which allow the measurement of movements in two directions simultaneously. The frequency stabilised HeNe laser ( $\lambda = 632.8 \text{ nm}$ ) and the signal processing electronics are separated in an external enclosure to reduce the thermal drift inside the sensor heads. The laser beam is guided to the sensor heads by an optical fibre. The effective laser wavelength used for the displacement calculations is corrected for changes of the ambient temperature and the atmospheric pressure, as these parameters influence the index of refraction of air [17]. The electronics are able to read out the photo diodes used for intensity measurements of the interference signals with a sampling rate of 1 Hz – 1 MHz. These signals are then averaged by the electronics and sent via a USB connection to the measuring and recording software *InfasNTC* on a PC. The interferometer achieves a resolution of 0.3 nm [16]. Each sensor head is mounted on a rigid post where it can be adapted in height and tilt for a proper alignment of the laser beam.

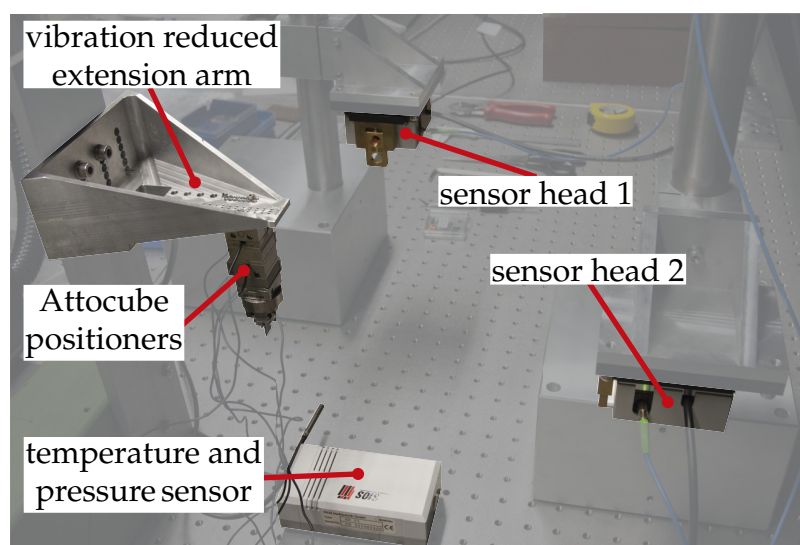


Figure A.1: Experimental setup for the characterisation of the attocube positioners with the SIOS SP-S 120 interferometer. The positioners were mounted upside down on the vibration reduced extension arm, as used in the GINI-X instrument at the P10 beamline. All positions were measured at the place of the waveguide at the lower end of the hanging positioner stack.



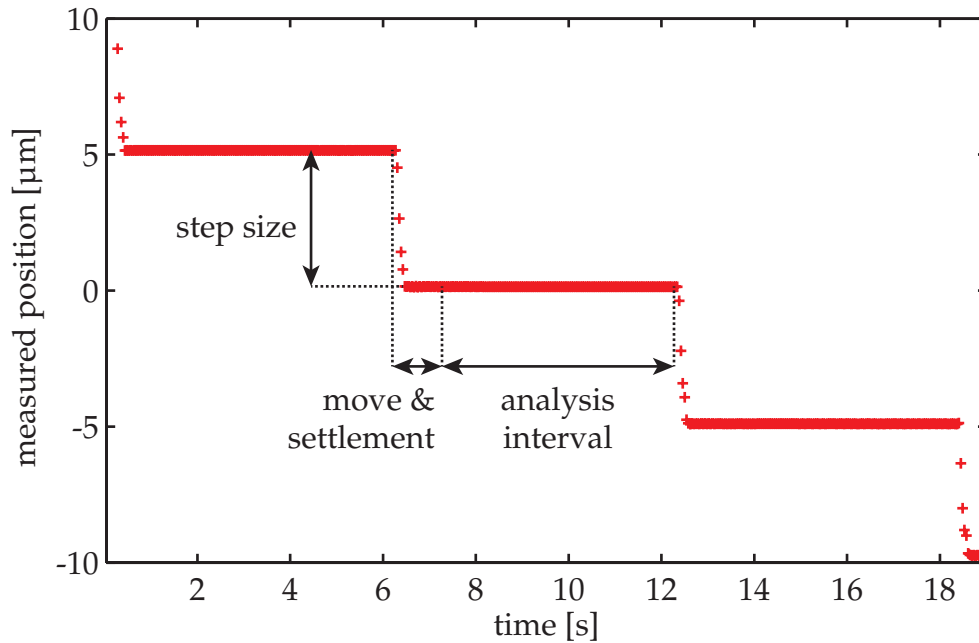


Figure A.2: Example of a linearity measurement: After the movement of a step size the positions in the analysis interval were averaged.

## A.2 Linearity Measurements

The linearity was determined by comparing the position reported by the encoder system with the position measured at the waveguide position at the lower end of the hanging positioners with the SIOS interferometer. As the working principle of the encoder system may show non-linear effects on both long and short length scales, the linearity was studied over the whole travel range and over a short interval of 100  $\mu\text{m}$  in the middle of the travel range.

The full travel range was discretised in points with a step size of 5  $\mu\text{m}$ . The nanopositioner was then moved to each of these points and remained there for 5 s. The measured interferometer positions were recorded during the whole movement sequence with a sampling rate of 6410.25 Hz, while 256 positions were averaged by the electronics, i.e. 25.04 positions per second were recorded. An example is shown in Figure A.2. The elapsed time between two points is approx. 6 s, reflecting the move and the settlement of the closed loop operation. For the analysis the recorded positions over a time interval of 5 s were averaged and compared with the targeted

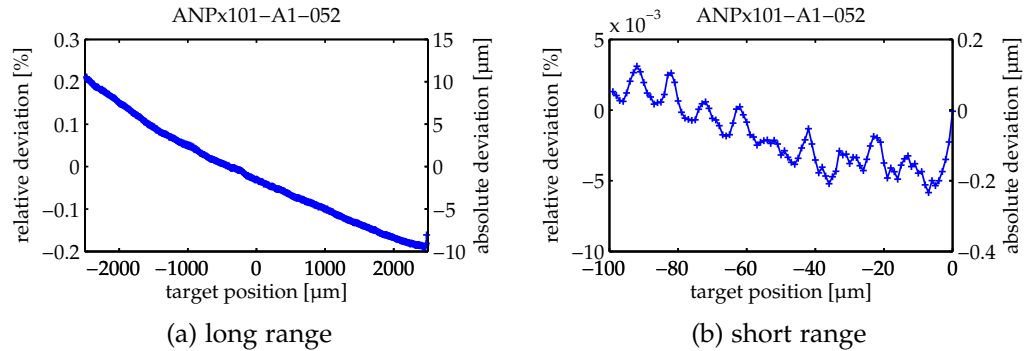


Figure A.3: Linearity of the ANPx101/NUM.

position.

For the short range linearity the interval of  $100\ \mu\text{m}$  was discretised in points with a step size of  $1\ \mu\text{m}$ . The data was recorded and averaged as for the full travel range described above.

The relative linearity is calculated from the absolute deviation between the target positions and the measured positions with reference to the full travel range of the positioner. Figures A.3 to A.6 show the results of the four linear attocube positioners<sup>1</sup>.

The short range linearity for the ANPx101/NUM (serial number ANPx101/NUM-A1-052), ANPz101/NUM, and ANPx51/NUM show oscillations, whose length scales correspond to the grating constant of  $20\ \mu\text{m}$  of the optical encoder system. The absolute deviations are in the order of  $200\ \text{nm}$ , which is close to the diameter of the KB focus.

The linearity of the second ANPx101/NUM (serial number ANPx101-/NUM-A1-055) in Figure A.4b shows oscillations with a shorter periodicity and larger absolute deviations up to  $500\ \text{nm}$ . For further investigations on this effect the SIOS setup was modified: One sensor head was directed to the waveguide position at the lower end of the positioner stack, as before. The second sensor head was directed directly to the moving part of the affected ANPx101/NUM positioner, see Figure A.7. The distance between the two reflecting points on the positioners was  $43\ \text{mm}$ . Very small tilts could be measured with this setup. The ANPx101/NUM was then moved over a range of  $6\ \mu\text{m}$  in  $100\ \text{nm}$  steps. The measured positions of both sensor heads are plotted in Figure A.8. Over the measurement

<sup>1</sup>This SIOS setup can only measure linear displacements. Thus, the rotational positioners could not be characterised.

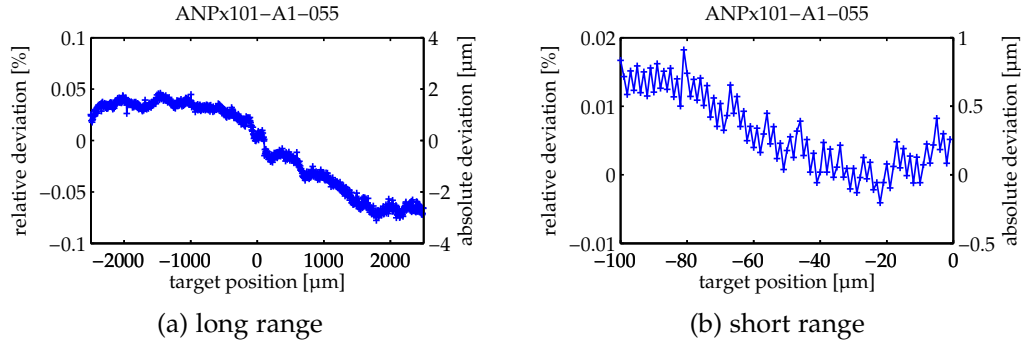


Figure A.4: Linearity of the ANPx101/NUM with abnormal behaviour.

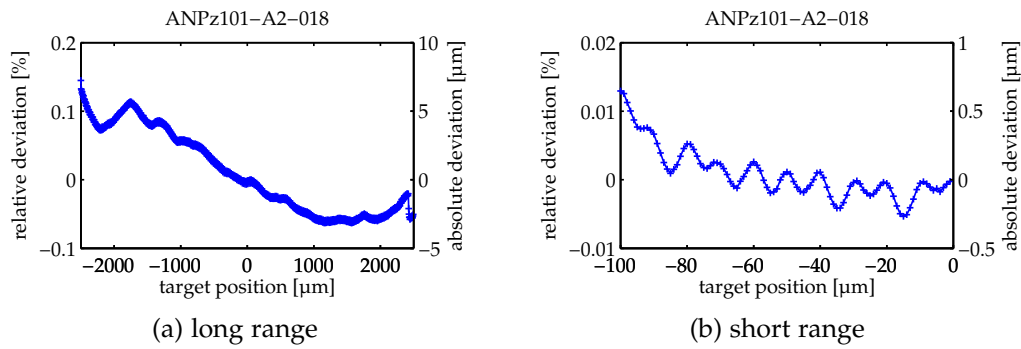


Figure A.5: Linearity of the ANPz101/NUM.

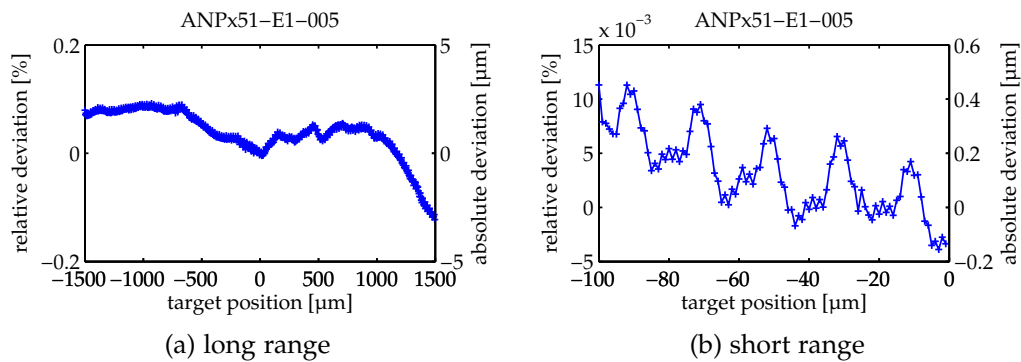


Figure A.6: Linearity of the ANPx51/NUM.

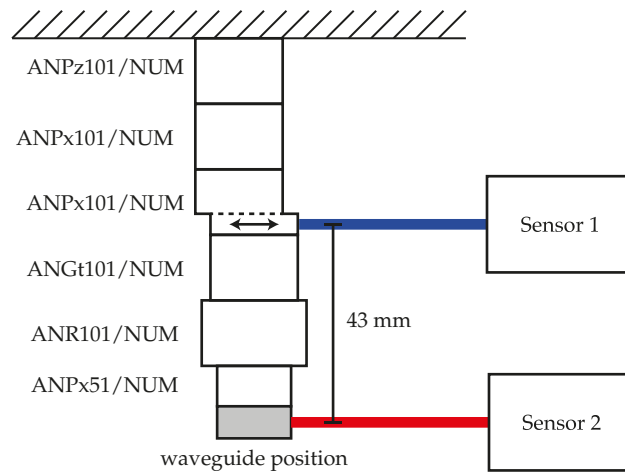


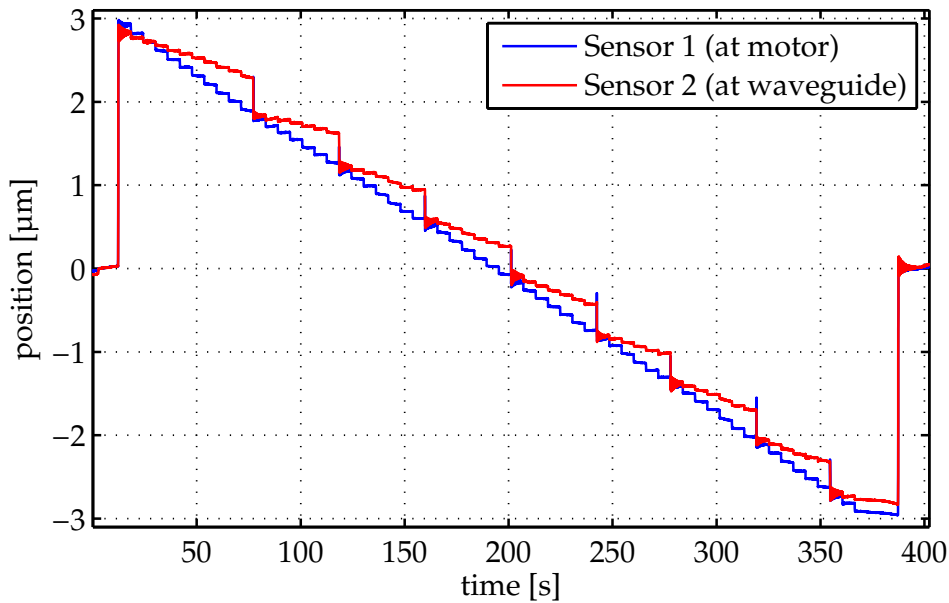
Figure A.7: Schematic setup for the in-depth linearity study of the ANPx101/NUM. The first sensor measured the positions directly at the positioner, the second sensor measured at the waveguide position. The distance in between was 43 mm.

range the discrepancy of the recorded positions between the two sensor heads increases slowly and drops down within one step. At these points the whole stack of positioners starts to vibrate, e.g. at time 160 s or 202 s in Figure A.8b. Between the “drop down” points the measured step size at the waveguide position is only the half of the de facto performed steps of the ANPx101/NUM. This indicates that the positioners below the ANPx101/NUM are tilted up to  $7 \mu\text{rad}$  while moving.

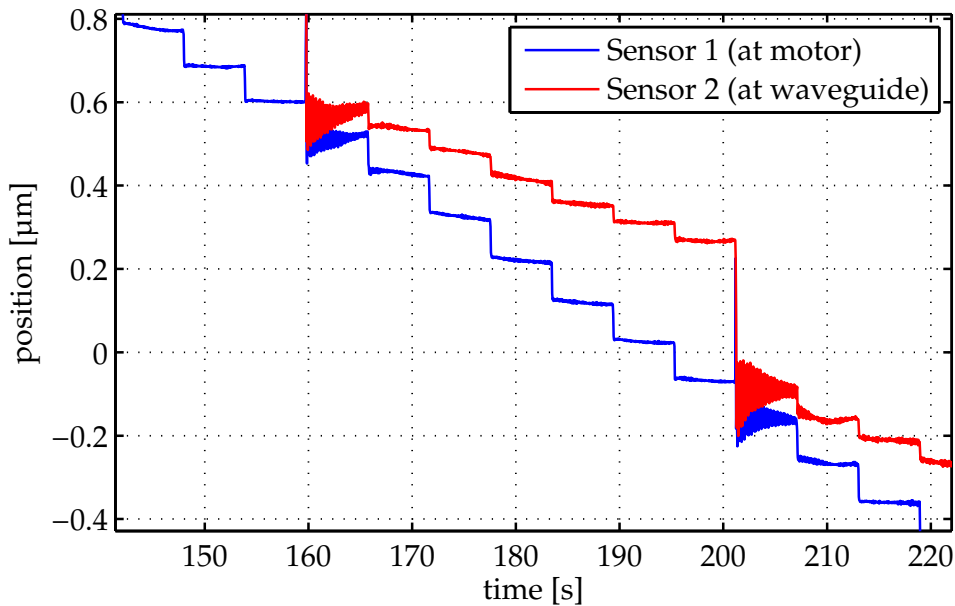
Although the effect of the reduced step size at the waveguide positions is within the vendor’s specifications, it has implications on the validity of measurements made with this ANPx101/NUM, which are mainly focus measurements during the first commissioning experiments.

The effect originates from the characteristics of the particular piezo used in the affected ANPx101/NUM and can be explained by the working principle of the attocube positioners. The slip stick piezos expand slowly and contract rapidly. While the piezos expand, they also bent slightly, which becomes visible due to the long distance between the positioner and the point of measurement at the waveguide position. When the piezos contract, the stack of positioners is accelerated, which excites its vibration.

The second positioner of this type and the ANPX51/NUM do not show this effect on the observed scale. After getting of this behaviour, the affected ANPx101/NUM was then used as translation in x direction,



(a) Full measured range



(b) Zoomed range of (a)

Figure A.8: Measured positions at the ANPx101/NUM positioner (blue) and at the position of the waveguide at the lower end of the positioner stack (red). The distance between these two interferometer beams was 43 mm.

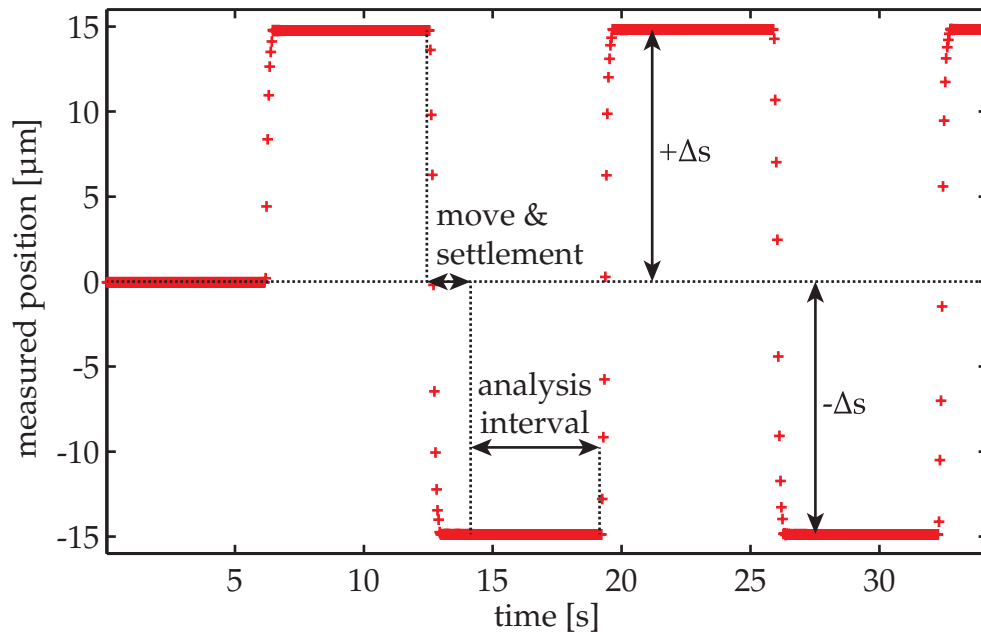


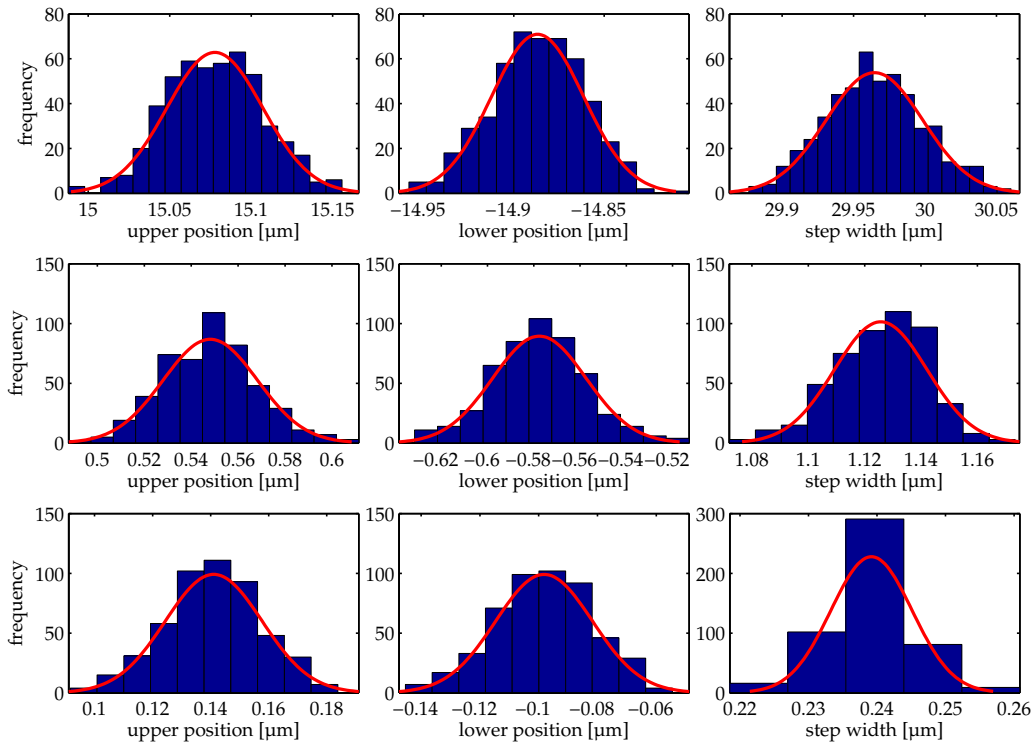
Figure A.9: Example of a repeatability measurement: After the movement of  $2 \Delta s$  the positions in the analysis interval were averaged.

as movements along the X-ray beam are not remarkably sensitive to this effect. Also the angular tilt in the range of  $\mu\text{rad}$  is negligible for waveguide alignment.

### A.3 Repeatability Measurements

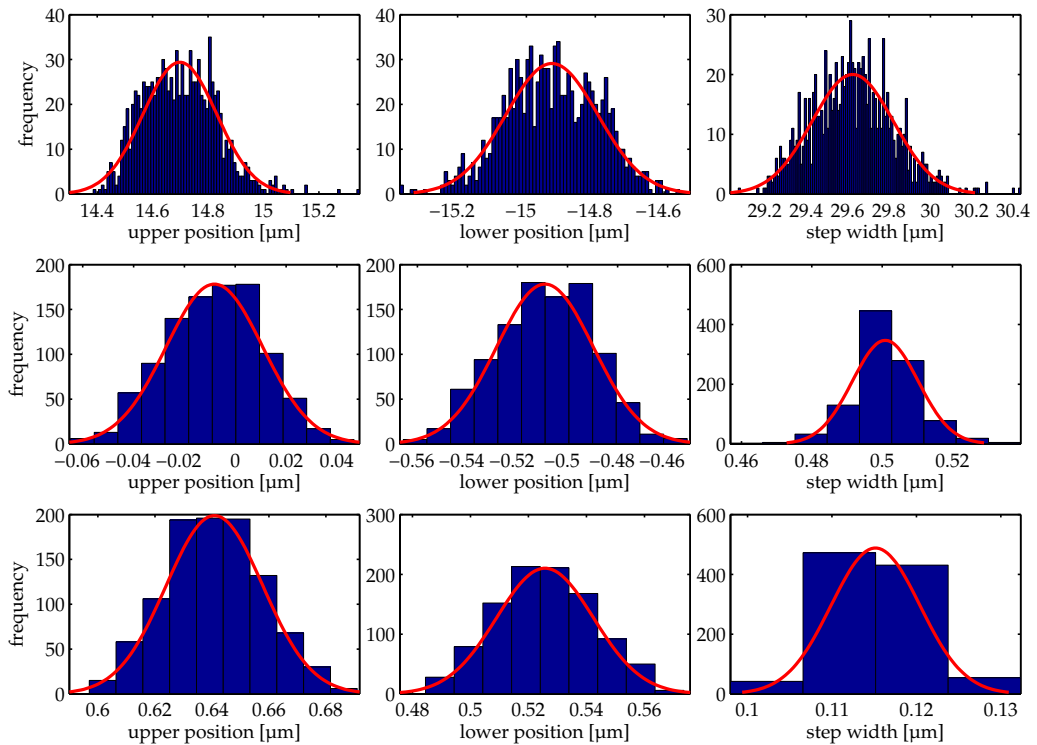
The repeatability was determined by the following procedure: Starting from an arbitrary zero position close to the middle of its travel range the positioner was moved to an upper point  $+\Delta s$ . Then it was moved to a lower position  $-\Delta s$ , so that the total step size is  $2 \Delta s$ . The sequence of moving from  $+\Delta s$  to  $-\Delta s$  and back to  $+\Delta s$  was repeated several hundred times. At each point it remained there for 5 s. The elapsed time between two movements is again approx. 6 s, caused by the move and the closed-loop settlement. An example is shown in Figure A.9. The measured interferometer positions were recorded with the same sampling and averaging parameters as the linearity measurements in Section A.2.

The positions in the analysis intervals were averaged and then plotted in separate histograms with a bin size of 10 nm for the upper and lower



$\Delta s$ [μm]	upper position [μm]	lower position [μm]	step size [μm]
15.0	$15.078 \pm 0.029$	$-14.887 \pm 0.026$	$29.965 \pm 0.034$
0.5	$0.548 \pm 0.020$	$-0.578 \pm 0.020$	$1.126 \pm 0.016$
0.1	$0.141 \pm 0.017$	$-0.098 \pm 0.016$	$0.239 \pm 0.006$

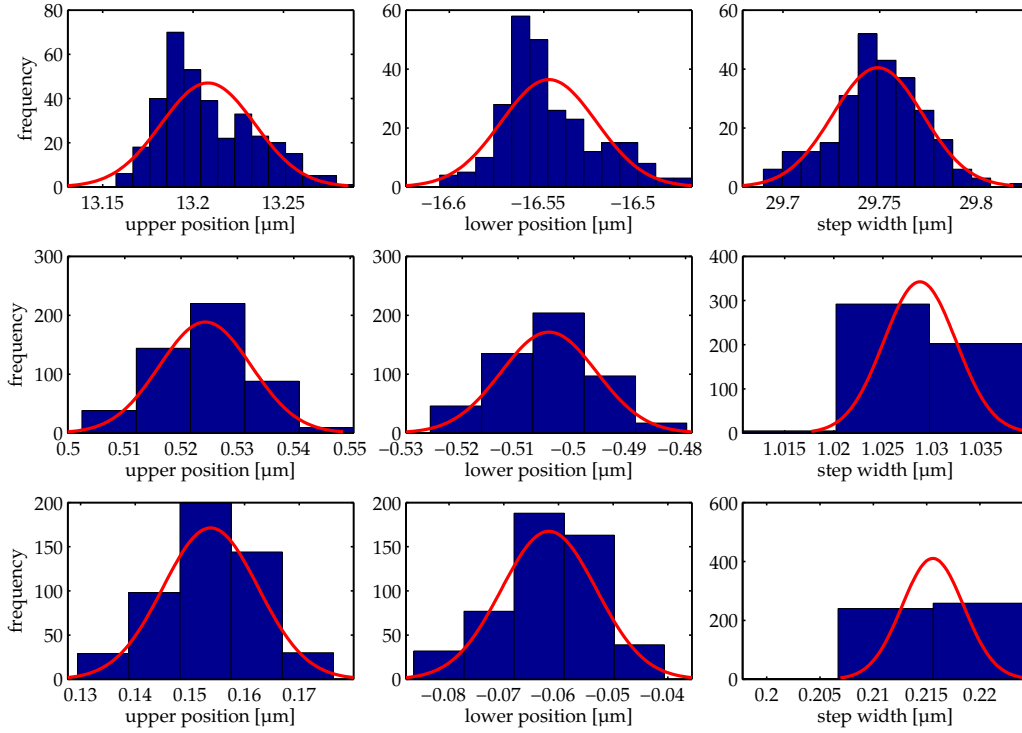
Figure A.10: ANP×101/NUM repeatability results.



$\Delta s$ [ $\mu\text{m}$ ]	upper position [ $\mu\text{m}$ ]	lower position [ $\mu\text{m}$ ]	step size [ $\mu\text{m}$ ]
15.0	$14.698 \pm 0.133$	$-14.926 \pm 0.134$	$29.625 \pm 0.197$
0.5	$-0.008 \pm 0.019$	$-0.509 \pm 0.019$	$0.501 \pm 0.009$
0.1	$0.641 \pm 0.017$	$0.526 \pm 0.017$	$0.115 \pm 0.005$

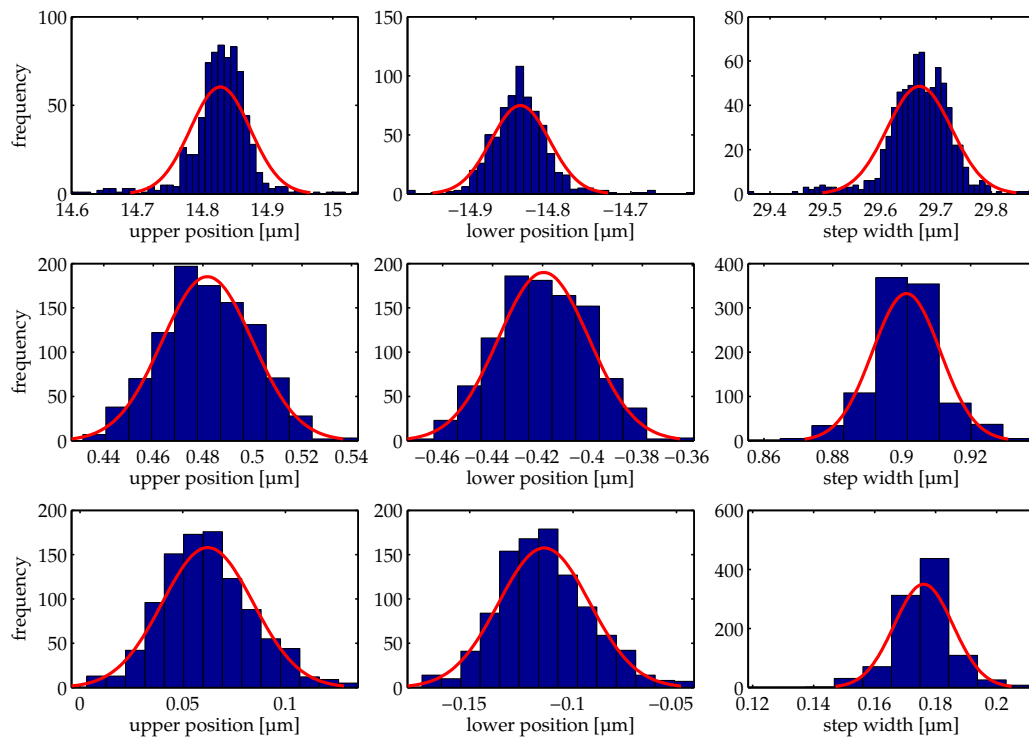
Figure A.11: ANPx101/NUM with abnormal behaviour repeatability results.





$\Delta s$ [ $\mu\text{m}$ ]	upper position [ $\mu\text{m}$ ]	lower position [ $\mu\text{m}$ ]	step size [ $\mu\text{m}$ ]
15.0	$13.208 \pm 0.025$	$-16.547 \pm 0.025$	$29.749 \pm 0.023$
0.5	$0.524 \pm 0.008$	$-0.504 \pm 0.008$	$1.029 \pm 0.004$
0.1	$0.154 \pm 0.009$	$-0.062 \pm 0.009$	$0.216 \pm 0.003$

Figure A.12: ANPz101/NUM repeatability results. The values of the upper and lower positions differ from  $\pm\Delta s$  due to thermal drift of the setup during the measurements.



$\Delta s$ [μm]	upper position [μm]	lower position [μm]	step size [μm]
15.0	$14.827 \pm 0.046$	$-14.843 \pm 0.038$	$29.670 \pm 0.058$
0.5	$0.482 \pm 0.018$	$-0.420 \pm 0.018$	$0.901 \pm 0.010$
0.1	$0.062 \pm 0.022$	$-0.114 \pm 0.022$	$0.176 \pm 0.010$

Figure A.13: ANPx51/NUM repeatability results.

points. The performed step sizes were calculated from the distances between subsequent upper and lower points, and again plotted as a histogram with a bin size of 10 nm. A Gaussian was then fitted to the histogram data. The histograms for  $\Delta s = 15 \mu\text{m}$ ,  $\Delta s = 1 \mu\text{m}$  and  $\Delta s = 0.1 \mu\text{m}$  with the corresponding step sizes are shown in Figure A.10 – A.13. In each figure a table with the fit results including the  $1\sigma$  error are given. The ANPx101/NUM with the abnormal linearity behaviour shows also abnormalities in the repeatability on short step sizes of  $\Delta s = 0.5 \mu\text{m}$  and  $\Delta s = 0.1 \mu\text{m}$ . Here the achieved step sizes are again just the half of the commanded step size.

For the other positioners the deviation between the achieved step size and the commanded step size is smaller than 250 nm for  $2 \Delta s = 30 \mu\text{m}$ . On shorter step sizes the deviation is typically smaller than 100 nm. As these measurements took several hours, the whole setup was affected by thermal drift. Thus, the measured upper and lower positions may differ from the values of  $\pm \Delta s$ .



# Appendix B

## Source Code

### B.1 spec Macros

Listing B.1: Macro for the use of the MAXIPIX detector with spec via a TANGO server.

```
#####  
# #  
# MAXIPIX INTERFACE FOR SPEC USING TANGO #  
# #  
5 #####  
# for use at P10@PETRA III #  
# S. Kalbfleisch, 2.04.2011 #  
#####  
  
10 # COMMANDS  
# =====  
# maxipixon enables Maxipix acquisitions during scans/ct  
# maxipixoff disables Maxipix acquisitions during scans/ct  
# mpxacq [exposuretime] [images [exposure_period]]  
15 # mpxacqset [exposuretime] [images [exposure_period]]  
# maxipixstatus show status information on Maxipix  
# maxipixhelp  
  
20 # GLOBALS  
# =====  
global MAXIPIXTANGO MAXIPIXCCDTANGO MAXIPIX_SCANON MAXIPIXFILE[]  
global MAXIPIX_EXPOSURETIME  
global MAXIPIX_FILENUM  
25 MAXIPIXTANGO = "mpxdesy1:20000/desy/limampx/mpx2"
```

```

MAXIPIXCCDTANGO = "mpxdesy1:20000/desy/limaccd/mpx2"

MAXIPIX_SCANON = 0
30
MAXIPIXFILE["PATH"] = tango_get(MAXIPIXCCDTANGO, "saving_directory")
MAXIPIXFILE["PREFIX"] = tango_get(MAXIPIXCCDTANGO, "saving_prefix")
MAXIPIXFILE["POSTFIX"] = tango_get(MAXIPIXCCDTANGO, "saving_suffix")
MAXIPIX_FILEENUM = tango_get(MAXIPIXCCDTANGO, "saving_next_number")
35
MAXIPIX_EXPOSURETIME = 0.1
MAXIPIX_IMAGES = 1

40 # =====
# USER COMMANDS
# =====

def maxipixhelp '{
45   printf("maxipixon \n\t Enables maxipix acquisitions during scans and
      ct\n")
   printf("maxipixoff \n\t Disables maxipix acquisitions during scans and
      ct\n")
   printf("mpxacq [exposuretime] [images] \n\t Acquires maxipix images(s)
      with global or given parameters\n")
   printf("mpxacqset [exposuretime] [images] \n\t Shows/sets the
      acquisition parameters for \'mpxacq\', scans and ct\n")
   printf("maxipixstatus \n\t Shows status information on maxipix\n")
50   printf("maxipixthreshold \n\t Shows/sets the threshold of the maxipix
      [in keV]\n")
   printf("maxipixhelp \n\t shows this help text\n")
   printf("\n")
}'
# activate maxipix exposure during counting
55 def maxipixon '{
   cdef("user_precount", "", "maxipix_precount", "delete")
   cdef("user_precount", "
      _maxipix_precount
      ", "maxipix_precount", 0x10)
60
   cdef("user_getcounts", "", "maxipix_getcounts", "delete")
   cdef("user_getcounts", "
      _maxipix_getcounts
      ", "maxipix_getcounts", 0x10)
65
   cdef("measure0", "", "maxipix_measure0", "delete")
   cdef("measure0", "
      MAXIPIX_SCANON=1

```

```
    ", "maxipix_measure0")
70
cdef ("measure2", "", "maxipix_measure2", "delete")
cdef ("measure2", "
    MAXIPIX_SCANON=0
    ", "maxipix_measure2")
75
cdef ("_cleanup2", "", "maxipix_cleanup", "delete")
cdef ("_cleanup2", "
    _maxipix_cleanup
    ", "maxipix_cleanup")
80
printf ("maxipix acquisition during scans with given settings for
    accumulations and frames enabled.\n")
}'

# deactivate maxipix exposure during counting
85 def maxipixoff '{
    cdef ("user_precount", "", "maxipix_precount", "delete")
    cdef ("user_getcounts", "", "maxipix_getcounts", "delete")
    cdef ("measure0", "", "maxipix_measure0", "delete")
    cdef ("measure2", "", "maxipix_measure2", "delete")
90 cdef ("_cleanup2", "", "maxipix_cleanup", "delete")
    printf ("maxipix acquisition during scans disabled.\n")
}'

def mpxacq '{
95 if ($# == 0){
    eprint "Usage: mpxacq exposuretime [images [exposure_period]]"
    printf ("Now using global settings of mpxacqset.\n")
    }

100 if ($# == 1){
    MAXIPIX_EXPOSURETIME = $1
    MAXIPIX_IMAGES = 1
    }

105 if ($# == 2){
    MAXIPIX_EXPOSURETIME = $1
    MAXIPIX_IMAGES = $2
    }

110 if ($# == 3){
    MAXIPIX_EXPOSURETIME = $1
    MAXIPIX_IMAGES = $2
    }
    # prepare maxipix Deviceserver for exposure
```

```

115 | _maxipix_prepare_acq(MAXIPIX_EXPOSURETIME, MAXIPIX_IMAGES)
      | # start exposure
      | _maxipix_start
      | # wait until exposure is finished
      | _maxipix_wait
120 | # increase FileNumber for filename of next exposure
      | _maxipix_filename_inc(MAXIPIX_IMAGES)
      | }'

def mpxacqset '{
125 |   if ($# == 0){
      |     eprint "Usage: mpxacqset exposuretime [images]"
      |     printf("Current settings:\n")
      |     printf("Exposure time:   %g s\n",MAXIPIX_EXPOSURETIME)
      |     printf("Number of Images: %g\n",MAXIPIX_IMAGES)
130 |   }

      |
      |   if ($# == 1){
      |     MAXIPIX_EXPOSURETIME = $1
      |   }
135 |
      |   if ($# == 2){
      |     MAXIPIX_EXPOSURETIME = $1
      |     MAXIPIX_IMAGES = $2
      |   }
140 |
      | # prepare maxipix deviceserver for exposure
      | _maxipix_prepare_acq(MAXIPIX_EXPOSURETIME, MAXIPIX_IMAGES)
      | printf("\n")
      | }'
145 |
def maxipixstatus '{
      | # get current settings of maxipix device server for
      | exposuretime = tango_get(MAXIPIXCCDTANGO, "acq_expo_time")
      | images = tango_get(MAXIPIXCCDTANGO, "acq_nb_frames")
150 | delaytime = tango_get(MAXIPIXCCDTANGO, "latency_time")
      | shuttermode = tango_get(MAXIPIXCCDTANGO, "shutter_mode")
      | triggermode = tango_get(MAXIPIXCCDTANGO, "acq_trigger_mode")
      | filedir = tango_get(MAXIPIXCCDTANGO, "saving_directory")
      | fileformat = tango_get(MAXIPIXCCDTANGO, "saving_format")
155 | fileprefix = tango_get(MAXIPIXCCDTANGO, "saving_prefix")
      | filestartnumber = tango_get(MAXIPIXCCDTANGO, "saving_next_number")
      | filepostfix = tango_get(MAXIPIXCCDTANGO, "saving_suffix")
      | threshold = tango_get(MAXIPIXCCDTANGO, "energy_threshold")

160 | printf("Current settings on maxipix device server %s
      | are:\n",MAXIPIXCCDTANGO)

```



```
printf("Exposure period (spec controlled): %f s\n", exposureperiod)
printf("Number of images (spec controlled): %g\n", images)
printf("Delay time: %f s\n", delaytime)
printf("Shutter mode: %s\n", shuttermode)
165 printf("Trigger mode: %s\n", triggermode)
printf("File path (spec controlled): %s on remote host\n", filedir)
printf("File format (spec controlled): %s\n", fileformat)
printf("File name (spec controlled): %s\n", fileprefix)
printf("File number (spec controlled): %s\n", filestartnumber)
170 printf("File postfix (spec controlled): %s\n", filepostfix)
printf("Threshold (spec controlled): %g keV\n", threshold)
}'

def maxipixthreshold '{
175 local threshold threshold_new
threshold = tango_get(MAXIPIXTANGO, "energy_threshold")
printf("Current maxipix threshold is set to %g keV.\n", threshold)

if ($# == 1) {
180 threshold_new = $1
if (threshold_new <= 0) {
printf("maxipix threshold must be a positive.\n")
exit
}
185 printf("Now setting maxipix threshold to %g keV.\n", threshold_new)
tango_put(MAXIPIXTANGO, "energy_threshold", threshold_new)
printf("The loading of the new threshold may take several seconds.
Please wait.\n")
exit
}
190 }'

# INTERNAL COMMANDS
# =====
def _maxipixnewfile '{
195 #create subfolder in haspp10wg data folder
#to be able to move files from buffer of mpxdesy 1 correctly
directoryname= getdatafilename()
cmd=sprintf("mkdir -p %s/maxipix/%s", DATA_BASEDIR, directoryname)
200 unix(cmd)

MAXIPIXFILE["PATH"] = "/buffer/mpxdesy12/p10/" # buffer with automatic
mv script does not support sub-folderds
MAXIPIXFILE["PREFIX"] = sprintf("%s_",getdatafilename())

MAXIPIXFILE["FORMAT"] = "edf"
205 MAXIPIXFILE["POSTFIX"] = "." MAXIPIXFILE["FORMAT"]
```

```

MAXIPIX_FILENUM = 0
# send new settings to maxipix device server
tango_put(MAXIPIXCCDTANGO, "saving_mode", "Auto_frame")
tango_put(MAXIPIXCCDTANGO, "saving_directory", MAXIPIXFILE["PATH"])
210 tango_put(MAXIPIXCCDTANGO, "saving_format", MAXIPIXFILE["FORMAT"])
tango_put(MAXIPIXCCDTANGO, "saving_prefix", MAXIPIXFILE["PREFIX"])
tango_put(MAXIPIXCCDTANGO, "saving_suffix", MAXIPIXFILE["POSTFIX"])
tango_put(MAXIPIXCCDTANGO, "saving_next_number", MAXIPIX_FILENUM)
# activate new settings
215 tango_io(MAXIPIXCCDTANGO, "prepareAcq")

text_bf
printf("\nWARNING: Existing maxipix files will be overwritten without
notice!\n\n")
text_non_bf
220 }'

# init maxipix with default values (beamtime hack).
def _maxipix_init '{
p "Use _maxipix_init with care, threshold is hardcoded (beamtime hack)"
225 # set energy threshold
tango_put(MAXIPIX_TANGO, "Energy_Threshold", 7.5) # half of x-ray
energy in keV

# general detector settings
230 tango_put(MAXIPIXCCDTANGO, "acq_mode", "single")
tango_put(MAXIPIXCCDTANGO, "acq_trigger_mode", "Internal_trigger")
tango_put(MAXIPIXCCDTANGO, "acc_time_mode", "Real")
tango_put(MAXIPIXCCDTANGO, "shutter_mode", "Auto_sequence")

# setup file information
235 tango_put(MAXIPIXCCDTANGO, "saving_mode", "Auto_frame")
tango_put(MAXIPIXCCDTANGO, "saving_directory",
"/buffer/mpxdesy12/p10/")
tango_put(MAXIPIXCCDTANGO, "saving_prefix", getdatafilename())
tango_put(MAXIPIXCCDTANGO, "saving_suffix", ".edf")
tango_put(MAXIPIXCCDTANGO, "saving_format", "edf")
240 tango_put(MAXIPIXCCDTANGO, "saving_overwrite_policy", "Overwrite")
tango_put(MAXIPIXCCDTANGO, "saving_frame_per_file", 1)

# activate new settings
245 tango_io(MAXIPIXCCDTANGO, "prepareAcq")

}'

def _maxipix_prepare_acq(ExposureTime, NbFrames) '{
if (ExposureTime > 0) {

```

```
250     tango_put(MAXIPIXCCDTANGO, "acq_expo_time", ExposureTime)
    } else {
        eprint "Exposuretime must be >0 s."
        exit
    }
255 if (NbFrames >= 1) {
    tango_put(MAXIPIXCCDTANGO, "acq_nb_frames", NbFrames)
} else {
    eprint "Number of Images must be >=1."
    exit
260 }
    # activate new settings
    tango_io(MAXIPIXCCDTANGO, "prepareAcq")
}'

265 def _maxipix_start '{
    # wait for motors and counters to finish
    waitall

    printf("Starting maxipix acquisition for %g s each of %g images.\n",
        MAXIPIX_EXPOSURETIME, MAXIPIX_IMAGES)
270 # prepare cleanup macro
    cdef("cleanup", "
        _maxipix_cleanup
        ", "maxipix_cleanup")
    # start the maxipix
275 # open P10 fast shutter
    _osh
    tango_io(MAXIPIXCCDTANGO, "startAcq")
}'

280 def _maxipix_wait '{
    while (!(tango_get(MAXIPIXCCDTANGO, "ready_for_next_acq"))) {}
    # delete cleanup macro
    cdef("cleanup", "", "maxipix_cleanup", "delete")
    # close P10 fast shutter
285 _csh
}'

def _maxipix_filename_inc(inc) '{
    if (inc < 1){
290     inc = 1
    }
    mpxfilename = sprintf("%s%s%04d%s", tango_get(MAXIPIXCCDTANGO,
        "saving_directory"), tango_get(MAXIPIXCCDTANGO, "saving_prefix"),
        tango_get(MAXIPIXCCDTANGO, "saving_next_number") - 1,
        tango_get(MAXIPIXCCDTANGO, "saving_suffix"))
```

```

mpxfilenameshort = sprintf("s%04d", tango_get(MAXIPIXCCDTANGO,
    "saving_prefix"), tango_get(MAXIPIXCCDTANGO, "saving_next_number")
    - 1, tango_get(MAXIPIXCCDTANGO, "saving_suffix"))

295  MAXIPIX_FILENUM = MAXIPIX_FILENUM + inc
if (MAXIPIX_FILENUM == tango_get(MAXIPIXCCDTANGO,
    "saving_next_number")) {
    printf("Last image of a sequence of %i saved on remote host as
        %s.\n", inc, mpxfilename)
} else {
    printf("Image sequence of %i interrupted. Last saved image is %s.\n",
        inc, mpxfilename)
300  MAXIPIX_FILENUM = tango_get(MAXIPIXCCDTANGO, "saving_next_number")
}

    # save dump of motor positions
    dumpfile = sprintf("s/dump/s.txt",DATA_DIR, mpxfilenameshort )
305  _save_motordump(dumpfile)
}'

def _maxipix_precount '{
    # set the maxipix exposure time to specs counting time
310  MAXIPIX_EXPOSURETIME = COUNT_TIME

    # prepare maxipix Deviceserver for exposure
    _maxipix_prepare_acq(MAXIPIX_EXPOSURETIME, MAXIPIX_IMAGES)
    # start exposure
315  _maxipix_start
}'

def _maxipix_getcounts '{
    # wait until exposure is finished
320  _maxipix_wait

    # increase FileNumber for filename of next exposure
    _maxipix_filenum_inc(MAXIPIX_IMAGES)
}'
325

def _maxipix_cleanup '{
    # stop running acquisitions
    tango_io(MAXIPIXCCDTANGO, "StopAcq")
    # close P10 fast shutter
330  p "cleanup close shutter"
    _csh
    # wait until running exposure can be finished
    sleep(MAXIPIX_EXPOSURETIME)
    # get FILENUM for next image (needed?)

```

```

335 MAXIPIX_FILENUM = tango_get(MAXIPIXCCDTANGO, "saving_next_number")
# get the server status
if (tango_get(MAXIPIXTANGO, "State") == 8 ) {
# reset the tango device server if in FAULT state
tango_io(MAXIPIXTANGO, "reset")
340 printf("maxipix device server reset...\n")
}

mpxfilename = sprintf("%s%s%04d%s", tango_get(MAXIPIXCCDTANGO,
"saving_directory"), tango_get(MAXIPIXCCDTANGO, "saving_prefix"),
tango_get(MAXIPIXCCDTANGO, "saving_next_number") - 1,
tango_get(MAXIPIXCCDTANGO, "saving_suffix"))
printf("Image acquisition interrupted. Last saved image is %s.\n",
mpxfilename)
345 MAXIPIX_FILENUM = tango_get(MAXIPIXCCDTANGO, "saving_next_number")
# remove maxipix_cleanup macro from the queue
cdef("cleanup","", "maxipix_cleanup", "delete")
}'

350 #####
# hook to spec #
#####

cdef("user_newfile","", "maxipix_newfile", "delete")
355 cdef("user_newfile" , "_maxipixnewfile\n", "maxipix_newfile", 0x10)

### load P10 Shutter macro
qdo /usr/local/lib/spec.d/macros/P10shutter.mac

```

Listing B.2: Macro for the use of the PILATUS detector with spec via a TANGO server.

```

#####
# #
# PILATUS INTERFACE FOR SPEC USING TANGO #
# #
5 #####
# for use at P10@PETRA III #
# S. Kalbfleisch, 1.04.2011 #
#####

10 # COMMANDS
# =====
# pilatuson enables Pilatus acquisitions during scans/ct
# pilatusoff disables Pilatus acquisitions during scans/ct
# piacq [exposuretime] [images [exposure_period]]
15 # piacqset [exposuretime] [images [exposure_period]]

```

```
# pilatusstatus      show status information on Pilatus
# pilatusinit       init the Pilatus with spec settings
# pilatushelp

20
# GLOBALS
# =====
global PILATUSTANGO PILATUS_SCANON PILATUSFILE[]
global PILATUS_EXPOSURETIME PILATUS_PERIOD PILATUS_EXPOSUREPERIOD
25 global PILATUS_FILENUM

PILATUSTANGO = "haspp10opt:10000/p10/pilatus/300k"

PILATUS_SCANON = 0
30
PILATUSFILE["PATH"] = tango_get(PILATUSTANGO, "FileDir")
PILATUSFILE["PREFIX"] = tango_get(PILATUSTANGO, "FilePrefix")
PILATUSFILE["POSTFIX"] = tango_get(PILATUSTANGO, "FilePostfix")
PILATUS_FILENUM = tango_get(PILATUSTANGO, "FileStartNum")
35
PILATUS_EXPOSURETIME = 0.1
PILATUS_IMAGES = 1
PILATUS_EXPOSUREPERIOD = 0.1 + 0.003

40
# =====
# USER COMMANDS
# =====

45 def pilatushelp '{
    printf("pilatuson \n\t Enables Pilatus acquisitions during scans and
        ct\n")
    printf("pilatusoff \n\t Disables Pilatus acquisitions during scans and
        ct\n")
    printf("piacq [exposuretime] [images [exposure_period]] \n\t Acquires
        Pilatus images(s) with global or given parameters\n")
    printf("piacqset [exposuretime] [images [exposure_period]] \n\t
        Shows/sets the acquisition parameters for \'piacq\', scans and
        ct\n")
50    printf("pilatusstatus \n\t Shows status information on Pilatus\n")
    printf("pilatusthreshold \n\t Shows/sets the threshold of the
        Pilatus\n")
    printf("pilatushelp \n\t shows this help text\n")
    printf("\n")
    }'
55
# activate Pilatus exposure during counting
```

```
def pilatuson '{
  cdef("user_precount","", "pilatus_precount", "delete")
  cdef("user_precount" , "
60   _pilatus_precount
   ", "pilatus_precount", 0x10)

  cdef("user_getcounts","", "pilatus_getcounts", "delete")
  cdef("user_getcounts" , "
65   _pilatus_getcounts
   ", "pilatus_getcounts", 0x10)

  cdef("measure0","", "pilatus_measure0", "delete")
  cdef("measure0" , "
70   PILATUS_SCANON=1
   ", "pilatus_measure0")

  cdef("measure2","", "pilatus_measure2", "delete")
  cdef("measure2", "
75   PILATUS_SCANON=0
   ", "pilatus_measure2")

  cdef("_cleanup2","", "pilatus_cleanup", "delete")
  cdef("_cleanup2", "
80   _pilatus_cleanup
   ", "pilatus_cleanup")

  printf("Pilatus acquisition during scans with given settings for
        accumulations and frames enabled.\n")
}'
85
# deactivate Pilatus exposure during counting
def pilatusoff '{
  cdef("user_precount","", "pilatus_precount", "delete")
  cdef("user_getcounts","", "pilatus_getcounts", "delete")
90  cdef("measure0","", "pilatus_measure0", "delete")
  cdef("measure2","", "pilatus_measure2", "delete")
  cdef("_cleanup2","", "pilatus_cleanup", "delete")
  printf("Pilatus acquisition during scans disabled.\n")
}'
95
def piacq '{
  if ($# == 0){
    eprint "Usage: piacq exposuretime [images [exposure_period]]"
    printf("Now using global settings of piacqset.\n")
100  }

  if ($# == 1){
```

```

    PILATUS_EXPOSURETIME = $1
    PILATUS_IMAGES = 1
105   PILATUS_EXPOSUREPERIOD = PILATUS_EXPOSURETIME + 0.003
    }

    if ($# == 2){
        PILATUS_EXPOSURETIME = $1
110     PILATUS_IMAGES = $2
        PILATUS_EXPOSUREPERIOD = PILATUS_EXPOSURETIME + 0.003
    }

    if ($# == 3){
115     PILATUS_EXPOSURETIME = $1
        PILATUS_IMAGES = $2
        PILATUS_EXPOSUREPERIOD = $3
    }

120   # prepare Pilatus Deviceserver for exposure
    _pilatus_prepare_acq(PILATUS_EXPOSURETIME, PILATUS_IMAGES,
        PILATUS_EXPOSUREPERIOD)
    # start exposure
    _pilatus_start
    # wait until exposure is finished
125   _pilatus_wait
    # increase FileNumber for filename of next exposure
    _pilatus_filenum_inc(PILATUS_IMAGES)
}'

130 def piacqset '{
    if ($# == 0){
        eprint "Usage: piacqset exposuretime [images [exposure_period]]"
        printf("Current settings:\n")
        printf("Exposure time:    %g s\n",PILATUS_EXPOSURETIME)
135     printf("Number of Images: %g\n",PILATUS_IMAGES)
        printf("Exposure periode: %g s\n",PILATUS_EXPOSUREPERIOD)
    }

    if ($# == 1){
140     PILATUS_EXPOSURETIME = $1
        PILATUS_EXPOSUREPERIOD = PILATUS_EXPOSURETIME + 0.003
    }

    if ($# == 2){
145     PILATUS_EXPOSURETIME = $1
        PILATUS_IMAGES = $2
        PILATUS_EXPOSUREPERIOD = PILATUS_EXPOSURETIME + 0.003
    }
}
```



```
150  if ($# == 3){
        PILATUS_EXPOSURETIME = $1
        PILATUS_IMAGES = $2
        PILATUS_EXPOSUREPERIOD = $3
    }
155  # prepare Pilatus Deviceserver for exposure
    _pilatus_prepare_acq(PILATUS_EXPOSURETIME, PILATUS_IMAGES,
        PILATUS_EXPOSUREPERIOD)
    printf("\n")
}'

160  def pilatusstatus '{
        # get current settings of pilatus device server for
        exposuretime = tango_get(PILATUSTANGO, "ExposureTime")
        exposureperiod = tango_get(PILATUSTANGO, "ExposurePeriod")
        images = tango_get(PILATUSTANGO, "NbFrames")
165  delaytime = tango_get(PILATUSTANGO, "DelayTime")
        shutterenable = tango_get(PILATUSTANGO, "ShutterEnable")
        triggermode = tango_get(PILATUSTANGO, "TriggerMode")
        filedir = tango_get(PILATUSTANGO, "FileDir")
        fileprefix = tango_get(PILATUSTANGO, "FilePrefix")
170  filestartnumber = tango_get(PILATUSTANGO, "FileStartNum")
        filepostfix = tango_get(PILATUSTANGO, "FilePostfix")
        threshold = tango_get(PILATUSTANGO, "Threshold")
        gain = tango_get(PILATUSTANGO, "Gain")

175  printf("Current settings on Pilatus device server %s
        are:\n",PILATUSTANGO)
        printf("Exposure time (spec controlled): %f s\n", exposuretime)
        printf("Exposure period (spec controlled): %f s\n", exposureperiod)
        printf("Number of images (spec controlled): %g\n", images)
        printf("Delay time: %f s\n", delaytime)
180  printf("Shutter Enable: %g\n", shutterenable)
        printf("Trigger mode: %g\n", triggermode)
        printf("File path (spec controlled): %s on remote host\n", filedir)
        printf("File name (spec controlled): %s\n", fileprefix)
        printf("File number (spec controlled): %s\n", filestartnumber)
185  printf("File postfix (spec controlled): %s\n", filepostfix)
        printf("Threshold (spec controlled): %g\n", threshold)
        printf("Gain mode: %i\n", gain)
    }'

190  def pilatusthreshold '{
        local threshold threshold_new
        threshold = tango_get(PILATUSTANGO, "Threshold")
        printf("Current Pilatus threshold is set to %g.\n", threshold)
```

```

195  if ($# == 1) {
        threshold_new = int($1)
        if (threshold_new <= 0) {
            printf("Pilatus threshold must be a positive integer.\n")
            exit
200  }
        printf("Now setting Pilatus threshold to %g.\n", threshold)
        tango_put(PILATUSTANGO, "Threshold", threshold_new)
        printf("The loading of the new threshold may take several seconds.
            Please wait.\n")
        exit
205  }
    }'

# INTERNAL COMMANDS
210  # =====
def _pilatusnewfile '{
    #create subfolder in haspp10wg data folder
    #to be able to move files from ramdisk of haspp10pilatus 1 correctly

215  directoryname= getdatafilename()
    cmd=sprintf("mkdir -p %s/pilatus/%s", DATA_BASEDIR, directoryname)
    unix(cmd)

    #PILATUSFILE["PATH"] = "/disk2/images/goettingen2/" getdatafilename()
220  PILATUSFILE["PATH"] = "/ramdisk/" # ramdisk with automatic mv script
        does not support sub-folders
    PILATUSFILE["PREFIX"] = getdatafilename()
    PILATUSFILE["POSTFIX"] = ".cbf"
    PILATUS_FILENUM = 0
    # send new settings to Pilatus device server
225  tango_put(PILATUSTANGO, "FileDir", PILATUSFILE["PATH"])
        tango_put(PILATUSTANGO, "FilePrefix", PILATUSFILE["PREFIX"])
        tango_put(PILATUSTANGO, "FilePostfix", PILATUSFILE["POSTFIX"])
        tango_put(PILATUSTANGO, "FileStartNum", PILATUS_FILENUM)

230  text_bf
        printf("WARNING: Existing Pilatus files will be overwritten without
            notice!\n\n")
        text_non_bf
    }'

235  def _pilatus_prepare_acq(ExposureTime, NbFrames, ExposurePeriod) '{
        if (ExposureTime > 0) {
            tango_put(PILATUSTANGO, "ExposureTime", ExposureTime)

```

```
    } else {
240     eprint "Exposuretime must be >0 s."
        exit
    }
    if (NbFrames >= 1) {
        tango_put(PILATUSTANGO, "NbFrames", NbFrames)
    } else {
245     eprint "Number of Images must be >=1."
        exit
    }
    if (ExposurePeriod >= ExposureTime + 0.003) {
        tango_put(PILATUSTANGO, "ExposurePeriod", ExposurePeriod)
250    } else {
        eprint "Exposureperiod - Exposuretime must be >=0.003 s."
        exit
    }
}'
255
def _pilatus_start '{
    # wait for motors and counters to finish
    waitall

260    printf("Starting Pilatus aquisition for %g s each of %g images.\n",
        PILATUS_EXPOSURETIME, PILATUS_IMAGES)
    # prepare cleanup macro
    cdef("cleanup", "
        _pilatus_cleanup
265        ", "pilatus_cleanup")
    # start the pilatus
    # open P10 fast shutter
    _osh
    tango_io(PILATUSTANGO, "StartStandardAcq")
    # close P10 fast shutter
270 }'

def _pilatus_wait '{
    while (tango_get(PILATUSTANGO, "State") !=0) {}
    # delete cleanup macro
275    cdef("cleanup", "", "pilatus_cleanup", "delete")
    _csh
}'

def _pilatus_filenum_inc(inc) '{
280    if (inc < 1){
        inc = 1
    }
    PILATUS_FILENUM = PILATUS_FILENUM + inc
```

```
285 tango_put(PILATUSTANGO, "FileStartNum", PILATUS_FILENUM)
printf("Last image of a sequence of %i saved on remote host as
      %s%s.\n", inc, tango_get(PILATUSTANGO, "FileDir"),
      tango_get(PILATUSTANGO, "LastImageTaken"))
# save dump of motor positions
if (is_macro("_save_motordump")){
    dumpfile = sprintf("s/dump/%s.txt",DATA_DIR,
        tango_get(PILATUSTANGO, "LastImageTaken") )
    _save_motordump(dumpfile)
290 }
}'

def _pilatus_precount '{
# set the pilatus exposure time to specs counting time
295 PILATUS_EXPOSURETIME = COUNT_TIME
if (PILATUS_EXPOSUREPERIOD < PILATUS_EXPOSURETIME + 0.003) {
    PILATUS_EXPOSUREPERIOD = PILATUS_EXPOSURETIME + 0.003
    eprint "PILATUS_EXPOSUREPERIOD reset."
}
300 # prepare Pilatus Deviceserver for exposure
_pilatus_prepare_acq(PILATUS_EXPOSURETIME, PILATUS_IMAGES,
    PILATUS_EXPOSUREPERIOD)
# start exposure
_pilatus_start
}'

305 def _pilatus_getcounts '{
# wait until exposure is finished
_pilatus_wait
# increase FileNumber for filename of next exposure
310 _pilatus_filename_inc(PILATUS_IMAGES)
}'

def _pilatus_cleanup '{
# stop running acquisitions
315 tango_io(PILATUSTANGO, "StopAcq")
# wait until running exposure can be finished, since this will not be
    stopped be "StopAcq"
sleep(PILATUS_EXPOSURETIME)
# get the server status
if (tango_get(PILATUSTANGO, "State") == 8 ) {
320 # reset the tango device server if in FAULT state
    tango_io(PILATUSTANGO, "Reset")
    printf("Pilatus device server reset.\n")
}
}

325 text_bf
```

```

printf("WARNING: All acquired images of this run will be overwritten
      by next acquisitions.\n")
text_non_bf
printf("Next filename will be %s%s%05g%s.\n",tango_get(PILATUSTANGO,
      "FileDir"), tango_get(PILATUSTANGO, "FilePrefix"),
      tango_get(PILATUSTANGO, "FileStartNum"), tango_get(PILATUSTANGO,
      "FilePostfix"))
# remove pilatus_cleanup macro from the queue
330 cdef("cleanup","", "pilatus_cleanup", "delete")
}'

#####
335 # hook to spec #
#####

cdef("user_newfile","", "pilatus_newfile", "delete")
cdef("user_newfile" , "_pilatusnewfile\n", "pilatus_newfile", 0x10)

```

Listing B.3: Macro for the use of P10 beamline motors connected to a VMO OMS58 motor controller with spec via a TANGO server.

```

#####
#
# USE P10 BEAMLINE MOTORS WITH VME OMS58 #
#           IN SPEC VIA TANGO           #
5 #
#####
# for use at P10@PETRA III             #
# S. Kalbfleisch, 8.03.2010           #
#####
10
def vme58tango_config(mne, key, unit, module, chan) '{
  local tangodevice
  if (key == "mot") {
    tangodevice = motor_par(mne, "tangodevice")
15   if (tangodevice == 0) {
      text_bf
      eprint motor_mne(mne) " not configured porperly. No tangodevice
        given as optional motor parameter."
      text_non_bf
      return(".error.")
20   }
  # display status information on config/reconfig
  deviceid = motor_par(mne, "device_id")
  motorname = motor_mne(mne)
  printf("Using %s tango motor on \"%s\" for \"%s\"\\n", deviceid,

```

```
    tangodevice, motorname)
25 }
}'

def vme58tango_cmd(mne, key, p1, p2) '{
    local tangodevice
30 # get the tango device of mne
    if (mne == "..") {
    }
    else {
35     tangodevice = motor_par(mne, "tangodevice")
    }

    if (key == "position") {
        # read the position from the tango server
40     pos = tango_get(tangodevice, "Position")
        return(pos)
    }

    if (key == "set_position") {
45     # may be set with tango_io(tangodevice,"Calibrate", newvalue)
        # before this is implemented, discuss with beamline staff (M. Sprung
            et al.)
        printf("Setting of dial value not supported on %s. Ignore next
            line.",motor_name(mne))
        return
    }

50     if (key == "start_one") {
        tango_put(tangodevice, "Position", p1)
        return
    }

55     if (key == "get_status") {
        local returnstate state
        returnstate = 0x00
        # check for moving
        state = tango_get(tangodevice, "State" )
60         if (state != 0) {
            returnstate = returnstate + 0x02
        }
        # check for clockwise limit
        state = tango_get(tangodevice, "CwLimit" )
65         if (state != 0) {
            returnstate = returnstate + 0x04
        }
        # check for counter clockwise limit
```

```
state = tango_get(tangodevice, "CcwLimit" )
70 if (state != 0) {
    returnstate = returnstate + 0x08
}
return(returnstate)
}
75
if (key == "abort_one") {
    # send abort command to tango
    tango_io(tangodevice, "StopMove", ans )
    return
80 }
}'

def vme58tango_par(mne, key, type, p1, p2) '{
85 if (mne == "..") {
}
else {
    tangodevice = motor_par(mne, "tangodevice")
}
if (key == "limits") {
90 local flag tangolowlimit tangohighlimit
    tangolowlimit = tango_get(tangodevice, "UnitLimitMin")
    tangohighlimit = tango_get(tangodevice, "UnitLimitMax")
    flag = 0
if (p1 < tangolowlimit) {
95 eprint "Setting of low limit below Tango\'s low limit not
        permitted."
eprint "Using Tango\'s low limit."
        flag++
        p1 = tangolowlimit
}
100 if (p2 > tangohighlimit) {
eprint "Setting of high limit above Tango\'s high limit not
        permitted."
eprint "Using Tango\'s high limit."
        flag++
        p2 = tangohighlimit
105 }
if (flag != 0) {
    set_lim(mne, dial(mne,p1), dial(mne, p2))
}
return
110 }

if (key == "offset") {
    return(0)
}
```

```

115 }
    }'

```

Listing B.4: Macro for the monitoring parameters of the storage ring via a TANGO server.

```

#####
#                                     #
# STORAGE RING PARAMETERS FOR SPEC USING TANGO #
#                                     #
5 #####
# for use at P10@PETRA III             #
# S. Kalbfleisch, 3.02.2010           #
#####

10 def petra_config(mne, key, p1, p2, unit) '{
    return
}'

def petra_cmd(mne, key, p1, p2, unit) '{
15   local tangoattribute tangocommand

   # get the tango device of mne
   if (mne == "..") {
       # suitable commands for all counters
20   }
   else {
       tangoattribute = counter_par(mne, "tangoattribute")
       if (counter_par(mne, "tangoattribute")){
           tangoattribute = counter_par(mne, "tangoattribute")
25   } else {
           eprint "No valid tango attribute given"
       }
   }

30   if (key == "start_one") {
       # commands sent before counting is startet, use for init of counting
       run
       return
   }

35   if (key == "get_status") {
       # get status of counter, relevant for polled counters
       return
   }

40   if (key == "counts") {

```



```

    # reading counts
    local current
    current = tango_get(tangodevice, tangoattribute)
    return(current)
45 }

    if (key == "halt_one") {
        # abort counting
        return
50 }
}'

```

Listing B.5: Macro for switching the P10 I/O registers via a TANGO server.

```

#####
# REMOTE CONTROL VIA TANGO OF P10 FASTSHUTTER #
# for use at P10@PETRA III #
# S. Kalbfleisch, 4.04.2011 #
#####
10 # tango device name for shutter register (SIS3610 I/O Register hardware
    in VME Crate)
    global P10SHUTTERREGISTER
    P10SHUTTERREGISTER = "haspp10opt:10000/p10/register/e2.out01"

    # opens P10 Piezo fast shutter (Cedrat) and enables remote control for
    CCDs via Logic board
15 def osh '{
    # enable shutter
    tango_put(P10SHUTTERREGISTER, "Value", 1)
    printf("P10 fast shutter is now open.\n\n")
}'

20 # closes P10 Piezo fast shutter (Cedrat) and disables remote control for
    CCDs via Logic board
    def csh '{
    # close and disable shutter
    tango_put(P10SHUTTERREGISTER, "Value", 0)
25 printf("P10 fast shutter is now closed.\n\n")
}'

    # opens P10 Piezo fast shutter (Cedrat) and enables remote control for
    CCDs via Logic board
    # silent verison without message

```

```
30 def _osh '{
    # enable shutter
    tango_put(P10SHUTTERREGISTER, "Value", 1)
}'

35 # closes P10 Piezo fast shutter (Cedrat) and disables remote control for
    CCDs via Logic board
# silent verison without message
def _csh '{
    # close and disable shutter
    tango_put(P10SHUTTERREGISTER, "Value", 0)
40 }'

### control of P10 Beamline shutter
# Tango device
45 global P10BSSHUTTER
P10BSSHUTTER = "haspp10opt.desy.de:10000/p10/shutter/1"

def shopen '{
    local state statestring
50 # get current state of shutter
    state = -1
    statestring = "undefined"
    state = tango_get(P10BSSHUTTER, "StateBS2")
    if (state == 0) {
55     statestring = "closed"
    } else if (state == 1) {
        statestring = "open"
    }
    printf("P10 BS2 was %s.\n",statestring)
60

    # open BS2
    tango_io(P10BSSHUTTER, "CloseOpen_BS_2",1)
    sleep(2)

65 # get new state of shutter
    state = -1
    state = tango_get(P10BSSHUTTER, "StateBS2")
    if (state == 0) {
        statestring = "closed"
70 } else if (state == 1) {
        statestring = "open"
    }
    printf("Now is %s.\n",statestring)
}'
75
```

```

def shclose '{
    local state statestring
    # get current state of shutter
80    state = -1
    statestring = "undefined"
    state = tango_get(P10BSSHUTTER, "StateBS2")
    if (state == 0) {
        statestring = "closed"
85    } else if (state == 1) {
        statestring = "open"
    }
    printf("P10 BS2 was %s.\n",statestring)

90    # close BS2
    tango_io(P10BSSHUTTER, "CloseOpen_BS_2", 0)
    sleep(2)

    # get new state of shutter
95    state = -1
    state = tango_get(P10BSSHUTTER, "StateBS2")
    if (state == 0) {
        statestring = "closed"
    } else if (state == 1) {
100    statestring = "open"
    }
    printf("Now is %s.\n",statestring)
}'

```

Listing B.6: Macro for control of SCX/LCX CCDs with a ST133 controller via spec2WinView server.

```

#####
# SPEC MACROS FOR SCX/LCX with PRINCETON ST133 CONTROLLER #
#           WITH Spec2WinView SERVER           #
# SK last change 18.05.2010           #
5 # based on ccd2.mac by chol           #
#####

# ===== user commands =====
# ccdacqset [time accumulation frames] # displays the current standard
#           settings or [set CCD standard settings for ccdacq, ccdfocuson]
10 # ccdacq [time accumulation frames] # exposure with CCD standard
#           settings or specified time number of accumulations and number of frames
# ccdfocuson [time accumulation frames] # start focus mode with CCD
#           standard settings or specified time number of accumulations and number
#           of frames

```

```
# ccdfocusoff # stop focus mode
# ccdon # exposure acquisition with current standard CCD settings during
      counting swichted on
# ccdoff # exposure acquisition during counting swichted off
15 # ccdroiset [top left right bottom [groupx groupy]] # displays the
      current RoI settings or [set hardware based RoI on CCD]
# ccdroiclear # clear RoI
# ccdconnect # (re)establish UDP connection to SpecToWinView server

#####
20 # global variables #
#####
global CCD_IP CCD_DIMX CCD_DIMY CCD_TIME CCD_ACCNUM CCD_FRAMES
      CCD_FILECOUNT
global CCD_CMD_START CCD_CMD_FOCUS CCD_CMD_EXIT CCD_CMD_CANCEL
      CCD_CMD_SETPARAM
global CCD_ROION CCD_ROI_TOP CCD_ROI_LEFT CCD_ROI_RIGHT CCD_ROI_BOTTOM
      CCD_ROI_GROUPX CCD_ROI_GROUPY
25 global CCD_SCANON
global CCD_REGISTER

# IP of spec2WinVIEW server, UDP port is fixed to 24637
CCD_IP="tingeltangelbob:24637"
30

# CCD chip dimensions
CCD_DIMX=1340
CCD_DIMY=1300

35 # CCD binning groups
CCD_ROI_GROUPX=1
CCD_ROI_GROUPY=1

# tango device name for CCD register (SIS3610 I/O Register hardware in
      VME Crate)
40 global CCD_REGISTER
CCD_REGISTER = "haspp10opt.desy.de:10000/p10/register/e2.out02"

# Suffix of CCD filename for spec's DATAFILE
CCD_FILE_SUFFIX="_ccd_"
45 # Flag for exposures during scans
CCD_SCANON=0
#FLAG for RoIC
CCD_ROION=0

50 # CCD default parameters
CCD_TIME=0.1
CCD_ACCNUM=1
```

```
CCD_FRAMES=1

55 # CCD command strings
CCD_CMD_START="start"
CCD_CMD_FOCUS="focus"
CCD_CMD_EXIT="exit"
CCD_CMD_CANCEL="cancel"
60 CCD_CMD_SETPARAM="setparam"
CCD_CMD_SETROI="setroi"
CCD_CMD_CLEARROI="clearroi"

65 #####
# user commands #
#####

def ccdconnect '{
70 sock_par(CCD_IP,"connect_udp")
}'

def ccdacqset '{
    if ($#!=3) {
75 p "normal usage:\n $0 time accumulations frames\n"
printf("current settings:\n")
printf("%s second(s)\n", CCD_TIME)
printf(" %s accumulation(s)\n", CCD_ACCNUM)
printf(" %s frame(s)\n", CCD_FRAMES)
80 _ccd_set
} else {
CCD_TIME=$1
CCD_ACCNUM=$2
CCD_FRAMES=$3
85 _ccd_set
}
}'

def ccdacq'{
90 if ($#!=3) {
p "normal usage:\n $0 time accumulations frames\n"
p "now using standard settings:"
printf(" %s second(s)\n", CCD_TIME)
printf(" %s accumulation(s)\n", CCD_ACCNUM)
95 printf(" %s frame(s)\n", CCD_FRAMES)
}
if ($#==3) {
CCD_TIME = $1
CCD_ACCNUM = $2
```

```
100     CCD_FRAMES = $3
      }
      _ccd_set
      _ccd_start
      printf("Exposure done:\n")
105     printf(" %s second(s)\n", CCD_TIME)
      printf(" %s accumulation(s)\n", CCD_ACCNUM)
      printf(" %s frame(s)\n", CCD_FRAMES)
      _ccd_save
    }'
110 # start exposures in focus mode
    def ccdfocuson' {
      if ($#!=3) {
        p "normal usage:\n $0 time accumulations frames\n"
115     p "now using standard settings:"
        printf(" %s second(s)\n", CCD_TIME)
        printf(" %s accumulation(s)\n", CCD_ACCNUM)
        printf(" %s frame(s)\n", CCD_FRAMES)
      }
120     if ($#==3) {
        CCD_TIME = $1
        CCD_ACC = $2
        CCD_FRA = $3
      }
125     _ccd_set
      _ccd_put(CCD_CMD_FOCUS)
      p "FOCUS mode enabled."
    }'
130 # stop exposures in focus mode
    def ccdfocusoff' {
      _ccd_put(CCD_CMD_CANCEL)
      p "FOCUS mode disabled."
    }'
135 # activate CCD exposure during counting
    def ccdon '{
      \cxon
140     cdef("user_precount", "", "ccd_precount", "delete")
      cdef("user_precount" , "
        _ccd_precount
        ", "ccd_precount", 0x10)
145     cdef("user_getcounts", "", "ccd_getcounts", "delete")
      cdef("user_getcounts" , "
```

```

    _ccd_getcounts
    ", "ccd_getcounts", 0x10)

150  cdef("measure0", "", "ccd_measure0", "delete")
    cdef("measure0" , "
        CCD_SCANON=1
        ", "ccd_measure0")

155  cdef("measure2", "", "ccd_measure2", "delete")
    cdef("measure2", "
        CCD_SCANON=0
        ", "ccd_measure2")

160  cdef("_cleanup2", "", "ccd_cleanup", "delete")
    cdef("_cleanup2", "
        _ccd_cleanup
        ", "ccd_cleanup")

165  p "CCD exposure during counting with given settings for accumulations
    and frames enabled."
}'

# deactivate CCD exposure during counting
def ccdoff '{
170  lcxoff
    cdef("user_precount", "", "ccd_precount", "delete")
    cdef("user_getcounts", "", "ccd_getcounts", "delete")
    cdef("measure0", "", "ccd_measure0", "delete")
    cdef("measure2", "", "ccd_measure2", "delete")
175  cdef("_cleanup2", "", "ccd_cleanup", "delete")
    p "CCD exposure during counting disabled."
}'

# setting roi
180  def ccdroiset'{
    if (($!=4) && ($!=6) && CCD_ROION>=1) {
        p "normal usage:\n $0 top left bottom right [groupx groupy]\n"
        p "current RoI settings:"
        printf("top: %s\n", CCD_ROI_TOP)
185  printf("left: %s\n", CCD_ROI_LEFT)
        printf("bottom: %s\n", CCD_ROI_BOTTOM)
        printf("right: %s\n", CCD_ROI_RIGHT)
        printf("groupx: %s\n", CCD_ROI_GROUPX)
        printf("groupy: %s\n", CCD_ROI_GROUPY)
190  _ccd_roiset
    }
    if (CCD_ROION==0 && $!=4 && $!=6){

```

```

    p "normal usage:\n $0 top left bottom right [groupx groupy]\n"
    p "RoI is disabled."}
195  if ($#==4) {
        CCD_ROI_TOP = $1
        CCD_ROI_LEFT = $2
        CCD_ROI_BOTTOM = $3
        CCD_ROI_RIGHT = $4
200  _ccd_roiset
    }
    if ($#==6){
        CCD_ROI_TOP = $1
        CCD_ROI_LEFT = $2
205  CCD_ROI_BOTTOM = $3
        CCD_ROI_RIGHT = $4
        CCD_ROI_GROUPX = $5
        CCD_ROI_GROUPY = $6
        _ccd_roiset
210  }
    }'

# clear roi
def ccdroiclear '{
215  _ccd_put(CCD_CMD_CLEARROI)
        _ccd_wait
        CCD_ROION=0
    }'

220 # ccdmenu
def ccdmenu '{
        printf("ccdmenu: ")
        text_bf; printf("I\'ll be back.\n");text_non_bf
    }'
225

# stop all running ccd commands
def ccdstop'{
        _ccd_put(CCD_CMD_CANCEL)
    }'
230

#####
# internal commands #
#####

235 # send string ccdcmd to CCD_IP
def _ccd_put(ccdcmd){
        # clear socket
        sock_par(CCD_IP,"flush")
        # send ccdstring to CCD_IP

```



## Source Code

---

```
240  sock_put(CCD_IP,ccdcmd)
    }'

    # read string from CCD_IP
    def _ccd_get(){
245      local ccdstring
          # read from CCD_IP
          ccdstring=sock_get(CCD_IP)
          return ccdstring
    }'

250  # wait until spec2WinView finishes current command
    def _ccd_wait '{
          while (_ccd_get()!="done")
    }'

255  # send CCD parameters to s2WV
    def _ccd_set '{
          _ccd_put(sprintf("set exp_time %s",CCD_TIME))
          _ccd_put(sprintf("set num_acc %s",CCD_ACCNUM))
260      _ccd_put(sprintf("set num_frames %s",CCD_FRAMES))
          _ccd_put(CCD_CMD_SETPARAM)
          _ccd_wait
          if (CCD_ROION>=1){
265              _ccd_roiset
          }
    }'

    # start CCD exposure
    def _ccd_start '{
270      cdef("_cleanup1","", "ccd_cleanup", "delete")
          cdef("_cleanup1",
              _ccd_cleanup
              , "ccd_cleanup")
          waitmove
275      _ccd_put(CCD_CMD_START)
          _ccd_wait
          cdef("_cleanup1","", "ccd_cleanup", "delete")
    }'

280  # abort all running processes
    def _ccd_cancel'{
          _ccd_put(CCD_CMD_CANCEL)
          text_bf
          p "CCD acquisition canceled"
285      p "CHECK YOUR CCD for overexposure and/or communication problems !!!!!"
          text_non_bf
```

```

}'
# reset CCD_FILECOUNT when new DATAFILE is started
290 def _ccd_filecount_reset'{
    # reset ccd-file counter
    CCD_FILECOUNT=0
    # use nae of DATAFILE as directoryname
    directoryname= getdatafilename()
295 # create directory for ccd files
    cmd=sprintf("mkdir -p %s/CCD/%s", DATA_BASEDIR, directoryname)
    unix(cmd)
}'

300 # save exposure
def _ccd_save '{
    datafilename = getdatafilename()
    # when scan is active
    if (CCD_SCANON>=1) {
305 # compose ccd filename
        ccd_savefile=sprintf("%s\\CCD\\%s\\%s_CCD_SCAN_%04i_%04i.spe",
            WIN_BASEDIR, datafilename, datafilename, SCAN_N, NPTS)
        dumpfile = sprintf("%s/dump/%s_CCD_SCAN_%04i_%04i.spe.txt",DATA_DIR,
            datafilename, SCAN_N, NPTS)
    }
    else { # scan is not active
310 CCD_FILECOUNT++
        # compose ccd filename
        ccd_savefile=sprintf("%s\\CCD\\%s\\%s_CCD_%04i.spe", WIN_BASEDIR,
            datafilename, datafilename, CCD_FILECOUNT)
        dumpfile = sprintf("%s/dump/%s_CCD_%04i.spe.txt",DATA_DIR,
            datafilename, CCD_FILECOUNT)
    }
315 # send save command
    _ccd_put(sprintf("save %s",ccd_savefile))
    _ccd_wait
    p
    # save dump of motor positions
320 if (is_macro("_save_motordump")){
        _save_motordump(dumpfile)
    }
    printf("Acquisition saved as: %s\n",ccd_savefile)
}'

325 def _ccd_precount '{
    CCD_TIME=COUNT_TIME
    _ccd_set
    _ccd_start

```

```

330 }'

def _ccd_getcounts '{
    _ccd_save
}'

335 def _ccd_cleanup '
    _ccd_cancel
    CCD_SCANON=0
    ,

340 # send ROI settings to S2WV
def _ccd_roiset '{
    _ccd_roicheck
    roicmd=sprintf("%s %s %s %s %s %s %s", CCD_CMD_SETROI, CCD_ROI_TOP,
        CCD_ROI_LEFT, CCD_ROI_BOTTOM, CCD_ROI_RIGHT, CCD_ROI_GROUPX,
        CCD_ROI_GROUPY)
345 _ccd_put(roicmd)
    _ccd_wait
    CCD_ROION=1
}'

350 # check ROI settings
def _ccd_roicheck '{
    if (CCD_ROI_TOP<1 || CCD_ROI_LEFT<1 || CCD_ROI_RIGHT<1 ||
        CCD_ROI_BOTTOM<1){
        p "RoI borders must be greater equal 1!"
        exit}
355 if (CCD_ROI_GROUPX<1 || CCD_ROI_GROUPY<1){
        p "RoI groups must be greater equal 0!"
        exit}
    if ((CCD_ROI_TOP >= CCD_ROI_BOTTOM) || (CCD_ROI_LEFT >=
        CCD_ROI_RIGHT)){
        p "RoI dimensions incorrect!"
360 exit}
    if ((fabs(CCD_ROI_TOP - CCD_ROI_BOTTOM)>CCD_DIMY) ||
        (fabs(CCD_ROI_LEFT - CCD_ROI_RIGHT)>CCD_DIMX)){
        printf("RoI size must be below %s and %s pixels\n",CCD_DIMX,CCD_DIMY)
        exit}
}'

365 #####
# hook to spec #
#####

cdef("user_newfile","","ccd_filecount","delete")
370 cdef("user_newfile" , "_ccd_filecount_reset\n","ccd_filecount",0x10)

```

```

#####
# P10 fastshutter control #
#####
375
def lcxon '{
  # enable remote control of P10 fast shutter for ST133 controller of CCD
  tango_put(CCD_REGISTER, "Value", 1)
  # enable P10 fast shutter for CCD control
380  osh
}'

def lcxoff '{
  # close and disable P10 fast shutter for CCD control
385  csh
  # disable remote control of P10 fast shutter for ST133 controller of
  CCD
  tango_put(CCD_REGISTER, "Value", 0)
}'

```

Listing B.7: Macro for usage of the APD with ACE pulse processor.

```

#####
#
# MACROS FOR ACE PULSE PROCESSOR OF APD #
#
5 #####
# S. Kalbfleisch, 9.03.2010 #
#####

#####
10 # Commands for Macro Motors #
#####

def APD_motor_config(mne, key, unit, module, chan) '{
  local serialport
15  global _APD_INIT

  if (mne == "..") {
    # reset init status of current unit
    _APD_INIT[unit] = 0
20  }

  if (key == "mot") {
    serialport = motor_par(mne, "serialport")
    type = _apd_gettype(motor_par(mne, "type"))
25  if (serialport < 0) {
    text_bf

```

```
    eprint motor_mne(mne) " not configured porperly. No serialport
        given as optional motor parameter."
    text_non_bf
    return(".error.")
30 }
    if (type < 0) {
        text_bf
        eprint motor_mne(mne) " not configured porperly. No valid type
            given as optional motor parameter."
        text_non_bf
35     return(".error.")
    }
    # init device
    if (_APD_INIT[unit] == 0) {
        _apd_sendcmd(serialport,"CSRC SCA")
40     _apd_sendcmd(serialport,"GLVL TTL")
        _apd_sendcmd(serialport,"OUT 30")
        _apd_sendcmd(serialport,"SOUT GATE POS")
        _apd_sendcmd(serialport,"LCD WIN LOG")
        # set init status of current unit to done
45     _APD_INIT[unit] = 1
    }
    # display status information on config/reconfig
    deviceid = motor_par(mne,"device_id")
    motorname = motor_mne(mne)
50     serialdevice = ser_par(serialport, "device_id")
    printf("Using %s on %s as \"%s\"\n", deviceid,serialdevice,motorname)
}
}'
55
def APD_motor_cmd(mne, key, p1, p2, unit) '{
    if (key == "set_position") {
        _apd_set_position(mne, key, p1, p2, unit)
        return
60     }
    if (key == "start_one") {
        _apd_set_position(mne, key, p1, p2, unit)
        return
    }
65     if (key == "position") {
        local pos
        pos = _apd_position(mne, key, p1, p2, unit)
        return(pos)
    }
70     if (key == "get_status") {
        local status
```

```

    status = _apd_get_status(mne, key, p1, p2, unit)
    return(status)
}
75 }'

def APD_motor_par(mne, key, type, p1, p2) '{
    if (mode == "get") {
        return(-1)
80    }

    if (mode == "set") {
        return(-1)
    }
85 }'

#####
# internal functions #
#####
90

# returns type of APD macro motor
def _apd_gettype(type) '{
    # motor for SCA lower threshold
    if (type == "LTH"){
95     type = "LTH"
        return(type)
    }
    # motor for SCA window
    if (type == "WIN"){
100    type = "WIN"
        return(type)
    }
    # motor for High Voltage
    if (type == "HVVOLT"){
105    type = "HVVOLT"
        return(type)
    }
    return(0)
}'
110

# sets the position
def _apd_set_position(mne, key, p1, p2, unit) '{
    # get the serialport
    local serialport
115    serialport = motor_par(mne, "serialport")
    # get the motor type
    local type
    type = _apd_gettype(motor_par(mne, "type"))
```

```
120  if (type == "HVOLT") {
    _apd_sendcmd(serialport, sprintf("HVOLT %.3f ON",p1))
    return
  }
  if (type == "LTH") {
125  _apd_sendcmd(serialport, sprintf("SCA %.3f",p1))
    return
  }
  if (type == "WIN") {
    # get the current value of lth first
130  local current
    current = _apd_readcmd(serialport,"?SCA")
    split(current, values)
    _apd_sendcmd(serialport, sprintf("SCA %.3f %.3f",values[1], p1))
  }
135 }'

# reads the position
def _apd_position(mne, key, p1, p2, unit) '{
  local ans values
140  # get the serialport
  local serialport
  serialport = motor_par(mne, "serialport")
  # get the motor type
  local type
145  type = _apd_gettype(motor_par(mne, "type"))

  if (type == "HVOLT") {
    local ans values
    ans = _apd_readcmd(serialport,"?HVOLT")
150  split(ans,values)
    return(values[0])
  }
  if (type == "LTH") {
    _apd_sendcmd(serialport,"?SCA")
155  ans = _apd_readcmd(serialport,"?SCA")
    split(ans, values)
    return(values[1])
  }
  if (type == "WIN") {
160  _apd_sendcmd(serialport,"?SCA")
    ans = _apd_readcmd(serialport,"?SCA")
    split(ans, values)
    return(values[2])
  }
165 }'
```

```

# return the status
def _apd_get_status(mne, key, p1, p2, unit) '{
    local ans values
170 # get the serialport
    local serialport
    serialport = motor_par(mne, "serialport")
    # get the motor type
    local type
175 type = _apd_gettype(motor_par(mne, "type"))
    # read the alarm status (TODO)
    # TODO
    return(0x00)
}'

180 # send cmd over spec's serial line no. 'serial'
def _apd_sendcmd(serial, cmd) '{
    ser_par(serial, "flush")
    ser_put(serial, sprintf("%s\r",cmd))
185 }'

# send cmd over spec's serial line no. 'serial' and returns its answer
def _apd_readcmd(serial, cmd) '{
    ser_par(serial, "flush")
190 ser_put(serial, sprintf("%s\r",cmd))
    ans = ser_get(serial)
    return(ans)
}'

195 #####
# Commands for Macro Counters #
#####

def APD_counter_config(mne, key, unit, p0, chan) '{
200 local serialport type
    if (key == "cnt") {
        serialport = counter_par(mne, "serialport")
        type = counter_par(mne, "type")
        if (serialport < 0) {
205 text_bf
            eprint cnt_mne(mne) " not configured porperly. No serialport given
                as optional counter parameter."
            text_non_bf
            return(".error.")
        }
210 if ((type == "PULSES") || (type == "TEMPERATURE") || (type ==
        "HVCUR") || (type == "HVMON")) {

```



```
    }
    else {
        text_bf
        eprint cnt_mne(mne) " not configured porperly. No valid counter
215         type given as optional counter parameter."
        return(".error.")
        text_non_bf
    }
}
}'
220
def APD_counter_cmd(mne, key, p1, p2, unit) '{
    local serialport type
    # get counter type and serial port for mne, if mne != ..
    if (mne != "..") {
225         serialport = counter_par(mne, "serialport")
         type = counter_par(mne, "type")
         # check for proper config of serialport and type
    }

230     # arm the counter
    if (key == "start_one") {
        if (type = "PULSES") {
            _apd_sendcmd(serialport,"ECT")
        }
235         return
    }

    # get status
    if (key == "get_status") {
240         return(0)
    }

    # read counts
    if (key == "counts") {
245         # photon counts
        if (type == "PULSES") {
            local countdata countrawdata
            rawcountdata = _apd_readcmd(serialport,"?CT DATA")
            split(rawcountdata, countdata)
250             countresult = countdata[4]
        }

        # temperature of APD head
        if (type == "TEMPERATURE") {
255             countresult = _apd_readcmd(serialport,"?HTEMP")
        }
    }
}
```

```

# High voltage current
if (type == "HVCUR") {
260     countresult = _apd_readcmd(serialport,"?HCURR")
}

# measuerd high voltage
if (type == "HVMON") {
265     countresult = _apd_readcmd(serialport,"?HVMON")
}

return(countresult)
}
270

# stop the counter
if (key == "halt_one") {
    if (type == "PULSES") {
        # stop counter
275         _apd_sendcmd(serialport,"STCT")
    }
}
}'

280

def APD_counter_par(mne, key, mode, p1, p2) '{
    if (mode == "get") {
285         return(-1)
    }
    if (mode == "set") {
        return(-1)
    }
}'

```

Listing B.8: Macro for usage of the Cyberstar X2000 pulse processor.

```

#####
#                                     #
# MACROS FOR CYBERSTAR X2000 PULSE PROCESSOR #
#                                     #
5  #####
# S. Kalbfleisch, 13.03.2011          #
#####

#####
10 # Commands for Macro Motors #
#####

```

```
def cyberstar_config(mne, key, unit, module, chan) '{
15  local serialport
    if (key == "mot") {
        serialport = motor_par(mne, "serialport")
        movecmd = _cyberstar_getmovecmd(motor_par(mne, "type"))
        if (serialport < 0) {
            text_bf
20         eprint motor_mne(mne) " not configured porperly. No serialport
            given as optional motor parameter."
            text_non_bf
            return(".error.")
        }
        if (movecmd == -1) {
25         text_bf
            eprint motor_mne(mne) " not configured porperly. No valid type
            given as optional motor parameter."
            text_non_bf
            return(".error.")
        }
30     # display status information on config/reconfig
        deviceid = motor_par(mne,"device_id")
        motorname = motor_mne(mne)
        serialdevice = ser_par(serialport, "device_id")
        printf("Using %s on %s as \"%s\"\n", deviceid,serialdevice,motorname)
35     }
}

def cyberstar_cmd(mne, key, p1, p2) '{
40  local serialport
    # get the serial port of mne
    if (mne == "..") {
    }
    else {
45         serialport = motor_par(mne, "serialport")
        movecmd = _cyberstar_getmovecmd(motor_par(mne, "type"))
    }

    if (key == "position") {
50     # read the positon from the CBY-2202 module
        ser_par(serialport,"flush")
        ser_put(serialport,sprintf("%s?\n",movecmd))
        sleep(0.05)
        pos = ser_get(serialport)
        # delete trailing ACK (ASCII char 6)
55     pos_cleaned = substr(pos,2)
        return(pos_cleaned)
    }
```

```
    }  
  
    if (key == "set_position") {  
60     ser_par(serialport,"flush")  
     ser_put(serialport,sprintf("%s %.2f\n",movecmd,p1))  
     return  
    }  
  
65     if (key == "start_one") {  
     # pause because of slow cyberstar electronics, caues problems when  
     # two parameters are changed subsequently (SCA window scan)  
     sleep(0.1)  
     ser_par(serialport,"flush")  
70     ser_put(serialport,sprintf("%s %.2f\n",movecmd,p1))  
     sleep(0.1)  
     return  
    }  
  
75     if (key == "get_status") {  
     local returnstate  
     returnstate = 0x00  
     return(returnstate)  
    }  
80 }'  
  
def cyberstar_par(mne, key, type, p1, p2) '{  
    if (mode == "get") {  
85     return(-1)  
    }  
  
    # Called to set the user-defined parameter key to p1 for counter  
    # number mne.  
    if (mode == "set") {  
90     return(-1)  
    }  
}'  
  
def _cyberstar_getmovecmd(type) '{  
95     if (type == "highvoltage"){  
     movecmd = ":SOUR0:VOLT"  
     return(movecmd)  
    }  
    if (type == "scalow"){  
100     movecmd = ":SENS0:SCA:LOW"  
     return(movecmd)  
    }  
}'
```

```

105  if (type == "scaup"){
      movecmd = ":SENS0:SCA:UPP"
      return(movecmd)
    }
      if (type == "gain"){
      movecmd = ":INP0:GAIN"
      return(movecmd)
110  }
      if (type == "peakingtime"){
      movecmd = ":SENS0:PKT"
      return(movecmd)
    }
115  return(-1)
  }'

```

Listing B.9: Macro for periodic experimenter maintenance.

```

def prompt_mac '{
  local hour, minute, seconds
  local lunch_opening, lunch_closing, dinner_opening, dinner_closing
  hour = date("%H")
5  minute = date("%M")
  seconds = (hour*60 + minute) * 60

  # lunch time
  lunch_opening = (12.0 * 60) * 60 # opening at 12:00
10  lunch_closing = (13.5 * 60) * 60 # closing at 13:30

  # dinner time
  dinner_opening = (18.5 * 60) * 60 # opening at 18:30
  dinner_closing = (20.0 * 60) * 60 # closing at 20:00
15

  # at lunch time
  if (( seconds >= lunch_opening) && (seconds <= lunch_closing)) {
    _lunch_message
  }
20

  # at dinner time
  if (( seconds >= dinner_opening) && (seconds <= dinner_closing)) {
    _dinner_message
  }
25 }'

def _lunch_message '{
  global LAST_MESSAGE, LAST_MESSAGE_COUNTER
  local message_count
30

```

```

messages["0"] = "Lunch time approaching."
messages["1"] = "The canteen is now open."
messages["2"] = "Lunch time."
messages["3"] = "Getting hungry?"
35 messages["4"] = "\"Ask not what you can do for your country. Ask
    what\'s for lunch.\"\n\tOrson Welles"
messages["5"] = "\"I only eat one meal a day. Lunch, not
    dinner.\"\n\tRobert Goulet"
messages["6"] = "\"Lunch is for wimps.\"\n\tOliver Stone"
messages["7"] = "Aren\'t you hungry?"
messages["8"] = "There is no free lunch."
40 messages["9"] = "Are you hungry? Shall I close the shutters?"
messages["10"] = "DO NOT FEED THE TROLLS!"
messages["11"] = "\"Lunch kills half of Paris, supper the other
    half.\"", Charles de Montesquieu"
messages["12"] = "\"When ordering lunch, the big executives are just
    as indecisive as the rest of us.\" William Feather"
messages["13"] = "\"Time is an illusion, lunchtime doubly so.\"",
    Douglas Adams"
45
# get number of messages
message_count = 0
for (i in messages) {
50     message_count++
}

intervall = 5*60 # display a message every 5 minutes
if (LAST_MESSAGE + intervall < time()) {
55     printf("\n%s\n",messages[LAST_MESSAGE_COUNTER % message_count])
    LAST_MESSAGE_COUNTER++
    LAST_MESSAGE = time()
}
}'

60 def _dinner_message '{
    global LAST_MESSAGE, LAST_MESSAGE_COUNTER
    local message_count

messages["0"] = "Dinner time approaching."
65 messages["1"] = "Who is preparing dinner today?"
messages["2"] = "Dinner time."
messages["3"] = "Getting hungry?"
messages["4"] = "\"I would have loved to have met Marilyn Monroe and
    have dinner with her.\"\n\tJuliette Binoche"
messages["5"] = "\"My body is like breakfast, lunch, and dinner. I
    don\'t think about it, I just have it.\"\n\tArnold Schwarzenegger"
70 messages["6"] = "\"Burning dinner is not incompetence but

```

```

    war.\n\tMarge Piercy"
    messages["7"] = "Aren\'t you hungry?"
    messages["8"] = "There is no free lunch. But free dinner?"
    messages["9"] = "Are you hungry? Shall I close the shutters?"
    messages["10"] = "DO NOT FEED THE TROLLS!"
75
    # get number of messages
    message_count = 0
    for (i in messages) {
80
        message_count++
    }

    intervall = 5*60 # display a message every 5 minutes
    if (LAST_MESSAGE + intervall < time()) {
        printf("\n%s\n",messages[LAST_MESSAGE_COUNTER % message_count])
85
        LAST_MESSAGE_COUNTER++
        LAST_MESSAGE = time()
    }
}'

```

Listing B.10: Macro for usage of the Keithley 6485 Picoammeter as a counter in spec.

```

#####
#                                     #
# MACRO COUNTER FOR KEITHLEY 6485 PICOAMMETER #
#                                     #
5 #####
# S. Kalbfleisch, 28.11.2010          #
#####

def keithley_config(mne, key, p1, p2, unit) '{
10 # send init and config commands to device
    if (key == "cnt") {
        serialport = counter_par(mne, "serialport")
        if (serialport <= 0) {
            eprint cnt_mne(mne) " not configured porperly. No serial port number
                given as optional counter parameter."
15         return(".error.")
        }
        # init device
        if (!(ser_par(serialport,"responsive"))) {
            eprint "Serial device for \" cnt_mne(mne)\" not configured
                properly."
20         return(".error.")
        } else {
            printf("Using Keithley 6485 on serial port \"%s\" for counter

```

```

        \"%s\".\n", ser_par(serialport, "device_id"), cnt_mne(mne))
    }
    # init/reset device
25  _keithley_reset(serialport)
    }
    return
}'

30 def keithley_cmd(mne, key, p1, p2, unit) '{
    local serialport
    global KEITHLEY_COUNTABORT KEITHLEY_READINGS
    # get the serialport of mne
    if (mne == "..") {
35     }
    else {
        serialport = counter_par(mne, "serialport")
    }

40  if (key == "start_one") {
        # commands sent before counting is started, use for init of counting
        run
        local freq nplc readings

        # query power line frequency (PLC)
45     freq = _keithleyread(serialport,"SYST:LFR?")

        nplc = 1 # set reading time of 20 ms (one period of 50 Hz, 1 NPLC =
            1/50 Hz = 20 ms)
        # calculate readings that can be done in the COUNTTIME p1
        readings = int( p1*freq/nplc )
50     # limit the number of readings to maximum buffersize of device
        if (readings > 2500) {
            readings = 2500
        }
        _keithley_send(serialport,"*CLS")
55     _keithley_send(serialport,"TRIG:DEL 0")
        _keithley_send(serialport,sprintf("TRIG:COUN %i",readings))
        _keithley_send(serialport,sprintf("NPLC %f",nplc))
        _keithley_send(serialport,sprintf("TRAC:POIN %i",readings))
        _keithley_send(serialport,"TRAC:CLE")
60     _keithley_send(serialport,"TRAC:FEED:CONT NEXT")
        _keithley_send(serialport,"STAT:MEAS:ENAB 512")
        _keithley_send(serialport,"INIT")
        _keithley_send(serialport,"*OPC?")
        return
65     }
}

```



```
70  if (key == "get_status") {
    # get status of counter, relevant for polled counters
    return
  }

  if (key == "counts") {
    # read counts
    local ans
75    if (KEITHLEY_COUNTABORT != 1) {
        # wait until measurement has finished
        # this does not work properly in server mode
        local timeout
        timeout = ser_par(serialport,"timeout")
80    ser_par(serialport,"timeout",0.05)

        while (ser_get(serialport) != 1) {
        }

85

        ser_par(serialport,"timeout",timeout)
        ser_par(serialport,"flush")
        _keithleysend(serialport,"CALC3:FORM MEAN")
        local maxreadings
90    ans = ""
        maxreadings = 0
        while ((ans == "") && (maxreadings < 15)) {
            ans=_keithleyread(serialport,"CALC3:DATA?")
            maxreadings++
95    sleep(0.02)
        }
    }
    KEITHLEY_READINGS++
    # reset Keithley after 250 readings
100   if (KEITHLEY_READINGS >= 250) {
        _keithley_reset(serialport)
        KEITHLEY_READINGS = 0
    }
    KEITHLEY_COUNTABORT = 0
105   return(ans)
  }

  if (key == "halt_one") {
    # abort counting
110   if (p1!=0) {
        printf("aborting counting on Keithley...")
        KEITHLEY_COUNTABORT = 1
        _keithleysend(serialport,"ABOR")
    }
  }
}
```

```
115     _keithley_reset(serialport)
        KEITHLEY_READINGS = 0
        sleep(3)
        printf("done\n")
    }
    return
120 }
}'

def _keithley_reset(serialport) '{
    # reset to factory defaults
125     _keithleysend(serialport,"SYST:PRES")
    # reset
    _keithleysend(serialport,"*RST")
    # enable autoranging
    _keithleysend(serialport,"CURR:RANGE:AUTO ON")
130     # disable zero check
    _keithleysend(serialport,"SYST:ZCH OFF")
    # disable autozero
    _keithleysend(serialport,"SYST:AZER:STAT OFF")
    # set output format to measured value only
135     _keithleysend(serialport,"FORM:ELEM READ")
    _keithleysend(serialport,"*CLS")
    # clear current data in buffer
    _keithleysend(serialport,"TRAC:CLE")
    _keithleysend(serialport,"SYST:KEY 32")
140 }'

def _keithleysend(port, cmd) '{
    ser_par(port,"flush")
145     ser_put(port, sprintf("%s\r", cmd) )
    return
}'

def _keithleyread(port, cmd) '{
150     local ans
    ser_par(port,"flush")
    ser_put(port, sprintf("%s\r", cmd) )
    ans = ser_get(port)
    return ans
155 }'

def _keithley_cleanup(serialport) '{
    p "cleaning diode"
    _keithley_reset(serialport)
160 }'
```

Listing B.11: Macro for usage of the home built high voltage supply for ionisation chamber.

```
#####  
# #  
# spec support for home built HV supply #  
# with serial line #  
5 # #  
#####  
# S. Kalbfleisch, 04.12.2009 #  
#####  
  
10 ### GLOBAL VARIABLES ###  
global TTYHV HVCALIB  
# serial line of HV supply  
TTYHV = 0  
  
15 # Calibration data of HV Supply (device specific)  
HVCALIB["m"] = 4.5391E+01 # slope  
HVCALIB["b"] = -5.4828E+03 # intersection with y axis  
  
#####  
20 # user commands #  
#####  
  
# show current high voltage  
def hvshow '{  
25 printf("current high voltage set to %i V.\n",_get_hv())  
'  
  
# set new high voltage  
def hvset '{  
30 local voltage  
if ($#!=1) {  
printf("Usage: hvset VOLTAGE\n\n")  
} else {  
voltage = int($1)  
35 _set_hv(voltage)  
printf("high voltage set to %i V.\n",_get_hv())  
}  
}'  
  
40 #####  
# internal commands #  
#####
```

```
# read the current HV from the device
45 def _get_hv()'{
    # init byte array for voltage value in 16 bit=2 bytes
    ubyte array hvraw[2]
    # clean up serial line
    ser_par(TTYHV,"flush")
50 # ask for HV
    ser_put(TTYHV,2)
    # read the answer (2 bytes)
    ser_get(TTYHV,hvraw)
    voltage = _raw2volt(hvraw[0]*256 + hvraw[1])
55 return sprintf("%.f",voltage) # round to integer
}'

# set the HV
def _set_hv(voltage)'{
60 # convert voltage to byte
    local hvraw
    array hvraw[2]
    hvraw = _volt2raw(voltage)

65 # prepare command string
    ubyte array hvsend[3]
    hvsend[0] = \061
    # \061 = 1 in ASCII
    # 1 is the command for setting the voltage
70 # followed by 2 bytes for the 16bit value of the DA-converter
    hvsend[1] = hvraw[0]
    hvsend[2] = hvraw[1]

    # clean up serial line
75 ser_par(TTYHV,"flush")
    # set HV
    ser_put(TTYHV,hvsend)
    return 1
}'

80 # calculate the 16bit raw value from given voltage
def _volt2raw(voltage)'{
    # raw = m*voltage + b
    m = HVCALIB["m"]
85 b = HVCALIB["b"]
    rawdecimal = int(m*voltage + b)
    # check if voltage is within accesible range
    if (rawdecimal < 0 || rawdecimal > 65535) {
        fprintf "Voltage out of range"
```

```

90     exit
    }
    ubyte array raw[2]
    raw[1] = rawdecimal%256
    raw[0] = (rawdecimal-raw[1])/256
95     return raw
}'

# calculate voltage from the given 16bit raw value as read from the
  device
def _raw2volt(raw) '{
100   # voltage = 1/m*raw - b/m
    m = HVCALIB["m"]
    b = HVCALIB["b"]
    voltage = 1/m*raw - b/m
    return voltage
105 }'
```

Listing B.12: Macro for calculational macro motors for vertical and horizontal KB slit gaps and offsets.

```

# Definition of calculational macro motors for scatterless slits
# horizontal direction

def skbh_config(mne, type, unit, module, chan) '{
5   if (type == "mot")
    return "skbl skbr"
}'

def skbh_calc(mne, mode) '{
10   if (mode == 0) {
    if (mne == skbhg)
        A[mne] = A[skbr] + A[skbl]
    if (mne == skbho)
        A[mne] = (A[skbl] - A[skbr])/2
    } else {
15     if (mne == skbl)
        A[mne] = A[skbho] + A[skbhg]/2
    if (mne == skbr)
        A[mne] = -A[skbho] + A[skbhg]/2
    }
20 }'
```

```

# Definition of calculational macro motors for scatterless slits
# vertical direction

25 def skbv_config(mne, type, unit, module, chan) '{
    if (type == "mot")
```

```

    return "skbb skbt"
}'
def skbv_calc(mne, mode) '{
30  if (mode == 0) {
    if (mne == skbvg)
        A[mne] = A[skbb] + A[skbt]
    if (mne == skbvo)
        A[mne] = (A[skbt] - A[skbb])/2
35  } else {
    if (mne == skbb)
        A[mne] = -A[skbvo] + A[skbvg]/2
    if (mne == skbt)
        A[mne] = A[skbvo] + A[skbvg]/2
40  }
}'

```

Listing B.13: Macro for calculational macro motor for height of optical table.

```

#####
# macromotor functions for height (z) of IDT optical table #
#####

5  def IDTtable_height_config(mne, type, unit, module, chan) '{
    global TABLE_HEIGHT
    # insert here motor mnemonics of used real motors
    # INSERT SAME INFORMTION ALSO IN IDTtable_height_calc()
    z1_mne = otz1
10  z2_mne = otz2
    z3_mne = otz3
    height_mne = otz
    if (type == "mot") {
        ### height ###
15  if (mne == height_mne ) {
        real_motors = sprintf("%s %s %s", motor_mne(z1_mne),
            motor_mne(z2_mne), motor_mne(z3_mne))
        # display status information on config/reconfig
        deviceid = motor_par(mne,"device_id")
        motorname = motor_mne(mne)
20  printf("Using %s macro motor with motor(s) \"%s\" for \"%s\"\\n",
            deviceid, real_motors, motorname)
        return real_motors
    }
    }
25  }'

```

```

def IDTtable_height_calc(mne, mode) '{
  # insert here motor mnemonics of used real motors
  # USE SAME INFORMATION AS IN IDTtable_height_calc()
30  z1_mne = otz1
    z2_mne = otz2
    z3_mne = otz3
    height_mne = otz
  # table geometry
35  tab_length = 1100.0
    tab_width = 600.0
    if (mode == 0) {
      if (mne == height_mne)
        A[mne] = (A[z1_mne] + A[z2_mne] + A[z3_mne])/3
40  } else {
      if (mne == "..")
        TABLE_HEIGHT = (A[z1_mne]+A[z2_mne]+A[z3_mne])/3
      else if (mne == z1_mne){
        delta = A[height_mne] - TABLE_HEIGHT
45      A[mne] = A[mne] + delta
      }
      else if (mne == z2_mne) {
        delta = A[height_mne] - TABLE_HEIGHT
        A[mne] = A[mne] + delta
50      }
      else if (mne == z3_mne) {
        delta = A[height_mne] - TABLE_HEIGHT
        A[mne] = A[mne] + delta
55      }
    }
  }'

```

Listing B.14: Macro for calculational macro motor for translation of optical table.

```

#####
# macromotor functions for translation (y) of IDT optical table #
#####
5  def IDTtable_trans_config(mne, type, unit, module, chan) '{
    global TABLE_TRANS
    # insert here motor mnemonics of used real motors
    # INSERT SAME INFORMTION ALSO IN IDTtable_trans_calc()
10  y1_mne = oty1
    y2_mne = oty2
    trans_mne = oty
    if (type == "mot") {
      ### translation (y) ###

```

```

15     if (mne == trans_mne ) {
        real_motors = sprintf("%s %s", motor_mne(y1_mne),
            motor_mne(y2_mne))
        # display status information on config/reconfig
        deviceid = motor_par(mne,"device_id")
        motorname = motor_mne(mne)
        printf("Using %s macro motor with motor(s) \"%s\" for \"%s\"\\n",
20         deviceid, real_motors, motorname)
        return real_motors
    }
}
}'

25
def IDTtable_trans_calc(mne, mode) '{
    # insert here motor mnemonics of used real motors
    # USE SAME INFORMATION AS IN IDTtable_trans_calc()
    y1_mne = oty1
30    y2_mne = oty2
    trans_mne = oty
    # table geometry
    tab_length = 1100.0
    tab_width = 600.0
35    if (mode == 0) {
        if (mne == trans_mne)
            A[mne] = (A[y1_mne] + A[y2_mne])/2
        } else {
            if (mne == "..")
40             TABLE_TRANS = (A[y1_mne]+A[y2_mne])/2
            else if (mne == y1_mne){
                delta = A[trans_mne] - TABLE_TRANS
                A[mne] = A[mne] + delta
            }
45            else if (mne == y2_mne) {
                delta = A[trans_mne] - TABLE_TRANS
                A[mne] = A[mne] + delta
            }
        }
50    }
}'

```

Listing B.15: Macro for calculational macro motor for pitch angle of optical table.

```

#####
# macromotor functions for pitch angle (yrot) of IDT optical table #
#####

```



```

5  def IDTtable_pitch_config(mne, type, unit, module, chan) '{
    global TABLE_PITCH
    # insert here motor mnemonics of used real motors
    # INSERT SAME INFORMTION ALSO IN IDTtable_pitch_calc()
    z1_mne = otz1
10  z2_mne = otz2
    z3_mne = otz3
    pitch_mne = otyrot
    if (type == "mot") {
        ### pitch (yrot) ###
15  if (mne == pitch_mne ) {
            real_motors = sprintf("%s %s %s", motor_mne(z1_mne),
                                motor_mne(z2_mne), motor_mne(z3_mne))
            # display status information on config/reconfig
            deviceid = motor_par(mne,"device_id")
            motorname = motor_mne(mne)
20  printf("Using %s macro motor with motor(s) \"%s\" for \"%s\"\\n",
            deviceid, real_motors, motorname)
            return real_motors
        }
    }
}
25 }'

def IDTtable_pitch_calc(mne, mode) '{
    # insert here motor mnemonics of used real motors
    # USE SAME INFORMATION AS IN IDTtable_pitch_calc()
    z1_mne = otz1
30  z2_mne = otz2
    z3_mne = otz3
    pitch_mne = otyrot
    # table geometry
    tab_length = 1100.0
35  tab_width = 600.0

    if (mode == 0) {
        if (mne == pitch_mne)
            A[mne] = deg(atan( ((A[z2_mne] + A[z3_mne])/2 - A[z1_mne]) /
                               tab_length ))
40  } else {
        if (mne == "..")
            TABLE_PITCH = deg(atan( ((A[z2_mne] + A[z3_mne])/2 - A[z1_mne]) /
                                       tab_length ))
        else if (mne == z1_mne){
            delta = A[pitch_mne] - TABLE_PITCH
45  delta_z = tan(rad(delta)) * tab_length/2
            A[mne] = A[mne] - delta_z
        }
    }
}

```

```

    else if (mne == z2_mne) {
        delta = A[pitch_mne] - TABLE_PITCH
50     delta_z = tan(rad(delta)) * tab_length/2
        A[mne] = A[mne] + delta_z
    }
    else if (mne == z3_mne) {
        delta = A[pitch_mne] - TABLE_PITCH
55     delta_z = tan(rad(delta)) * tab_length/2
        A[mne] = A[mne] + delta_z
    }
}
}'

```

Listing B.16: Macro for calculational macro motor for roll angle of optical table.

```

#####
# macromotor functions for roll angle (xrot) of IDT optical table #
#####

5 def IDTtable_roll_config(mne, type, unit, module, chan) '{
    global TABLE_ROLL
    # insert here motor mnemonics of used real motors
    # INSERT SAME INFORMTION ALSO IN IDTtable_roll_calc()
    z1_mne = otz2
10    z2_mne = otz3
    roll_mne = otxrot
    if (type == "mot") {
        ### roll (xrot) ###
        if (mne == roll_mne ) {
15            real_motors = sprintf("%s %s", motor_mne(z1_mne),
                motor_mne(z2_mne))
            # display status information on config/reconfig
            deviceid = motor_par(mne,"device_id")
            motorname = motor_mne(mne)
            printf("Using %s macro motor with motor(s) \"%s\" for \"%s\"\\n",
20                deviceid, real_motors, motorname)
            return real_motors
        }
    }
}'

25 def IDTtable_roll_calc(mne, mode) '{
    # insert here motor mnemonics of used real motors
    # USE SAME INFORMATION AS IN IDTtable_roll_calc()
    z1_mne = otz2
    z2_mne = otz3

```

```

30  roll_mne = otxrot
    # table geometry
    tab_length = 1100.0
    tab_width = 600.0
    if (mode == 0) {
35      if (mne == roll_mne)
          A[mne] = deg(atan( (A[z2_mne] - A[z1_mne]) / tab_width ))
    } else {
        if (mne == "..")
            TABLE_ROLL = deg(atan( (A[z2_mne] - A[z1_mne]) / tab_width ))
40      else if (mne == z1_mne){
          delta = A[roll_mne] - TABLE_ROLL
          delta_z = tan(rad(delta)) * tab_width/2
          A[mne] = A[mne] - delta_z
        }
45      else if (mne == z2_mne) {
          delta = A[roll_mne] - TABLE_ROLL
          delta_z = tan(rad(delta)) * tab_width/2
          A[mne] = A[mne] + delta_z
        }
    }
50  }
}'

```

Listing B.17: Macro for calculational macro motor for yaw angle of optical table.

```

#####
# macromotor functions for yaw angle (zrot) of IDT optical table #
#####
5  def IDTtable_yaw_config(mne, type, unit, module, chan) '{
    global TABLE_YAW
    # insert here motor mnemonics of used real motors
    # INSERT SAME INFORMTION ALSO IN table_yaw_calc()
    y1_mne = oty1
10   y2_mne = oty2
    yaw_mne = otzrot
    if (type == "mot") {
        ### yaw (zrot) ###
        if (mne == yaw_mne ) {
15         real_motors = sprintf("%s %s", motor_mne(y1_mne),
            motor_mne(y2_mne))
            # display status information on config/reconfig
            deviceid = motor_par(mne,"device_id")
            motorname = motor_mne(mne)
            printf("Using %s macro motor with motor(s) \"%s\" for \"%s\"\n",
                deviceid, real_motors, motorname)

```

```

20     return real_motors
      }
    }
  }'

25 def IDTtable_yaw_calc(mne, mode) '{
    # insert here motor mnemonics of used real motors
    # USE SAME INFORMATION AS IN IDTtable_pitch_calc()
    y1_mne = oty1
    y2_mne = oty2
30   yaw_mne = otzrot
    # table geometry
    tab_length = 1100.0
    tab_width = 600.0

35   if (mode == 0) {
      if (mne == yaw_mne)
        A[mne] = deg( atan((A[y2_mne] - A[y1_mne])/tab_length) )
    } else {
      if (mne == "..")
40     TABLE_YAW = deg( atan((A[y2_mne] - A[y1_mne])/tab_length) )
      else if (mne == y1_mne){
        delta = A[yaw_mne] - TABLE_YAW
        delta_y = tan(rad(delta))*tab_length/2
        A[mne] = A[mne] - delta_y
45     }
      else if (mne == y2_mne) {
        delta = A[yaw_mne] - TABLE_YAW
        delta_y = tan(rad(delta))*tab_length/2
        A[mne] = A[mne] + delta_y
50     }
    }
  }'

```

Listing B.18: Macro for beam-following macro motors for sample, OAV microscope and piezos.

```

#####
#                                     #
# BEAM-FOLLOWING MACRO MOTORS FOR #
#           STX, OAVX, PX           #
5 #                                     #
#####
# for use at P10@PETRA III           #
# S. Kalbfleisch, 25.11.2010        #
#####
10

```

```
def stxb_config(mne, type, unit, module, chan) '{
    global DELTA_stx ALPHA_stx BETA_stx
    # uplift angle, in degree
    # vertical deflecting angle is ALPHA_stx, in degree
15  # horizontal deflecting angle is BETA_stx, in degree
    BETA_stx = 3
    if (type == "mot")
        return "stx sty stz"
}'
20
def stxb_calc(mne, mode) '{
    if (mode == 0) {
        if (mne == stxb) {
25             A[mne] = A[stx]
        }
    } else {
        if (mne == "..") {
            # calculate the distance to be move
            DELTA_stx = A[stxb] - A[stx]
30        } else if (mne == stx) {
            A[mne] = A[stxb]
        } else if (mne == sty) {
            A[mne] += DELTA_stx * sin(rad(BETA_stx))
        } else if (mne == stz) {
35        A[mne] += DELTA_stx * sin(rad(ALPHA_stx))
        }
    }
}'
40
def oavxb_config(mne, type, unit, module, chan) '{
    global DELTA_oavx ALPHA_oavx BETA_oavx
    # vertical deflecting angle is ALPHA_oavx, in degree
    # horizontal deflecting angle is BETA_oavx, in degree
    if (type == "mot")
45        return "oavx oavy oavz"
}'
50
def oavxb_calc(mne, mode) '{
    if (mode == 0) {
        if (mne == oavxb) {
            A[mne] = A[oavx]
        }
    } else {
        if (mne == "..") {
55        # calculate the distance to be move
            DELTA_oavx = A[oavxb] - A[oavx]
        } else if (mne == oavx) {
```

```

        A[mne] = A[oavxb]
    } else if (mne == oavy) {
60     A[mne] += DELTA_oavx * sin(rad(BETA_oavx))
    } else if (mne == oavz) {
        A[mne] += DELTA_oavx * sin(rad(ALPHA_oavx))
    }
}
65 }'

def pxb_config(mne, type, unit, module, chan) '{
    global DELTA_px ALPHA_px BETA_px
    # vertical deflecting angle is ALPHA_px, in degree
70    # horizontal deflecting angle is BETA_px, in degree
    if (type == "mot")
        return "px py pz"
}'

75 def pxb_calc(mne, mode) '{
    if (mode == 0) {
        if (mne == pxb) {
            A[mne] = A[px]
        }
80    } else {
        if (mne == "..") {
            # calculate the distance to be move
            DELTA_px = A[pxb] - A[px]
        } else if (mne == px) {
85         A[mne] = A[pxb]
        } else if (mne == py) {
            A[mne] += DELTA_px * sin(rad(BETA_px))
        } else if (mne == pz) {
90         A[mne] += DELTA_px * sin(rad(ALPHA_px))
        }
    }
}'

```

Listing B.19: Helper macros for file name creation in detector macros and restoring information after a spec restart.

```

global DATA_BASE DATA_BASEDIR WIN_DRIVE WIN_BASEDIR

# path to general location of data on disk
5 DATA_BASE = "/home/data"
  WIN_DRIVE = "Z:"

```

```

def newrun '{
10  if ($# != 1) {
    eprint "Usage: newrun name_of_run"
    printf(" Current DATA_BASEDIR is: %s\n",DATA_BASEDIR)
    exit
  }
15  local _run _year _date _settingsfile
    _run = "$1"
    _year = date("%Y")
    _date = date("%Y_%m_%d")

20  # disable current logging files
    off(sprintf("%s/log/tlog",DATA_DIR))
    off(sprintf("%s/log/dlog",DATA_DIR))
    close(sprintf("%s/log/tlog",DATA_DIR))
    close(sprintf("%s/log/dlog",DATA_DIR))

25  # form new data directory variables
    DATA_BASEDIR = sprintf("%s/%s/%s_%s", DATA_BASE, _year, _date, _run)
    DATA_DIR = sprintf("%s/spec",DATA_BASEDIR)
    WIN_BASEDIR = sprintf("%s\\%s\\%s_%s", WIN_DRIVE, _year, _date, _run)

30  # create directories
    unix(sprintf("mkdir -p %s",DATA_BASEDIR))
    # dir for spec files
    unix(sprintf("mkdir -p %s",DATA_DIR))
35  # dir for spec logging
    unix(sprintf("mkdir -p %s/log",DATA_DIR))
    # dir for position dumps
    unix(sprintf("mkdir -p %s/dump",DATA_DIR))

40  # enable current logging files
    on(sprintf("%s/log/tlog",DATA_DIR))
    on(sprintf("%s/log/dlog",DATA_DIR))

    #####
45  # save settings for a fresh start of spec
    #####
    _settingsfile = sprintf("%s/%s/current_run.mac",SPEC, SPEC)

    # first delete existing file (only way to clear the existing content
    # since spec always appends to existing files)
50  if (file_info(_settingsfile, "-f"))
    unix(sprintf("rm %s", _settingsfile))

    # create file and insert values
    # first replace \ in WIN_BASEDIR to survive reloading at fresh start

```

```

55  local numel # number of backslashes in WIN_BASEDIR
    numel = split(WIN_BASEDIR, slashcut, "\\")
    slashed = ""
    for (i=0; i<numel-1; i++) {
60      slashed = slashed sprintf("%s\\", slashcut[i])
    }
    i = i++
    slashed = slashed sprintf("%s", slashcut[i]) # this is now WIN_BASEDIR
        with doubled backslashes

    s = sprintf("DATA_BASEDIR = \"%s\"\n", DATA_BASEDIR)
65  s = s sprintf("DATA_DIR = \"%s\"\n", DATA_DIR)
    s = s sprintf("WIN_BASEDIR = \"%s\"\n\n", slashed)
    s = s sprintf("%s\n", "on(sprintf(\"%s/log/tlog\",DATA_DIR))" )
    s = s sprintf("%s\n", "on(sprintf(\"%s/log/dlog\",DATA_DIR))" )
    fprintf(_settingsfile, "%s\n", s)
70  # close file
    close(_settingsfile)

    # call newfile
    newfile
75  }'

#####
# reload settings after a fresh start of spec #
#####
80  if (FRESH) {
    local _settingsfile
    _settingsfile = sprintf("%s/%s/current_run.mac",SPEC, SPEC)
    if (file_info(_settingsfile, "-r")) {
85      qdofile(_settingsfile, label)
    } else {
        eprintf("Can't read command file \"%s\".\n", _settingsfile)
    }
    }
90  }

#####
# create user_newfile as hook for other macros #
#####
95  cdef("user_newfile")
    # definition of newfile as in standard.mac plus user_newfile
    def newfile '{
        local dfile, scan_num, prev
100  prev = DATAFILE

```



```
    if ($# == 0)
        dfile = getsval("\nData file", DATAFILE)
    else
        dfile = "$1"
105
    if (dfile == "")
        exit

    if (dfile == "null")
110        dfile = "/dev/null"

    if (!index(dfile, "/") && file_info(DATA_DIR, "-d"))
        dfile = sprintf("%s/%s",DATA_DIR,dfile)

115    if ((dfile = user_filecheck(dfile)) == "")
        exit

    if (!file_info(dfile, "-r")) {
        if (dfile == DATAFILE) {
120            printf(\
"Warning: Same name as before, but file isn't in %s directory.\n",\
                index(dfile, "/")? "same":"current")
            printf("A new version of the file will be created.\n")
        }
125        DATAFILE="/dev/null"
    } else if (dfile != "/dev/null") {
        local s
        if (file_info(s = SPECD "/chk_file", "-x"))
            if (unix(sprintf("%s %s",s,dfile)))
130            print "Note: file already exists."
    }
    #
    if (open(dfile)) {
        DATAFILE="/dev/null"
135        exit
    }
    if (dfile != "tty" && dfile != "/dev/tty")
        close(dfile)

140    scan_num = $2
    if (dfile == DATAFILE) {
        if (open(dfile)) {
            DATAFILE="/dev/null"
            exit
145        }
    }
    if ($# == 2)
        SCAN_N = scan_num
```

```

} else {
  if ($# == 0)
150     scan_num = getval("Last scan number",scan_num)

  if (prev != "" && prev != "0" && prev != "tty" && prev != "/dev/tty")
    close(prev)

155  if (open(dfile)) {
    DATAFILE="/dev/null"
    exit
  }

160  DATAFILE = dfile
  SCAN_N = scan_num

  if (SCAN_N == 0 || !file_info(DATAFILE, "-s")) {
    constant EPOCH time()
165    fprintf(dfile, "#F %s\n", DATAFILE)
  }
  fprintf(dfile, "#E %d\n",EPOCH)
  fprintf(dfile, "#D %s\n",date())
  fprintf(dfile, "#C %s User = %s\n",TITLE,USER)

170  ond; offt
  user_filehead
  ont; offd

175  local i,j,s
  for (i=0; i<MOTORS; i+=8) {
    s = sprintf("#0%d ", i/8)
    for (j=i; j<i+8 && j<MOTORS;) {
      if (motor_name(mA[j]) != "unused")
180        s = s sprintf("%8s", motor_name(mA[j]))
      if (j%8 == 7)
        break
      s = s " "
      j++
185    }
    fprintf(dfile, "%s\n", s)
  }
  user_newfile
}
190 printf("Using \"%s\". Next scan is number %d.\n",DATAFILE,SCAN_N+1)
}'

```

```
#####
```

```

195 # extract the filename out of DATAFILE #
# (cut directories and extensions) #
#####
def getdatafilename(){
200   local datafilename slashcounter datafileparts

# cut trailing directories
if (index(DATAFILE, "/") > 0) {
  slashcounter = split(DATAFILE,datafileparts,"/")
  datafilename = datafileparts[slashcounter-1]
205 }

# remove pending extensions (use only the part before the first ".")
if (index(datafilename, ".") > 0) {
  pointcounter = split(datafilename,filenameparts, ".")
210   datafilename = filenameparts[0]
}

return datafilename
}'

```

Listing B.20: Helper macro for dump of motor positions at acquisitions of detectors.

```

def _save_motordump(filename) '{
# filename should be something like this:
# sprintf("%s/dump/%s.txt",DATA_DIR, FILENAME_USED_DETECTOR_WITHOUT_PATH)
# e.g.
  /data/2010/2010_11_17_dumptesting/spec/dump/dumptesting_CCD_4.spe.txt
5 # with DATA_DIR = "/home/data/2010/2010_11_17_dumptesting/spec" (set by
  "newrun" command
# and FILENAME_USED_DETECTOR_WITHOUT_PATH = "dumptesting_CCD_4.spe"
  fprintf(filename,"%s", _motordump())
  close(filename)
}'
10

def _motordump() '{
  local dumpstring, i
  dumpstring = sprintf("# Motor positions at %s\n# mne user
    dial\n",date())
  for (i=0;i<MOTORS;i++) {
15   dumpstring = dumpstring sprintf("%s %.6f %.6f\n", motor_mne(i),
    A[i], dial(i,A[i]))
  }
  return(dumpstring)
}'

```

## B.2 Temperature Logger Source Code

Listing B.21: Python code for the communication with the temperature logger DK8010

```
#!/usr/bin/python
# driesenkern.py
#
# Container to hold the Driesen&Kern temperature logger object
5
# basic configuration
HOST = '192.168.1.13'
PORT = 4006
LOG_FILENAME =
10     '~/data/2010/2010_11_24_I-20100111/TemperatureLogger/gini.log'
readintervall = 5 # intervall in seconds of reading the temperature
    logger

import math
import time
import datetime
15 import logging
import socket
from logfile import *

class DriesenkernLogger:
20
    # logger attributes

    HOST = None
    PORT = None
25    requeststring = '\r\n'
    delay = 0.15
    __rawdata = str()
    __timestamp = float()
    loggerstring = str()
30    __errorstring = "-9999"

    # Function used to create a logfile object

    def __init__(self, HOST, PORT):
35        self.HOST = HOST
        self.PORT = PORT

    def ReadRawData(self):
        try:
```

```
40         # connect via tcp to HOST on port PORT
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect((self.HOST, self.PORT))
    except socket.error, (value, message):
45         if s:
            s.close()
            #TODO improve error handling if connection fails
            timeoferror = time.strftime("%Y-%m-%d %H:%M:%S")
            print('Error on connection at ' + timeoferror)
            self.__rawdata = rawdata = self.__errorstring
50            timestamp = datetime.datetime.now()
            self.__timestamp = timestamp
            return rawdata, timestamp

        time.sleep(self.delay / 10)
55        s.settimeout(0.1)

        reading = True
        while reading:
            try:
60                # send request strings until the answer contains the
                # temperature data, i.e. string contains \t and
                self.requeststring
                s.send(self.requeststring)
                timestamp = time.time()
                time.sleep(self.delay)
65                rawdata = s.recv(4096)
                #print repr(rawdata)
                if (rawdata.count('\t') >= 1) and
                    (rawdata.count(self.requeststring)>=1):
                    reading = False
            except socket.timeout:
70                reading = True

            self.__timestamp = timestamp
            # close tcp connection
            s.close()
75

            self.__rawdata = rawdata
            return rawdata, timestamp

    def FormatRawData(self):
80        rawdata = self.__rawdata
        timestamp = self.__timestamp

        # init temperature dict
        temperaturdict = dict()
```

```
85     #split data in data blocks
    splitdata = rawdata.split('\t')
    # for each data block get channel number and temperature and
    # store it on temperature
    for channelraw in splitdata:
        #deterime the end of the channel char
        channelend = channelraw.find(':')
90     #deterime the end of the temperature char
        temperatureend = channelraw.find('\n')
        #convert channel string to integer
        channel = int(channelraw[0:channelend])
95     #convert temperature string to float
        temperature = float(channelraw[channelend +
            1:temperatureend])
        # insert current channel in temperature list
        temperaturedict[channel] = temperature

100    # create logger string
    loggerstring = str()
    # write timestamp as EPOCH to loggerstring
    loggerstring = '%(#.3f' % {"#": timestamp}
    # now append the values of each channel
105    for i in temperaturedict.itervalues():
        loggerstring = loggerstring + '\t%(#.3f' % {"#": i}
    # return the logger string
    self.loggerstring = loggerstring

110    return loggerstring

def LoggerLoop():
    # Set up a specific logger with our desired output level
115    print "starting..."
    temperaturelogfile = logfile(LOG_FILENAME)

    dk8010 = DriesenkernLogger(HOST, PORT)

120    while True:
        # get starting time of action
        t0 = time.time()
        # read the logger
        dk8010.ReadRawData()
125    # format logger raw output
        dk8010.FormatRawData()
        #print dk8010.loggerstring
        # write to logfile
        temperaturelogfile.write(dk8010.loggerstring)
```

```
130     delta = math.fabs(t0 - time.time())
        #print delta
        # wait before next action is started
        if (readintervall - delta) > 0:
            time.sleep(readintervall - delta)
135
def SingleRun():
    '''only for testing purposes'''
140     temperaturelogfile = logging
        temperaturelogfile.basicConfig(filename=LOG_FILENAME,
                                     format="%(message)s",
                                     level=logging.INFO)
        # new instance of Driesen & Kern Temperature Logger
        dk8010 = DriesenkernLogger(HOST, PORT)
145
        # get starting time of action
        t0 = time.time()
        # read the logger
        dk8010.ReadLogger()
150     # format logger raw output
        dk8010.FormatRawData()
        # write to logfile
        temperaturelogfile.info(dk8010.loggerstring)
        # calculate the elapsed time since start of action
155     delta = math.fabs(t0 - time.time())
        # wait before next action is started
        time.sleep(readintervall - delta)
160 if __name__ == "__main__":
        LoggerLoop()
        #SingleRun()
```

Listing B.22: Python code for the logging of the temperature data.

```
import os.path
# logfile.py
#
# Container to hold the logfile object
5 # import needed classes
import os
import time
10
class logfile:
```

```
# logfile attributes
# name of log file, may include also relative or absolute path
filename = str()
# current logfile
15 currentlogfile = str()
# file identifier
fid = None

# Function used to create a logfile object
20
def __init__(self, filename):
    #check existance for file
    dirname = os.path.dirname(filename)
    if dirname == '':
25         dirname = os.getcwd()

    if not os.path.exists(dirname):
        os.makedirs(dirname)

30     # store file name
    self.filename = filename

def __del__(self):
    # close fid on exit
35     print 'closing file'
    self.fid.close()

def write(self, data):
40     # first write access after init of object
    if self.fid == None:
        self.currentlogfile = self.form_logfilename()
        self.fid = file(self.currentlogfile, 'a')

45     # rotate current logfile
    if self.currentlogfile != self.form_logfilename():
        self.fid.close()
        self.currentlogfile = self.form_logfilename()
        self.fid = file(self.currentlogfile, 'a')

50     # write data in file
    self.fid.writelines(data + '\n')
    # write data in buffer to disk
    self.fid.flush()

55
def form_logfilename(self):
    # create extension with timestamp for logfile name with format
```



## Source Code

---

```
        YYYYMMDD_HH.log, e.g. 20100306_17.log
        newext = time.strftime("_%Y%m%d_%H.log", time.localtime())
60      # replace logfile extension by extension with timestamp
        # first cut current extension
        ext = os.path.splitext(self.filename)[1]
        newfilename = self.filename[0:-len(ext)]
        # now attach new extension
65      newfilename = newfilename + newext
        return newfilename
```

Listing B.23: Python code to run the temperature logger program in daemon mode.

```
#!/usr/bin/python
#####
LOGFILE = '/tmp/pydaemon.log'
PIDFILE = '/tmp/pydaemon.pid'
5 #####
from driesenkern import *
import sys
import os

10 class Log:
    """file like for writes with auto flush after each write
    to ensure that everything is logged, even during an
    unexpected exit."""
    def __init__(self, f):
15         self.f = f
    def write(self, s):
        self.f.write(s)
        self.f.flush()

20 def main():
    #change to data directory if needed
    #os.chdir("/tmp")
    #redirect outputs to a logfile
    #sys.stdout = sys.stderr = Log(open(LOGFILE, 'a+'))
25 #ensure the that the daemon runs a normal user
    #os.setegid(103) #set group first "pydaemon"
    #os.seteuid(103) #set user "pydaemon"
    #start the user program here:
    LoggerLoop()

30 if __name__ == "__main__":
    # do the UNIX double-fork magic, see Stevens' "Advanced
    # Programming in the UNIX Environment" for details (ISBN 0201563177)
```

```
35     try:
        pid = os.fork()
        if pid > 0:
            # exit first parent
            sys.exit(0)
40     except OSError, e:
        print >>sys.stderr, "fork #1 failed: %d (%s)" % (e.errno,
            e.strerror)
        sys.exit(1)

        # decouple from parent environment
        os.chdir("/") #don't prevent unmounting...
45     os.setsid()
        os.umask(0)
        # do second fork
        try:
            pid = os.fork()
50     if pid > 0:
            # exit from second parent, print eventual PID before
            #print "Daemon PID %d" % pid
            open(PIDFILE, 'w').write("%d"%pid)
            sys.exit(0)
55     except OSError, e:
        print >>sys.stderr, "fork #2 failed: %d (%s)" % (e.errno,
            e.strerror)
        sys.exit(1)

        # start the daemon main loop
60     main()
```

# Bibliography

- [1] J. Als-Nielsen and D. McMorrow. *Elements of Modern X-ray Physics*. John Wiley & Sons, 2001.
- [2] D. Attwood. *Soft X-rays and extreme ultraviolet radiation : principles and applications*. Cambridge University Press, 2000.
- [3] K. Balewski, W. Brefeld, W. Decking, H. Franz, H. Röhlberger, and E. Weckert. PETRA III: A Low Emittance Synchrotron Radiation Source. Technical report, Deutsches Elektronen-Synchrotron DESY, 2004.
- [4] M. Bartels. Phasenkontrast-Mikrotomographie an einer Laborröntgenquelle. Diplomarbeit, Universität Göttingen, 2010.
- [5] M. Bartels, M. Priebe, R. N. Wilke, S. P. Krüger, K. Giewekemeyer, S. Kalbfleisch, C. Olendrowitz, M. Sprung, and T. Salditt. Low-dose three-dimensional x-ray imaging of bacterial cells. *unpublished*, 2012.
- [6] A. Beerlink. *Black lipid membranes studied by x-ray phase contrast imaging*. PhD thesis, Universität Göttingen, 2011.
- [7] A. Beerlink, M. Mell, M. Tolkiehn, and T. Salditt. Hard x-ray phase contrast imaging of black lipid membranes. *Applied Physics Letters*, 95(20):203703, 2009.
- [8] C. Bergemann, H. Keymeulen, and J. F. van der Veen. Focusing x-ray beams to nanometer dimensions. *Physical Review Letters*, 91(20):204801, 2003.
- [9] P. Bleuet, P. Cloetens, P. Gergaud, D. Mariolle, N. Chevalier, R. Tucoulou, J. Susini, and A. Chabli. A hard x-ray nanoprobe for scanning and projection nanotomography. *Review of Scientific Instruments*, 80(5):056101, 2009.

- [10] U. Bonse and M. Hart. AN X-RAY INTERFEROMETER. *Applied Physics Letters*, 6:155, 1965.
- [11] Certified Scientific Software. <http://certif.com>.
- [12] W. Chao, B.D. Harteneck, J.A. Liddle, E.H. Anderson, and D.T. Attwood. Soft x-ray microscopy at a spatial resolution better than 15nm. *Nature*, 435(7046):1210–1213, 2005.
- [13] P. Cloetens, R. Barrett, J. Baruchel, J.P. Guigay, and M. Schlenker. Phase objects in synchrotron radiation hard x-ray imaging. *Journal of Physics D: Applied Physics*, 29(1):133–146, 1996.
- [14] J.M. Cowley. *Diffraction Physics*. North Holland Publishing Company, 1981.
- [15] S.B. Dierker, R. Pindak, R.M. Fleming, I.K. Robinson, and L. Berman. X-Ray Photon Correlation Spectroscopy Study of Brownian Motion of Gold Colloids in Glycerol. *Physical Review Letters*, 75(3):449–452, 1995.
- [16] D. Dontsov. Schnelle Analyse. *Quality Engineering*, 6, 2005.
- [17] B. Edlén. The refractive index of air. *Metrologia*, 2(2):71, 1966.
- [18] ESRF. ID22NI homepage. <http://www.esrf.eu/UsersAndScience/Experiments/Beamlines/beamline-snapshot?BeamlineID=ID22NI> (last access 2.2.2012).
- [19] U. Fano. Ionization Yield of Radiations. II. The Fluctuations of the Number of Ions. *Physical Review*, 72:26–29, 1947.
- [20] J.R. Fienup. Phase retrieval algorithms: a comparison. *Applied Optics*, 21(15):2758–2769, 1982.
- [21] C. Fuhse. *X-ray waveguides and waveguide-based lensless imaging*. PhD thesis, Universität Göttingen, 2006.
- [22] C. Fuhse, C. Ollinger, and T. Salditt. Waveguide-based off-axis holography with hard x rays. *Physical Review Letters*, 97(25):254801, 2006.
- [23] C. Fuhse and T. Salditt. Finite-difference field calculations for two-dimensionally confined x-ray waveguides. *Applied Optics*, 45(19):4603–4608, 2006.

- [24] D. Gabor. A new microscopic principle. *Nature*, 161:777–778, 1948.
- [25] K. Giewekemeyer. *A study on new approaches in coherent x-ray microscopy of biological specimens*. PhD thesis, Universität Göttingen, 2011.
- [26] K. Giewekemeyer, S.P. Krüger, S. Kalbfleisch, M. Bartels, C. Beta, and T. Salditt. X-ray propagation microscopy of biological cells using waveguides as a quasipoint source. *Physical Review A*, 83(2):023804, 2011.
- [27] K. Giewekemeyer, H. Neubauer, S. Kalbfleisch, S.P. Krüger, and T. Salditt. Holographic and diffractive x-ray imaging using waveguides as quasi-point sources. *New Journal of Physics*, 12(3):035008, 2010.
- [28] GigE Standard. <http://www.machinevisiononline.org/vision-standards-details.cfm?type=5>.
- [29] S. Gorelick, J. Vila-Comamala, V.A. Guzenko, R. Barrett, M. Salomé, and C. David. High-efficiency Fresnel zone plates for hard X-rays by 100keV e-beam lithography and electroplating. *Journal of Synchrotron Radiation*, 18(3):442–446, 2011.
- [30] T.E. Gureyev. Composite techniques for phase retrieval in the fresnel region. *Optics Communications*, 220(1-3):49 – 58, 2003.
- [31] C. Gutt, T. Ghaderi, V. Chamard, A. Madsen, T. Seydel, M. Tolan, M. Sprung, G. Grübel, and SK Sinha. Observation of Heterodyne Mixing in Surface X-Ray Photon Correlation Spectroscopy Experiments. *Physical Review Letters*, 91(7):76104, 2003.
- [32] D.C. Harris. History of magnetorheological finishing. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 8016, 2011.
- [33] E. Hecht. *Optik*. Oldenbourg, 2005.
- [34] B. L. Henke, E. M. Gullikson, and J. C. Davis. X-Ray Interactions: Photoabsorption, Scattering, Transmission, and Reflection at  $E = 50\text{--}30,000$  eV,  $Z = 1\text{--}92$ . *Atomic Data and Nuclear Data Tables*, 54(2):181–342, July 1993.

- [35] M.R. Howells, T. Beetz, H.N. Chapman, C. Cui, J.M. Holton, C.J. Jacobsen, J. Kirz, E. Lima, S. Marchesini, H. Miao, D. Sayre, D.A. Shapiro, J.C.H. Spence, and D. Starodub. An assessment of the resolution limitation due to radiation-damage in x-ray diffraction microscopy. *Journal of Electron Spectroscopy and Related Phenomena*, 170(1-3):4–12, 2009.
- [36] A. Jarre. *Hard X-ray Waveguide Optics*. PhD thesis, Universität Göttingen, 2005.
- [37] A. Jarre, C. Fuhse, C. Ollinger, J. Seeger, R. Tucoulou, and T. Salditt. Two-dimensional hard x-ray beam compression by combined focusing and waveguide optics. *Physical Review Letters*, 94(7):074801, 2005.
- [38] A. Jarre, J. Seeger, C. Ollinger, C. Fuhse, C. David, and T. Salditt. X-ray waveguide nanostructures: Design, fabrication, and characterization. *Journal of Applied Physics*, 101(5):054306, 2007.
- [39] P. Kirkpatrick and A.V. Baez. Formation of Optical Images by X-Rays. *Journal of the Optical Society of America*, 38(9):766–773, 1948.
- [40] J. Kirz and C. Jacobsen. The history and future of x-ray microscopy. *Journal of Physics: Conference Series*, 186(1):012001, 2009.
- [41] A. Kohlstedt, S. Kalbfleisch, T. Salditt, M. Reiche, U. Gösele, E. Lima, and P. Willmott. Two-dimensional x-ray waveguides: fabrication by wafer-bonding process and characterization. *Applied Physics A*, 91:7–12, 2008. 10.1007/s00339-007-4374-1.
- [42] P. Kraft, A. Bergamaschi, Ch. Broennimann, R. Dinapoli, E. F. Eikenberry, B. Henrich, I. Johnson, A. Mozzanica, C. M. Schlepütz, P. R. Willmott, and B. Schmitt. Performance of single-photon-counting pilatus detector modules. *Journal of Synchrotron Radiation*, 16(3):368–375, 2009.
- [43] S. P. Krüger. *Optimization of waveguide optics for lensless x-ray imaging*. PhD thesis, Universität Göttingen, 2010.
- [44] S. P. Krüger, K. Giewekemeyer, S. Kalbfleisch, M. Bartels, H. Neubauer, and T. Salditt. Sub-15 nm beam confinement by twocrossed x-ray waveguides. *Optics Express*, 18(13):13492–13501, 2010.

- [45] W. Lauterborn and T. Kurz. *Coherent optics: fundamentals and applications*. Springer, 2003.
- [46] O. Leupold and M. Sprung. Coherence Beamline at PETRA III - Beamline Design Report. Technical report, Deutsches Elektronen-Synchrotron DESY, 2008.
- [47] Youli Li, Roy Beck, Tuo Huang, Myung Chul Choi, and Morito Divinagracia. Scatterless hybrid metal–single-crystal slit for small-angle x-ray scattering and high-resolution x-ray diffraction. *Journal of Applied Crystallography*, 41(6):1134–1139, Dec 2008.
- [48] M. Mell. Phase Contrast Imaging of Lipid Bilayer Model Membranes using Hard X-Rays. Diplomarbeit, Universität Göttingen, 2009.
- [49] J. Miao, P. Charalambous, J. Kirz, and D. Sayre. Extending the methodology of x-ray crystallography to allow imaging of micrometre-sized non-crystalline specimens. *Nature*, 400(6742):342–344, 1999.
- [50] J. Miao, T. Ishikawa, B. Johnson, E.H. Anderson, B. Lai, and K.O. Hodgson. High resolution 3d x-ray diffraction microscopy. *Physical Review Letters*, 89(8):088303, Aug 2002.
- [51] MiCos GmbH. <http://www.micos-online.com/web2/en/1,5,030,upr160a.html>.
- [52] H. Mimura, H. Yumoto, S. Matsuyama, Y. Sano, K. Yamamura, Y. Mori, M. Yabashi, Y. Nishino, K. Tamasaku, T. Ishikawa, and K. Yamauchi. Efficient focusing of hard x rays to 25 nm by a total reflection mirror. *Applied Physics Letters*, 90(5):051903, 2007.
- [53] New Focus. <http://www.newport.com/Vacuum-and-Ultrahigh-Vacuum-Picomotor-Actuators/918309/1033/catalog.aspx>.
- [54] J.D. Nicolas. Röntgenphasenkontrast an Lipid-Wasser-Grenzschichten. Bachelorarbeit, Universität Göttingen, 2011.
- [55] K.A. Nugent. Coherent methods in the x-ray sciences. *Advances in Physics*, 59(1):1–99, 2010.
- [56] C. Olendrowitz. *A micro-scope on intracellular trafficking*. PhD thesis, Universität Göttingen, 2011.

- [57] C. Ollinger. *A waveguide-based lens-less x-ray microscope*. PhD thesis, Universität Göttingen, 2006.
- [58] C. Ollinger, C. Fuhse, S. Kalbfleisch, R. Tucoulou, and T. Salditt. Object localization with 10 nm accuracy by x-ray phase contrast projection imaging. *Applied Physics Letters*, 91(5):051110, 2007.
- [59] M. Osterhoff. *Wave optical simulations for x-ray nano-focusing optics*. PhD thesis, Universität Göttingen, 2012.
- [60] M. Osterhoff and T. Salditt. Real structure effects in x-ray waveguide optics: The influence of interfacial roughness and refractive index profile on the near-field and far-field distribution. *Optics Communications*, 282(16):3250 – 3256, 2009.
- [61] M. Osterhoff and T. Salditt. Coherence filtering of x-ray waveguides: analytical and numerical approach. *New Journal of Physics*, 13(10):103026, 2011.
- [62] M. Osterhoff and T. Salditt. Partially coherent x-ray beam simulations: mirrors and more. *Proc. SPIE Vol. 8141, Advances in Computational Methods for X-Ray Optics II*, 2011.
- [63] R.L. Owen, J.M. Holton, C. Schulze-Briese, and E.F. Garman. Determination of X-ray flux using silicon pin diodes. *Journal of Synchrotron Radiation*, 16(2):143–151, 2009.
- [64] D.M. Paganin. *Coherent X-Ray Optics*. Oxford University Press, 2006.
- [65] S. Panknin, A. Hartmann, and T. Salditt. X-ray propagation in tapered waveguides: Simulation and optimization. *Optics Communications*, 281:2779, 2008.
- [66] F. Pfeiffer, M. Bech, O. Bunk, P. Kraft, E.F. Eikenberry, C. Brönnimann, C. Grünzweig, and C. David. Hard-x-ray dark-field imaging using a grating interferometer. *Nature Materials*, 7:134–137, 2008.
- [67] F. Pfeiffer, O. Bunk, C. Schulze-Briese, A. Diaz, T. Weitkamp, C. David, J.F. van der Veen, I. Vartanyants, and I.K. Robinson. Shearing Interferometer for Quantifying the Coherence of Hard X-Ray Beams. *Physical Review Letters*, 94(16):164801, 2005.



- [68] C. Ponchut, J. Clément, J.M. Rigal, E. Papillon, J. Vallerger, D. LaMarra, and B. Mikulec. Photon-counting x-ray imaging at kilohertz frame rates. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 576(1):109–112, 2007.
- [69] J. Postel. User Datagram Protocol. RFC 768 (Standard), August 1980.
- [70] J.R. R. Janesick. *Scientific Charge-Coupled Devices (SPIE Press Monograph Vol. PM83)*. SPIE Publications, 2001.
- [71] J.M. Rodenburg. Ptychography and related diffractive imaging methods. *Advances in Imaging and Electron Physics*, Volume 150:87–184, 2008.
- [72] Wilhelm Conrad Röntgen. Über eine neue Art von Strahlen. *Aus den Sitzungsberichten der Würzburger Physik.-medic. Gesellschaft Würzburg*, 1895.
- [73] Roper Scientific. LCX:1300 Datasheet. [http://www.princetoninstruments.com/Uploads/Princeton/Documents/Datasheets/Princeton\\_Instruments\\_PI\\_LCX\\_1300\\_Rev\\_C1.pdf](http://www.princetoninstruments.com/Uploads/Princeton/Documents/Datasheets/Princeton_Instruments_PI_LCX_1300_Rev_C1.pdf) (last access 6.2.2012).
- [74] Roper Scientific. SCX:1300 Datasheet. [http://www.princetoninstruments.com/Uploads/Princeton/Documents/Datasheets/Princeton\\_Instruments\\_SCX\\_1300\\_Rev\\_E1.pdf](http://www.princetoninstruments.com/Uploads/Princeton/Documents/Datasheets/Princeton_Instruments_SCX_1300_Rev_E1.pdf) (last access 6.2.2012).
- [75] L. Rossi, P. Fischer, T. Rohe, and N. Wermes. *Pixel Detectors: From Fundamentals to Applications*. Springer, 2006.
- [76] V. Saile. *LIGA and its applications*, volume 7 of *Advanced micro & nanosystems*. Wiley-VCH, Weinheim, 2009.
- [77] T. Salditt, K. Giewekemeyer, C. Fuhse, S.P. Krüger, R. Tucoulou, and P. Cloetens. Projection phase contrast microscopy with a hard x-ray nanofocused beam: Defocus and contrast transfer. *Physical Review B*, 79(18):184112–6, 2009.
- [78] T. Salditt, S.P. Krüger, C. Fuhse, and C. Bahtz. High-transmission planar x-ray waveguides. *Physical Review Letters*, 100(18):184801, 2008.

- [79] C. G. Schroer, P. Boye, J. M. Feldkamp, J. Patommel, A. Schropp, A. Schwab, S. Stephan, M. Burghammer, S. Schoder, and C. Riekel. Coherent x-ray diffraction imaging with nanofocused illumination. *Physical Review Letters*, 101(9):090801, 2008.
- [80] F. Siewert, T. Noll, T. Schlegel, T. Zeschke, and H. Lammert. The Nanometer Optical Component Measuring Machine: a new Sub-nm Topography Measuring Device for X-ray Optics at BESSY. *AIP Conference Proceedings*, 705(1):847–850, 2004.
- [81] A. Snigirev, I. Snigireva, V. Kohn, S. Kuznetsov, and I. Schelokov. On the possibilities of x-ray phase contrast microimaging by coherent high-energy synchrotron radiation. *Review of Scientific Instruments*, 66(12):5486–5492, 1995.
- [82] C. Song, H. Jiang, A. Mancuso, B. Amirbekian, L. Peng, R. Sun, S.S. Shah, Z.H. Zhou, T. Ishikawa, and J. Miao. Quantitative Imaging of Single, Unstained Viruses with Coherent X-Rays. *Physical Review Letters*, 101(15):158101, 2008.
- [83] M. Sprung. Private communication.
- [84] E. Taurel. <http://www.tango-controls.org>.
- [85] IDT Instrument Design Technology. *DES-104 Optical Tables*.
- [86] P. Thibault, M. Dierolf, A. Menzel, O. Bunk, C. David, and F. Pfeiffer. High-resolution scanning x-ray diffraction microscopy. *Science*, 321(5887):379–382, 2008.

# Danksagung

Zum Gelingen dieser Arbeit haben zahlreiche Unterstützer beigetragen, denen ich sehr danke.

Mein ganz besonderer Dank gilt Herrn Prof. Dr. Tim Salditt für die Ermöglichung dieser Arbeit am Institut für Röntgenphysik. Ich bedanke mich für das interessante Thema, die vielfältige Förderung und das entgegengebrachte Vertrauen.

Herrn Prof. Dr. Hans Hofsäss danke ich herzlich für das Koreferat dieser Arbeit.

Bastian Hartmann hat mit seinem Können, technischen Geschick und unermüdlicher Unterstützung wesentlich zu den Aufbauten beigetragen. Hierfür zolle ich ihm höchste Anerkennung und spreche ihm meinen tief empfundenen Dank aus.

Mein weiterer Dank gilt den Mitgliedern des Instituts für Röntgenphysik für die gute Zusammenarbeit und die angenehme Arbeitsatmosphäre. Dem Wellenleiter- und Imagingteam Klaus Giewekemeyer, Markus Osterhoff, Henrike Neubauer, Sven Philip Krüger, Robin Wilke, Matthias Bartels, Marius Priebe und Christian Olendrowitz danke ich für viele aufschlussreiche Diskussionen, Anregungen und umfangreiche Hilfe bei unzähligen Messzeiten.

Bei Jochen Herbst, Thorsten Gronemann und Mike Kanbach bedanke ich mich für ihre hervorragende Unterstützung bei der Probenpräparation. Peter Nieschalk und Carsten Wulf danke ich für die feinmechanischen Arbeiten, ebenso den Mitarbeitern der Zentralwerkstatt und der Elektronikwerkstätten. Jan Goeman spreche ich meine Dank aus für seine vielfältige Unterstützung bei Computerfragen. Christina Emser, Sabine Balder, Kerstin Grüny, Eva Hetzel, Kerstin Pluschke, Kerstin Reinecke und Anita Jaehnke danke ich für ihre Hilfe bei den umfangreichen Verwaltungsangelegenheiten.

Klaus Giewekemeyer, Jörg Hallmann, Henrike Neubauer, Markus Osterhoff, Marius Priebe und Julia Sedlmaier danke ich für die sorgfältige

Durchsicht und Korrektur des Manuskripts, sowie für ihre wertvollen Anmerkungen und Anregungen.

Michael Sprung und seinem Team vom Strahlrohr P10 an PETRA III danke ich herzlich für die fruchtbaren Diskussionen und die umfangreiche Hilfe während des Aufbaus und der Inbetriebnahme des Instruments.

Remi Tucoulou vom Strahlrohr ID22/ID22NI der ESRF danke ich für die hervorragende Unterstützung bei den Messzeiten. Der ESRF danke ich für die Bereitstellung der Strahlzeit.

Das Projekt wurde finanziert vom Bundesministerium für Bildung und Forschung unter den Förderkennziffern 05KS7MGA und 05K10MGA. Weitere finanzielle Unterstützung wurde von Sonderforschungsbereich 755 "Nanoscale Photonic Imaging" und dem Courant Forschungszentrum "Nano-Spectroscopy and X-Ray Imaging" gewährt.

Meinen Eltern möchte ich für ihre umfassende und vorbehaltlose Unterstützung während meines gesamten Studiums herzlich danken. Mein liebster Dank gilt Gudrun Lotze.

# Lebenslauf

SEBASTIAN KALBFLEISCH

geboren am 23. August 1980 in Schwalmstadt-Ziegenhain,  
Staatsangehörigkeit: deutsch

## Schulbildung

1987 – 1990 Grundschule Willingshausen  
1990 – 2000 Schwalmshule, Schwalmstadt-Treysa,  
abgeschlossen mit der Allgemeinen Hochschulreife

## Studium

2000 – 2012 Physik-Studium an der Georg-August-Universität  
Göttingen  
2006 Diplom in Physik, Titel der Arbeit: “Fresnelsche  
Reflexionszonenplatten für harte Röntgenstrahlung”,  
Institut für Röntgenphysik, Georg-August-Universität  
Göttingen  
seit 2006 wissenschaftlicher Mitarbeiter am Institut für  
Röntgenphysik, Georg-August-Universität Göttingen

