Shaping Light, Measuring Space:

Effects of Beam Jitter and Detector Misalignment on QPD Response to Gaussian and Tophat Beams in LISA

Martijn Moorlag

MSc Applied Physics
Physics for Instrumentation
TU Delft

October 2025
Supervisors: Ernst-Jan Buis
Timesh Mistry
Niels van Bakel

TU Delft

Nikhef

Abstract

This work quantifies how detector misalignment (tip, tilt, clocking) and beam jitter (rotation and lateral offset) couple into the standard quadrant photodiode (QPD) observables used in LISA-like heterodyne interferometry: Differential Wavefront Sensing (DWS), Longitudinal Pathlength Sensing (LPS), Differential Power Sensing (DPS), and fringe contrast. Measured QPD responsivity maps (DC and RF) were acquired by scanning a focused, modulated 1064,nm beam over the detector; an auto-clocking routine fitted the gap lines and found a small in-plane misalignment of $\sim 0.293^\circ$. Simulations in IfoCAD used both ideal and measured maps, a pivot-at-QPD convention to separate angular and lateral effects, and two numerical integration schemes: GSL (Gauss–Kronrod 1D) and DCUHRE (adaptive 2D with smoothed apertures). Non-Gaussian, flat-top beams were generated by three methods: Gaussian Beam Decomposition (GBDC), Mode Expansion Method (MEM), and Numerical Mode Fitting (NMF). The study also assessed how detector geometry (radius and 20/40μm gap widths) scales the couplings. Limitations include sparse parameter sampling for computational feasibility, a fixed beam origin that noticeably influences absolute coefficient values, responsivity mapping at 10V bias (reliable for DC but suboptimal for RF that prefers near-full depletion at ~ 24V) for the gap/size study, and a 20μm scan step that under-resolves gap edges.

DWS was modeled as the sum of intended terms (rotation$\parallel$ and offset$\parallel$) and cross terms (rotation$\perp$ and offset$\perp$). Sequential fits delivered $R^2 \approx 0.999996$ overall for GSL. For Gaussian beams with GSL, the intended rotation-to-DWS gain $I^{\mathrm{rot}}_{\mathrm{DWS}}$ is $\sim (2308 \pm 1)\mathrm{rad/rad}$; RF cases are slightly higher: up to $(2346 \pm 2)\mathrm{rad/rad}$ at 30MHz and $(2340 \pm 2)\mathrm{rad/rad}$ at 3MHz. The ideal map yields $\sim (2382 \pm 1)\mathrm{rad/rad}$. Tophat beams show lower intended gains: MEM and truncated NMF cluster near $1660, \mathrm{rad/rad}$ (about -30% vs. Gaussian), and the non-truncated NMF can be even lower (near $1098\mathrm{rad/rad}$; about -52%). For the clocking-induced cross term $e^{\mathrm{clock}}_{\mathrm{DWS}}$ is consistent across Gaussian, MEM, and truncated NMF, with ideal values exceeding real ones by roughly 40–$60, \mathrm{rad}^{-1}$ (e.g., Gaussian: 2386 vs. $2325, \mathrm{rad}^{-1}$; MEM: 2307 vs. $2268, \mathrm{rad}^{-1}$; truncated NMF: 2304 vs. $2263, \mathrm{rad}^{-1}$), while non-truncated NMF is $\sim 20\%$ higher than Gaussian (ideal 2826, real $2781, \mathrm{rad}^{-1}$). GBDC returns much larger but noisy values ($\sim 4.8{\times}10^3 \pm 10^3, \mathrm{rad}^{-1}$). Frequency effects on the clocking–rotation coupling are negligible under GSL and remain within uncertainty under DCUHRE. Tip–tilt cross coupling is well captured by a bilinear model $e^{\mathrm{tip/tilt}}_{\mathrm{DWS}} \times \mathrm{tip} \times \mathrm{tilt}$ with $e^{\mathrm{tip/tilt}}_{\mathrm{DWS}} \approx (1.592 \pm 0.00014) \times 10^{-3}\mathrm{rad}^{-2}$ ($R^2 = 0.9986$); neither tip nor tilt alone introduces extra cross coupling, but their product does. Setting QPD tilt to 0° versus 2° leaves intended and cross DWS slopes essentially unchanged; the main difference is numerical: GSL gives compact, repeatable slopes, whereas DCUHRE inflates variability and can be nondeterministic even for ideal maps.

LPS was modeled as bilinear angle–offset terms in both axes, quadratic offset terms, and small linear angle errors. Layered fits converged to $R^2 \approx 0.999998$ (real) and $R^2 = 1$ (ideal). The first two bilinear terms dominate. Across beam families, truncated beams yield $k_{\mathrm{LPS}} \approx 0.70\,\mathrm{pm}/(\mu\mathrm{m}\mu\mathrm{rad})$, nearly 20% below the Gaussian references (~ 0.83 with GSL, ~ 0.81 with DCUHRE). Non-truncated beams cluster near 0.87, slightly above Gaussian. Under GSL, the spread could not be attributed to a single parameter; under DCUHRE, the spread correlates with tilt but without a reliable trend. For DPS, the intended offset term fits with $R^2 \approx 0.99998$ and the orthogonal error adds only $\Delta R^2 \approx 0.16$. Contrast follows a quadratic loss model in angle and offset; layered fits achieved $R^2 \approx 0.99994$.

A size and gap-size study (Gaussian beams) used laboratory maps for detectors with radii 500, 750, and 1000μm and gaps 20 and 40μm. Power-law fits versus radius $y = r^a + b$ show intended couplings increase with size: $a \approx 0.95$ (DWS rotation), $a \approx 0.38$ (DWS offset), and $a \approx 0.91$ (DPS). Contrast terms scale inversely ($a \approx -1$). LPS families decrease with size (angular: $a \approx -0.15$, quadratic: $a \approx -1$), implying reduced longitudinal sensitivity for larger detectors but slightly higher contrast loss. Cross-coupling coefficients fluctuate around zero without systematic size or gap dependence. Real maps show lower sensitivity than ideal: intended DWS/DPS are reduced by ~ 4–7% (larger gaps yield bigger shortfalls), LPS cross terms by $\sim 2\%$, and contrast by ~ 1–2% at

20 μm gaps and up to ~ 5–6% at 40μm gaps.

Integration method is decisive. GSL consistently delivers stable, reproducible coefficients across Gaussian, MEM, and NMF; DCUHRE fails on LPS, inflates DWS variability, and in several cases cannot reproduce expected results even for ideal QPDs. GBDC is incompatible with GSL in this framework and, under DCUHRE, returns noisy and sometimes unusable DWS/LPS fits except at tip=tilt$= 0$, limiting its utility for realistic sweeps. MEM and NMF are both viable; MEM's hard aperture limits profile tunability and can introduce shoulders, while NMF is flexible (supports soft or sharp edges), fast to build, and numerically stable. For future IfoCAD studies aimed at LISA-level error budgeting, GSL should be used as the integration method, and NMF is recommended as the preferred tophat-beam construction, with MEM as a consistent alternative when a fixed hard aperture is desired.

Acknowledgements

This project was conducted as part of the collaboration between TU Delft and Nikhef within the Detector R&D group.

I would like to thank Ernst-Jan for first sparking my interest in Nikhef during his lectures on elementary particles, where he subtly acted as an ambassador for the institute and its many research projects. His enthusiasm for physics, and for the LISA mission in particular, was contagious. Throughout this project his excitement and encouragement often came at exactly the right time—especially during long stretches of analysis when it was easy to lose sight of the bigger picture. Conversations with him always put things in perspective and reminded me that I was indeed working on something meaningful and extraordinary.

I am also grateful to Niels, who made it possible for me to join the Detector R&D group at Nikhef, and shared his enthusiasm for the LISA mission early on. His support in the early stages and our occasional conversations about geopolitics or bike-packing were a welcome break form the technical work.

Finally, a special thanks to Timesh, my daily supervisor and guide throughout this work. His deep understanding of the LISA mission and its intricate physics made every discussion valuable. Whenever I was stuck or puzzled by a confusing plot, he would immediately dive into the physics behind it, linking details I hadn't even considered. His explanations always led to insights that later proved essential. Beyond his knowledge, I appreciated his patience, humor, and the relaxed yet focused atmosphere he brought to my project.

List of Abbreviations and Symbols

Abbreviations

DC	Direct Current
DCUHRE	adaptive 2D integration of optical fields (Deterministic Cubature Integration Routine)
DPS	Differential Power Sensing
DWS	Differential Wavefront Sensing
ESA	European Space Agency
GBDC	Gaussian Beam Decomposition
GSL	adaptive 1D quadrature integration of optical fields (GNU Scientific Library)
IfoCAD	Interferometer simulation framework
LISA	Laser Interferometer Space Antenna
LPS	Longitudinal Phase / Pathlength Signal
MEM	Mode Expansion Method
NMF	Numerical Mode Fitting
QPD	Quadrant Photodiode
RF	Radio Frequency
TTL	Tilt-to-Length coupling
trunc	Truncated; hard-edged tophat beam (aperture-limited)
notrunc	Non-Truncated; flat-top Gaussian (soft-edged)

Symbols

HG_{mn}	Hermite–Gaussian mode
N	number of grid points in MSE (tophat context)
N_{modes}	number of spatial modes / beamlets
θ_{clock}	QPD clocking angle about z
$x_{\mathrm{off}}, y_{\mathrm{off}}$	beam lateral offsets on the QPD
α, β	beam rotations about x and y
tip, tilt	QPD rotations about x / y axes
$\mathrm{rotation}_{\parallel,\perp}$	components parallel / orthogonal to sensed axis
$\mathrm{offset}_{\parallel,\perp}$	components parallel / orthogonal to sensed axis
$I_{\mathrm{DWS}}^{\mathrm{rot}}$	intended beam-rotation-to-DWS coupling coefficient
$I_{\mathrm{DWS}}^{\mathrm{off}}$	intended beam-offset-to-DWS coupling coefficient
$I_{\mathrm{DPS}}, e_{\mathrm{DPS}}$	intended DPS coupling; DPS cross-coupling
$e_{\mathrm{DWS}}^{\mathrm{rot}}$	cross-coupled beam-rotation-to-DWS coefficient
$e_{\mathrm{DWS}}^{\mathrm{off}}$	cross-coupled beam-offset-to-DWS coefficient
$e_{\mathrm{DWS}}^{\mathrm{clock}}$	clocking-induced cross-coupling coefficient
$e_{\mathrm{DWS}}^{\mathrm{tip/tilt}}$	tip–tilt-induced cross-coupling coefficient
$C_{(\mathrm{tip,tilt})}$	tip–tilt-induced beam–rotation–to–DWS coupling
$k_{\mathrm{LPS}}^{\alpha,\beta,x,y}$	LPS coupling coefficients
$k_{c}^{\alpha,\beta,x,y}$	contrast-loss coefficients
$R^2, \Delta R^2$	coefficient of determination; incremental improvement

Contents

"We are very, very small, but we are profoundly capable of very, very big things"

— Stephen Hawking

1. Introduction

This thesis presents simulations and analyses of wavefront sensing for the Laser Interferometer Space Antenna (LISA). The work focuses on the response of quadrant photodiodes (QPDs) to heterodyne signals and examines how detector properties and beam distortions affect sensing performance. The aim is to understand how these effects influence the stability and accuracy of LISA's interferometric measurements during real operation in space.

In short, the performance of LISA's detectors will be investigated through lab measurements and simulations. More specifically, the study investigates the performance of the Quadrant Photo Diodes (epitaxial stacks of the Indium-Gallium-Arsenide diodes) by simulating heterodyne signals and different beam shapes. The simulations and models are used to gauge the impact on sensing techniques, i.e. Differential Wavefront Sensing and Longitudinal Path Sensing. All simulations are done in IfoCAD. The goal of the study is to acquire a better understanding of the stability of the LISA detectors during real operation in space.

1.1. On the Shoulders of Giants

In 1687, Isaac Newton described gravity as a universal force of nature, and later specified it with his famous inverse-square law [1]. It is still used today as a good approximation for a variety of physics applications. However, after a few centuries, Albert Einstein revealed a deeper reality [2]. In his Theory of General Relativity, he postulated the following: gravity is not a force or a force field that propagates through space. Instead, gravity is the consequence of the deformation of space-time. The geometry of spacetime is inherently related to the arrangement of matter in it, and its relation is described by the Einstein Field Equation:

$$G_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu} \tag{1}$$

Here, $G_{\mu\nu}$ is the Einstein Tensor, which describes how matter and energy cause the curvature of space-time, in 3 spatial dimensions and 1 time dimension. Λ is the cosmological constant; it describes the energy density of space and is related to the expansion of space. $g_{\mu\nu}$ is the metric tensor and describes the curvature of space-time, G is the gravitational constant, the same as in Newton's law for gravity. Lastly, the mass, energy, momentum, pressure, and tension are all encapsulated in the stress-energy tensor $T_{\mu\nu}$.

Einstein opened possibilities to study large mechanics in the universe. His field equation directly describes the existence of black holes [3][4], gravitational lensing [5][6], and last but not least; it postulates the existence of Gravitational Waves (GWs) [7].

1.1.1. Gravitational Waves

According to the Einstein Field Equation in equation (1), it is predicted that accelerated matter will emit gravitational quadrupole radiation [7]. Electromagnetic radiation arises from the acceleration of charged particles, whereas gravitational radiation is generated by accelerated mass or energy. Due to momentum conservation, the leading order of gravitational radiation is quadrupole in nature, rather than dipole. This radiation, more generally labeled as Gravitational Waves (GWs), is expected to travel through the universe at the speed of light. GWs induce stretching and compression of spacetime, perpendicular to their direction of motion. This movement is determined by the polarization of the GWs, which is typically defined as plus ($+$) and cross ($\times$) polarization (or a combination of the two). Figure 1 portrays this concept, where a ring is influenced by GW coming from the viewer's perspective. The magnitude of a GW is measured by the dimensionless metric-strain amplitude $h \approx \Delta L/L$, perpendicular to their direction of propagation. This strain h is detectable as changes in distances L. Systems like merging binary neutron stars generate 10^{28} Watt of energy. However, these

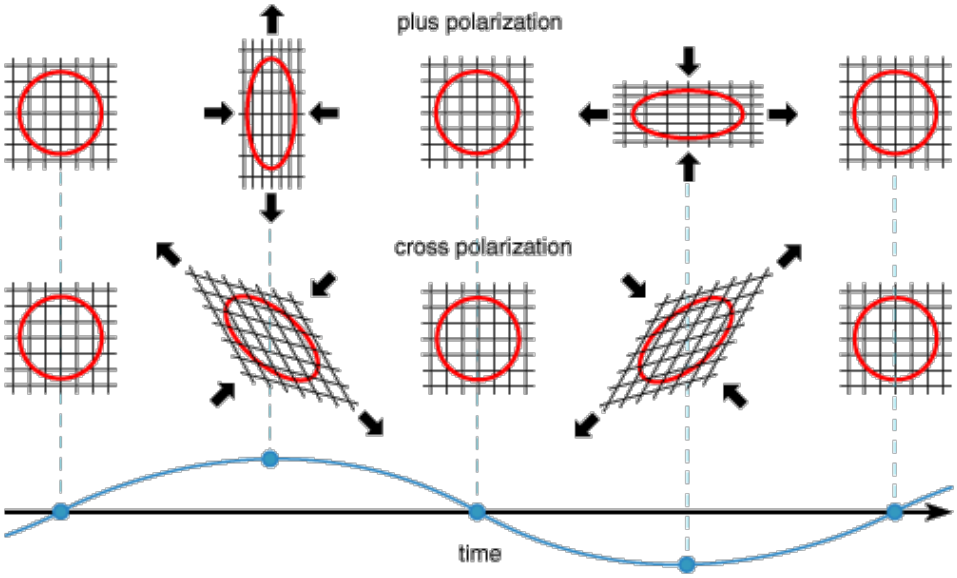


Figure 1: Schematic depiction of how different polarizations for Gravitational Waves change distances as the wave oscillates over time. [9]

events would generate GW with amplitudes of the order 10^{-21}: the distance from the earth to the sun would change by a diameter of an atom [8]. Einstein himself thought that GWs would never be measured.

1.1.2. Gravitational Waves Detection

Within the century after Einstein's proposition, multiple direct and indirect observations of gravitational waves took place. Indirect observations include the Hulse-Taylor binary pulsar [10] which was awarded the Nobel Prize in 1993, and the detection of B-mode Polarization at the BICEP2 telescope [11]. Detectors that can measure GWs directly come in a variety of ways:

Resonant Mass Detectors, (example: AURIGA [12]) were first built by Weber [13], and will oscillate at their resonant frequency when excited by GWs of specific frequencies.

Pulsar Timing Arrays (example: European Pulsar Timing Array [14]) measure changes in milliseconds of pulsar signals.

Interferometric Gravitational Wave Detectors measure the relative distance between two test masses by the phase shift of light in large interferometers. An interferometer, an instrument first proposed by Michelson and Morley [15], splits a laser via a beam splitter into two perpendicular paths (arms) and reflects them back. The returning beams are recombined, creating interference patterns. Gravitational waves will change the distance of the laser arms of the detector, as shown in Figure 2. This change in distance will result in a phase shift in each of the lasers.

Heterodyne detection employs two frequencies and measures the phase difference between the signals. In heterodyne interferometers, the interference pattern varies over time, even when the arm lengths remain unchanged [8]. The detected signal is a harmonic function of time but changes when a GW passes through the detector.

In 2015, the first *direct* observation of gravitational waves was detected using an interferometric gravitational wave detector [16]. This technique therefor seems most promising, and several detectors are operational today, including LIGO (USA) [16], VIRGO (Italy) [17], GEO600 (Germany) [18], and KAGRA (Japan) [18].

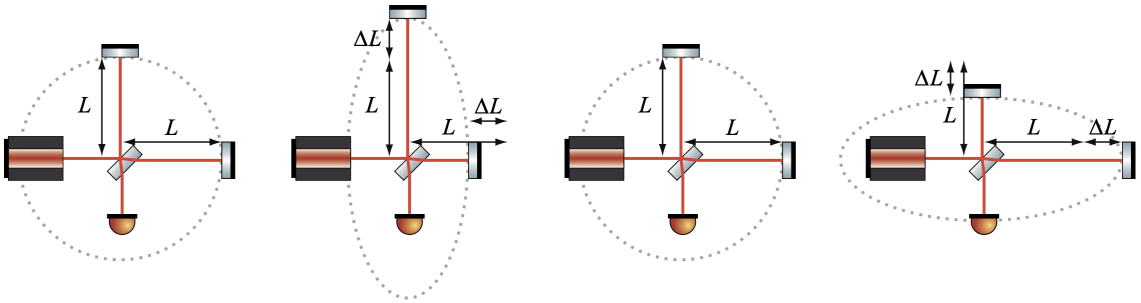


Figure 2: A Gravitational Wave passing through a Michelson interferometric detector will change the path lengths of the laser. [8]

1.2. LISA

In 2035, the European Space Agency (ESA) plans to launch the space-based gravitational wave antenna called the Laser Interferometer Space Antenna (LISA) [19]. The mission consists of three spacecraft. The arm length between any two spacecraft is about 2.5 million km (approximately 8.3 light-seconds). The distances between the antennas are precisely monitored in order to detect gravitational waves. To minimize distance-alterations by other non-gravitational forces, e.g. solar wind, the antennas are constructed as a drag-free satellite [20][21], where a test mass floats in free-fall inside the spacecraft, while the surrounding spacecraft absorbs all non-gravitational forces. In other words, the spacecraft actively follows the free-falling test mass so that its trajectory is unaffected.

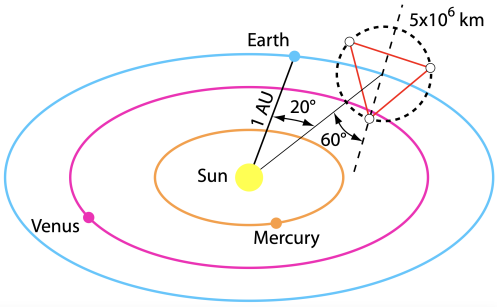


Figure 3: Three satellites in an equilateral triangle, 20° ahead of Earth, $\sim 50 \times 10^6$,km away; plane inclined $\sim 1°$ to ecliptic and rotated 60°. ecliptic plane, [8].

Laser links between the spacecraft continuously measure the separation fluctuations of these test masses (46 mm gold-platinum cubes) with picometer-level precision [22]. A passing gravitational wave will stretch and squeeze the triangle's arms, imprinting a tiny phase shift onto the laser beams. Figure 3 shows a schematic of the LISA orbit. The three satellites are arranged in an equilateral triangle. The center follows a heliocentric orbit with approximately 20° between Earth and the LISA center, which is about 50×10^6 km from the Earth. Their orbits are slightly inclined (1°) with respect to the ecliptic plane such that the plane of the LISA spacecrafts (marked with the dotted line) is shifted 60° with respect to the ecliptic [23]. Each arm length is so large that the finite speed of light must be accounted for: the transmitted laser from one spacecraft must be pointed slightly ahead of the apparent position of the distant spacecraft using a Point Ahead Angle Mechanism [24], anticipating where the receiver will be after the 8 seconds light travel time. LISA is designed to detect a signal from about 0.1 mHz to 0.1 Hz. However, the constellation is not rigidly controlled; natural orbital motion causes slow inter-spacecraft drifts of a few m/s, leading to Doppler shifts of the measured MHz beatnote in the received laser frequencies over a year [25].

1.2.1. Gravitational Wave Sources and Signals for LISA

LISA (Laser Interferometer Space Antenna) will operate in the millihertz gravitational-wave band ($\sim 10^{-4} - 10^{-1}$ Hz), bridging the gap between nanohertz pulsar-timing arrays and the 10–1000 Hz band of ground-based detectors [26]. LISA's strongest signals will come from coalescing massive black holes in the mass range $\sim 10^3$–$10^7\ M_\odot$. A pair of $\sim 10^6\ M_\odot$ black holes at $z \approx 1$ radiates in the 10^{-3}–10^{-1} Hz band during the final hours of inspiral, right in LISA's sensitivity window [26]. This

enables tests of general relativity in the strong-field regime. Dozens of signals are expected during the mission, providing insight into black hole and galaxy growth, with multi-messenger follow-up probing their environments [27][28].

Extreme Mass Ratio Inspirals (EMRIs): involve a stellar-mass compact object ($\sim 10\,M_\odot$) slowly inspiraling into a 10^5–$10^7\,M_\odot$ black hole. They produce complex, harmonic-rich waveforms at frequencies of a few mHz, near LISA's most sensitive band [29]. Signals last months to years, allowing detailed mapping of spacetime around the central object. By resolving multiple harmonics, LISA can measure mass and spin and test fundamental predictions of general relativity [26][30].

Galactic Ultracompact Binaries, e.g. compact white dwarf, neutron star, or black hole binaries in the Milky Way will be numerous LISA sources. With orbital periods of minutes to hours, they emit nearly monochromatic waves in the 0.1–10 mHz band. LISA will resolve $\sim 10^4$ of the brightest systems, while millions more form a confusion foreground dominating noise between 0.4–2 mHz [29]. These populations provide insight into binary evolution and Galactic structure through both resolved and unresolved signals [26].

Stellar-Mass Black Hole Binaries (Early Inspirals) are binaries of tens of $M_\odot$. They can emit in LISA's band years before their merger in the LIGO/Virgo range. At orbital periods of minutes, they appear in the $\sim$ 0.01–0.1 Hz regime with slowly oscillating signals [26]. LISA observations will precede ground-based detections, enabling multi-band studies and constraining sky localizations for multi-messenger astronomy.

Stochastic Backgrounds and Other Sources: LISA will also measure stochastic backgrounds, including the Galactic binary foreground and potential extragalactic contributions [31][26]. Unresolved EMRIs or compact binaries may form a diffuse background in the mHz range [32]. Cosmological processes like inflation, cosmic strings, or phase transitions could generate relic signals at these frequencies. A detection would open a window onto physics at TeV energy scales and early-universe dynamics. At the same time, LISA could also reveal unexpected transient bursts from new or exotic sources.

1.2.2. Heterodyne readout on the Optical Bench

Unlike ground-based Michelson interferometers such as LIGO and Virgo, which operate with highly stabilized lasers in fixed kilometer-scale arms, LISA performs inter-spacecraft interferometry. At each spacecraft, an outgoing laser (measurement beam) is sent to a distant spacecraft while a faint incoming laser (received measurement beam) from that distant partner is received. As depicted in Figure 4 a portion of the local laser (reference beam) is superposed with the incoming beam on a detector, producing an interference signal. The two lasers are at slightly different frequencies, and their interference yields a heterodyne beat note [25]. The phase of this beat note carries the length information: any change in the light travel time (optical path) between the spacecrafts, such as a passing gravitational wave, will manifest as a phase shift of the heterodyne signal. Each of the six one-way links in the constellation produces such a beat.

A major challenge is that laser frequency noise is many orders of magnitude larger than the expected gravitational-wave phase shifts. LISA addresses this by Time Delay Interferometry (TDI) [33, 34, 35]. In essence, TDI algorithmically combines the phase data from all six one-way links, time-shifting them to create interference signals equivalent to a Michelson interferometer with equal-arm lengths . This cancels out the common laser frequency noise to first order, since noise fluctuations affect both arms equally after appropriate delays [25]. TDI thus synthesizes a virtual equal-arm interferometer, recovering the gravitational wave signal while suppressing laser noise.

1.2.3. LISA's Optical Bench

Each LISA spacecraft carries two nominally identical optical assemblies (often called MOSA, Movable Optical Subassembly) pointing toward the other two spacecraft. Next to the two laser links (one

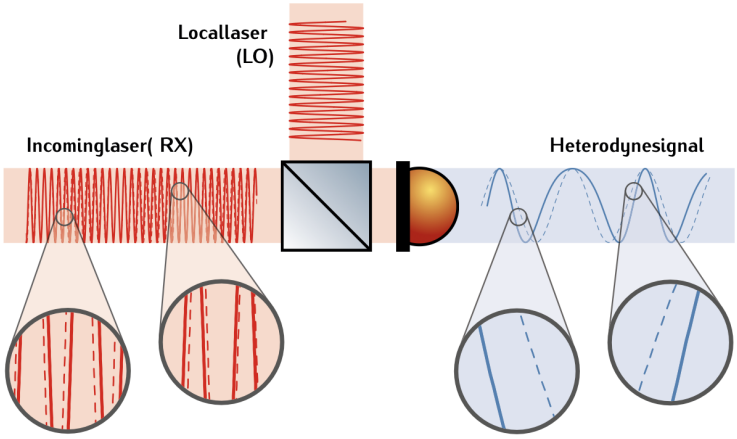


Figure 4: Heterodyne interferometry: the incoming measurement beam interferes with the local laser on a photodiode, producing a beat note whose phase encodes displacement information [36]

per opposite arm) on board each spacecraft, a reference interferometer on the optical bench is present that compares the frequencies of the two lasers. This measurement ties the arms together and provides input for TDI. A test mass interferometer monitors the position of each test mass relative to the optical bench by reflecting a beam off the mass and interfering it with the reference beam. Together with the science links and reference interferometer, these readouts separate gravitational-wave signals from spacecraft jitter, laser noise, and clock noise [25].

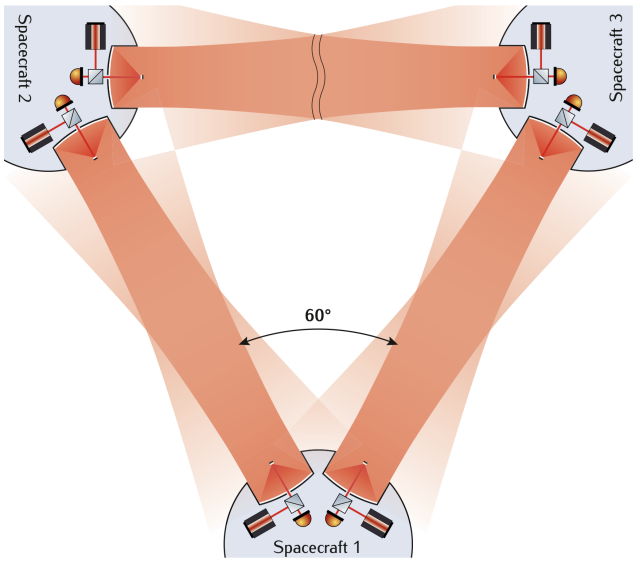


Figure 5: The LISA constellation of three identical spacecraft, each transmitting and receiving two laser beams and hosting two onboard interferometers [36]

A 30 cm off-axis optical telescope transmits and receives laser light, sending out 1 W of optical power at 1064 nm and detecting approximately 370 pW at the photoreceiver [36]. The photoreceiver (i.e. the detector) is a quadrant photodiode that is described in more detail in the next section 1.3. It converts the weak incoming beam into an electrical heterodyne signal, which is digitized by the phasemeter. The phasemeter's optimal range is between 3 and 30 MHz. The telescope is an afocal two-mirror design (magnification 135×) that takes a 2.2 mm diameter beam from the optical bench and expands it to a 0.3 m outbound beam, which greatly reduces diffraction over 2.5 million km. The optical bench is (on the order of 0.5–0.6 m in size) made of a low thermal-expansion glass-ceramic (e.g. Zerodur) [37]. The bench uses flexure mounts and stable bonding to handle small expansions or vibrations without disturbing the interferometer [37].

LISA's metrology is extremely sensitive, so even nanometer-scale distortions can influence the measurement. The mounts keep the detectors aligned to the optical axis, but residual effects such as thermal warping can still cause the incoming beam to move. To monitor this, the optical bench uses quadrant photodiodes (QPDs). These not only record the heterodyne phase but also provide information on the alignment of the beam.

1.3. Quadrant Photo Diodes

The laser sensors that will be used in the LISA mission are called Quadrant Photo Diodes (QPDs) or Quadrant Photo Receivers (QPR's). These are being tested by Nikhef in Amsterdam. QPDs will measure the incoming heterodyne signal and can determine the position of an incident optical beam [38].

1.3.1. Device Structure

A photodiode converts incident light into an electrical current via the internal photoelectric effect. In a p–i–n photodiode, incident photons are absorbed in a depletion (intrinsic) region, generating electron-hole pairs. When operated under a reverse bias, these pairs are swept by the internal electric field to produce a photocurrent. A quadrant photodiode implements four photodiode segments, with a small gap isolating each quadrant [39].

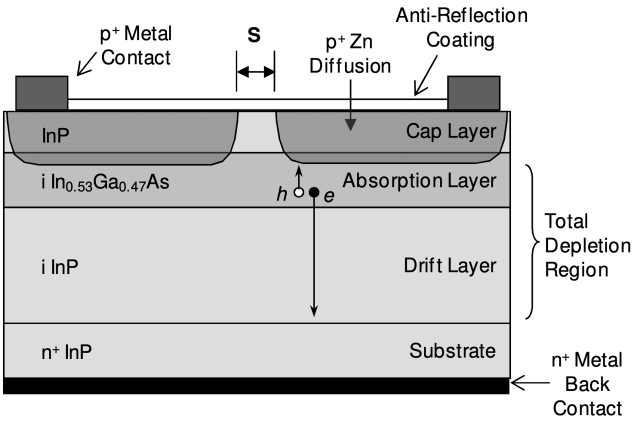


Figure 6: Structure and band diagram of an InGaAs/InP p–i–n photodiode. Layers include anti-reflection coating, p^+ InP cap, $In_{0.53}Ga_{0.47}As$ absorber, InP drift, and n^+ InP substrate with metal contacts [38].

The device structure consists of several functional layers, as depicted in Figure 6. On the surface, an anti-reflection coating reduces reflection losses. Beneath it lies The InP cap layer, which zinc diffused to make the metal electrical connection. The layer absorbs some light but it is highly doped so that the recombination time is short and a current is not measured.

Below the cap is the absorption region, about 2 µm thick, where the incoming photons are absorbed to generate carriers. For near-infrared operation around 1064 nm, InGaAs is the semiconductor of choice for the photodiode material with a bandgap of about 0.75 eV [40]. This bandgap corresponds to a cutoff wavelength of about 1.7 µm, encompassing 1064 nm, while exhibiting high quantum efficiency in the NIR (often >80–90%, and up to 99% with anti-reflection coatings) [40][41].

Under the absorption region lies a thick ($\approx$ 17 µm) InP drift layer. When the diode is biased to full depletion (around 024 V), this region establishes the electric field that drives carrier collection from the absorption layer. Capacitance is the main noise source in the QPD. It depends on the depletion region between the p and n layers and increases with diode area and doping level. To reduce it, the device is made thick, and both the absorption and drift regions use ultra-low doping due to the large area of the QPD.

At the bottom, the n^+ InP substrate provides mechanical support and forms the back electrical contact for the reverse bias. The top contact is used to read out the photocurrent.

The photodiode is one device with a top cap layer divided into four segments, each collecting the photocurrent from its part of the active area. A phase meter then processes these currents to extract phase information for the interferometer readout. Figure 7 highlights the position and direction sensing capacity of the QPDs.

Ideally, all quadrants are identical and perform uniformly. In reality, they will produce slightly different current outputs, causing errors. In addition, production imperfections will impose errors in QPDs' orientation (pitch, yaw, and roll). All these errors will accumulate in the noise produced by the QPDs. A key objective of this thesis is to gain insight into how these parameters influence the noise characteristics of the QPDs.

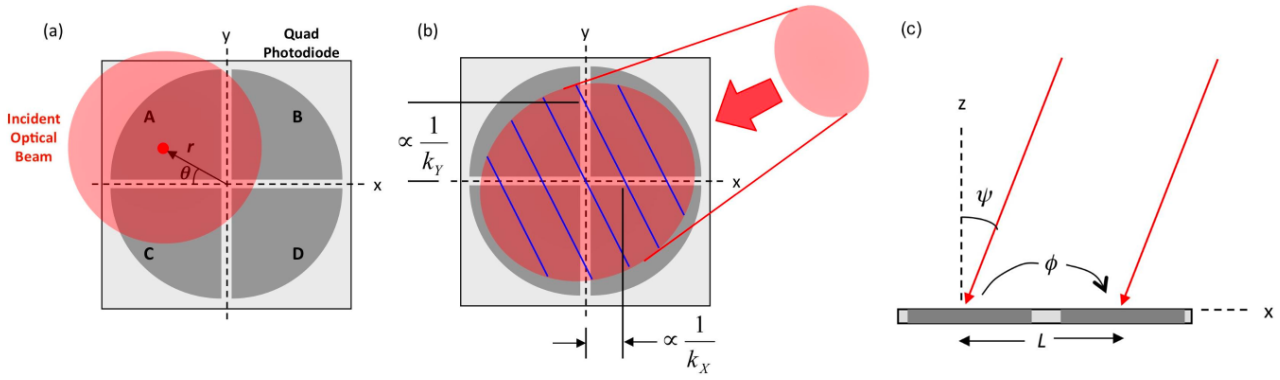


Figure 7: Schematic of the position and direction sensing of a Quadrant Photo Diode. (a) The displacement of an incident optical beam from the center of the device. The QPD is shown in grey with the quadrants labeled A,B,C,D and the optical beam in red. (b) The directional sensing of the QPD, with the isophase lines displayed blue. (c) An off-normal angle of incidence Ψ along the x-axis of the QPD of two optical beam paths with phase difference ϕ. [38]

1.3.2. Simulated Quadrant Photodiode Interferometric Readout

As described in section 1.2.2; for each of the six links of LISA, a stable *reference beam* and the long-arm *measurement beam* are superposed on a photodetector. We model the measurement (m) and reference (r) beams by their complex electric fields at the detector surface:

$$E_b(t) = A_b \exp\!\big[-i\Phi_b + i\omega_b t\big], \qquad b \in \{m, r\}, \tag{2}$$

where A_b is the field amplitude, Φ_b the optical phase, and ω_b the angular frequency. When both beams superpose on the detector, the local intensity is

$$I \simeq \big|E_m + E_r\big|^2, \tag{3}$$

By inserting Eq. (2) into Eq. (3) and expanding, the intensity can be written explicitly as

$$\begin{aligned} I(t) &= \frac{1}{2Z}\left(A_r^2 + A_m^2\right)\left[1 + \frac{2A_m A_r}{A_m^2 + A_r^2}\cos\!\big(\omega_{\rm het}t - \Delta\Phi\big)\right] \\ &= \overline{I}\left[1 + c_I \cos\!\big(\omega_{\rm het}t - \Delta\Phi\big)\right], \end{aligned} \tag{4}$$

Here, $\overline{I}$ denotes the time-averaged local intensity, c_I represents the local fringe visibility (contrast), the phase term $\Delta\Phi \equiv \Phi_m - \Phi_r$ accounts for the optical path length difference between the two interfering beams, and $\omega_{\rm het} \equiv \omega_m - \omega_r$ is the heterodyne frequency [39]. The parameter Z is the medium impedance (included here for completeness; it cancels in phase/contrast demodulation). Defining the local mean intensity and local fringe visibility (contrast),

$$\bar{I} = A_m^2 + A_r^2, \qquad c_I = \frac{2A_m A_r}{A_m^2 + A_r^2}. \tag{5}$$

The power on a given photodiode (PD) is the surface integral,

$$P(t) = \int_{\rm PD} dS\, I \quad = \overline{P}\big[1 + c\cos\!\big(\omega_{\rm het}t - \phi\big)\big], \tag{6}$$

where the last equality yields the generic heterodyne form, with $\overline{P}$ the mean power on the quadrant surface, c the contrast of the whole PD, and ϕ the demodulated phase. Equation (6) is valid for a single-element PD and, applied per quadrant, for a QPD. The PD converts optical power into photocurrent,

$$J(t) = \frac{\eta e}{\hbar\omega}\, P(t) = \rho\, P(t), \tag{7}$$

with quantum efficiency η, elementary charge e, photon angular frequency ω, Planck constant $\hbar$, and responsivity ρ. Therefore, mapping the QPD responsivity lets us compute (simulate) the segment photocurrents from the incident power distribution.

Phase and contrast by quadrature projection. By integrating the power we found in equation 6 with cosine and sine at the heterodyne frequency, we get two time-independent terms [42]:

$$C^Q = \frac{1}{\pi}\int_0^{2\pi} P(\omega_{\text{het}}t)\,\cos(\omega_{\text{het}}t)\,d(\omega_{\text{het}}t) = \overline{P}c\cos\phi, \tag{8}$$

$$C^I = \frac{1}{\pi}\int_0^{2\pi} P(\omega_{\text{het}}t)\,\sin(\omega_{\text{het}}t)\,d(\omega_{\text{het}}t) = \overline{P}c\sin\phi. \tag{9}$$

Here we use the mean power defined as

$$\overline{P} = \frac{1}{2\pi}\int_0^{2\pi} d(\omega_{\text{het}}t)P. \tag{10}$$

C^Q is the in-phase component (cosine projection) and C^I is the quadrature component (sine projection) [42]. Together they form the complex amplitude, defined as

$$C = C^Q + iC^I = \frac{1}{\pi}\int_0^{2\pi} d(\omega_{\text{het}}t)\,P(t)\,e^{-i\omega_{\text{het}}t} = \overline{P}\,c\,e^{-i\phi}. \tag{11}$$

so that the contrast is defined by

$$c = \frac{\sqrt{(C^Q)^2 + (C^I)^2}}{\overline{P}} = \frac{|C|}{\overline{P}}, \tag{12}$$

and then the phase can be computed with

$$\phi = \arctan\Big(\frac{C^I}{C^Q}\Big) = \arg(C). \tag{13}$$

Using the terms derived above, we can now describe how different signals processed by the QPD.

Phase information . Let A, B, C, D denote the QPD quadrants, and are labeled as viewed along beam propagation: A: top-left, B: top-right, C: bottom-left, D: bottom-right). The global phase on the full QPD is obtained by summing complex amplitudes,

$$\phi = -\arg\big(C_A + C_B + C_C + C_D\big), \tag{14}$$

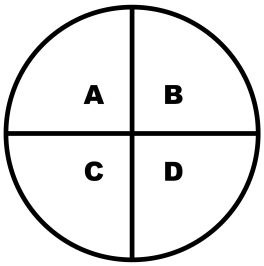


Figure 8: QPD-labeling

When the two beams (measurement and reference) interfere on a QPD, the light is not a uniform DC signal but a time-varying fringe pattern at the heterodyne frequency (MHz), (figure 4 from section 1.2.2). If the incoming wavefront is perfectly parallel to the local reference beam, the fringe pattern is aligned across all quadrants and the oscillations are in phase; all four currents rise and fall together.

Differential Power Sensing (DPS). If the beam is off-center but still parallel, the oscillations remain phase-aligned, but the average power per quadrant is unbalanced: the side with more light produces a larger DC current than the opposite side. This effect is used for **Differential Power Sensing** (DPS), where left–right imbalances indicate horizontal offset and top–bottom imbalances indicate vertical offset. So beam *offset* (centroid shift on the QPD) is read out from DC power imbalances,

$$\text{DPS}_h := \frac{\overline{P}_\ell - \overline{P}_r}{\overline{P}_\ell + \overline{P}_r} = \frac{\overline{P}_A + \overline{P}_C - \overline{P}_B - \overline{P}_D}{\overline{P}_A + \overline{P}_B + \overline{P}_C + \overline{P}_D}, \tag{15}$$

$$\text{DPS}_v := \frac{\overline{P}_t - \overline{P}_b}{\overline{P}_t + \overline{P}_b} = \frac{\overline{P}_A + \overline{P}_B - \overline{P}_C - \overline{P}_D}{\overline{P}_A + \overline{P}_B + \overline{P}_C + \overline{P}_D}. \tag{16}$$

DPS is the correct observable for lateral centering; it is not suitable for wavefront tilt because it is insensitive to the RF phase.

Differential Wavefront Sensing (DWS). By contrast, if the incoming beam is slightly tilted, the interference fringes sweep across the detector, so one side of the QPD reaches its oscillation maxima slightly earlier (or later) than the other. The phasemeter digitizes these oscillating currents and extracts their relative phase; comparing left vs. right or top vs. bottom reveals angular misalignment of the wavefront. This is known as **Differential Wavefront Sensing** (DWS). The angle of incidence between the beam and the QPD (relative wavefront angle) is read out from phase differences between detector halves formed by complex sums,

$$DWS_h = \phi_{\text{right}} - \phi_{\text{left}} = \arg\big(C_A + C_C\big) - \arg\big(C_B + C_D\big) = \arg\left(\frac{C_A + C_C}{C_B + C_D}\right), \tag{17}$$

$$DWS_v = \phi_{\text{top}} - \phi_{\text{bottom}} = \arg\big(C_A + C_B\big) - \arg\big(C_C + C_D\big) = \arg\left(\frac{C_A + C_B}{C_C + C_D}\right). \tag{18}$$

These expressions match the LPF/LISA demodulation used in phasemeters and are the natural complex-sum counterparts of the DPS definitions above.

Longitudinal Path Sensing (LPS). This signal is obtained by summing all four quadrants. In this case, a change in optical path length between beams (due to test mass motion or a passing gravitational wave) produces a global phase shift of the heterodyne beat, seen identically in all quadrants. The longitudinal signal is the common phase converted to length,

$$\text{LPS} = \frac{1}{k}\,\phi = \frac{1}{k}\,\arg\big(C_A + C_B + C_C + C_D\big), \qquad k = \frac{2\pi}{\lambda}. \tag{19}$$

In a heterodyne interferometer, the two beams have slightly different optical frequencies, but since the heterodyne frequency $f_{\text{het}} \ll f$ is negligible compared to the laser frequency, their wavelengths are practically equal, such that $k_1 \simeq k_2 = 2\pi/\lambda$. The phasemeter tracks the phaseshift with high precision, with a 2π phase change corresponding to one laser wavelength (1064 nm). Thus, LPS provides the primary length readout, DPS measures beam offset, and DWS measures angular tilt, together enabling precise interferometric sensing in LISA.

1.3.3. Tilt-to-length (TTL) coupling

Tilt-to-Length (TTL) coupling is the unwanted conversion of angular jitter (tilt/pointing fluctuations) into spurious longitudinal pathlength changes in an interferometer readout. It therefore acts as a noise source for LISA's interferometry, affecting its sensitivity to gravitational waves [43, 44, 45]. Essentially, if a beam is misaligned or tilted, it travels a slightly different distance than a perfectly aligned beam, resulting in a phase shift in the interference signal that appears as a fictitious length change. TTL noise was a major limiting factor in the earlier LISA Pathfinder mission and remains a primary challenge for LISA [42][46][47].

Geometric TTL Coupling In the LISA test mass interferometer, a tilted test mass can introduce a light-lever effect, where a lever arm between the test mass and the detector means a tiny tilt yields an extra optical path length. Similarly, if the laser beam does not reflect at the test mass's center of rotation (center of mass), a tilt causes a *piston effect* – a small longitudinal shift of the reflection point that adds extra path length [43][42][48][49]. Apart from the test mass, a tilt of a flat mirror or microscopic warping of the optical bench could change the projection of the beam path along the sensitive axis, thereby altering the distance traveled.

Non-Geometric TTL Coupling These arise from wave-optical and detector-related factors rather than simple path geometry [50][49]. These can relate to wavefront properties of the interfering beams, i.e. if the two interfering beams have mismatched optical profiles or wavefront curvatures, a tilt will cause an imperfect overlap of the wavefronts on the detector, leading to an uneven interference pattern. Any wavefront error or aberration in the incoming beam can couple with pointing jitter to induce phase

shifts across the beam that perturb the interference signal [51]. Furthermore, the detector geometry plays a role. In essence, the measured phase is the integrated interference phase over the detector's active area, so any factor that alters the distribution of light/phase on that area (misalignment, beam shape mismatch, detector positioning) will produce TTL coupling. The angle and placement of the photodiode can affect the relationship between incoming beam wavefronts and the signal readout [50]. Non-ideal detector surfaces, shape, and finite size can further introduce phase noise, as real detectors fail to perfectly integrate the incoming beam's phase profile. Analytical models demonstrate that these geometric properties must be considered when estimating non-geometric TTL effects [52].

Recent studies show that TTL noise can be accurately modeled and subtracted without degrading gravitational-wave signals, demonstrating the robustness of the approach across diverse source types [53]. A deeper understanding of TTL coupling remains essential to further improve its correction and achieve LISA's intended sensitivity.

1.4. Beam Models

Gaussian Beams Most optical simulations for LISA assume *Gaussian beams*, which are the fundamental solutions of the paraxial wave equation. A Gaussian beam is characterized by its waist radius w_0, Rayleigh range $z_R = \pi w_0^2/\lambda$, and divergence angle $\theta \approx \lambda/(\pi w_0)$. The intensity profile at a plane z from the waist is

$$I(r,z) = I_0 \left(\frac{w_0}{w(z)}\right)^2 \exp\left(-2\frac{r^2}{w^2(z)}\right), \tag{20}$$

with beam radius $w(z) = w_0\sqrt{1+(z/z_R)^2}$. Gaussian beams preserve their functional form under propagation, making them mathematically convenient for modeling. In LISA, the *reference beam* generated locally on each optical bench is assumed Gaussian, and most alignment and sensing models are based on Gaussian overlap integrals.

Tophat and Clipped Beams In contrast, the *measurement beam* propagates across 2.5 million km, during which it becomes diffraction-broadened. At the spacecraft, it is transmitted through a 30 cm telescope, where the finite aperture clips the outer Gaussian wings. The combined effect of diffraction during propagation and aperture clipping at the telescope means that the received far-field beam can be approximated as a flat-top (tophat-like) profile [54]. Unlike a pure Gaussian, a tophat beam has a flat central intensity with sharp edges, and its far-field diffraction produces side lobes due to the discontinuity at the aperture edge.

Impact on Interferometry Over 40 different noise signals have been investigated [23]. So far, limited studies have explored the impact of the TTL-noise of the *top-hat beam* on the long arm lasers. This is because there is no analytical solution to top-hat beams since the shape of the beam is unknown.

The difference between local Gaussian and received tophat-like beams is not only a matter of shape but also of *interference properties*. A truncated Gaussian can be represented as a superposition of higher-order modes, and the resulting mode content modifies the interference at the QPD. This impacts precision readouts: diffraction alters the interference pattern and directly influences tilt-to-length (TTL) coupling [55][42]. Since TTL noise arises when angular jitter couples into the longitudinal phase measurement, the non-Gaussian nature of the received beam makes it more challenging to predict how beam jitter will manifest in the QPD signals.

Thus, while Gaussian beams remain the default model for analytical studies and system design, incorporating tophat beam models is necessary to capture realistic effects in LISA's long-arm interferometry.

1.5. Problem Statement and Research Objectives

In LISA's interferometry, Quadrant Photodiodes (QPDs) are essential for extracting both the longitudinal phase information (LPS) and angular alignment signals (DWS). However, the response of a QPD depends not only on the idealized Gaussian beam assumption but also on practical effects: detector orientation, incident beam pointing, optical bench stability, and the non-Gaussian character of received beams. These factors introduce couplings and distortions that complicate the interpretation of QPD data and can generate systematic errors such as tilt-to-length (TTL) noise.

The central problem is that the relationship between beam/detector parameters and QPD output is *nonlinear and multidimensional*. Variations in detector alignment (tip, tilt, clocking), incoming beam angle or offset, and beam profile (Gaussian vs. tophat-like) are not independent. Their combined influence defines a large parameter space in which isolating the effect of a single variable becomes nontrivial. The challenge is to quantify how deviations from the ideal system propagate into the QPD signals, and to determine to what extent these deviations must be modeled or corrected in LISA simulations.

The objectives of this thesis are therefore:

1. **Misalignment effects:** Quantify how deviations in alignment influence DWS and LPS signals. This includes

   (i) QPD misalignment due to tip, tilt, or clocking, and

   (ii) beam/bench misalignment arising from angular jitter, lateral offsets, or thermal deformations.

2. **Parameter space analysis:** Characterize the interdependence of orientation parameters and how their combined variations shape the DWS/LPS response.

3. **Deviation from ideal:** Compare QPD laboratory measurements with idealized simulation results, and determine the level of error or uncertainty that must be introduced into simulations to match experimental data.

4. **Impact of tophat beams:** Study how clipped or diffracted beams change the QPD signals, considering different beam generation methods, modal decompositions, and comparisons against both measured data and idealized simulations.

5. **Detector geometry study:** Evaluate the influence of QPD size and inter-quadrant gap size on the stability and fidelity of DPS, DWS, and LPS signals.

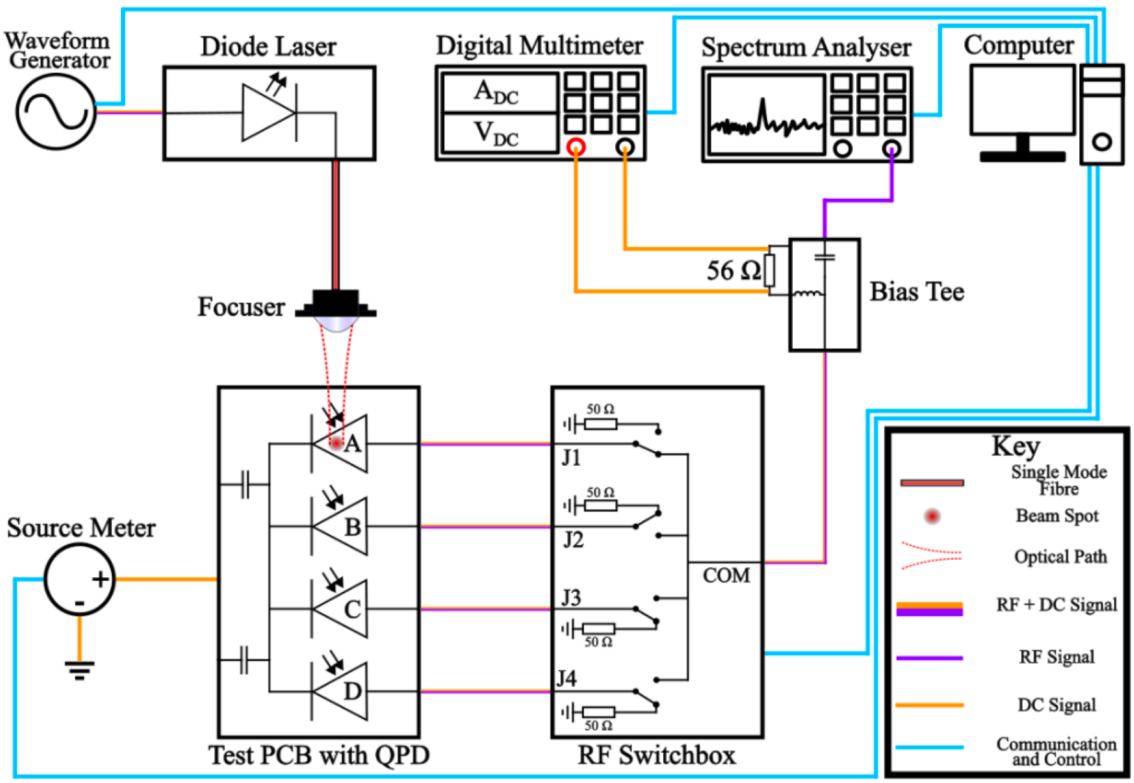


Figure 9: Experimental setup for QPD responsivity measurements. A focused, modulated laser beam illuminates the QPD mounted on a passive PCB. Quadrant outputs are routed through an RF switch box and split by a bias-tee into DC and RF components, measured by a multimeter and spectrum analyzer. [56]

2. Methods

2.1. Experimental Setup

The responsivity of a quadrant photodiode (QPD) was mapped using a focused laser beam (with a waist of approximately $4\,\mu$m at the diode) modulated at radio frequencies of 3, 15, or 30 MHz. A fiber-coupled diode laser, current-modulated for intensity control, delivered the beam through a focusing lens onto the QPD. The QPD, mounted on a custom printed circuit board (PCB) with passive circuitry, provided four quadrant current outputs. These outputs were routed via an RF switch box that selected one quadrant at a time for readout. Downstream of the switch, a bias-tee split each selected quadrant's signal into a DC branch and an RF branch. The DC photocurrent was measured with a digital multimeter (for average power and position calibration), while the RF component at the modulation frequency was measured with a spectrum analyzer. A programmable source meter supplied a stable 24 V reverse bias to the QPD during operation. The entire setup was computer-controlled: the beam was scanned over the QPD active area using precision stages (with finer spatial steps near the quadrant boundaries to capture steep gradients). At each position the DC and RF readings were recorded, with approximately 50 samples averaged per point to reduce noise. [56]

2.1.1. Data Acquisition

At each grid point, the photodiode's DC output (proportional to local intensity) and RF amplitude (proportional to the local modulated responsivity at the drive frequency) were logged. The raw data is stored in binary format (.pkl files), which include the coordinate, DC, and RF values for each sample. The scan grid had non-uniform spacing, so the measured responsivity maps were interpolated onto a uniform grid for use in simulations. This interpolation allows the data to interface with the simulation

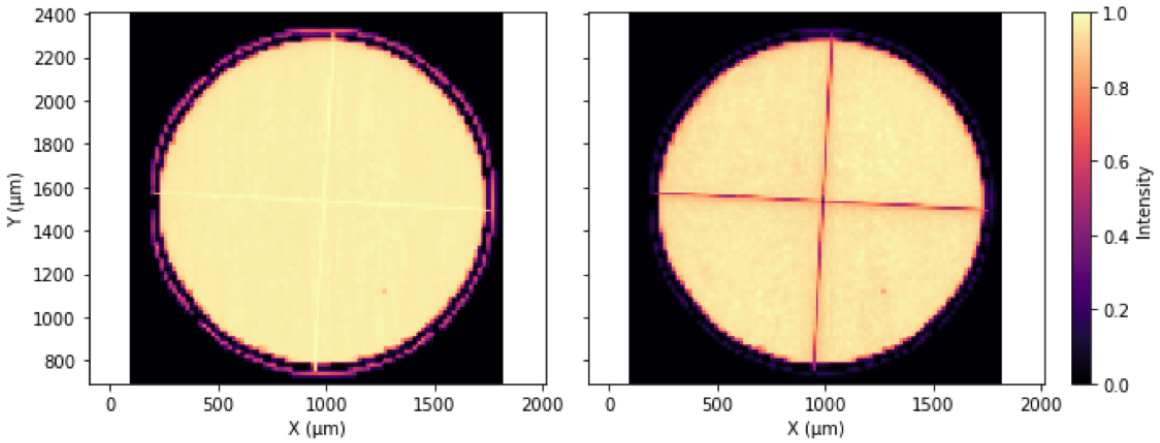


Figure 10: Laboratory responsivity mapping of the QPD. Left: normalized DC intensity. Right: normalized RF intensity at 3 MHz. Both maps correspond to a single measurement instance.

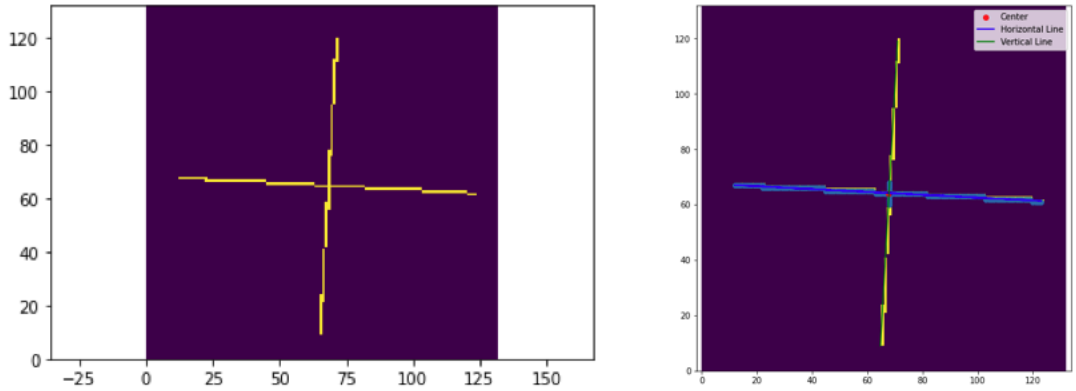


Figure 11: Left: Mask of the RF signal isolating the QPD gaps. Right: Auto-clocking algorithm fitting slopes of the horizontal and vertical gap lines. The fitted angles differ slightly: –3.22° vs. –2.95°, corresponding to a 0.273° difference.

software's grid requirements while preserving the fine features around quadrant boundaries. The outcome of this process is a set of spatial responsivity maps (for both DC response and RF response at each frequency) over the QPD's surface. These maps can then be uploaded into the interferometer simulation (IfoCAD) as a custom detector response.

Crosstalk, i.e. unintended coupling between photodiode quadrants, can influence responsivity measurements [38], but is not explicitly analyzed in this thesis. However, since the laboratory responsivity maps explicitly included crosstalk present for each quadrant, its effect is implicitly carried into the simulations. Previous work using spatially resolved QPD response maps found crosstalk contributions to be negligible [57], though further dedicated studies are recommended to confirm QPD accuracy.

In addition to crosstalk, imperfections in QPD mounting or fabrication can introduce small orientation errors, such as clocking, tip, or tilt, which affect the alignment between the detector and the incoming beam. To quantify this, an auto-clocking algorithm was developed that detects the quadrant gap lines in the responsivity maps and fits straight lines through them. From this, the effective clocking angle of the device is determined. Interestingly, the algorithm yielded two slightly different angles for the orthogonal gaps, indicating a small asymmetry. The average difference was $0.273°$ This discrepancy is revisited later in Section 3.1.1 and in Appendix A.

2.2. Simulation framework (IfoCAD)

For the simulations, the IfoCAD software library [58] was used to model the QPD and beam alignment degrees of freedom. IfoCAD is a C++ simulation toolkit originally developed by Gerhard Heinzel at the Max Planck Institute for Gravitational Physics (Albert Einstein Institute, Hanover) [59]. It enables detailed modeling of interferometric systems by providing beam propagation routines, Hermite–Gaussian mode decomposition, and modules for realistic optical elements and detectors. In this thesis, IfoCAD is used to propagate Gaussian and non-Gaussian beams to the QPD plane and to compute DPS, DWS, and LPS signals, which allows direct comparison between idealized and experimentally measured QPD responses [39][60][61].

All coordinates are defined such that the QPD surface lies in the x–y plane at the origin, and the nominal beam propagates along the $-z$ direction (normal incidence onto the QPD). Several possible misalignment parameters (degrees of freedom) were included in the model, representing both photodiode alignment errors and beam jitter:

- **QPD tip and tilt:** small rotations of the QPD about its x- or y-axes, altering the pitch or yaw orientation relative to the beam.
- **QPD clocking:** in-plane rotation of the QPD about the z-axis, which rotates the quadrant layout.
- **Beam lateral offsets:** translations of the incoming beam axis in the QPD plane (along x or y), i.e., a beam that hits off center.
- **Beam angular offsets:** rotations of the beam direction (pivoting around a fixed point in the QPD) such that the beam arrives at a slight angle, introducing a pointing error.

By convention, a pure rotation of the beam (angular misalignment) in the model does not induce any lateral displacement of the beam spot on the QPD. The beam is pivoted about the QPD center, so that a small angular tilt changes only the incidence angle while keeping the spot centered. In this way, angle-induced effects can be studied separately from lateral displacement effects, consistent with the approach in [50].

Other studies adopt different conventions. Some place the pivot a finite distance upstream of the detector, so that a beam rotation also introduces additional optical path length [39, 62, 48]. More detailed models include the beam splitter and imaging system: here, an initial offset or rotation at the source propagates into combined angles and lateral displacements at the QPD [54]. The most complete treatments extend the pivot to the test mass itself, including the entire optical chain from test mass, beam splitter, and imaging optics down to the QPD [61][63].

In this work, the simplified pivot-at-QPD convention is used. Any additional optical path length arising from an angled beam (a geometric path difference across the diode surface) is ignored in these simulations. Effectively, all beams are treated as arriving planar to the QPD, with the analysis focused on overlap and phase differences rather than absolute path-length changes. This isolates the so-called "non-geometric" tilt-to-length effects from the geometric phase shift.

2.2.1. Parameterization and Signal Calculation

The simulations were performed using a beam wavelength of $\lambda = 1064\,\text{nm}$ and a waist radius of $w_0 = 0.5\,\text{mm}$. Two beam profiles were considered: a Gaussian TEM_{00} mode, serving as the reference case, and tophat-like beams constructed through GBDC, MEM, or NMF decomposition methods. The QPD was modeled with an active area of $0.75\,\text{mm}$ and an inter-quadrant gap of $20\,\mu\text{m}$. For realistic simulations, responsivity maps obtained from laboratory measurements were interpolated and applied, covering both DC and RF responses at 3, 15, and 30 MHz.

For all simulations, the beam origin was defined at a distance of $-L = 5\,\text{mm}$ upstream of the QPD plane. This choice places the QPD slightly beyond the waist, but still well within the Rayleigh range ($z_R \approx 0.74\,\text{m}$). The motivation was twofold: first, to set a consistent reference point for all runs, and second, to avoid positioning the QPD exactly at the waist, which would correspond to a

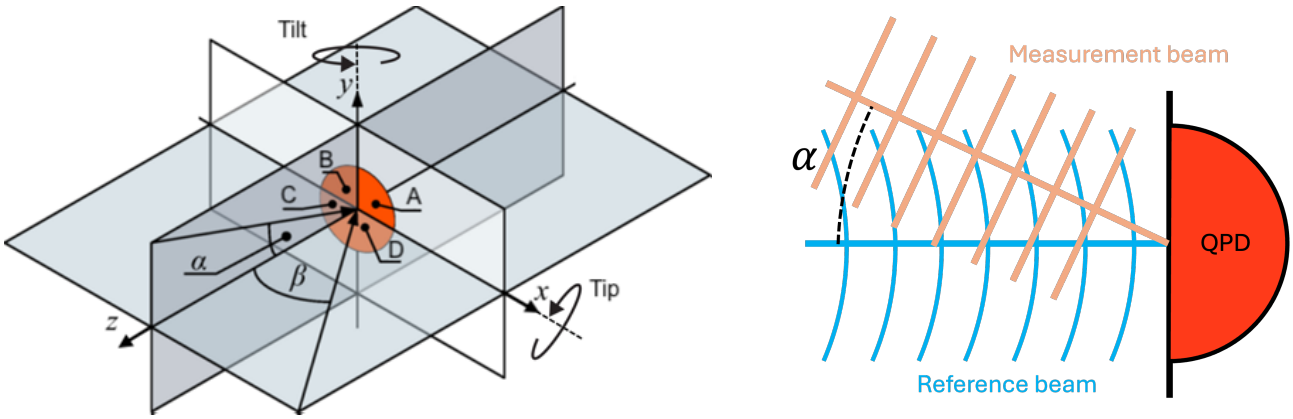


Figure 12: Left: Geometry of the QPD in the x–y plane facing $+z$. Misalignments are tip (rotation about x), tilt (rotation about y), and clocking (rotation about z). Angles α and β denote beam incidence relative to the QPD normal. Right: Interference of the reference beam (blue) and tilted measurement beam (orange) on the QPD, where the incidence angle creates a phase gradient used for differential wavefront sensing.

perfectly flat wavefront. By shifting the waist a few millimetres upstream, the simulated beam at the QPD naturally carries a small but finite curvature, better reflecting realistic experimental conditions where the detector is rarely located exactly at the waist.

Misalignment and jitter effects were introduced by applying controlled transformations to the beams and the QPD. Beam rotations about the x and y axes (α, β) were varied within $\pm 100\,\mu$rad. Beam lateral offsets in the x and y directions were scanned within $\pm 30\,\mu$m. The QPD itself could be tilted by up to $\pm 4^\circ$ and clocked (in-plane rotation) within $\pm 3^\circ$. For each configuration, the incident beam field was generated at the QPD plane and transformed according to the prescribed misalignment parameters. Beam rotation and QPD tip/tilt are not equivalent perturbations, since the reference beam always remains aligned to the optical z-axis; thus, beam rotation and QPD tip/tilt represent two independent degrees of freedom.

The evaluation of QPD signals then required numerical integration of intensity and interference terms over the detector quadrants. In the software, this was implemented through two alternative routines: GSL, and DCUHRE [64][65]. The exact details of the numerical integration are not discussed in this thesis, but the main differences between the implemented algorithms are outlined here.

GSL uses a nested one-dimensional adaptive quadrature strategy: the integral is split into an outer loop over the vertical coordinate and an inner loop over the horizontal coordinate, with exact analytic limits matching the quadrant geometry. It applies Gauss–Kronrod rules with embedded error estimates; both refine the integration grid until user-defined absolute and relative error tolerances are satisfied. In contrast, DCUHRE performs a fully two-dimensional adaptive quadrature over a rectangular bounding box. DCUHRE expects smooth functions, so the sharp detector edges and quadrant slits are represented by a smoothed mask (a damper function), enabling stable convergence. This flexibility makes DCUHRE advantageous when measured responsivity maps are included, while the 1D methods remain more efficient for idealized detector geometries. In both cases, the outcome of the integration is a set of quadrant-resolved DC powers and complex amplitudes, which subsequently form the basis of the longitudinal and differential wavefront sensing signals.

2.3. Top-Hat Beam Construction

Long-arm interferometers like LISA motivate the need for *flat-top (top-hat) beam* simulations, since a fundamental Gaussian beam transmitted over millions of kilometers expands and is effectively clipped by the receiving aperture, yielding a flat-topped intensity profile [62]. In this study, three different methods were implemented to construct or represent a flat-top beam: (1) Gaussian Beam Decomposition, (2) Mode Expansion Method, and (3) Numerical Mode Fitting (NMF). Each method decomposes

or approximates the flat-top beam in terms of Hermite–Gaussian (HG) mode components, but with different techniques, as described below.

The term "higher harmonics" in this context refers to spatial modes rather than temporal or frequency domain harmonics.

2.3.1. Gaussian Beam Decomposition (GBDC)

The Gaussian Beam Decomposition (GBDC) method represents an arbitrary beam profile as a superposition of many fundamental Gaussian beams. It breaks down complex optical fields, like a top-hat beam with uniform intensity, into a weighted sum of Gaussian beams. Each Gaussian is defined by parameters such as amplitude, phase, and waist size, allowing their combination to approximate the original beam profile. Using the established properties of Gaussian beams simplifies the modeling and propagation of these shapes. In the original formulation by Greynolds (1985) [66], the beam's wavefront at a chosen "decomposition plane" is sampled on a grid, and at each grid point a Gaussian beamlet is launched (all beamlets having the same waist size, located in that plane, initially parallel). As each beamlet propagates independently, their re-superposition in the far field or at a detector can reconstruct the diffraction pattern of the original beam.

This approach, also referred to as *Gaussian beam summation* or *Gaussian beamlet summation*, has been implemented in several optical simulation tools and refined in subsequent work [67, 68]. The method's main limitation is parameter tuning: too few beamlets or poorly chosen waists yield large errors, whereas too many beamlets increase computational cost.

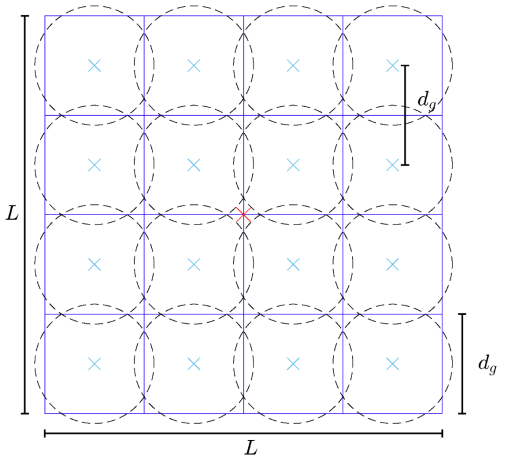


Figure 13: Illustration of the GBDC method square grid structure [62]

In practice, parameters are tuned so that neighboring beamlets overlap sufficiently to avoid blind spots. The implementation used here follows prior guidelines: a grid of beamlets spanning the aperture, with waists chosen such that their $1/e^2$ diameter is comparable to the grid spacing. This method is computationally heavier than a single Gaussian propagation, but it directly captures diffraction from hard edges and can propagate through complex optical systems. In our application, the GBDC-generated flat-top served as one input beam in IfoCAD to simulate the received far-field beam in LISA-like scenarios.

2.3.2. Mode Expansion Method (MEM)

The Mode Expansion Method (MEM) uses higher-order spatial modes, such as Hermite-Gaussian modes, to represent complex optical fields. These modes extend beyond the fundamental Gaussian beam by incorporating intensity and phase structures to approximate beam profiles with sharp edges or non-uniform intensity distributions. By combining multiple spatial modes with tailored amplitudes and phases, a composite beam matching the target profile can be constructed.

MEM constructs the flat-top (or any arbitrary beam) by expanding it into an orthonormal basis of Hermite–Gaussian modes. The HG modes $HG_{mn}(x, y; w_{0d})$ are transverse eigenmodes of a Gaussian resonator and form a complete basis for square-integrable fields. The electric field of the beam is expressed as a sum:

$$E(x, y) \approx \sum_{m,n} a_{mn}\, HG_{mn}(x, y; w_{0d}), \tag{21}$$

where w_{0d} is a chosen waist parameter for the basis functions. The coefficients a_{mn} are determined

by overlap integrals:

$$a_{mn} = \iint HG^{*}_{mn}(x, y; w_{0d})\, E(x, y)\, dx\, dy. \tag{22}$$

This analytic projection ensures that for a given mode order N (where $m+n \leq N$), the expansion is optimal in a least-squares sense. The choice of N and w_{0d} controls accuracy and computational cost: larger N captures sharper edges but requires more modes, while poor choice of w_{0d} slows convergence. In this work, w_{0d} was set equal to the $1/e$ radius of the input Gaussian before truncation, which distributes modal power efficiently without requiring excessive high-order modes. The expansion was truncated once the normalized mean-squared error fell below 10^{-6} in power.

MEM provides an analytical, precise representation of smooth beams and is efficient for Gaussian-like inputs. However, representing sharply truncated beams requires very high order expansions, making MEM less efficient in such cases [69, 70].

2.3.3. Numerical Mode Fitting (NMF)

The third method developed in this work is the numerical, grid-based expansion technique **Numerical Mode Fitting (NMF)**, or *Grid-based Hermite–Gaussian decomposition*. In this approach, an ideal Gaussian beam is sampled on a two-dimensional grid, and an aperture or clipping function is applied to impose truncation. For example, to simulate a flat-top beam of diameter D, we multiply a TEM_{00} Gaussian $E_0(x, y)$ (which is just a fundamental Gaussian mode) by an aperture function $A(x, y)$ that equals 1 within radius $D/2$ and 0 outside. The resulting field is

$$E_{\text{clip}}(x, y) = E_0(x, y)\, A(x, y). \tag{23}$$

This clipped field has no closed-form Hermite–Gaussian expansion, so it is instead fitted numerically. The sampled field $E_{\text{clip}}(x_i, y_j)$ is compared to a truncated HG expansion, and coefficients b_{mn} are solved by minimizing the least-squares error:

$$\min_{b_{mn}} \sum_{i,j} \left| E_{\text{clip}}(x_i, y_j) - \sum_{m,n} b_{mn}\, HG_{mn}(x_i, y_j) \right|^2. \tag{24}$$

The fitted coefficients are then pruned: modes with negligible contributions (below a chosen power threshold) are discarded, and the remaining set is renormalized to match the input beam power. This yields a compact modal representation of the truncated beam.

Both MEM and NMF expand the beam into the same Hermite–Gaussian modes. The difference is that MEM computes the coefficients analytically via integrals, while the clipped Gaussian obtains them numerically from sampled and modified field.

3. Results

3.1. Gaussian Beams

3.1.1. Beam Rotation

The first investigation concerns the DWS signal obtained from responsivity measurements in the simulation, studied as a function of beam tilt. We begin by varying only α, the rotation of the beam around the x-axis.

A clear distinction must be made between two components: In Figure 14, the green slope represents the intended or **primary DWS coupling**: the response of the correct channel to beam rotation ($\alpha \rightarrow \mathrm{DWS}_v$, $\beta \rightarrow \mathrm{DWS}_h$). The blue slope represents the **cross-coupled DWS**, i.e. the undesired response in the orthogonal channel that ideally remains zero ($\alpha \rightarrow \mathrm{DWS}_h$, $\beta \rightarrow \mathrm{DWS}_v$). Solid lines represent simulations using the measured QPD responsivity, and dashed lines correspond to an idealized QPD model.

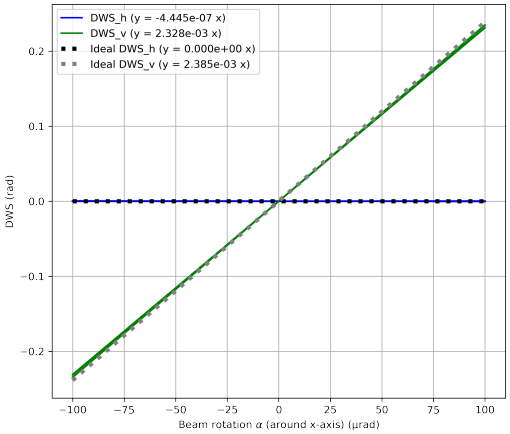


Figure 14: Differential wavefront sensing (DWS) as a function of beam rotation around the x-axis (α). Vertical DWS is in green, horizontal DWS is in blue. Solid lines use measured QPD response; dashed lines show the ideal case.

We define the *primary beam-rotation-to-DWS coupling coefficient* $I^{\mathrm{rot}}_{\mathrm{DWS}}$ as the slope between the applied beam rotation and its corresponding DWS signal (green). Similarly, *the cross-coupled beam-rotation-to-DWS coefficient* $e^{\mathrm{rot}}_{\mathrm{DWS}}$ is defined as the slope between the same beam rotation and the orthogonal DWS signal (blue).

A small deviation is observed between the ideal QPD and the real responsivity data: the vertical DWS slope is reduced from $I^{\mathrm{rot}}_{\mathrm{DWS}} = 2.39 \times 10^{-3} \pm 3 \times 10^{-7}$ [rad/μrad] (ideal) to $I^{\mathrm{rot}}_{\mathrm{DWS}} = 2.33 \times 10^{-3} \pm 1 \times 10^{-7}$ [rad/μrad] (measured), a difference of $5.7 \times 10^{-5} \pm 3 \times 10^{-7}$ [rad/μrad], or about 2.4%. The horizontal (cross-coupled) DWS remains close to zero in both cases.

The analysis is extended to the cross-coupled DWS components that ideally should remain close to zero (again: DWS_h for α-rotation and DWS_v for β-rotation). The QPD clocking was chosen to minimize cross-coupled DWS_h under α-rotation (clocking angle $= 2.95°$).

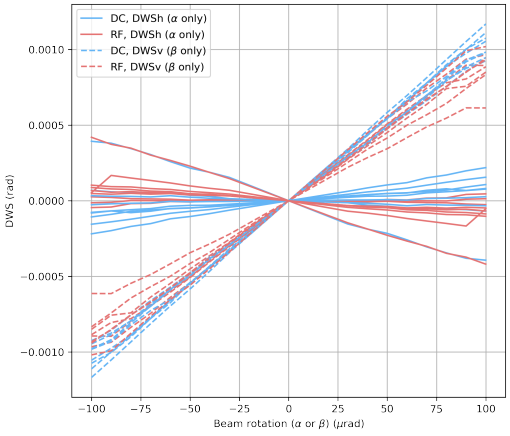


Figure 15: Cross-coupled DWS signals. Left: DWS_h under α-rotation. Right: DWS_v under β-rotation. In total 18 curves are shown (DC and RF signals at 3, 15, and 30 MHz, with three repeated measurements each).

As seen in Figure 15, the two signals behave differently: DWS_h is nearly flat, whereas DWS_v exhibits a small but noticeable slope. This asymmetry matches the difference in gap orientations revealed by the auto-clocking algorithm (Section 2.1.1). To fully capture the experimental variability, all 18 curves are plotted: two signal types (DC and RF), three RF modulation frequencies (3, 15, 30 MHz), and three independent measurements for each condition. These repetitions are labeled as *channel number*; that is, channel 1 corresponds to the first measurement, channel 2 to the second, and channel 3 to the third. The spread across these curves reflects both detector non-uniformity and small measurement-to-measurement variations. The DC and RF responses show slight differences, indicating that frequency-dependent effects also influence the cross-coupled DWS signals.

It is not possible to simultaneously minimize both DWS_h and DWS_v. Correcting the orientation to straighten one set of gap lines inevitably increases the slope of the opposing DWS under beam

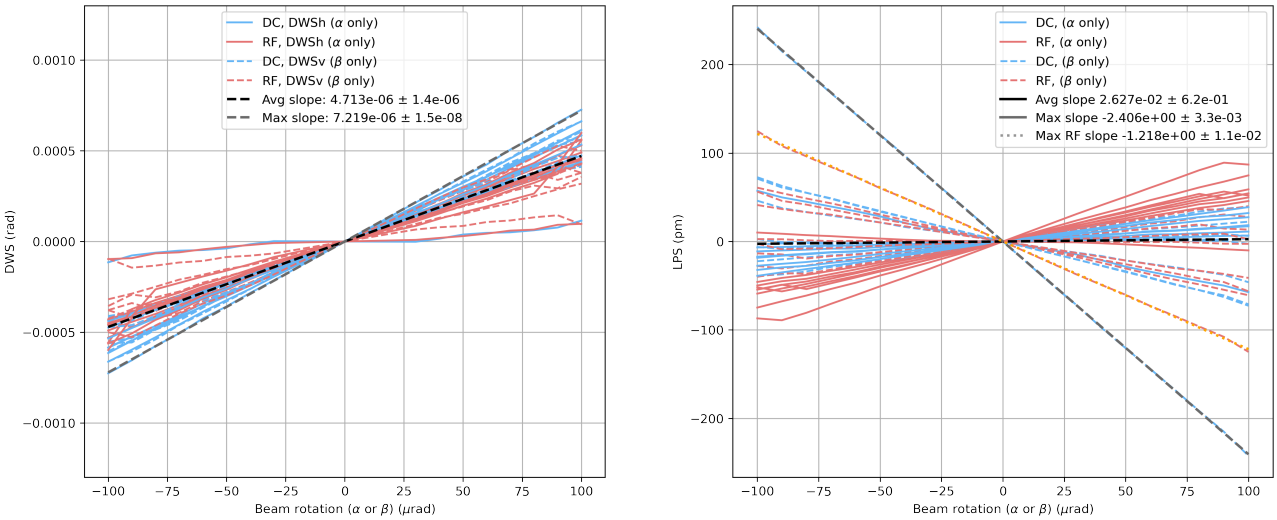


Figure 16: Left: Residual DWS errors after QPD clocking optimization. The QPD was rotated by $3.08°$ to minimize the combined slopes of DWS_h and DWS_v. The average slope is $e^{\mathrm{rot}}_{\mathrm{DWS}} = 4.7 \times 10^{-6} \pm 1.4 \times 10^{-6}$ [rad/μrad], while the maximum slope reaches $e^{\mathrm{rot}}_{\mathrm{DWS}} = 7.22 \times 10^{-6} \pm 1.5 \times 10^{-8}$ [rad/μrad]. Shaded regions indicate the range across 18 measurements (DC and RF at 3, 15, and 30 MHz, three repetitions each). Right: Longitudinal pathlength signal (LPS) versus beam rotation. The average slope of $2.6 \times 10^{-2} \pm 6.23 \times 10^{-1}$ [pm/μrad] is close to the ideal regime, but worst-case scenarios reach $2.41 \pm 3 \times 10^{-3}$ [pm/μrad] (including DC) or $1.22 \pm 1 \times 10^{-2}$ [pm/μrad] (RF only). These bounds define the added error budget LISA must tolerate from beam rotation, even under otherwise optimal conditions.

rotation. In the left plot of Figure 16, the QPD is instead clocked to minimize the combined effect of both signals, which occurs at a clocking angle of $3.08°$. Under this configuration, the average slope across all measurements is $e^{\mathrm{rot}}_{\mathrm{DWS}} = 4.7 \times 10^{-6} \pm 1.4 \times 10^{-6}$ [rad/μrad], representing the residual cross-coupled DWS that arises purely from beam rotation. Since the ideal case would yield zero, this slope can be interpreted as an additional error contribution from the detector geometry. For this plot, tip, tilt, and beam offset were all set to zero, and the QPD clocking was optimized, isolating the influence of beam rotation alone. The dotted black line in the plot indicates the average of residual DWS errors obtained from all 18 measured cases (again: DC and RF signals at 3, 15, and 30 MHz, with three repetitions each). The worst-case scenario corresponds to the largest slope of $e^{\mathrm{rot}}_{\mathrm{DWS}} = 7.22 \times 10^{-6} \pm 2 \times 10^{-8}$ [rad/μrad], and could be interpreted as the upper bound. This represents the maximum additional DWS error that could be introduced into the LISA measurement chain even under otherwise stable conditions. The actual error in practice is expected to lie below this line.

Next, we examine the longitudinal pathlength signal (LPS) as a function of beam rotation about the x-axis (α) and y-axis (β). In the right image of Figure 16, LPS is plotted in picometers against rotation angle in microradians. The spread of results is much larger than for DWS, reflecting both responsivity non-uniformities and frequency dependence.

From previous studies it is known that LPS ideally follows a parabolic dependence on beam tilt [61, 39], leading to additional pathlength signals in the order of tens of $10^{-6}\,\mathrm{pm}$. However, even small lateral beam offsets of a few micrometers can significantly alter the response of LPS versus beam rotation [56]. Combined with non-ideal quadrant responsivity, the result is a broader error distribution. For simplicity in the present analysis, the added LPS error due to beam rotation is characterized by its effective linear slope.

The average slope across all simulations is $2.6 \times 10^{-2} \pm 6.23 \times 10^{-1}$ [pm/μrad]. While this is far above the ideal regime, where added LPS from beam tilt would remain in the $10^{-6}\,\mathrm{pm}$ range, it provides a practical characterization of the residual error under measured QPD responsivity. Worst-case slopes reach $2.41 \pm 3 \times 10^{-3}$ [pm/μrad] when DC measurements are included, or $1.22 \pm 1 \times 10^{-2}$ [pm/μrad] for RF only. These bounds define the additional LPS error that LISA must account for.

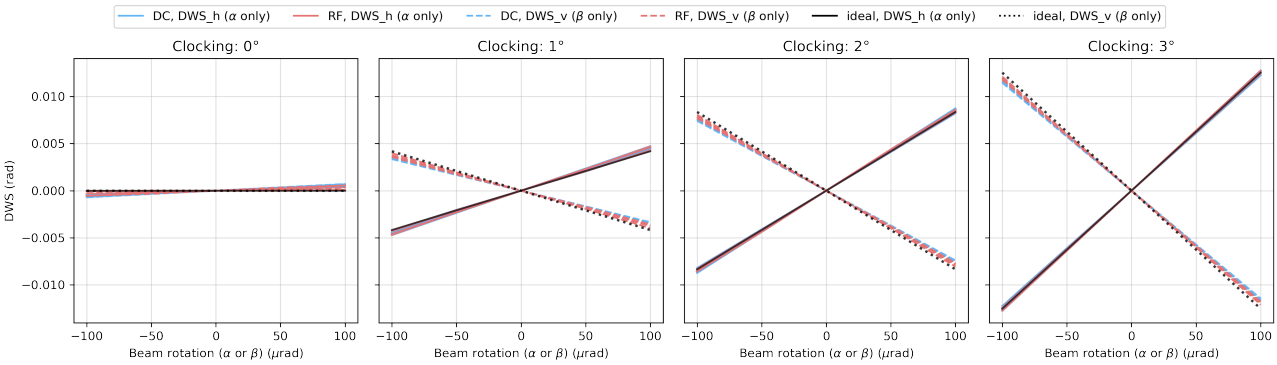


Figure 17: Effect of QPD clocking on cross coupled DWS signals. The detector is rotated from $0°$ to $3°$ in simulation. At zero clocking, beam rotation about one axis produces DWS only in the corresponding direction. With increasing clocking, the gap lines misalign with the beam axes, and each rotation induces components in both DWS_h and DWS_v. Ideal detector responses (black) show the same trend.

3.1.2. QPD clocking

Parameter space:

Responsivity mapping	**QPD**	**Beam**
Type: DC, RF	Tip: $0°$	Rotation axes: α or β
Frequencies: 3, 15, 30 MHz	Tilt: $0°$	Rotation: $[-100,\ 100]$ μrad
Repetitions: 3 per setting	Clocking: $[-3°, 3°]$	Offset: 0

Clocking Angle Figure 17 shows the evolution of the cross-coupled DWS signals when a positive clocking angle is applied to the QPD in simulation. The same dataset as in Figure 16 is plotted, but here the detector is rotated from $0°$ to $3°$. At zero clocking, the response behaves as expected: rotation about the x-axis (α) produces a vertical DWS_v while DWS_h remains near zero, and conversely for y-axis rotation (β). As the QPD is clocked, the gap lines no longer align with the beam rotation axes. The result is that each beam rotation now generates a component in both DWS directions. In other words, the intended separation of horizontal and vertical sensing is gradually lost as clocking increases.

This behavior is consistent with geometric expectations: when the measurement beam wavefronts are no longer parallel to the quadrant gaps, even a pure tilt around one axis projects partially onto the orthogonal direction. The effect is symmetric and grows with the applied clocking, as seen in the stronger slopes of both DWS_h and DWS_v at higher clocking angles. The ideal detector (dashed black lines) shows the same trend, confirming that the effect arises directly from QPD orientation rather than non-ideal responsivity.

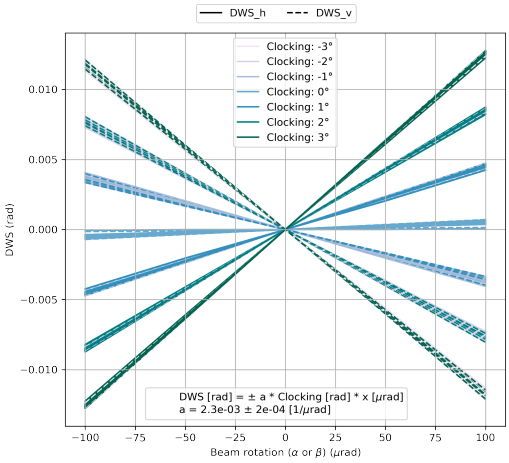


Figure 18: Cross-coupled DWS signals as a function of beam rotation for QPD clocking angles from $-3°$ to $+3°$, and zero QPD tip/tilt and zero beam offset. Solid lines show DWS_h, dashed lines DWS_v.

Figure 18 combines the results of Figure 17 into a single plot, again showing DWS responses for QPD clocking and beam-rotation. Now perhaps more clearly; as the QPD is clocked away from zero, the slopes of both DWS components increase symmetrically with the magnitude of the clocking angle.

To quantify this effect, the slopes of DWS_h were extracted for clocking angles between $-3°$ and $+3°$. The values change almost linearly from $e^{\mathrm{rot}}_{\mathrm{DWS}} = -1.17 \times 10^{-4}$ [rad/μrad] at $-3°$ to $e^{\mathrm{rot}}_{\mathrm{DWS}} = +4.62 \times 10^{-5}$

[rad/μrad] at $+3°$. A linear regression

$$e_{\text{DWS}}^{\text{rot}} = e_{\text{DWS}}^{\text{clock}} \cdot \theta_{\text{clock}} + b \tag{25}$$

across this range was fitted, with b as the intercept. This gave a slope change of $e_{\text{DWS}}^{\text{clock}} = 2.35 \times 10^{-3} \pm 3 \times 10^{-4}$ [μrad^{-1}] per radian of clocking.

This coefficient is the *clocking induced cross-coupling coefficient* $e_{\text{DWS}}^{\text{clock}}$. It quantifies how much beam rotation is coupled into the (cross-coupled) DWS signal, per radian of clocking.

To summarize: the cross-coupled DWS signal should ideally remain zero when the measurement beam is rotated. However, Figure 18 demonstrates that QPD clocking introduces a measurable increase in this coupling. Quantifying this effect is therefore essential to the objectives of this thesis, as it characterises how clocking amplifies the otherwise negligible cross-coupling between beam rotation and the DWS signal.

It is important to note that, unlike the DWS signals, the longitudinal pathlength signal (LPS) shown in the right panel of Figure 16 remains unchanged when a clocking angle is applied to a QPD with zero tip and tilt. This is expected, since LPS reflects the optical phase accumulated along the propagation direction and is therefore insensitive to a pure in-plane rotation of the detector. In other words, clocking alters the mapping of transverse beam rotation onto the quadrant signals, but does not affect the longitudinal phase measurement.

3.1.3. Beam Jitter

Responsivity mapping	**QPD**	**Beam**
Type: DC, RF	Tip: $0°$	Rotation axes: α
Frequencies: 3, 15, 30 MHz	Tilt: $0°$	Rotation range: $[-100,\ 100]$ μrad
Repetitions: 3 per setting	Clocking: 0	Beam offset: $[-30,\ 30]$ μm

For the next investigation, the parameter space is restricted to the nominal alignment of the QPD, i.e. with tip and tilt fixed at $0°$, and clocking also set to $0°$ The beam is only allowed to rotate about the y-axis (α) within ± 100 μrad, and an additional horizontal beam offset in the range $[-30, 30]$ μm is introduced.

In this configuration, we examine the relation between horizontal beam offset and DWS$_h$. Figure 19 (a) shows the results: the x-axis denotes the horizontal beam displacement, the y-axis the measured DWS$_h$, and the color scale encodes the angle α. The legend indicates that both tip, tilt, and clocking are set to zero. A clear linear proportionality is observed between horizontal offset and DWS$_h$. The black points, representing the ideal detector, follow the same linear dependence, confirming that this coupling arises naturally from geometry rather than non-ideal responsivity. An additional linear correlation is visible between beam rotation about the α-axis and DWS$_h$. However, this does not originate from the offset itself but reflects the residual DWS error identified earlier in Section 3.1.1 (Figure 16). The ideal detector, for which the correlation between α-rotation and DWS$_h$ is strictly zero and does not show this effect. The linear dependency between horizontal beam offset and DWS$_h$ will be examined in greater depth in Section 3.1.6, where its implications for error budgeting in LISA is discussed.

Figure 19 (b) shows the longitudinal pathlength signal (LPS) as a function of horizontal beam offset. The x-axis corresponds to beam displacement, the y-axis to the measured LPS in picometers, and the color scale encodes the applied beam rotation angle α. The legend indicates that tip, tilt, and clocking are fixed at zero. Both real responsivity data (colored points) and the ideal detector response (black markers) are included.

In this case, no clear dependence of LPS on horizontal beam offset is visible. Instead, the plot is dominated by the contribution of beam rotation, which produces the strong LPS response already observed in Section 3.1.1, Figure 16. This confirms that under ideal alignment conditions, horizontal

beam offset does not significantly affect LPS, and the residual signal is primarily driven by beam rotation.

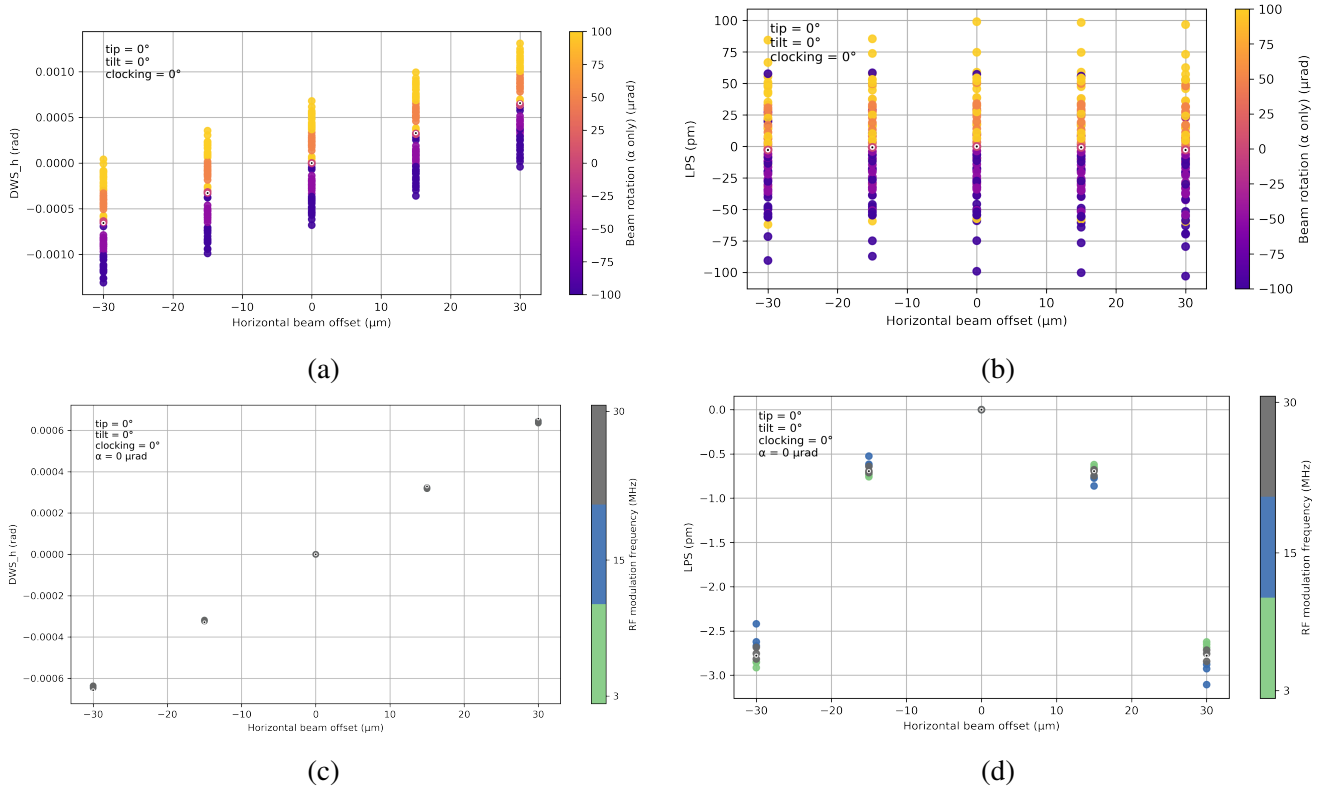


Figure 19: QPD tip $= 0°$, tilt $= 0°$, clocking $= 0°$. Colored markers use measured responsivity; black/grey markers show the ideal detector. (a) DWS_h as a function of horizontal beam offset. Color bar encodes beam rotation α (in μrad). (b) LPS vs horizontal beam offset with the same conditions as in (a). Color bar again denotes α (in μrad). (c) DWS_h vs horizontal beam offset with beam rotation suppressed ($\alpha = 0$) to isolate offset effects. Color bar encodes RF frequency (3, 15, 30 MHz); DC excluded. (d) LPS vs horizontal beam offset under the same constraints as (c) ($\alpha = 0$, tip/tilt/clocking $= 0°$). Color bar again encodes RF frequency (3, 15, 30 MHz); DC excluded.

In plots (c) and (d), the role of beam offset is isolated, without the added complexity of beam rotation. The color mapping has been switched to represent the three RF frequencies (3, 15, and 30 MHz), DC measurements are excluded. In Figure 19(c), a clear linear dependency between beam offset and DWS_h is visible. The data follows the ideal very closely, and no systematic differences between the three RF frequencies are apparent. This highlights that, in the absence of rotation, beam offset alone couples linearly into the horizontal DWS signal. In Figure 19(d), the horizontal axis again represents beam offset, but now the vertical axis shows LPS. The data reveals a shallow parabolic dependency, as expected from theory: LPS grows quadratically with displacement away from the detector center. Although this effect is much smaller than the combined influence of beam rotation and offset reported in earlier studies [56], it is still relevant given the high sensitivity requirements of LISA. The figure indicates small apparent differences between RF frequencies, which might suggest frequency-dependent coupling. In reality, these variations arise from the spread among responsivity measurements (channel numbers), as is investigated in sections 3.1.9 and 3.3.6. At this stage, the result mainly serves to illustrate the sensitivity of the simulations and data.

3.1.4. QPD Misalignment and Beam Jitter

Responsivity mapping	**QPD**	**Beam**
Type: DC, RF	Tip: $[-4°, 4°]$	Rotation axes: α or β
Frequencies: 3, 15, 30 MHz	Tilt: $[-4°, 4°]$	Rotation range: $[-100, 100]$ μrad
Repetitions: 3 per setting	Clocking: $[-3°, 3°]$	Beam offset: $[-30, 30]$ μm

As a limited number of studies have systematically investigated QPD tip and tilt, the simulated detector was deliberately subjected to relatively large rotations $-4°$ to $+4°$. This large simulated range serves to enlarge the resulting effects, allowing them to be quantified and better understood.

In addition to the tip and tilt range of $[-4°, 4°]$, the simulated QPD was also varied in clocking within $[-3°, 3°]$. The beam rotation and horizontal offset were scanned as in the previous section: $\alpha, \beta \in [-100, 100]$ μrad and x-offset $\in [-30, 30]$ μm.

3.1.5. Explanation of Plot Format

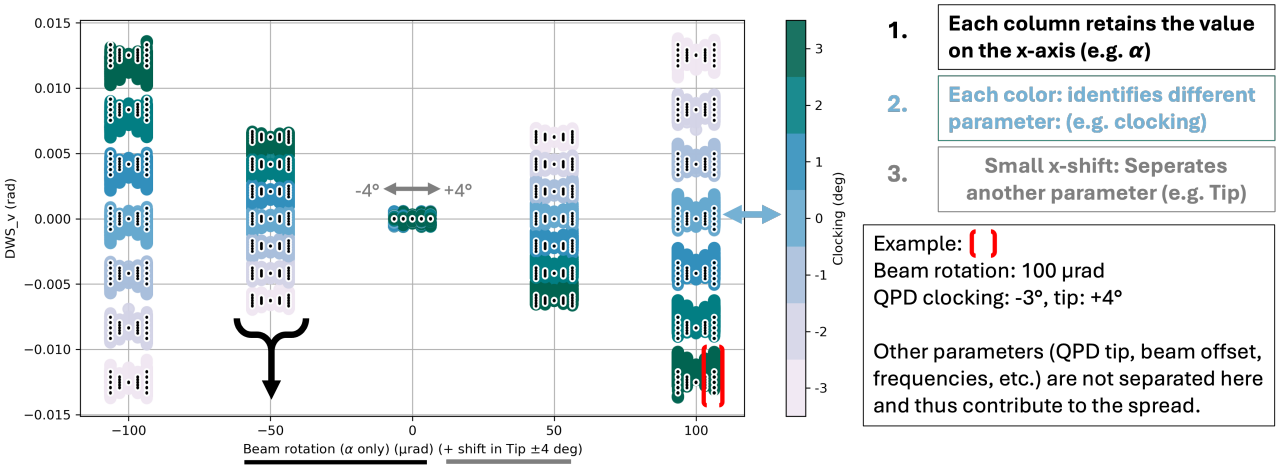


Figure 20: Example plot format used throughout this thesis and appendix. The underlying parameters are interdependent, so the parameter space is segmented to allow multiple relations to be shown in one figure. This approach makes it possible to visualize correlations between beam jitter and QPD alignment without requiring separate plots for each parameter combination.

The parameter space is now almost fully displayed, which makes the resulting plots dense. To make the interdependencies more interpretable, the data are visualised using multiple layers of information:

- Each plot retains the conventional x–y axes.
  - x-axis: driving parameter (beam or QPD)
  - y-axis: QPD signal
- Color coding is added to represent an additional parameter.
- A small artificial shift along the x-axis is applied to disentangle different settings. These shifts do not indicate x-values; instead, all points within a shifted cluster share the same global x-axis value. For instance, in Figure 20, all points at the rightmost position correspond to 100 μrad beam rotation. By color you can identify the clocking of the QPD (pale white = $+3°$). The datapoints are offset slightly in x to distinguish their respective tip values ranging from $-4°$ to $+4°$, this does not alter their 100 μrad beam rotation.

Each cluster can be thought of as a small graph of its own. The y-axis still shows the true simulated values. On the x-axis of the cluster, you only see the range of the shifted parameter. The global x-axis tells you where the cluster belongs overall, while the local spread within the cluster helps to compare how the secondary parameter changes the result.

This approach allows several interconnected parameters to be displayed in a single figure. For example, in Figure 20, the influence of three parameters—beam rotation, QPD clocking, and tip

angle—can be interpreted at once: Beam rotation and clocking strongly affect the DWS response. Tip alone has little effect when the beam rotation is zero. However, when combined with a finite beam rotation, tip introduces an additional spread within the clusters, making the interaction between parameters visible.

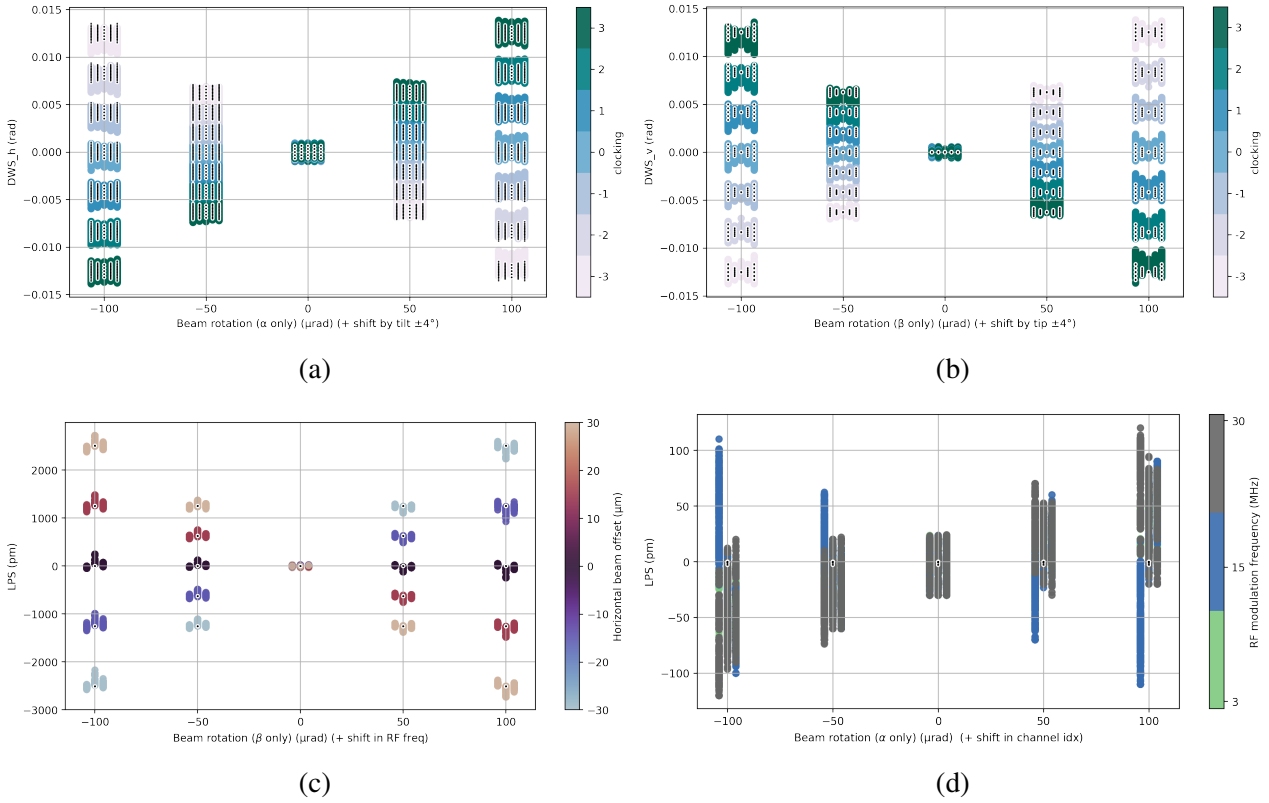


Figure 21: Colored markers use measured QPD responsivity, black markers show the ideal detector. (a) DWS_h [rad] as a function of beam rotation (α) [μrad]. Colormap indicates QPD clocking angle ($-3°$ to $+3°$). Small x-shifts separate variations in tilt [°]. (b) DWS_v [rad] as a function of beam rotation (β) [μrad]. Color indicates clocking angle, with artificial x-shifts separating tip variations. (c) LPS [pm] as a function of beam rotation (β) [μrad]. Small horizontal shifts correspond to the different modulation frequencies (3,15,30 MHz). The markers are color-coded by horizontal beam offset. (d) LPS as a function of beam rotation (α). Small horizontal shifts correspond to channel number (measurement repetition), and color-code corresponds to modulation frequency (3,15,30 MHz).

Results For Figure 21: Plot (a): This plot shows the differential wavefront sensing signal in the horizontal channel, DWS_h [rad], as a function of beam rotation α [μrad]. Small artificial x-shifts represented changes in tilt [°]. The color scale encodes the QPD clocking angle [°]. The black markers show the ideal reference model.

Figure 21 (b): This plot shows the vertical DWS channel, DWS_v [rad], as a function of beam rotation around the other axis β [μrad]. Small artificial x-shifts represent tip variations [°]. The color coding again represents QPD clocking angle. Unlike in DWS_h, no vertical stacking effect is visible here. This is likely caused by the varied horizontal beam offset that does not significantly couple into the vertical channel, this investigated in Section 3.1.6.

Figure 21 (c): This plot shows the LPS in pm as a function of beam rotation β [μrad], with small artificial x-shifts corresponding to the different modulation frequencies (3,15,30 MHz). The datapoints are color-coded by horizontal beam offset in μm.

Figure 21 (d): This plot shows the LPS in pm as a function of beam rotation α [μrad]. Small horizontal shifts now corresponding to the channel numbers (i.e. the 3 repetitions of the measurements). Color-code corresponds to modulation frequency (3, 15, 30 MHz)

Turning to the results: Figure 21 (a) and (b) show the effect of beam rotation on the DWS signals. The influence of clocking is clearly dominant, consistent with Section 3.1.2, even a clocking of 1° produces a stronger effect than a tip/tilt of 4°. Nevertheless, within each cluster of points (same beam rotation, varying tilt), one can already see that increasing tip or tilt gradually enhances the DWS error. Moreover, this influence becomes stronger at larger beam rotations, indicating that tip/tilt contributes a secondary but amplifying effect. The data overall follow the trend predicted by the ideal model, confirming that the observed dependencies are expected and not artefacts.

In the DWS_h plot (a), the influence of horizontal beam offset becomes visible. This is represented by the vertical stacking of datapoints within each cluster. While the primary variation in each cluster arises from different tilt values, the stacked arrangement indicates that horizontal offsets also contribute systematically to the DWS_h signal. This is consistent with the linear dependency between horizontal beam displacement and DWS response discussed earlier (Figure 19). By contrast, in the DWS_v plot (b), no comparable offset-related structure is present, reflecting the fact that horizontal displacement predominantly couples into the horizontal sensing channel.

Figure 21(c–d) show that LPS is not only sensitive to beam rotation, as discussed in Section 3.1.1, but is strongly influenced by the combination of beam rotation and beam offset. When offset and rotation lie in the same plane (e.g. horizontal offset with rotation about the y-axis, Fig. 21 (c)), beam rotation couples much more strongly into LPS. In contrast, when offset and rotation are in orthogonal planes (e.g. horizontal offset with rotation about the x-axis, Fig. 21(d)), the coupling of beam rotation appears unaffected by the offset. The results indicate that DWS is influenced by both QPD misalignments and beam jitter, with offset and rotation contributing independently but without a combined amplification. For LPS, however, the situation differs: it is largely insensitive to QPD misalignments alone, but shows a strong coupling when beam offset and rotation act together in the same plane.

The artificial x-shift of Figure 21 (c) distinguishes the three modulation frequencies used during the responsivity mapping measurements, namely 3, 15, and 30 MHz. Viewed on its own, this plot might suggest that the modulation frequency itself changes how beam rotation couples into LPS. In plot (d), where frequencies are color-coded and the three measurement repetitions (channels) are separated, it is shown that the situation is more complex. Channel 1 (the first measurement set) exhibits strong variations across modulation frequencies, while channels 2 and 3 (the later measurement sets) behave more consistently. Frequency dependent coupling will be further investigated in later sections.

3.1.6. Effects of QPD Tip & Tilt, and Beam Offset

In the previous sections, a possible linear relationship between horizontal beam offset and DWS_h was observed. Here, we explicitly test whether this suspected proportionality is real, quantify its magnitude, and examine to what extent tip/tilt misalignments modify or contribute to the observed DWS response.

Up to this point, the analysis has shown that beam rotation and QPD clocking dominate the differential wavefront sensing signals. These effects are so large that they make it difficult to determine whether smaller influences, such as lateral beam offset or tip/tilt of the detector, also play a role. Simply plotting raw DWS_h values does not resolve this problem, because the variations due to offset or tilt are buried under the much stronger trends from rotation and clocking.

To make these smaller contributions visible, a different approach is required. Instead of showing the absolute DWS_h signal, the analysis now considers how much the signal changes when a specific misalignment is introduced relative to a perfectly aligned detector. In other words, the question is no longer *"what is the total DWS response?"* but rather *"what is the additional response caused purely by offset or tilt?"*

This is implemented by defining a baseline configuration with no horizontal offset, no tip, and no tilt. The DWS_h signal at this baseline is then subtracted from the DWS_h signal at every other

configuration, while all other parameters (beam rotations, clocking, frequency, etc.) are held constant. The resulting difference, or delta value, is given by

$$\Delta\mathrm{DWS}_h = \mathrm{DWS}_h(\alpha \text{ or } \beta,\ x_{\mathrm{off}},\ \mathrm{tip},\ \mathrm{tilt},\ \mathrm{clock},\ f) - \mathrm{DWS}_h(\alpha \text{ or } \beta,\ 0,\ 0,\ 0,\ \mathrm{clock},\ f). \tag{26}$$

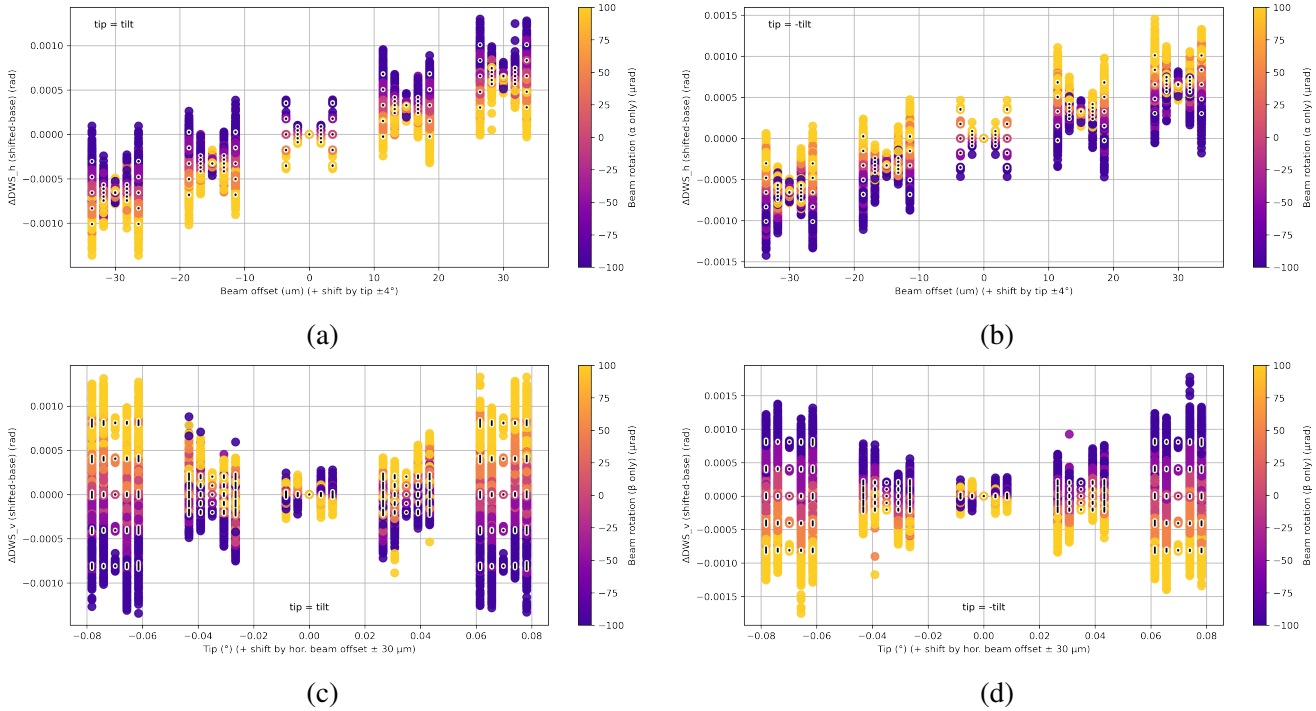


Figure 22: Differential DWS analysis for different combinations of beam offset and QPD tip/tilt. In all plots, black markers with white edges represent the ideal reference, while colored markers show simulated data for lab measured QPD responsivity. The y-axis gives the differential DWS_h or DWSv relative to the perfectly aligned baseline ($x$off $= 0, ;\mathrm{tip} = 0°, ;\mathrm{tilt} = 0°$). (a–b) $\Delta\mathrm{DWS}_h$: the x-axis corresponds to horizontal beam offset, with small x-shifts indicating different tilt values. Color encodes beam rotation α (rotation about the x-axis). Panel (a) shows the case tip = tilt, and panel (b) shows tip = –tilt. (c–d) $\Delta\mathrm{DWS}_v$: the x-axis corresponds to tip angle, with simultaneous tilt values (tip = tilt in (c), tip = –tilt in (d)). Color encodes beam rotation β (rotation about the y-axis), while small x-shifts indicate horizontal beam offset. Horizontal offset couples linearly into DWS_h, while its effect on DWS_v is negligible. By contrast, tip/tilt misalignments determine the regime in which beam rotation couples into both channels, with opposite signs in the tip = tilt and tip = –tilt cases.

For Figure 22 (a–b), DWS_h is plotted against x_{off}, with artificial x-shifts corresponding to the magnitude of the QPD tip. The color scale encodes the applied beam rotation α (x-axis). To further highlight the distinct roles of tip and tilt, the two panels split the parameter space: In (a), only cases with tip = tilt are shown. In (b), only cases with tip = –tilt are shown.

This separation reflects the different symmetry regimes these configurations impose, as explained geometrically in Appendix A.In short; tip = tilt and tip = –tilt lead to opposite sign DWS contributions due to the skewed projection of the QPD gap lines.

From Figure 22 (a-b) two main points emerge: 1. Horizontal beam offset has a clear, linear proportionality with DWS_h, independent of the other parameters. This confirms the earlier suspicion and shows that the proportionality holds robustly across different rotation and tilt regimes. 2. Misalignments in tip/tilt act as amplifiers of the effect of beam rotation. While beam offset and beam rotation represent continuous sources of jitter, tip and tilt are static misalignments that set the regime of sensitivity (as described in section 1.2.3). The stronger the static misalignment, the larger the amplification of beam rotation into the DWS signal.

This means that while beam offset alone produces a predictable linear effect, the presence of tip/tilt defines the scale at which beam jitter (rotation and offset together) couples into the interferometric

measurement.

For Figure 22 (c–d), the analysis is extended to investigate how tip/tilt misalignments affect the vertical channel, DWS_v.

In these panels, the x-axis again represents the applied tip angle, with tilt taken to be either equal (tip = tilt in (c)) or opposite (tip = –tilt in (d)). Beam rotation is now applied about the y-axis (β) and is color coded. Small artificial x-shift within each cluster correspond to variations in x_{off}.

The two plots reveal a marked difference in beam rotation dependence: the color gradient across (c) and (d) is reversed, again reflecting the opposite sign contributions of tip = tilt and tip = –tilt. In both cases, as tip/tilt increases, the influence of beam rotation grows, and the scaling is quadratic in the misalignment angle. This indicates that once a fixed tip/tilt misalignment is present, it sets the regime in which continuous beam jitter (rotation about y) couples into the DWS_v channel.

The additional x-shift encodes horizontal beam offset. From the data, and in agreement with the ideal reference, horizontal offsets have little to no effect on the vertical DWS response, confirming that this channel is primarily sensitive to angular misalignments rather than lateral displacement.

Beam offset to DWS coupling The horizontal beam offset couples linearly into the horizontal DWS signal. To quantify how strongly the offset couples to DWS, the data shown in Figure 22(a) and (b) were fitted with a linear model. For each combination of parameters in the explored parameter space, a linear fit of the form

$$\mathrm{DWS}_h = I_{\mathrm{DWS}}^{\mathrm{off}} \cdot x_{\mathrm{off}} + b \tag{27}$$

was applied to the data points ($R^2 = 0.981$). The resulting slope represents the coupling strength of horizontal beam offset into DWS_h, and is referred to as the *primary beam-offset-to-DWS coupling* $I_{\mathrm{DWS}}^{\mathrm{off}}$.

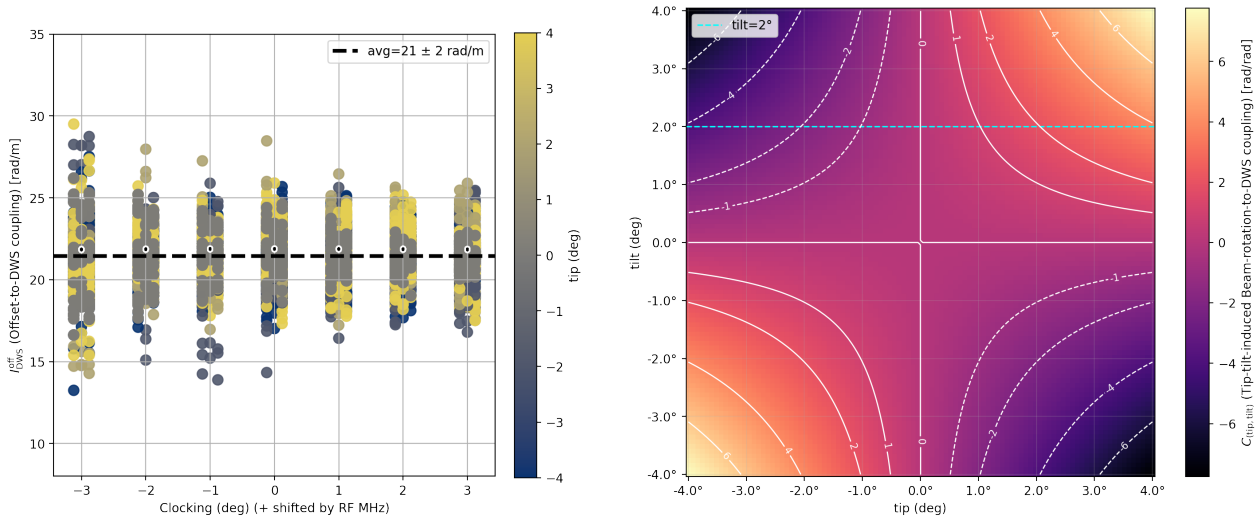
All the slopes $I_{\mathrm{DWS}}^{\mathrm{off}}$ are displayed in Figure 23 (a). On the x-axis the QPD clocking angle is shown, tip values are indicated by color, and modulation frequency is represented by small horizontal shifts of the points. In practice a QPD will always have specific values for clocking, tip, and tilt that are fixed once the QPD is mounted. A plot of the offset–DWS coupling slope across these parameters allows one to identify the expected coupling for any given geometry. By locating the relevant clocking and tip values in the plot, the corresponding slope can be read directly. This makes the plot a practical tool for translating detector misalignment into an expected offset–DWS response.

However, from Figure 23 (a) it can be seen that the offset-to-DWS coupling slope is largely unaffected by variations in clocking, tip, or tilt. Across the full parameter space, the slope remains approximately constant, with only a small spread. The mean coupling strength $I_{\mathrm{DWS}}^{\mathrm{off}}$ is about $21 \pm 2, \mathrm{rad\ m^{-1}}$, i.e. 21 radians of DWS phase per meter of horizontal beam displacement. Intuitively, one might expect that increasing clocking, tip, or tilt would reduce the effective coupling, since less of the beam displacement is projected onto the sensitive axis. This trend is visible in the ideal QPD simulations (see Appendix D), but the reduction is very small. In comparison, the spread caused by the measured QPD responsivity dominates, and the systematic dependence on clocking/tip/tilt is negligible.

Tip-tilt-induced Beam-rotation-to-DWS coupling The next step is to understand how tip and tilt misalignments change the way beam rotation couples into the DWS channel. From Figures 22 (c) and (d) we already saw that the spread in $\Delta\mathrm{DWS}_v$ increases as tip/tilt grows. Importantly, this does not mean that tip/tilt by itself creates extra DWS signal. Rather, the effect only appears when beam rotation is also present. In other words, tip and tilt act as a mechanism that allows beam rotation to leak into DWS.

To isolate this effect, we start from the differential signals defined earlier: for each configuration we form ΔDWS by subtracting the perfectly aligned baseline (as described in at the beginning of section 3.1.6). We take ΔDWS, and it fit with a linear functions for every combination of QPD tip and tilt:

$$\Delta\mathrm{DWS}_v = C_{(\mathrm{tip,tilt})} \cdot \beta, \quad \Delta\mathrm{DWS}_h = C_{(\mathrm{tip,tilt})} \cdot \alpha \tag{28}$$



(a) (b)

Figure 23: (a): The primary offset–to–DWS coupling slopes $I^{\text{off}}_{\text{DWS}}$ [rad m^{-1}] plotted against QPD clocking [°]. Tip [°] is indicated by color, and RF frequency (3, 15, 30 MHz) by small arteficial x-shifts. Each point is the slope of a linear fit of DWS$_h$ [rad] versus horizontal beam offset [m]. The average slope is 21.5 ± 1.6 rad m^{-1}.(b) 2D contour plot of Tip–tilt–induced beam–rotation–to–DWS coupling $C_{(\text{tip,tilt})}$ in [rad/rad]. x-axis: tip angle [deg]. y-axis: tilt angle [deg]. Color map and contours show the bilinear fit $C_{(\text{tip,tilt})} = e^{\text{tip/tilt}}_{\text{DWS}} \cdot \text{tip} \cdot \text{tilt}$, with the fitting coefficient $e^{\text{tip/tilt}}_{\text{DWS}} = (1.592 \pm 0.00014) \times 10^{-3}$ [1/rad^2]. The cyan dashed line indicates tilt $= 2°$. Both figures are made from the parameter space that includes DC and RF responsivity mappings, QPD tip/tilt in $[-4°, 4°]$, clocking in $[-3°, 3°]$, beam rotations α, β up to ± 100 μrad, and beam offsets up to ± 30 μm

Here we term the *tip–tilt–induced beam–rotation–to–DWS coupling* $C_{(\text{tip,tilt})}$. The value of $C_{(\text{tip,tilt})}$ in [rad/rad] tells us how much DWS signal is generated per unit of beam rotation, for a given combination of tip and tilt. If no misalignment is present (tip $=$ tilt $= 0$), then $C_{(\text{tip,tilt})}$ is nearly zero. As soon as tip and tilt deviate from zero, the same rotation produces a measurable DWS contribution, with the magnitude and sign set by the misalignment geometry.

By calculating $C_{(\text{tip,tilt})}$ for every (tip, tilt) pair, we map out the strength of this interaction. This makes it possible to separate the role of tip/tilt misalignment and describe the extra coupling in a compact form. To keep the channels clean from lateral effects we fix $x_{\text{off}} = 0$ in this analysis.

Figures 22 (c–d) already suggested the sign structure: for tip $=$ tilt the rotation–to–DWS$_v$ coupling appears positive; for tip $= -$tilt it appears negative. Mapping $C_{(\text{tip,tilt})}$ over all (tip, tilt) confirms this and reveals a simple surface. The values of $C_{(\text{tip,tilt})}$ lie on a saddle that is well described by the bilinear model:

$$C_{(\text{tip,tilt})} = e^{\text{tip/tilt}}_{\text{DWS}} \cdot \text{tip} \cdot \text{tilt} \tag{29}$$

Here we rename the fitting coefficient $e^{\text{tip/tilt}}_{\text{DWS}}$ as the *tip-tilt-induced cross–coupling coefficient* in [1/rad^2]. Fitting the model to all data gives $e^{\text{tip/tilt}}_{\text{DWS}} \approx 1.592 \times 10^{-3} \pm 1.4 \times 10^{-7}$ [1/rad^2] ($R^2 = 0.9986$) (see Appendix E).

$$\Delta\text{DWS} = e^{\text{tip/tilt}}_{\text{DWS}} \cdot \text{tip} \cdot \text{tilt} \cdot \text{Beam rotation} \tag{30}$$

Figure 23 (b) shows a plot of the tip-tilt-induced beam-rotation-to-DWS coupling $C_{(\text{tip,tilt})}$. Tip [°] and tilt [°] angles are plotted on the x- and y-axes, respectively. The color mesh gives the value of $C_{(\text{tip,tilt})}$ in [rad/rad]. To aid interpretation, contour lines have been added on top of the color mesh, and a horizontal dotted line marks the case tilt $= 2°$, as the QPD on the LISA spacecraft is expected to be mounted with a tilt of about 2° in order to reduce backscattering effects.

The role of this figure differs from the version shown in appendix. In the Appendix (Figure 38), the raw data points and the fitted surface are plotted together, demonstrating the bilinear fit. Here the focus is on readability: the fit surface for $e^{\text{tip/tilt}}_{\text{DWS}} = 1.592 \times 10^{-3}$ is shown on its own.

The figure illustrates several key points. First, the zero-lines of the surface coincide exactly with the tip $= 0$ and tilt $= 0$ axes. This confirms that only a combination of nonzero tip and nonzero tilt produces a nonzero $C_{\text{(tip,tilt)}}$. In other words, neither tip alone nor tilt alone is sufficient to induce extra beam-rotation–to–DWS coupling; it is their product that matters. This matches the qualitative behavior already visible in Figure 22 (c–d).

Second, the figure shows how the sensitivity grows when both misalignments are present. For example, if tilt is kept at $0°$, then changes in tip have no effect, since the surface remains at zero along that axis. But if tilt is increased, even slightly, then the same change in tip produces a measurable change in coupling. The same reasoning holds with tip and tilt interchanged. This means that once a fixed tilt misalignment is present, the detector becomes more sensitive to changes in tip, and vice versa.

Finally, the dotted line at tilt $= 2°$ illustrates the specific case expected for LISA. Along this line, the figure shows how the coupling grows with tip angle. It can be concluded the intended $2°$ tilt itself does not directly induce a coupling, but it sets the stage: any tip error added on top of it will translate into an additional DWS error through beam rotation.

Once the QPD is mounted and its tip and tilt are fixed, the corresponding coefficient $C_{\text{(tip,tilt)}}$ can be read directly from this surface, or equivalently obtained from Eq. 29. This coefficient quantifies how much additional DWS signal is produced per unit of beam rotation for that specific misalignment geometry.

3.1.7. QPD DWS Response: 0° vs 2° Tilt

Responsivity mapping	**QPD**	**Beam**
Type: DC, RF	Tip: $[-0.5°,\ 0.5°]$	Rotation axes: α or β
Frequencies: 3, 15, 30 MHz	Tilt: $0°$ or $2° \pm 0.5°$	Rotation range: $[-100,\ 100]\ \mu\text{rad}$
Repetitions: 3 per setting	Clocking: $[-3°,\ 3°]$	Beam offset: $[-30,\ 30]\ \mu\text{m}$
	Integration: GSL or DCUHRE	

This section tests whether a small fixed tilt of the QPD or different integration method modifies the way beam offset and beam rotation appear in the DWS and LPS readouts. Earlier chapters scanned wide parameter ranges to establish dominant effects; here the focus is narrower. We compare two practical detector settings—QPD tilt $0°$ and $2°$, using the same offset and rotation sweeps as elsewhere. To be precise: Tip was varied in the range $[-0.5°, 0.5°]$, and tilt centered at either $0°$ or $2°$ but varied within $\pm 0.5°$ around those bases (to assess whether the signals are more sensitive to tilt errors in the nominally flat or tilted case). The two different numerical integration schemes, GSL and DCUHRE, were also applied.

All figure grids for this comparison are placed in Appendix F. Figure 24 shows one representative example: moving the detector from $0°$ to $2°$ can make DWS look more sensitive to offset and rotation in some parts of the parameter space, consistent with Figures 23 (a–b). To test whether this effect depends on the numerical integration scheme, both GSL and DCUHRE were applied to the same datasets. Results computed with GSL show compact clusters and make subtle differences clearer, while DCUHRE increases dispersion and inflates a worst-case interpretation without revealing new trends. The outcome is therefore a robustness check: Within the tested ranges, the intended $2°$ tilt itself does not directly induce a coupling, so the main message of earlier chapters stands.

Clocking-induced beam-rotation-to-DWS coupling In Section 3.1.1 we identified a clear coupling between beam rotation and DWS, which increased approximately linearly with QPD clocking (schematically: cross coupled DWS $\approx e^{\text{clock}}_{\text{DWS}} \cdot$ clocking $\cdot$ beam-rotation). Having now confirmed that neither QPD tip, tilt, beam offset, nor integration scheme alter this conclusion, we now quantify the clocking-induced coupling across the full parameter space. This is operationally important, since the

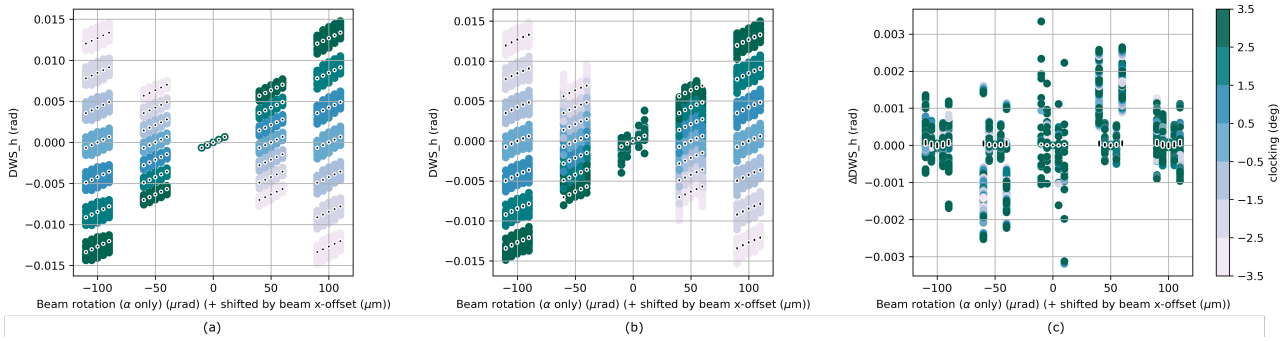


Figure 24: DWS_h plotted against beam rotation α for QPD response at tilt $= 0°$ vs. tilt $= 2°$ tilt, DCUHRE integration method. The three plots correspond to the two tilt settings and their difference: (a) shows the $0°$ case, (b) shows the $2°$ case, and (c) plots the difference between them. The rows correspond to different readouts. Artificial x-shift applied to separate different horizontal beam offsets. The vertical axis is the horizontal DWS signal DWS_h (rad). These panels show how DWS_h depends on α for various simulated parameters. The third panel highlights how this dependence changes when the QPD is tilted by $2°$. Here only DWS_h and DCUHRE are shown, for the other signal and integration types, see Appendix F

current mounting tolerance on clocking is $\pm 3°$, and clocking-induced cross-coupling directly affects how reliably DWS can be used to reconstruct beam orientation.

For every configuration in the parameter space, the applied beam rotation is plotted against the corresponding DWS response. A straight line is fitted to this relation, and its slope defines the beam-rotation-to-DWS coupling, expressed in [rad/rad]. This coefficient tells us how strongly the detector DWS changes per unit of beam rotation.

As described in Section 3.1.1, we separate the coefficients into two groups: 1. The primary DWS coupling coefficient $I^{\mathrm{rot}}_{\mathrm{DWS}}$: The DWS channel that *should* respond to the applied rotation (i.e. for α (about x-axis) $\rightarrow \mathrm{DWS}_v$; for β (about y-axis) $\rightarrow \mathrm{DWS}_h$). The cross-coupled DWS coefficient $e^{\mathrm{rot}}_{\mathrm{DWS}}$: The orthogonal DWS channel that ideally remains zero. This pair of coefficients is evaluated for α and β separately; the figures show the union.

Figure 25(a)–(d) summarize the beam-rotation-to-DWS coupling slopes and their variability as a function of QPD clocking.

Panel (a) shows the primary coupling coefficient $I^{\mathrm{rot}}_{\mathrm{DWS}}$. Panel (b) shows the cross-coupling coefficient $e^{\mathrm{rot}}_{\mathrm{DWS}}$. In both panels, the x-axis is the QPD clocking angle [°] and the y-axis is the fitted slope of DWS versus beam rotation [rad/rad] using the fit

$$\mathrm{DWS}_I = I^{\mathrm{rot}}_{\mathrm{DWS}} \cdot \mathrm{beam\_rotation}_{\parallel}, \quad \mathrm{DWS}_{\mathrm{cross}} = e^{\mathrm{rot}}_{\mathrm{DWS}} \cdot \mathrm{beam\_rotation}_{\perp} \tag{31}$$

Circles represent simulations with the GSL integration method, triangles represent DCUHRE. Filled markers denote $0°$ QPD tilt, open markers $2°$ QPD tilt; in both cases tilt is still scanned within $\pm 0.5°$. Solid lines correspond to DWS_h, dashed lines to DWS_v. Colors distinguish responsivity maps: DC, RF-3, RF-15, RF-30, and ideal.

Figure 25 (c) and (d) give the *error* of the slopes shown in (a) and (b), respectively, plotted against clocking. In principle, every datapoint in those panels could be shown with an error bar, but doing so would make the plots visually overloaded. Presenting the error separately therefore conveys the same information in a clearer way. The plotting conventions are identical to Figure 25 (a) and (b).

In Figure 25 (a) the primary coupling coefficient $I^{\mathrm{rot}}_{\mathrm{DWS}}$ is shown. The curves with smallest values correspond to GSL + DC: about 2308 ± 1 rad/rad (tilt $= 0°$) and 2307 ± 1 rad/rad (tilt $= 2°$) at zero clocking, rolling off to $\approx 2304 \pm 2$ and 2304 ± 1 rad/rad at $+3°$ clocking. The GSL + RF data lie slightly higher, for example $\sim 2346 \pm 2$ rad/rad for RF-30 at tilt $= 0°$, down to $\sim 2340 \pm 2$ rad/rad for RF-3 at tilt $= 2°$ at zero clocking, showing a small but measurable frequency dependence. The DCUHRE (ideal) data sit higher still (≈ 2370–2373 rad/rad at zero clocking) and already exhibit more error than GSL. DCUHRE combined with measured responsivity maps yields

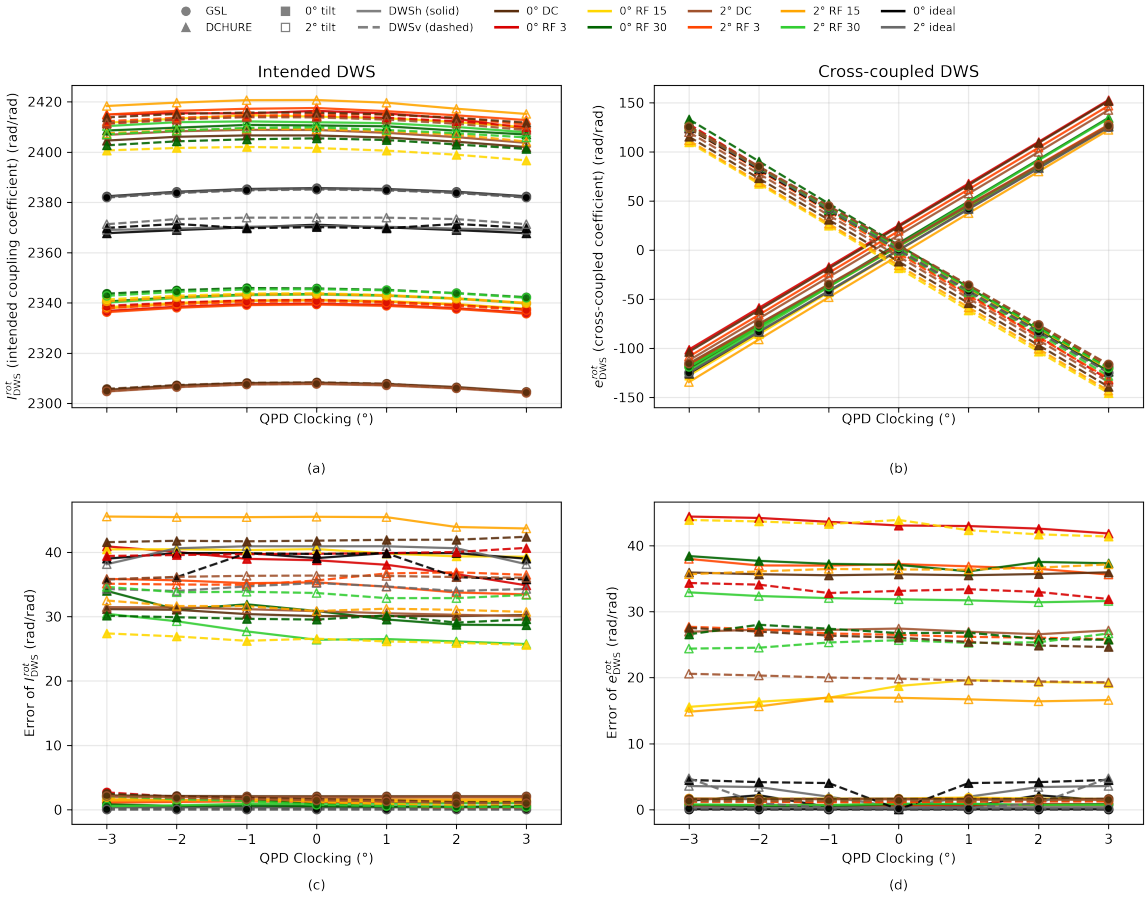


Figure 25: Beam-rotation-to-DWS coupling as a function of QPD clocking.(a) primary DWS slopes $I^{\text{rot}}_{\text{DWS}}$ [rad/rad]; (b) Cross-coupled DWS slopes $e^{\text{rot}}_{\text{DWS}}$[rad/rad];(c) Error of primary slopes [rad/rad]; (d) Error of cross-coupled slopes [rad/rad]. The x-axis in all panels is QPD clocking angle [°]. Circles: GSL integration; triangles: DCUHRE. Filled markers: QPD tilt base 0°; open markers: tilt base 2° (in both cases ±0.5° range scanned). Solid lines: DWSh; dashed lines: DWSv. Colors denote responsivity maps (DC, RF-3, RF-15, RF-30, and ideal). The fit $I^{\text{rot}}_{\text{DWS}} = (I^{\text{rot}}_{\text{DWS}})_{\max} \cdot \cos(\theta_{\text{clock}} + \phi)$ or $e^{\text{rot}}_{\text{DWS}} = (I^{\text{rot}}_{\text{DWS}})_{\max} \cdot \sin(\theta_{\text{clock}} + \phi)$ describes all datasets with $R^2 > 0.99$.

the highest and broadest values, with RF-3 reaching about 2421 ± 45 rad/rad for tilt $= 2°$ (DWSh) and $\approx 2402 \pm 27$ rad/rad for tilt $= 0°$ (DWSv). Across all methods and frequencies, the primary coupling spans roughly 113 rad/rad, with dataset-dependent errors of tens of rad/rad as quantified in Figure 25(c).

In Figure 25 (b), the cross-coupling coefficient $e^{\text{rot}}_{\text{DWS}}$ is shown. At zero clocking, GSL produces tight clusters near zero: e.g. 5.0 ± 0.6 rad/rad (RF-3, tilt 0°, DWSh) and 3.3 ± 1.2 rad/rad (RF-3, tilt 0°, DWSv). The DCUHRE results are much broader, with values scattered between roughly -20 and $+25$ rad/rad depending on frequency and tilt, and errors up to ~ 40 rad/rad. At $|\theta_{\text{clock}}| = 3°$, the cross-coupled slopes rise to $\sim \pm 125$ for GSL, with smalle errors (typically ≤ 2 rad/rad). DCUHRE again yields similar means but with wider dispersions, e.g. 152.6 ± 41.8 rad/rad (RF-3, DWSh) and -145.5 ± 41.4 rad/rad (RF-15, DWSv). The ideal data for both GSL and DCUHRE reach ± 125 rad/rad at $|\theta_{\text{clock}}| = 3°$.

From Figure 25 we observe two distinct trends. The primary DWS coupling coefficient $I^{\text{rot}}_{\text{DWS}}$ shows a shallow peak near zero clocking and then decreases slowly as the clocking angle is varied. By contrast, the cross-coupled coefficient $I^{\text{rot}}_{\text{DWS}}$ deviates rapidly as clocking moves away from zero. In other words, clocking has a minimal effect on the sensitivity of the primary channel (apart from the small peak and roll-off) but introduces a significant coupling into the orthogonal one. To quantify this behaviour, a simple geometric projection model was fitted, describing how clocking redistributes the DWS coupling between the two channels:

$$I_{\mathrm{DWS}}^{\mathrm{rot}} = I_{\mathrm{DWS}}^{\mathrm{clock}} \cdot \cos(\theta_{\mathrm{clock}} + \phi), \qquad e_{\mathrm{DWS}}^{\mathrm{rot}} = e_{\mathrm{DWS}}^{\mathrm{clock}} \cdot \sin(\theta_{\mathrm{clock}} + \phi) \tag{32}$$

Here $I_{\mathrm{DWS}}^{\mathrm{clock}}$ is the *clocking-induced primary coupling coefficient*, $e_{\mathrm{DWS}}^{\mathrm{clock}}$ is the *clocking-induced cross-coupling coefficient*, θ_{clock} is the clocking angle and ϕ is a fitted phase parameter that accounts for the slight effective shift of the optimal clocking origin. For $|\theta_{\mathrm{clock}}| \leq 3^\circ$, $\cos\theta_{\mathrm{clock}}$ deviates only slightly, which explains the gentle, nearly symmetric roll-off.

The cross-coupled component, by contrast, follows a $\sin\theta_{\mathrm{clock}}$ dependence: it grows approximately linearly with the clocking angle and flips sign when the clocking direction is reversed. For small clocking the values should vanish, but in practice the fitted slopes at $\theta_{\mathrm{clock}} = 0^\circ$ scatter around a few rad/rad rather than zero. It reflects a slight effective shift of the clocking origin, most pronounced in the DCUHRE runs ($|\phi| \lesssim 0.5^\circ$). In practice, this means that the integration method influences how the QPD responsivity map is interpreted in the simulation: GSL yields a more deterministic, static mapping with tight clustering, while DCUHRE introduces greater variability and spread, making the effective alignment point less sharply defined.

When both coefficients were set to equal amplitudes, $I_{\mathrm{DWS}}^{\mathrm{clock}} = e_{\mathrm{DWS}}^{\mathrm{clock}} = \left(I_{\mathrm{DWS}}^{\mathrm{clock}}\right)_{\mathrm{max}}$, where the latter represents the primary coupling at zero clocking, the resulting R^2 values confirm that this description is highly accurate: for all datasets the fits reach $R^2 > 0.99$, often essentially unity, except the primary DWS under DCUHRE with ideal responsivity maps, for which $R^2 = -1.24$. Strangely, it is the ideal QPD for which this integration method deviates most. This suggests that under DCUHRE the interpretation of even a perfectly symmetric responsivity map is less deterministic.

For all other cases, the high R^2 values confirm that the observed behaviour results from simple geometric projection: as the QPD is clocked, the beam-rotation response is projected onto the detector axes through simple sine and cosine dependence.

In the final LISA configuration, the QPD clocking will be fixed, meaning that any deviation from the optimal orientation will influence the DWS response. Figure 25 can therefore be used as a reference to estimate the sensitivity of the DWS channels to beam rotation for a given clocking angle. The fitted trigonometric model, supported by high R^2 values, offers a straightforward means of quantifying this dependence. Together, the figure and model provide a practical way to estimate the DWS error resulting from clocking misalignment, useful for both experimental checks and design evaluations.

When combined with Figure 23, which characterizes the added contributions from QPD tip and tilt, the full beam-rotation-to-DWS coupling can be determined for any final detector orientation. In other words, Figure 25 establishes the baseline beam-rotation-to-DWS coupling as a function of clocking. To obtain the total effective coupling, the additional tip/tilt-induced clocking contribution from Figure 23 must be added on top of this baseline. In this way, the two figures together provide the complete description of how QPD orientation (clocking + tip + tilt) perturbs the DWS response.

3.1.8. Emperical model of the Cross-Coupled DWS Response

The horizontal DWS under rotation about y (α) is well described by three geometry-driven terms:

$$\mathrm{DWS}_h = e_{\mathrm{DWS}}^{\mathrm{clock}} \cdot \theta_{\mathrm{clock}} \cdot \alpha + I_{\mathrm{DWS}}^{\mathrm{off}} \cdot x_{\mathrm{off}} + e_{\mathrm{DWS}}^{\mathrm{tip/tilt}} \cdot (\mathrm{tip} \cdot \mathrm{tilt}) \cdot \alpha, \tag{33}$$

with QPD clocking θ_{clock} (rad), beam rotation α (rad), and horizontal offset x_{off} (m). $e_{\mathrm{DWS}}^{\mathrm{clock}}$, $e_{\mathrm{DWS}}^{\mathrm{tip/tilt}}$, and $I_{\mathrm{DWS}}^{\mathrm{off}}$ are the corresponding coupling coefficients. For completeness, the trigonometric variant

$$\mathrm{DWS}_h = e_{\mathrm{DWS}}^{\mathrm{clock}}\, \sin\theta_{\mathrm{clock}}\, \alpha + I_{\mathrm{DWS}}^{\mathrm{off}}\, \cos\theta_{\mathrm{clock}}, x_{\mathrm{off}} + e_{\mathrm{DWS}}^{\mathrm{tip/tilt}}\, (\mathrm{tip} \cdot \mathrm{tilt} \cdot \alpha) \tag{34}$$

was tested. But for small angles for $|\theta_{\mathrm{clock}}| \leq 3^\circ$ both forms are equivalent in fit quality ($\Delta R^2 < 3 \times 10^{-4}$).

For DWS_v, the α and x_{off} terms in equation (33) have to be replaced with β and y_{off} respectively.

Validation. The validation was performed on the parameter space of Section 3.1.4, which covers larger tip/tilt ranges. The fit was performed hierarchically: Each new term was added to the residuals of the previous stage—so that the ΔR^2 quantifies the *fraction of previously unexplained variance* explained by that term, while the R^2 gives the total model fit so far[1] :

$$\mathrm{DWS}_h = \begin{cases} e_{\mathrm{DWS}}^{\mathrm{clock}}\,\theta_{\mathrm{clock}}\,\alpha \\ e_{\mathrm{DWS}}^{\mathrm{clock}}\,\theta_{\mathrm{clock}}\,\alpha + I_{\mathrm{DWS}}^{\mathrm{off}}\,\theta_{\mathrm{clock}}\,x_{\mathrm{off}} \\ e_{\mathrm{DWS}}^{\mathrm{clock}}\,\theta_{\mathrm{clock}}\,\alpha + I_{\mathrm{DWS}}^{\mathrm{off}}\,\theta_{\mathrm{clock}}\,x_{\mathrm{off}} + e_{\mathrm{DWS}}^{\mathrm{tip/tilt}}\,(\mathrm{tip}\cdot\mathrm{tilt}\cdot\alpha) \end{cases} \tag{35}$$

For the measured dataset: $R^2:\ 0.9927 \rightarrow 0.9989\ (\Delta R^2 = 0.849) \rightarrow 0.9994\ (\Delta R^2 = 0.455)$with $e_{\mathrm{DWS}}^{\mathrm{clock}} = 2.32 \times 10^{-3}$,$e_{\mathrm{DWS}}^{\mathrm{off}} = 2.13 \times 10^{-5}$,$e_{\mathrm{DWS}}^{\mathrm{tip/tilt}} = -7.48 \times 10^{-4}$

For the ideal dataset: $R^2:\ 0.993 \rightarrow 0.9996\ (\Delta R^2 = 0.932) \rightarrow 1.000\ (\Delta R^2 = 0.9998)$ with $e_{\mathrm{DWS}}^{\mathrm{clock}} = 2.39 \times 10^{-3}$, $e_{\mathrm{DWS}}^{\mathrm{off}} = 2.19 \times 10^{-5}$, $e_{\mathrm{DWS}}^{\mathrm{tip/tilt}} = -7.25 \times 10^{-4}$.

Thus, the clocking×rotation term dominates; the offset term adds a small but systematic correction; the tip×tilt×rotation term is minor in measured data yet cleanly visible in ideal conditions.

Robustness. Analysis on the parameter space of Section 3.1.7 indicates that RF frequency (3/15/30 MHz), DC vs. RF readout, and channel do not materially change DWS_h once θ_{clock}, α, x_{off}, tip, and tilt are included. The choice of numerical integration chiefly affects dispersion (GSL tight; DCUHRE broader) rather than the term hierarchy. See Appendix Fig. 41 for residual distributions.

3.1.9. QPD LPS-Offset-rotation Coupling

The characteristic dependence of longitudinal phase shift (LPS) on beam geometry was first suggested by the data shown earlier (Section 3.1.4 Figure 21, or Appendix, Fig. A), where horizontal beam offset and beam rotation (β) act jointly to produce strong LPS signals. This motivated a parametric fit of the form

$$\mathrm{LPS}\,[\mathrm{pm}] \;=\; k_{\mathrm{LPS}} \cdot \mathrm{beam\_offset}\,[\mu\mathrm{m}] \cdot \mathrm{beam\_rotation}\,[\mu\mathrm{rad}], \tag{36}$$

with k_{LPS} defined as the *LPS coupling coefficient*. The coefficient is reported in units of pm/(µm·µrad). By construction, multiplying µm and µrad yields 10^{-12}, which matches the scale of 1 pm, so fitted values of k_{LPS} are naturally of order unity.

For offsets and rotations in the same axis plane, e.g. horizontal offsets combined with rotation around the y-axis (β), the fit performs very well, with R^2 values consistently in the range 0.92–1.00. The choice of integration method strongly influences the recovered coupling coefficient. GSL produces clean and highly consistent results across all channels, RF frequencies, and detection modes (DC/RF), with a mean k_{LPS} of $-0.831 \pm 0.003 \quad \mathrm{pm}/(\mu\mathrm{m}\cdot\mu\mathrm{rad})$. In particular, the modulation frequency has no measurable influence, and the distributions confirm that DC versus RF readout and different channels yield essentially identical values within the statistical uncertainty. As a reference the ideal QPD has a $k_{LPS} = -0.8366 \pm 0.0001$.

DCUHRE yields a similar mean value $-0.884 \quad \pm 0.053\mathrm{pm}/(\mu\mathrm{m}\cdot\mu\mathrm{rad})$, but with substantially larger error. In this case, slopes vary more strongly across channels and detection modes. This broader distribution is evident in the coupling coefficient distributions shown in Appendix Fig. 42, where GSL clusters tightly while DCUHRE exhibits pronounced scatter.

Other experimental parameters were also examined. Varying the clocking angle and the applied tip/tilt values had no measurable influence on k_{LPS}, and changing the nominal tilt (0° vs. 2°) did not alter the coupling strength either. Overall, the results demonstrate that the LPS–offset–rotation coupling is a stable and reproducible feature of the simulation, with its numerical precision primarily limited by the integration method.

[1] As in Equation (32), $\sin(\theta_{\mathrm{clock}} + \phi)$ was used instead of $\sin\theta_{\mathrm{clock}}$, with ϕ accounting for a small shift of the clocking origin.

3.2. QPD size and gap analysis

Responsivity mapping	**QPD**	**Beam**
Type: DC	Tip, tilt, clocking: $0°$	Rotation axes: α and β
Frequencies: 30 MHz	Radius: 500, 750, 1000μm	Rotation range: $[-100,\ 100]$ μrad
Repetitions: 1 per setting	Gap size: 20, 40 μm	Beam offset axes: x and y
		Beam offset: $[-100,\ 100]$ μm

Up to this point, the responsivity maps used were based on QPD dimensions currently intended for LISA (a radius of 750 μm and an inter-quadrant gap of 20 μm). In this section, we extend the analysis by investigating how variations in detector size and gap width affect the signals. For this purpose, responsivity maps were obtained from real QPD measurements in the laboratory. The QPD radii studied were 500, 750, and 1000 μm, and the gap sizes considered were 20 and 40 μm. In all preceding sections, beam rotation and offset were assessed in isolation—typically evaluating α or β separately. The present analysis explores the full multidimensional parameter space. Both rotation axes (α, β) and beam offsets (x, y) are varied simultaneously over their full ranges, allowing interdependencies between these parameters to be captured.

During these measurements, the detector bias voltage was set to 10 V. However, full depletion of the diodes requires a bias of approximately 15 V, and the LISA requirement is 24 V. As a consequence, only DC responsivity could be characterized reliably: while DC signals are valid at 10 V, the RF signals require near-complete depletion to produce useful responsivity maps. For this reason, the present analysis focuses on DC.

As described in section 2.1.1 and section 3.1.1, the quadrant gap lines were not perfectly perpendicular: Ideally, a pure α rotation should couple only DWS_v, while a pure β rotation should couple only DWS_h. In practice, both signals showed small responses under single-axis rotation, meaning DWS_h and DWS_v were never both zero at the same clocking angle. Although the magnitude is small, it represents a systematic error contribution that must be considered.

Here, the effect of QPD size and inter-quadrant gap size is analyzed through four signals: DWS, LPS, DPS and contrast. The last two terms are included in this study because they are standard metrics for evaluating QPD performance [39][71]. The explicit definitions of these signals were given in Section 1.3.2.

Data fitting So far, models for how DWS and LPS couple to beam and detector parameters were derived in Section 3.1.8 (Eq. 33), Section 3.1.6 (Eq. 27 and 30), Section 3.1.7 (Eq. 31), and Section 3.1.9 (Eq. 36). Here, combine and extend these models to allow a full comparison across QPD radii and gap sizes. The corresponding signals, presented in Appendix I, reveal dependencies (linear, quadratic, bilinear, etc.) on the beam orientation and offset parameters.

$$\mathrm{DWS_h} = I^{\mathrm{rot}}_{\mathrm{DWS}} \cdot \beta + I^{\mathrm{off}}_{\mathrm{DWS}} \cdot x_{\mathrm{off}} + e^{\mathrm{rot}}_{\mathrm{DWS}} \cdot \alpha + e^{\mathrm{off}}_{\mathrm{DWS}} \cdot y_{\mathrm{off}} \tag{37}$$

$$\mathrm{DWS_v} = I^{\mathrm{rot}}_{\mathrm{DWS}} \cdot \alpha + I^{\mathrm{off}}_{\mathrm{DWS}} \cdot y_{\mathrm{off}} + e^{\mathrm{rot}}_{\mathrm{DWS}} \cdot \beta + e^{\mathrm{off}}_{\mathrm{DWS}} \cdot x_{\mathrm{off}} \tag{38}$$

Here, I denotes the primary coupling coefficients: I^{rot} in [rad/rad] and I^{off} in [rad/m]. The e coefficients represent error or cross-coupling contributions. These cross-couplings are small but non-negligible and therefore included to capture the expected residual errors in the signal. A more general expression is

$$\mathrm{DWS} = I^{\mathrm{rot}}_{\mathrm{DWS}} \cdot \mathrm{rotation}_{\parallel} + I^{\mathrm{off}}_{\mathrm{DWS}} \cdot \mathrm{offset}_{\parallel} + e^{\mathrm{rot}}_{\mathrm{DWS}} \cdot \mathrm{rotation}_{\perp} + e^{\mathrm{off}}_{\mathrm{DWS}} \cdot \mathrm{offset}_{\perp} \tag{39}$$

Where the last two terms represent the cross-coupled DWS:

$$\mathrm{DWS}_{\mathrm{cross}} = e^{\mathrm{rot}}_{\mathrm{DWS}} \cdot \mathrm{rotation}_{\perp} + e^{\mathrm{off}}_{\mathrm{DWS}} \cdot \mathrm{offset}_{\perp}, \tag{40}$$

with

$$e^{\mathrm{rot}}_{\mathrm{DWS}} = e^{\mathrm{clock}}_{\mathrm{DWS}} \cdot \theta_{\mathrm{clock}} + e^{\mathrm{tip/tilt}}_{\mathrm{DWS}} \cdot \mathrm{tip} \cdot \mathrm{tilt}, \tag{41}$$

as defined in section 3.1.2.

A layered fitting approach was applied to equation (39), in which each term was fitted sequentially to the data. After each fit, the residuals were analyzed and then used as input for fitting the next term. For DWS, the first term ($I^{\mathrm{rot}}_{\mathrm{DWS}}$) alone gave $R^2 = 0.99998$. Fitting the residuals with the second term ($I^{\mathrm{off}}_{\mathrm{DWS}}$) improved the agreement by $\Delta R^2 = 0.83$. The third and fourth terms ($e^{\mathrm{rot}}_{\mathrm{DWS}}$ and $e^{\mathrm{off}}_{\mathrm{DWS}}$) were added in the same way, contributing $\Delta R^2 = 0.35$ and 0.50, respectively. This stepwise procedure resulted in a final combined fit with $R^2 = 0.999996$.

For LPS, the following model was used:

$$\mathrm{LPS} = k^{\alpha}_{\mathrm{LPS}} \cdot (\alpha \cdot y_{\mathrm{off}}) + k^{\beta}_{\mathrm{LPS}} \cdot (\beta \cdot x_{\mathrm{off}}) + k^{x}_{\mathrm{LPS}} \cdot x^2_{\mathrm{off}} + k^{y}_{\mathrm{LPS}} \cdot y^2_{\mathrm{off}} + e^{\alpha}_{\mathrm{LPS}} \cdot \alpha + e^{\beta}_{\mathrm{LPS}} \cdot \beta \tag{42}$$

the coefficients k^{α} and k^{β} are expressed in pm/(μrad·μm), while k^{x} and k^{y} are given in 1/μm^2. Using the same layered R^2 analysis, the fits showed good agreement: the first term yielded $R^2 = 0.4997$, followed by improvements of $\Delta R^2 = 0.9994$, 0.1560, 0.0525, 0.4896, and 0.8093 across successive terms. The final fit reached $R^2 = 0.999998$ for real QPD responsivities and $R^2 = 1$ for the ideal case.

$$\mathrm{DPS_h} = I_{\mathrm{DPS}} \cdot x_{\mathrm{off}} + e_{\mathrm{DPS}} \cdot y_{\mathrm{off}} \tag{43}$$

$$\mathrm{DPS_v} = I_{\mathrm{DPS}} \cdot y_{\mathrm{off}} + e_{\mathrm{DPS}} \cdot x_{\mathrm{off}} \tag{44}$$

Here, I_{DPS} is the primary DPS coupling [1/m], while e_{DPS} denotes cross-coupling [1/m]. For DPS, the first term achieved $R^2 = 0.99998$, with the second term contributing only $\Delta R^2 = 0.16$, giving a total $R^2 = 0.99998$.

$$\mathrm{c} = 1 - \left(k^{\alpha}_{\mathrm{c}} \cdot \alpha^2 + k^{\beta}_{\mathrm{c}} \cdot \beta^2 + k^{x}_{\mathrm{c}} \cdot x^2_{\mathrm{off}} + k^{y}_{\mathrm{c}} \cdot y^2_{\mathrm{off}}\right) \tag{45}$$

For contrast, the coefficients k^{α} and k^{β} describe quadratic coupling to angular rotations [1/μrad^2], while k^{x} and k^{y} describe quadratic coupling to beam offsets [1/μm^2]. The layered fit gave partial R^2 values of 0.63, 0.43, 0.63, and 0.992, ending with a total $R^2 = 0.99994$ for both real and ideal responsivities.

In summary, the layered fitting procedure confirms that the proposed models describe the real QPD responsivity data with very high accuracy: for DWS and LPS, final R^2 values exceeded 0.99999, while for DPS and contrast the models reached 0.99998 and 0.99994, respectively. These results show that the models reproduce the measured behavior very well and provide a reliable foundation for the analysis that follows.

QPD size and gap-size study Figure 26 shows the coefficient, as described above, vs QPD-size. They provide an overview of how the main couplings scale with QPD radius and for two gap sizes (20 and 40 µm). Each panel shows one pair of related coefficients on the same axes. Colors separate mode ("real" vs. "ideal") and gap size. Several panels display a monotonic pattern that increases steadily with size but with a decreasing rate (the primary couplings), while others are flatter or more scattered (the error/cross terms).

The cross-coupling coefficients show no systematic dependence on detector radius or gap size. Their values fluctuate around zero without a consistent trend but remain finite, indicating a small residual coupling that persists across all configurations.

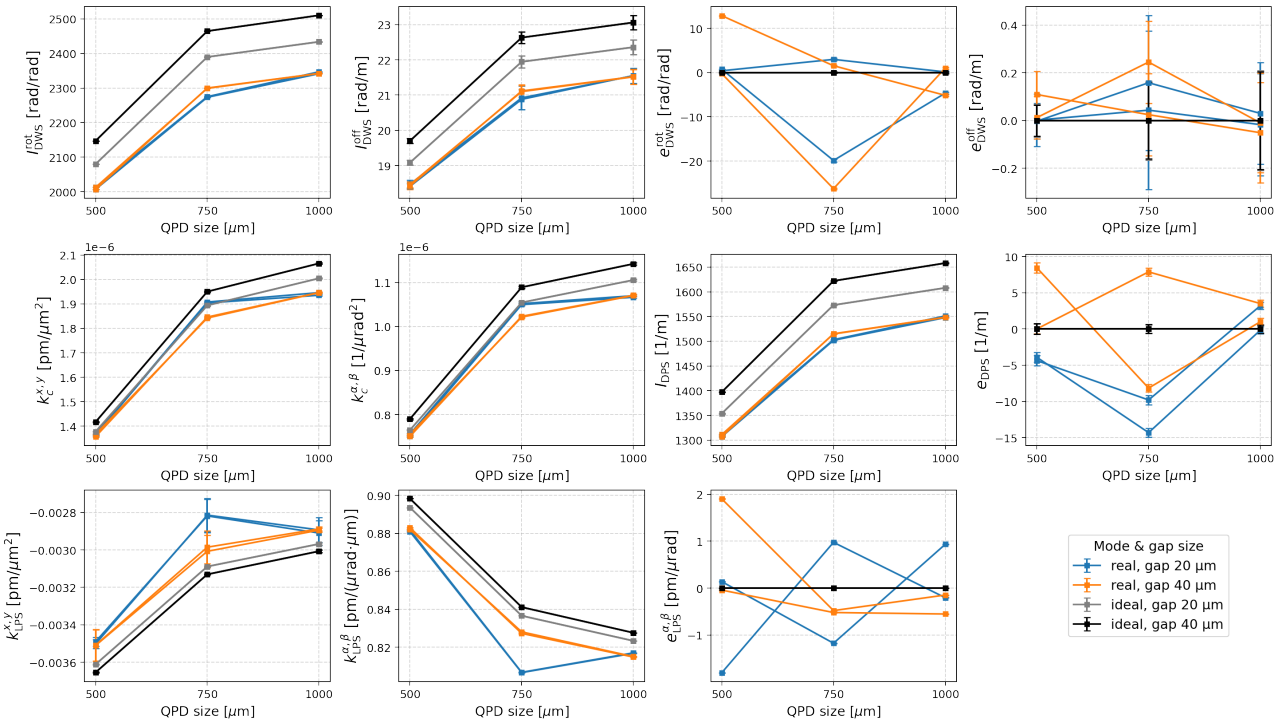


Figure 26: Fitted coupling coefficients versus QPD radius for two gap sizes (20 and 40 µm). Each panel shows one coefficient pair as a function of detector radius (500, 750, 1000 µm). Colors and markers distinguish between real and ideal responsivity maps and between the two gap sizes.

In summary: As the detector radius increases, the primary DWS and DPS couplings both strengthen, while the corresponding cross-coupling terms show no systematic dependence on size or gap width and fluctuate randomly around zero. This indicates that larger QPDs become more sensitive to the desired signals without a proportional increase in spurious coupling. The contrast terms, however, also increase in magnitude with detector size, implying slightly greater susceptibility to contrast loss. The LPS couplings are the only families with negative exponents, meaning their magnitude decreases for larger detectors; an advantageous trend, as it reduces the coupling of the longitudinal phase signal to unwanted beam or alignment variations.

Each coefficient family was evaluated to quantify the differences between the real and ideal detectors. Three consistent patterns were observed:

- For the **primary DWS and DPS couplings**, the real responsivities are consistently smaller. The mean reduction is about -3.9% at 20 µm gap and -6.6% at 40 µm gap. This applies equally to DWS rotation, DWS offset, DPSh, and DPSv. In other words, real detectors deliver $\approx$ 4–7% lower primary sensitivity than predicted by the ideal responsivity maps, with the shortfall depending on gap width.

- For the **LPS cross terms**, the percentage differences are smaller, about -1.9% (20 µm gap) and -1.6% (40 µm gap). The effect is weak and nearly gap-independent.

- For the **contrast terms**, two patterns emerge. With 20 µm gap, the shortfall is about -1.2% to -1.8%. With 40 µm gap, the deficit increases to -5% to -6%. Thus, increasing the gap size systematically degrades contrast sensitivity in the real devices compared to the ideal maps.

These bands ($\approx -6.6\%$, $\approx -3.9\%$, and $\approx -1.7\%$) summarize the overall differences between real and ideal responsivities. The primary DWS and DPS couplings form the largest group, the LPS cross terms the smallest, and the contrast terms show a gap-dependent transition from the small to the large band.

3.3. Tophat Beams

3.3.1. IfoCAD

To create, gauge, and adjust the tophat beams for the desired application, changes had to be made to the existing IfoCAD software, these are all specified in Appendix J. These software modifications were required to support the Gaussian Beam Decomposition (GBDC), Multi-Element Mode (MEM), and Numerical Mode Fitting (NMF) methods.

The GBDC beam in IfoCAD is made up of many smaller Gaussian beams. Due to this construction, IfoCAD normally blocks any commands that would move or rotate the beam. Allowing such operations would disturb the relative spacing of the Gaussians and break the tophat composition. To work around this restriction, new functionality was implemented. A method was added to rotate the entire group of Gaussians around a chosen pivot point. Another method was introduced to translate the beam by shifting every Gaussian together, so their relative positions remain unchanged. In addition, helper functions were added to return the beam origin and direction, and to apply translations and rotations consistently to the beam and its sub-beams. All these changes are presented in Listing 1 and 2 in Appendix J.

MEM and NMF both used the Hermite Gaussian Beam (HG-Beams) library from IfoCAD. As described in section 2.3, the HG-Beams are then pruned: modes with negligible contributions are discarded. The library had to be edited so that mode coefficients in a contstructed sparse matrix could be removed. Second, a metadata loading function was implemented. This function reads in a text file containing the q-parameters, the beam origin and direction, and the waist radii. The q-parameters are complex beam parameters used in Gaussian beam optics to describe both the radius of curvature of the wavefront and the beam waist (spot size). With this information, the beam could be reconstructed later with the same geometry and propagation state. All these changes are presented in Listing 3 and 4.

Lastly, in the Photo Diode Signals (PD_signals) library, the selection of the phasemeter integration method was modified (Listing 5). The original code forced the use of the GSL single-dimension integrator whenever response data was present. This line was disabled, so the integration method is no longer overwritten. The change was necessary because GBDC beams could not be handled with the GSL integrator, and therefore a more flexible selection of integration methods was required.

3.3.2. Beam profiles

This section introduces the beam profiles obtained with the three construction methods: NMF, MEM, and GBDC. Each method has a tunable complexity set by the number of spatial modes or beamlets, N_{modes}. To study convergence and typical artefacts, The beams were generated and saved for $N_{\text{modes}} = 10, 20, 40, 80, 50, 100, 150, 200, 300, 400$.

Two variants were considered througout:

trunc (short for truncation; aperture at $r = 0.5$): The profile reproduces the desired footprint, with the intensity forced to zero outside the aperture. This gives the correct edge location, but the abrupt cut makes the reconstruction more difficult: just inside the radius the intensity often overshoots (shoulders), and the otherwise flat top develops small ripples.

notrunc (flat-top Gaussian): The profile has a smooth flat top and then decays slowly without a sharp cut. No hard aperture is imposed, the interior remains clean and free of strong ripple or shoulders, even at modest N_{modes}. The drawback is that the edge is soft: the footprint extends gradually outward, so the beam does not have a well-defined radius and the wings contribute extra power outside the intended disk.

The trade-off between prioritizing a perfectly flat top, or a sharply defined edge at the target radius is not yet defined, in the rest of this work I carry both variants forward and evaluate them separately.

In Figure 27, we zoom in on a few representative beam profiles to make the earlier points concrete.

The examples span the three construction routes—MEM, NMF, and GBDC—and include both *trunc* and *notrunc* variants. Together they show the basic trade-off we discussed: hard edges are good for footprint, but they invite ringing; soft edges keep the top clean, but blur the boundary. A full overview of all generated beams can be found in Appendix K.

Each row in the figure shows three maps on the same x-y grid: 1. The *intensity* $|E|^2$, 2. the *difference to the corresponding ideal* profile (Simulated - Ideal intensity), and 3. the Phase. For trunc, the ideal is a power-matched disk of radius $r = 0.5$. For $\mathrm{notrunc}$, the ideal is a power-matched flat-top Gaussian (Gaussian amplitude clipped at a fixed level). The difference panel is the most informative: red/blue rings and lobes tell you where the reconstruction deviates and in what sense (over/under).

MEM, 10 modes, trunc. At very low mode count, MEM's explicit aperture produces the desired footprint, but the interior is not perfectly flat. You can see gentle bulging just inside the edge and faint ripple across the top. In the difference map that becomes a set of concentric rings: overshoot inside the radius and undershoot outside. The phase is mostly smooth across the core and picks up curvature only near the aperture where the field is cut.

NMF, 10 modes, trunc. The NMF reconstruction shows the same basic edge behaviour, but the ripple pattern mirrors the Hermite–Gaussian basis: square-symmetry features are visible on top of the disk. The difference panel again highlights a ring at the edge, now with axis-aligned lobes that reflect the limited set of modes. This is the "finite basis" effect from before—like approximating a step with a small number of Fourier terms. The phase shows a faint cross-like structure only outside of the aperture.

GBDC, 300 modes, notrunc. Removing the hard stop changes the picture. The intensity rises to a broad plateau and rolls off smoothly; shoulders are much weaker because there is no abrupt discontinuity to reproduce. The difference to the flat-top Gaussian is small and spread over the transition region rather than concentrated at a sharp boundary. Phase is clean across the aperture with only a mild beamlet imprint. The phase is mostly uniform across the plateu with only mild, very fine oscillations, and a broader phase change near the edge.

GBDC, 300 modes, trunc. With many more components the structure tightens. The interior looks flatter than in the low-mode cases, and most of the mismatch collapses into two thin rings right at the aperture. In the difference panel you see a narrow red/blue rim rather than broad oscillations-evidence that increasing N_{modes} pushes the error to the edge, but does not completely remove the edge ringing itself: The phase is uniform over most of the top and only varies rapidly near the boundary.

Power & Mean-Squared Error (MSE) analysis

While varying the number of modes of the generated beams, the overal power was calculated over the grid. The generated tophat beam should have the same power as the Gaussian beam used in the previous simulation expiriments = 0.5 W.

In addition, we quantify how closely a simulated beam $E_{\text{sim}}(x, y)$ matches a reference profile $E_{\text{ideal}}(x, y)$ on the same discrete x–y grid. The (per–coordinate) mean–squared error on *amplitude* is

$$\mathrm{MSE} = \frac{1}{N} \sum_{i=1} \sum_{j=1} \left(E_{\text{sim}}(x_i, y_j) - E_{\text{ideal}}(x_i, y_j) \right)^2 . \tag{46}$$

Where N here is the total number of points on the grid (not to be confused with N_{modes}). It is computed on field amplitudes (not intensities), and it penalizes spatial mismatches everywhere on the grid.

We evaluate each simulated map against an ideal that reflects the design target. Two E_{ideal} targets are used throughout. 1. A Truncated (tophat) ideal: A disk of radius $r = 0.5$ with constant amplitude h inside and zero outside, So $E_{\text{ideal}}(x, y) = h$ for $x^2 + y^2 < r^2$. To keep the overall scale consistent

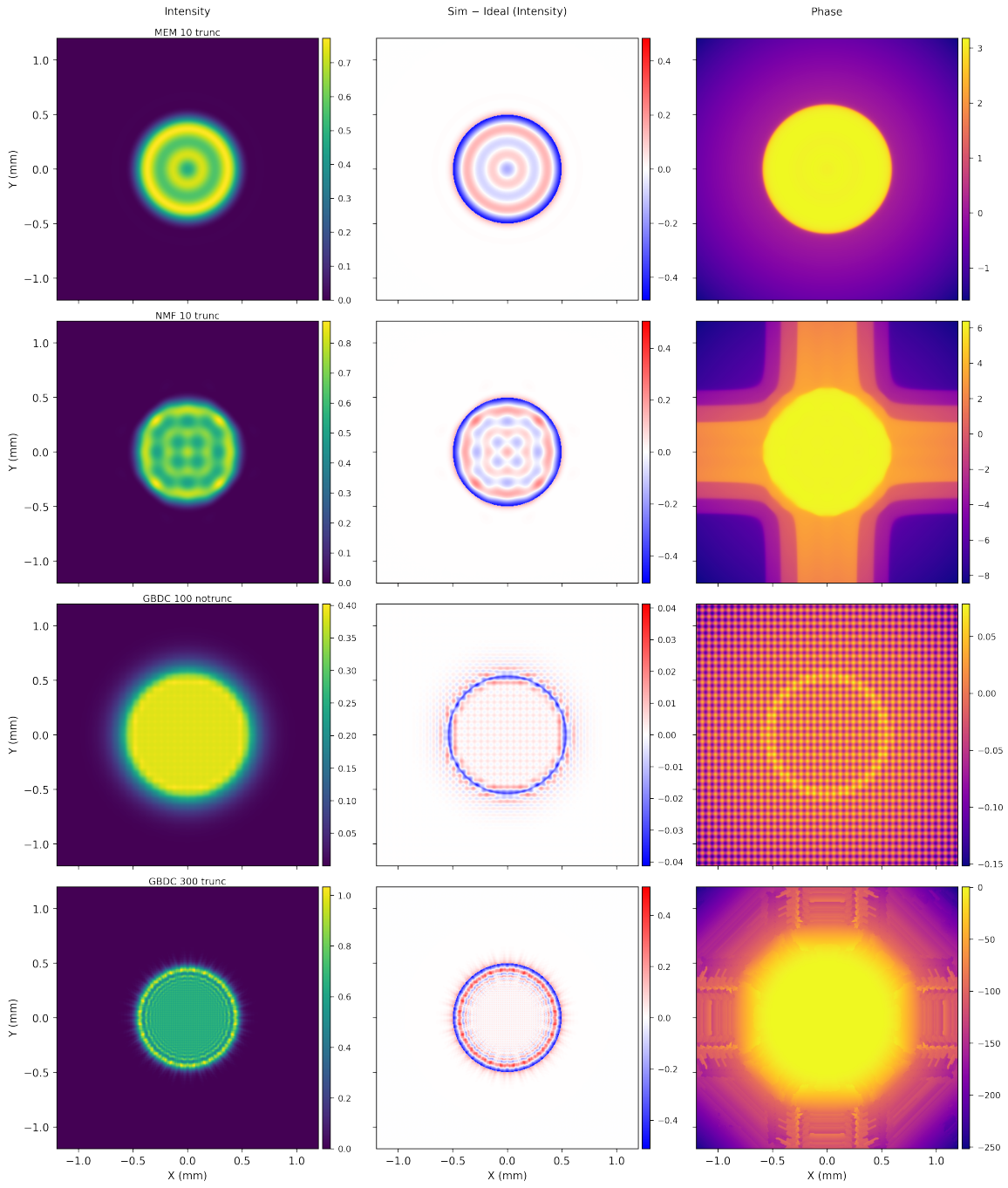


Figure 27: Beam maps on a common x–y grid (mm). Columns (left→right): intensity $|E|^2$, intensity difference "Sim − Ideal", and unwrapped phase (radians). Rows (top→bottom): MEM, $N_{\text{modes}} = 10$ (trunc); NMF, $N_{\text{modes}} = 10$ (trunc); GBDC, $N_{\text{modes}} = 50$ (notrunc); GBDC, $N_{\text{modes}} = 300$ (trunc). Ideals: trunc → power-matched disk of radius $r = 0.5$; notrunc → power-matched flat-top Gaussian (Gaussian amplitude clipped at a fixed level).

with our data products, $h = \sqrt{0.5/\pi r^2} \approx 0.798$ is chosen such that the ideal carries a total power of 0.5:

2. A Non–truncated (flat–top Gaussian) ideal: A circular Gaussian amplitude $E_b(r) = \exp(-r^2/r_a^2)$ with $r_a = 0.5$ is clipped at a fixed at level $a_c = 0.368 = E_b(r_a = 0.5)$ to form a flat plateau and Gaussian-like shoulders; $\tilde{E}(r) = s \cdot \min\big(E_b(r),\, a_c\big)$. A single global scale factor s is then applied so that the total power of the ideal equals 0.5. This construction preserves a flat center while retaining a Gaussian tail consistent with a standard Gaussian of "radius" $r_a = 0.5$.

Figure 28 contains nine panels. Columns from left to right represent NMF, MEM, and GBDC. The top row depicts the total power of the beam in Watts. The middle row the amplitude MSE

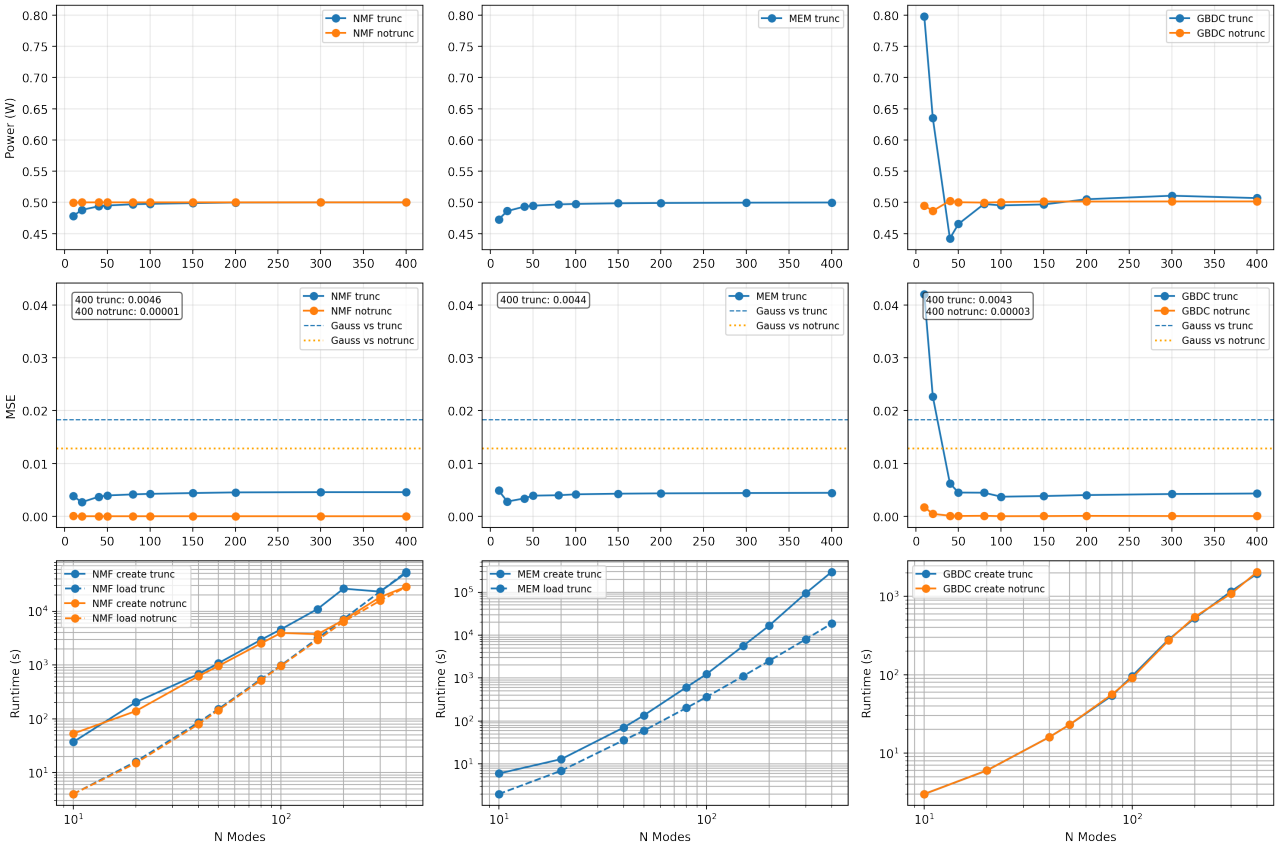


Figure 28: Top row: Beam power (W) versus number of modes N_{modes}. Middle row: Mean-squared error (MSE) of beam amplitude versus number of modes N_{modes}. Bottom row: runtime versus. Blue curves: trunc. Orange curves: notrunc. MSE is calculated for trunc — error relative to the ideal tophat (disk, $r = 0.5$, height $h \approx 0.798$, power 0.5). notrunc — error relative to the ideal flat-top Gaussian (Gaussian clipped at 0.3133 and globally scaled to power 0.5). Dashed horizontal line: MSE of a plain Gaussian vs. trunc ideal. Dotted horizontal line: MSE of a plain Gaussian vs. notrunc ideal.

described above in (46), and the bottom row the computation time. The horizontal axis is the number of modes N_{modes} used by the method. For NMF and GBDC, we have two colors; the truncated tophat (trunc) and the flat–top Gaussian ideal (notrunc). For MSE, two horizontal reference lines are drawn in every panel: "Gauss vs trunc" (dashed): MSE of a plain normalized Gaussian (no clipping) against the tophat ideal; here 0.0183136. "Gauss vs notrunc" (dotted): MSE of the same plain Gaussian against the flat–top Gaussian ideal; here 0.0128640. These lines serve as a baseline: if a purported tophat reconstruction cannot beat the "plain Gaussian" error, it is not achieving a genuinely tophat-like shape.

At low N_{modes} the curves move noticeably. As N_{modes} increases the values stabilize. In the low–mode regime, particularly for GBDC with small beamlet counts, the truncated curves can sit above 0.5 W, and above the Gaussian baseline to qualify as a good tophat. As the number of beamlets increases, the GBDC truncated curve drops below the baseline and continues down toward the 4×10^{-3} range. All 400-mode values are well below the "Gauss vs trunc" baseline; the notrunc values are also well below the "Gauss vs notrunc" baseline.

The number of modes affects not only how accurately a beam reproduces the desired tophat profile but also the computational cost. To evaluate the latter, the creation and loading times were measured for all three construction methods, recording both the time to build and to reload each beam.

The near-linear trends indicate an exponential increase in runtime with the number of modes. For NMF, loading remains faster than creation up to about 150 modes for the non-truncated case and up to roughly 300 modes for the truncated case. Although this effect might appear to be a one-time computational artifact, it persisted consistently across three independent runs. While individual runtimes

varied slightly, the overall scaling and relative timing remained stable. For MEM, the creation of the beam is significantly slower than the loading of the beam. For GBDC, no discrete way to save the beam into beam coefficient was found, however, the creation of the beam was orders of magnitude faster than the other models anyway. The load time for MEM with $N_{\text{modes}} = 10$ is approximately 1 second. Using this as a normalization point, all other runtimes in Figure 28 (bottom row) can be interpreted as a dimensionless scaling function that reflects how construction and loading times grow with increasing mode count and differ across methods.

This plot gives us a factor to weigh into the choice of what beams to use for our simulations, later we assess the actual performance of the beam types, so in combination with this will give guidance to what beam type to uses for both computational efficiency and result performance.

3.3.3. primary DWS

Responsivity mapping	**QPD**	**Beam**
Type: DC, RF	Tip: $[-0.5^\circ, 0.5^\circ]$	Rotation axes: α or β
Frequencies: 3, 15, 30 MHz	Tilt: 0° or $2^\circ \pm 0.5^\circ$	Rotation range: $[0, 100]$ μrad
Repetitions: 3 per setting	Clocking: $[0^\circ, 3^\circ]$	Beam offset: $[0, 30]$ μm
	Integration: GSL or DCUHRE	Type: [Gauss, GBDC, NMF, MEM]
		Truncation: [trunc, notrunc]
		N-modes: [10,20,50,100,200,300]

We now analyze the simulated interaction between the tophat beam generated by the GBDC, MEM and NMF methods and the QPD. In this analysis the QPD geometry is kept fixed at a radius 750 µm radius and 20 µm QPD, consistent with Sections 3.1.1 - 3.1.9. As before, we distinguish ideal vs real responsivity maps, for real responsivity maps we distinguish DC and RF signals for 3 MHz, 15 MHz, and 30 MHz and both integration methods (DCUHRE and GSL). In addition to these existing parameters, we now include both truncation settings (trunc/notrunc) and vary the mode count N_{modes}. The analysis first focuses on the primary beam-rotation-to-DWS coupling $I^{\text{rot}}_{\text{DWS}}$, (DWS$_v$ & α) and (DWS$_h$ & β).

The GBDC results require some clarification before proceeding. Compared to MEM and NMF, the GBDC data exhibit a unworkable broad spread and are only reliable under nominal alignment conditions (tip = 0°, tilt = 0°; see Appendix L). Even under these conditions, DWS$_v$ converges to 0 as N_{modes} increases, leaving only DWS$_h$ for analysis.

Figure 29 plots the primary beam-rotation–to–DWS coupling $I^{\text{rot}}_{\text{DWS}}$ in rad/rad. Panel (a) shows truncation split across beam types. The data includes by NMF (both variants), MEM (truncated only), and GBDC (with the constraints stated above).

- NMF notrunc: 1098.36 ± 44.24.
- NMF trunc: 1629.24 ± 38.13.
- MEM trunc: 1630.22 ± 39.71.
- GBDC notrunc: 1611.81 ± 257.61.
- GBDC trunc: 758.77 ± 152.68.

Truncation pushes NMF up to the same level as MEM (~ 1630). GBDC shows noticeably larger spread, and the trends are reversed: The notrunc beams shows the higher coupling, and the truncated version drops to roughly half that value.

Panel (b) shows the integration method by beam type, with ideal vs real maps. Here we only look at truncated NMF and MEM, and the non-truncated GBDC (so the ones with higher coupling). For

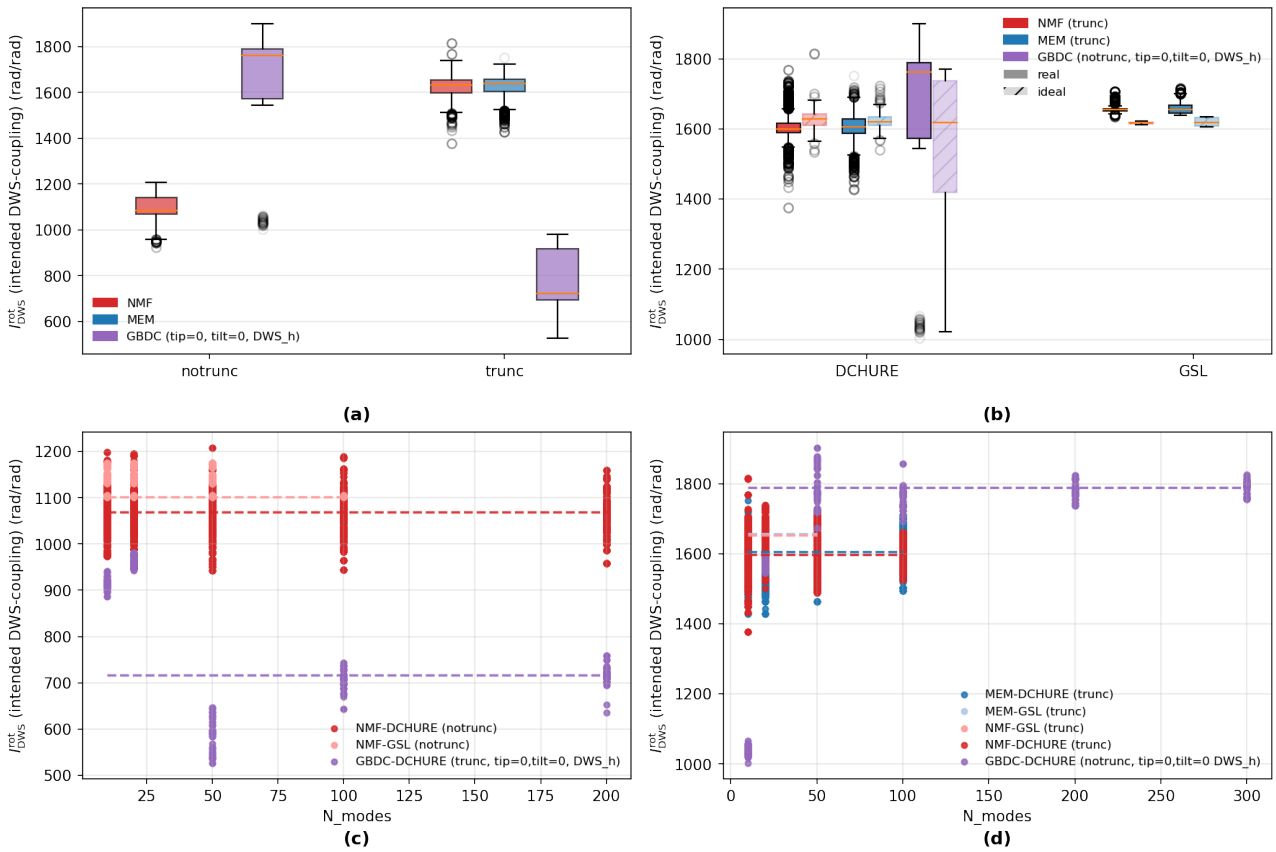


Figure 29: (a) Boxplots of primary DWS-coupling $I^{\text{rot}}_{\text{DWS}}$ by truncation for NMF, MEM, and GBDC (tip$=0$, tilt$=0$, DWS$_h$ only). (b) Boxplot of primary DWS-coupling by integration method (DCUHRE, GSL), split by Beam type and real vs. ideal responsivity; NMF/MEM shown for trunc, GBDC shown for notrunc (tip$=0$, tilt$=0$, DWS$_h$ only). (c-d) Scatter of primary DWS-coupling versus N_{modes}, dashed horizontal lines mark the mean at the largest N_{modes} shown. (c) plotted: NMF (notrunc) and GBDC (trunc, tip$=0$, tilt$=0$, DWS$_h$); (d) plotted: MEM (trunc), NMF (trunc), and GBDC (notrunc, tip$=0$, tilt$=0$, DWS$_h$).

each beam type we plot both the ideal QPD and the real responsivity map.

- **NMF (trunc):**
  - GSL — real: 1659.03 ± 11.98, ideal: 1618.32 ± 3.22
  - DCUHRE — real: 1607.64 ± 36.93, ideal: 1629.38 ± 36.42
- **MEM (trunc):**
  - GSL — real: 1660.66 ± 15.34, ideal: 1621.21 ± 10.67
  - DCUHRE — real: 1607.01 ± 38.36, ideal: 1624.39 ± 27.30
- **GBDC (notrunc):**
  - DCUHRE — real: $+1620.87 \pm 253.39$, ideal: $+1507.44 \pm 288.94$

As seen in the plot, GSL mean exceed DCUHRE for MEM/NMF by $\approx +52$. Ideal vs real differs by $\approx +40$ for GSL, and $\approx 20-60$ for DCUHRE. Additionally, GBDC shows extremely large spreads up to $\approx \pm 290$.

Panels (c) and (d) show how the primary coupling $I^{\text{rot}}_{\text{DWS}}$ values plotted against the number of modes N_{modes}. As seen in the plots, MEM and NMF are stable: truncated cases stay around 1630 and don't fluctuate; NMF notrunc stays near 1100. GBDC converges to two levels: about $+1771$ (notrunc) and about $+694$ (trunc), both with much larger fluctuation at low N_{modes}.

We also compared the real-map results across 3/15/30 MHz per data group. Comparing the 30 MHz results to the average of the 3 MHz and 15 MHz results, the difference was $\approx \pm 4$ [rad/rad]. The spread however in each data set much larger than the mean shift (± 4), so we do not observe a systematic frequency effect. Similar tests for tip and tilt dependencies showed no measurable influence.

When we compare these tophat results with the primary DWS couplings from the Gaussian-beam baseline (not plotted here), Section 3.1.7 Fig. 25(a,c), the difference is clear: The Gaussian GSL real value is about 2342 rad/rad. The truncated tophat models (NMF and MEM, GSL, real) sit near 1660 rad/rad, which is lower by 682 rad/rad ($\approx 30\%$ decrease). The non-truncated NMF GSL case is even lower, near 1098 rad/rad, i.e., a drop of 1244 rad/rad ($\approx 52\%$ decrease). Thus, the tophat beams show significantly lower primary coupling than the Gaussian, reflecting reduced rotational sensitivity.

3.3.4. Cross-coupled DWS

We now examine the cross-coupled DWS for the tophat beams. As before, cross-coupled is the pairings: DWS_h with the y-rotation (α) and DWS_v with the x-rotation (β).

Again, data from the GBDC method was only useable for tip = 0° and tilt = 0°. Previously, only DWS_h was analyzed as DWS_v converged to zero. Here however, the opposite occurs: DWS_h now vanishes, so only DWS_v is considered.

In Section 3.1.8 equation (33), DWS was expressed as a sum of three geometry-driven terms. The first term $e^{\text{clock}}_{\text{DWS}} \times$ clocking $\times$ beam-rotation was the dominant contributor for the Gaussian beams. This approximation was also tested for the tophat beams using the same fit:

$$\mathrm{DWS_h} = e^{\text{clock}}_{\text{DWS}} \times \theta_{\text{clock}} \times \alpha, \quad \text{and} \quad \mathrm{DWS_v} = e^{\text{clock}}_{\text{DWS}} \times \theta_{\text{clock}} \times \beta, \tag{47}$$

The fraction for $R^2 > 0.95$ for NMF and MEM methods are 70% and for GBDC (tip=0,tilt=0) 40%, compared to 82% for the Gaussian beams (see Appendix M for the R^2 distribution). These values indicate that the model fits the data well enough for analysis.

Figure 30 shows box plots of the coefficient $e^{\text{clock}}_{\text{DWS}}$. The x-axis lists beam types. Colors encode truncation state (blue = notrunc, orange = trunc). Gaussian has only a non-truncated variant; MEM has only a truncated variant. Colored boxes are the real responsivity-map results; the corresponding ideal results appear to the right of each real dataset. Panel (a) shows the GSL integration only.

- **Gaussian (notrunc):** Real: 2325 ± 18 rad^{-1}; Ideal: 2386 ± 1 rad^{-1}
- **MEM (trunc):** Real: 2268 ± 37 rad^{-1}; Ideal: 2307 ± 11 rad^{-1}
- **NMF (notrunc):** Real: 2781 ± 17 rad^{-1}; Ideal: 2826 ± 1 rad^{-1}
- **NMF (trunc):** Real: 2263 ± 68 rad^{-1}; Ideal: 2304 ± 8 rad^{-1}

For the Gaussian reference, the ideal value exceeds the real one by about $+61$ rad^{-1}. MEM and NMF (trunc) show similar behaviour, with their real-to-ideal differences around $+40$ rad^{-1} and overall coupling levels about 60–80 rad^{-1} below the Gaussian; about a 3% decrease. In contrast, NMF (notrunc) exhibits a stronger response, roughly $+450$ rad^{-1} above the Gaussian, $\approx 20\%$ increase. Overall, ideal values consistently exceed real ones by 40–60 rad^{-1}, and while absolute magnitudes differ across methods, the general cross-coupling trends remain comparable.

Panel (b) shows the DCUHRE integration only.

- **Gaussian:** Real: 2437 ± 80 rad^{-1}; Ideal: 2382 ± 75 rad^{-1}
- **MEM (trunc):** Real: 2304 ± 419 rad^{-1}; Ideal: 2262 ± 302 rad^{-1}
- **NMF (notrunc):** Real: 2841 ± 409 rad^{-1}; Ideal: 2851 ± 268 rad^{-1}
- **NMF (trunc):** Real: 2356 ± 725 rad^{-1}; Ideal: 2330 ± 464 rad^{-1}
- **GBDC (notrunc):** Real: 5126 ± 936 rad^{-1}; Ideal: 4960 ± 816 rad^{-1}
- **GBDC (trunc):** Real: 4461 ± 1068 rad^{-1}; Ideal: 4687 ± 1201 rad^{-1}

Across the beam types, ideal values generally exceed their real counterparts by about 20–60 rad^{-1}, though the differences are small relative to the observed spreads. MEM and NMF show broadly similar coupling magnitudes, both slightly below the Gaussian reference 2437 ± 80rad^{-1}. GBDC shows a much larger coupling of overall 4822 ± 1050 rad^{-1}, with a very large error. Truncation or non-truncation states show little consistent effect: For GBDC, the run-to-run error outweighs systematic offsets across configurations.

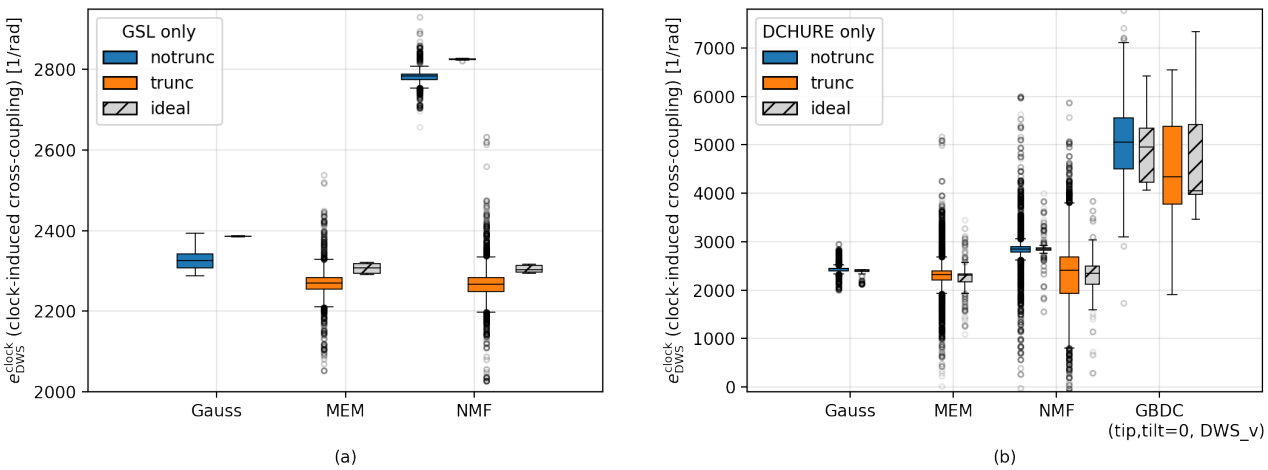


Figure 30: Box plots of the clocking-induced cross-coupling coefficient $e^{\text{clock}}_{\text{DWS}}$ [1/rad], defined by the fit $\text{DWS}_{\text{cross}} = e^{\text{clock}}_{\text{DWS}} \times \text{clocking} \times \text{beam-rotation}$. The x-axis lists beam types: Gaussian, NMF, MEM, and GBDC. Colors indicate truncation state (blue = notrunc, orange = trunc); hatched boxes mark ideal simulations; solid boxes show real responsivity-map results. Panel (a) shows results with GSL integration only. Panel (b) shows results with DCUHRE integration only.

A separate analysis was performed to examine how the measurement frequency (3, 15, or 30 MHz) affects the clocking rotation coupling: For Gaussian beams, the frequency-dependent difference occurs under DCUHRE, where the slope works out to about 1.6 rad^{-1}, though this sits within an error of at least 80 rad^{-1}. Under GSL the shifts are far smaller, between 0.03 rad^{-1} and 0.18 rad^{-1}. For NMF and MEM, the differences under GSL are essentially zero, and under DCUHRE they are only a few rad^{-1}, well inside the error. Even for GBDC, where differences can reach a couple hundred rad^{-1} across frequencies, the error are around a thousand, so the signal is buried in the noise. So to be explicit: frequency choice has only a very minor influence compared with integration method or truncation.

The influence of N_{modes} on the clocking–rotation coupling was also examined to see whether increasing the number of modes produces a systematic trend or whether the coupling remains constant within the statistical spread: Only the DCUHRE integration shows any meaningful dependence. For MEM, the coupling decreases by roughly 160 rad^{-1} between 10 and 100 modes, and for truncated NMF it drops by about 400 rad^{-1}. In all other cases the changes are small compared with the spreads, indicating no consistent mode dependence.

3.3.5. primary offset-to-DWS coupling

We estimate the primary offset–to–DWS coupling using the same linear model introduced in Section 3.1.6, Eq. (27),

$$\text{DWS}_h = I^{\text{off}}_{\text{DWS}} \cdot x_{\text{off}} + b$$

where $I^{\text{off}}_{\text{DWS}}$ denotes the primary offset-to-DWS coupling coefficient. Fits obtained with DCUHRE integration did not yield stable or interpretable results for this analysis and were therefore omitted. Only GSL integration is considered, meaning that the GBDC results are not analyzed in this subsection. We categorize by beam-construction method (MEM, NMF trunc, NMF notrunc), by responsivity map (ideal or real), and examine the dependence on N_{modes}.

The quality of the fits decreases with increasing N_{modes}. The per-category R^2 sequences below correspond to $N_{\text{modes}} = 10, 20,$ and 50 (and 100 where available), showing the mean R^2 value for each group:

- **ideal, MEM, trunc:** $0.989 \to 0.886 \to 0.704$
- **ideal, NMF, notrunc:** $0.998 \to 0.945 \to 0.928 \to 0.960$
- **ideal, NMF, trunc:** $0.991 \to 0.832 \to 0.977$

- **real, MEM, trunc:** 0.960 → 0.490 → 0.428
- **real, NMF, notrunc:** 0.787 → 0.490 → 0.527
- **real, NMF, trunc:** 0.982 → 0.463 → 0.467

For most categories, especially the real QPD maps, the linear model fits the data progressively worse at higher N_{modes}, indicating a weakening linear dependence on x_{off}. For comparison, the Gaussian-beam offset fit presented earlier had $R^2 = 0.981$ and could be regarded as linear; here, this linearity is largely lost.

Figure 31 shows the resulting coupling coefficients $I^{\text{off}}_{\text{DWS}}$ as a function of N_{modes} (mean ± std across groups). Colors distinguish MEM (blue) and NMF (red), with lighter shades for notrunc; dashed lines represent ideal maps and solid lines real maps. A grey dashed horizontal line marks the Gaussian reference mean ($\approx 21.05\ \mathrm{rad/m}$).

Numerically, the sequences (mean ± std, in $\mathrm{rad/m}$) are:

- **ideal, MEM, trunc:** 12.06±0.01 → -3.78±0.02 → -1.85±0.03
- **ideal, NMF, notrunc:** 4.41±0.01 → -0.93±0.08 → -0.01±0.01 → -0.08±0.01
- **ideal, NMF, trunc:** -38.41±0.71 → -7.02±0.33 → -3.54±0.58
- **real, MEM, trunc:** 16.74±2.51 → 2.55±2.47 → 0.18±6.12
- **real, NMF, notrunc:** 4.79±1.91 → -1.64±1.86 → 0.46±2.34
- **real, NMF, trunc:** -44.70±3.90 → 3.22±7.78 → -2.64±7.91

Across methods and map types, the magnitude of $I^{\text{off}}_{\text{DWS}}$ decreases with N_{modes} and stabilizing within a few $\mathrm{rad/m}$ of zero by 50 modes.

For sufficiently many modes, a lateral beam offset does not couple into $\mathrm{DWS_h}$ for tophat-like beams; this is in clear contrast to the Gaussian reference. Practically, this means the $\mathrm{DWS_h}$ channel becomes insensitive to pure offset, removing an error pathway and simplifying alignment control: residual responses are dominated by genuine angular terms rather than miscentering.

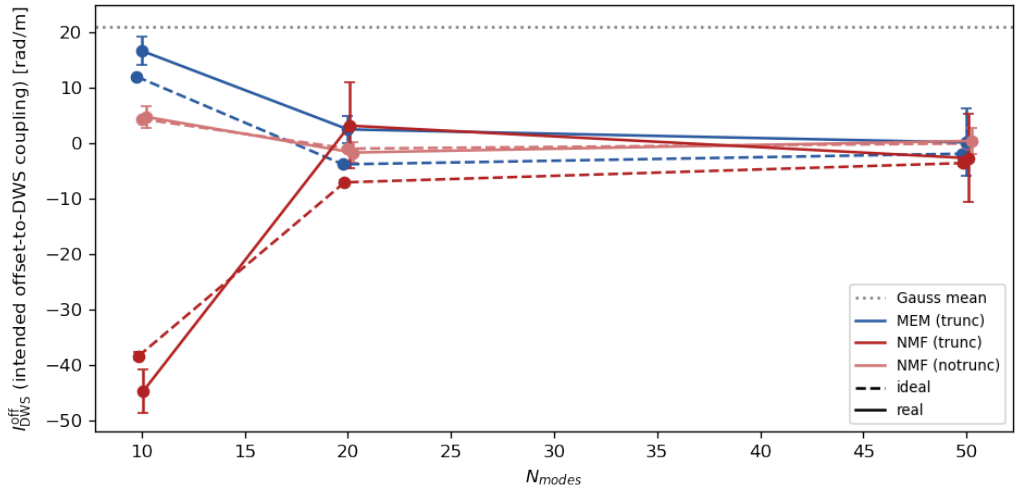


Figure 31: primary offset-to-DWS coupling coefficient $I^{\text{off}}_{\text{DWS}}$ $(\mathrm{rad/m})$ obtained from GSL integration. Colors denote beam type (blue = MEM, red = NMF), lighter shades mark notrunc, and dashed vs. solid lines indicate ideal and real responsivity maps respectively. The grey dashed line shows the Gaussian reference mean ($21.05\ \mathrm{rad/m}$).

3.3.6. LPS coupling for tophat beams

Following the Gaussian analysis in Section 3.1.9, we now probe the longitudinal pathlength signal (LPS) coupling for tophat-like beams. The GBDC variant is excluded here: even at tip$= 0$ and tilt$= 0$, the LPS scatter remained unusably large after phase wrapping. Therfor GBDC is not evaluated for LPS coupling in this section. The remaining analysis thus covers MEM (truncated) and NMF (both truncated and non-truncated), with the Gaussian shown as a reference.

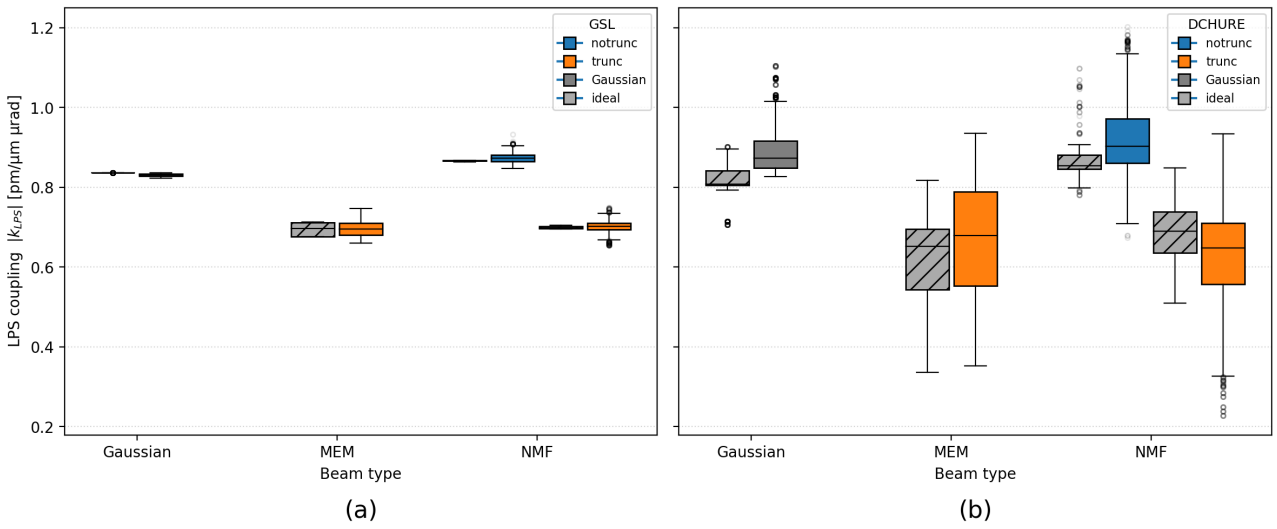


Figure 32: Boxplots of fitted $k_{\mathrm{LPS}}^{\mathrm{rot}}$ vs. beam type (Gaussian, MEM, NMF). The y-axis shows k_{LPS} $[\mathrm{pm}/(\mu\mathrm{m}\,\mu\mathrm{rad})]$. Panel (a): GSL; panel (b): DCUHRE. Blue and orange correspond to non-truncated and truncated tophat variants, respectively, and the Gaussian is plotted in grey as a reference. The fit model is LPS $\approx k_{\mathrm{LPS}} \times$ (beam rotation) $\times$ (beam offset).

In Section 3.2 equation (42), LPS was expressed a polynomial of beam rotation and offset terms. The first two bilinear terms were the most dominant contributors to the fit. This simplified form was therefore tested for the tophat beams using the model:

$$\mathrm{LPS} \;=\; k_{\mathrm{LPS}}^{\mathrm{rot}}\;(\text{beam offset}) \times (\text{beam rotation}), \tag{48}$$

Here the coupling coefficient $k_{\mathrm{LPS}}^{\mathrm{rot}}$ is estimated by linear regression (intercept allowed). Across all non-GBDC data, the fits are close to perfect for the GSL integrator: the minimum R^2 is 0.9981 and the mean R^2 is 0.9993. For DCUHRE the fits remain very strong but are modestly lower: minimum $R^2 = 0.7728$ and mean $R^2 = 0.8989$. In short, this bilinear model captures the LPS coupling moderately well for DCUHRE, and near-unity for GSL.

Figure 32 summarizes the distributions of the fitted coupling coefficients $k_{\mathrm{LPS}}^{\mathrm{rot}}$ as boxplots. The x-axis groups by beam type (Gaussian, MEM, NMF). Panel (a) shows GSL; panel (b) shows DCUHRE. (Full list of means and errors can be seen in Appendix N) Truncated beams generally show lower coupling values—around $k_{\mathrm{LPS}} \approx 0.70\ \mathrm{pm}/(\mu\mathrm{m}\,\mu\mathrm{rad})$—while non-truncated variants cluster near 0.87, slightly above the Gaussian reference of ≈ 0.83 for GSL and ≈ 0.81 for DCUHRE. This pattern is consistent across both integration methods.

The origin of the observed spread/error was investigated: For GSL, no single parameter in the scanned space provided a robust attribution; small differences between DC and RF responsivity maps were visible but did not explain variance in a consistent way. For DCUHRE, most variance traced back to *tilt*; no meaningful contributions from the other parameters were detected.

The results of these coupling coefficients matter, even if absolute differences appear small: the coupling is bilinear, so modest coefficient shifts can yield measurable LPS changes at realistic jitter bounds. In our parameter space, with maximum rotation $100\,\mu\mathrm{rad}$ and maximum offset $30\,\mu\mathrm{m}$, a coefficient $k_{\mathrm{LPS}} = 0.83$ implies an LPS excursion of $\approx 0.83 \times 100 \times 30 \approx 2.5 \times 10^3$ pm. A coefficient of 0.68 gives $\approx 2.0 \times 10^3$ pm—about **500** pm lower ($\sim 20\%$). While this sits in the picometer regime, multiple contributors can add coherently or broaden the error budget. Hence, given the near-unity R^2 values, the differences in $k_{\mathrm{LPS}}^{\mathrm{rot}}$ are likely systematic and should be accounted for when assessing the QPD path length readout.

4. Discussion

In this section the results and foremost limitations and imperfections of this research will be discussed, accompanied by recommendations for future research in this field of study.

It was observed that for Gaussian beams, a lateral beam offset produces a measurable DWS signal, whereas for tophat beams this coupling vanishes as the modal count increases. The origin of this large difference in coupling strength is not yet understood. Further investigation could be carried out through analytical modeling of the DWS response for laterally displaced Gaussian beams, complemented by numerical simulations or targeted experimental measurements.

The most immediate limitation of this study lies in the sampling resolution of the parameter space. For reasons of computational cost, only a small number of steps (typically 3, 5, or 7) were taken per dimension across broad ranges. This strategy ensured that the dominant, large-scale effects could be quantified, but it necessarily overlooked finer structure in the coupling behaviour. Earlier work has shown that when responsivity maps are sampled with much finer granularity, additional small-scale features can emerge [56]. In this work, the coarse sampling was used deliberately to emphasise how the parameters interact and to establish clear global trends, rather than to resolve subtle fluctuations.

Another important limitation concerns the beam origin. In all simulations the beam origin was fixed at a set distance of 5mm upstream of the QPD. Yet the coupling coefficients were found to depend strongly on this **origin**, not just on the **waist position**, even when beam waist was fixed at the QPD and both reference and measurement beams were moved in unison so their path lengths remained equal. This indicates that the way IfoCAD propagates beams—including the accumulated phase along the propagation path—directly influences the observed couplings. It is therefore possible that the absolute magnitudes of DWS and LPS coefficients reported here are contingent on this somewhat arbitrary modeling choice. To obtain more robust values, future work should include a systematic study of origin placement and phase evolution.

Another limitation concerns the GBDC results, which consistently showed abnormal DWS behaviour. In both the main and cross-coupled analyses, only one DWS component produced usable data at a time: previously DWS_v converged to zero, whereas in the later analysis DWS_h did. In both cases, this occurred for α (y-axis) rotations, suggesting that the issue may be linked to that specific rotational dimension. During data preparation, most phases could be unwrapped with `SciPy`'s tools, but the complex phase structure of the GBDC beams still contained large abrupt jumps that required manual correction before a smooth phase distribution could be obtained. It is plausible that these strong phase variations affected how IfoCAD simulated the beam–QPD interaction, causing one rotation axis to lose a consistent DWS response.

GBDC was computationally very efficient, but under GSL its fields could not integrated; in practice, GBDC worked only with DCUHRE. Even then, LPS and DWS fits showed large scatter, and applying tip/tilt yielded unstable, even unusable results. So GBDC analyses were effectively limited to tip=tilt=0, reducing realism of the optical setup. By contrast, MEM and NMF ran robustly with both GSL and DCUHRE and reproduced consistent trends for ideal and measured QPD responsivity maps. For practical analysis, MEM/NMF should be preferred; GBDC is useful only for fast, zero–tip/tilt sweeps under DCUHRE, and its poor interaction with GSL remains a methodological limitation of the present framework.

In this work, the tophat was represented as an ideal circular disk with a perfectly flat intensity profile. While this provides a clean baseline for comparing Gaussian and non-Gaussian beams, it does not capture finer structure that would be expected in a real optical system. In reality, the telescope and other upstream optics, although designed to be smooth, may introduce microscopic surface distortions or wavefront errors that the beam carries with it as it propagates to the QPD (Non-geometric TTL). These small-scale features could alter the responsivity-weighted signals and potentially change coupling behaviour. Future research should therefore simulate tophat beams with superimposed surface-error structures, once these errors are better characterized, to provide more realistic predictions of

QPD signals.

Another limitation concerns the QPD coverage; most analysis in this work were carried out on a single QPD, with the exception of the size and gap-size study. This means that all detailed investigations of DWS and LPS coupling—clocking, beam rotation, offsets, and their interactions—were based on the responsivity maps of one device. While those already captures nonuniform responsivity and any internal crosstalk of the diode, it also embeds all the nonuniformities of that specific detector. Without mapping additional QPD with a number of repetitions, it is not possible to establish whether the absolute coupling magnitudes observed here are representative of the broader population or merely characteristic of a single sample. A natural extension would be to repeat the responsivity mapping on several QPDs, which would allow confidence intervals to be assigned to responsivity-related effects and would help disentangle genuine physical trends from device-specific artefacts.

During the mapping of the different QPD sizes and gap sizes, the detector bias voltage was set to $10\ \mathrm{V}$. However, full depletion of the diodes typically requires a bias of about $24\ \mathrm{V}$. As a result, only the DC responsivity is expected to be reasonably reliable at this voltage, whereas accurate RF characterization would require near-complete depletion. For this reason, the present analysis focuses primarily on DC behaviour. Furthermore, the spatial resolution of the responsivity maps was limited. Although the laser spot had a sub-$3\ \mu\mathrm{m}$ diameter, the scan step size was $20\ \mu\mathrm{m}$, which is comparable to the QPD gap width. This means that fine details near the gap edges could not be fully resolved, and the resulting maps represent only the coarse responsivity structure. Future mapping should therefore use smaller step sizes to improve spatial accuracy and better capture the detailed QPD geometry.

One observation from this study is that the QPD gap lines appeared not completely perpendicular in the responsivity maps. This effect was noted repeatedly during the analysis and was also visible in the later size and gap-size study. I hypothesised that a residual tip/tilt of the QPD in the responsivity mapping setup could introduce such artefacts. However, this hypothesis was never tested: the setup was not re-aligned to check whether the non-perpendicular features would disappear or remain. As a result, the true origin of the effect is unresolved. A clear recommendation for future work is to perform systematic experiments in which the QPD's tip/tilt is deliberately varied, and the resulting responsivity maps and signals are compared. Such a study would determine whether the apparent gap-line misalignment is an intrinsic property of the diode or simply a consequence of setup alignment.

5. Conclusion

The goal of this research was to quantify how misalignments—both from QPD tip, tilt, and clocking, and from beam jitter, rotation/offsets, the QPD signals. It was also to characterize how these orientation parameters interact within a large parameter space and shape the combined response. A further aim was to compare laboratory QPD responsivity measurements with idealized simulations to establish the deviations and uncertainties relevant for modeling. In addition, the influence of detector geometry, such as overall size and inter-quadrant gap width, was evaluated with respect to DPS, DWS, LPS and contrast. Lastly, this work investigated how tophat-like beams from clipping or diffraction alter QPD performance when compared with Gaussian-based models.

QPD responsivity was mapped by scanning a focused, modulated 1064 nm beam over the diode and recording DC and RF outputs for each quadrant. The auto-clocking routine fitted straight lines through the horizontal and vertical gaps and found a small mismatch of about $0.293°$. IfoCAD was then used to calculate DPS, DWS, and LPS with both ideal and measured responsivity maps. The QPD was set in the x–y plane at $z = 0$. Beams were propagated along $-z$ with a waist radius of $0.5\,\mathrm{mm}$, an active area of $0.75\,\mathrm{mm}$, and $20\,\mu\mathrm{m}$ quadrant gaps. The pivot-at-QPD convention was chosen to keep angular and lateral effects independent. Beam rotations α, β were scanned within $\pm 100\,\mu\mathrm{rad}$, offsets within $\pm 30\,\mu\mathrm{m}$, and QPD tip/tilt and clocking up to $\pm 4°$ and $\pm 3°$. Quadrant powers and complex amplitudes were calculated by integration routines named GSL and DCUHRE. Flat-top beams were modeled in three ways. GBDC built the beam from a grid of equal-waist beamlets. MEM determined the coefficients of Hermite–Gaussian modes by evaluating overlap integrals with the target field. NMF obtained the coefficients through a discrete least-squares summation. The original IfoCAD code had to be altered to accompany the simulation needs of this study.

The comparison of tophat beam construction for MEM, NMF, and GBDC shows that all three methods can approximate tophat-like beams, but with distinct trade-offs. MEM and NMF reproduce the aperture footprint at low mode counts but leave shoulders, ripples, or basis artifacts, while GBDC produces smoother flat tops with either soft (notrunc) or sharper but ringed (trunc) edges. All methods converge to the target power of 0.5 W, and MSE values drop below the Gaussian baselines (0.0183, 0.0129) once $N_{\text{modes}} \gtrsim 100$. GBDC achieves the lowest MSE overall and is orders of magnitude faster to construct than MEM or NMF, though it lacks a coefficient-based save format. MEM is slower but provides clean interiors at moderate N_{modes}. NMF is orders of magnitude faster than MEM and has the ability to create trunc and notrunc beam models.

Integration method Because GBDC could not be evaluated using the GSL integration method, DCUHRE was included to enable assessment of GBDC beams, but also added to all other simulations for a fair comparison. The outcome was clear: Combining all the data from the Gaussian and tophat results, GSL emerges as the more reliable and reproducible integration method. DCUHRE, in contrast, not only failed to produce stable LPS values but also showed limited consistency across the other coefficient analyses—and, in several cases, could not even compute the expected results for ideally simulated QPDs. Therefor, GSL is suggested as the preferred integration method in IfoCAD for LISA interferometry studies.

Empirical model The following model for DWS was tested throughout this thesis:

$$\mathrm{DWS} = I^{\mathrm{rot}}_{\mathrm{DWS}} \cdot \mathrm{rotation}_{\parallel} + I^{\mathrm{off}}_{\mathrm{DWS}} \cdot \mathrm{offset}_{\parallel} + e^{\mathrm{rot}}_{\mathrm{DWS}} \cdot \mathrm{rotation}_{\perp} + e^{\mathrm{off}}_{\mathrm{DWS}} \cdot \mathrm{offset}_{\perp} \tag{49}$$

where the first two terms correspond to the primary DWS coupling, and the last two terms correspond to the cross-coupled DWS.

It was fitted sequentially, starting with the first term gave an ($R^2 = 0.99998$) and adding the remaining terms in order, which further improved the fit ($\Delta R^2 = 0.83$, 0.35, and 0.50), resulting in an overall $R^2 = 0.999996$.

For GSL, the primary beam rotation-to-DWS coupling coefficient ($I^{\text{rot}}_{\text{DWS}}$) was about $(2308 \pm 1\ \text{rad/rad})$, and slightly higher for RF data; up to $(2346 \pm 2\ \text{rad/rad})$ for 30MHz and $(2340 \pm 2\ \text{rad/rad})$ for 3 MHz The ideal case lies somewhat higher, near $(2382 \pm 1\ \text{rad/rad})$.

The truncated tophat models (NMF and MEM, GSL, real) sit near 1660 rad/rad, which is lower than the Gaussian by 682 rad/rad ($\approx 30\%$ decrease). The non-truncated NMF GSL case is even lower, near 1098 rad/rad, i.e., a drop of 1244 rad/rad ($\approx 52\%$ decrease). Ideal vs real differs by $\approx +40$ for GSL. Thus, the tophat beams show significantly lower primary coupling than the Gaussian, reflecting reduced rotational sensitivity.

The primary beam-offset-to-DWS coupling coefficient for Gaussian beam was fitted to 21.05 rad/m with R^2 9.81. However, it was found that For MEM and NMF, the magnitude of $I^{\text{off}}_{\text{DWS}}$ decreases with N_{modes} and stabilizing within a few rad/m of zero by 50 modes. This indicates that the DWS channel becomes largely insensitive to beam offset, reducing one potential source of alignment error.

The cross coupled DWS can be described more precisely as:

$$\text{DWS}_{\text{cross}} = e^{\text{clock}}_{\text{DWS}} \cdot \theta_{\text{clock}} \cdot \text{rotation}_{\perp} + e^{\text{off}}_{\text{DWS}} \cdot \text{offset}_{\perp} + e^{\text{tip/tilt}}_{\text{DWS}} \cdot \text{tip} \cdot \text{tilt} \cdot \text{rotation}_{\perp}, \tag{50}$$

Using a layered R^2 analysis, the fits showed good agreement: the first term yielded $R^2 = 0.9927$, followed by improvements of $\Delta R^2 = 0.849$ and 0.455, for the successive terms. The final fit reached $R^2 = 0.9994$ for real QPD responsivities and $R^2 = 1$ for the ideal case.

The clocking-induced coupling coefficient, $e^{\text{clock}}_{\text{DWS}}$, shows consistent behaviour across beam types. he ideal couplings exceed the real ones by roughly 40–60 rad^{-1}: for the Gaussian, 2386 vs. 2325 rad^{-1}; for MEM, 2307 vs. 2268 rad^{-1}; and for truncated NMF, 2304 vs. 2263 rad^{-1}. These three sit close together, within about 3% of each other, while the non-truncated NMF stands out at 2826 rad^{-1} (ideal) and 2781 rad^{-1} (real), roughly 20% higher than the Gaussian reference.

GBDC giving much higher but noisier results ($\sim 4.8 \times 10^3 \pm 10^3$ rad^{-1}). Frequency has little influence on the clocking–rotation coupling, with any observed variations well within measurement uncertainty. The number of modes likewise shows no clear effect, except for small decreases under DCUHRE that remain minor compared with the overall spread.

The tip–tilt-induced cross-coupling coefficient was well captured by the bilinear model in Eq. 29. Fitting across all data yielded $e^{\text{tip/tilt}}_{\text{DWS}} \approx (1.592 \pm 0.00014) \times 10^{-3}$ rad^{-2} with $R^2 = 0.9986$. Neither tip nor tilt alone caused additional coupling; only their product will. Hence, realistic detector geometries, such as the 2° tilt planned for LISA, tilt alone does not cause coupling, but any tip error on top of it amplifies the effect on cross-coupled DWS.

So when QPD tilt was set to either 0 or 2 in the simulations, the resulting DWS responses were nearly identical. However, GSL and DCUHRE behave differently: GSL produces tight clusters and consistent slopes, while DCUHRE inflates variability. For DCUHRE, the interpretation of even a perfectly symmetric responsivity map is less deterministic, and caused a lot of spread.

The tested LPS model was:

$$\text{LPS} = k^{\alpha}_{\text{LPS}} \cdot (\alpha \cdot y_{\text{off}}) + k^{\beta}_{\text{LPS}} \cdot (\beta \cdot x_{\text{off}}) + k^{x}_{\text{LPS}} \cdot x^2_{\text{off}} + k^{y}_{\text{LPS}} \cdot y^2_{\text{off}} + e^{\alpha}_{\text{LPS}} \cdot \alpha + e^{\beta}_{\text{LPS}} \cdot \beta \tag{51}$$

Using the same layered R^2 analysis, the fit starting from $R^2 = 0.4997$, each successive term improved the fit by $\Delta R^2 = 0.9994$, 0.1560, 0.0525, 0.4896, and 0.8093, reaching a final $R^2 = 0.999998$ for the real QPD responsivities and $R^2 = 1$ for the ideal case.

The first two bilinear terms were the most dominant contributors to the fit.

Across all simulations, truncated beams yield a lower pathlength coupling of approximately $k_{\text{LPS}} = 0.70\ \text{pm}/(\mu\text{m}\,\mu\text{rad})$, representing a decrease of nearly 20% compared to the Gaussian reference

(≈ 0.83 for GSL, ≈ 0.81 for DCUHRE). Non-truncated beams cluster near 0.87, slightly above the Gaussian reference.

The spread in coupling could not be linked to any specific parameter under GSL, while for DCUHRE it was primarily associated with variations in tilt, although no consistent trend was observed.

The DPS signals follow

$$\mathrm{DPS} = I_{\mathrm{DPS}} \cdot \mathrm{offset}_{\parallel} + e_{\mathrm{DPS}} \cdot \mathrm{offset}_{\perp}. \tag{52}$$

Here, the first term achieved $R^2 = 0.99998$, with the second term contributing only $\Delta R^2 = 0.16$.

Contrast is fitted as

$$\mathrm{c} = 1 - \left(k_c^{\alpha} \cdot \alpha^2 + k_c^{\beta} \cdot \beta^2 + k_c^{x} \cdot x_{\mathrm{off}}^2 + k_c^{y} \cdot y_{\mathrm{off}}^2\right). \tag{53}$$

The layered fit gave partial R^2 values of 0.63, 0.43, 0.63, and 0.992, ending with a total $R^2 = 0.99994$ for both real and ideal responsivities.

In the QPD size and gap-size study (for Gaussian beams), laboratory responsivity maps were taken for detectors with different radii and gap widths. The primary couplings (DWS and DPS) increase with detector radius. Contrast-related couplings decrease with radius, meaning larger detectors experience greater contrast loss. LPS couplings also decrease with radius, so the longitudinal phase readout becomes less sensitive to beam or alignment changes. Cross-coupling terms fluctuate around zero without systematic trends, showing that variations in QPD size or gap width mainly enhance desired the primary couplings without increasing unwanted ones.

Differences Real responsivity and ideal QPD Across the board, real QPDs show slightly lower DWS sensitivity than the ideal maps. primary DWS couplings are down by approximately 4 overall, with concrete examples such as a -2.4% drop in the Gaussian $I_{\mathrm{DWS}}^{\mathrm{rot}}$ slope and ≈ -1.6 to -2.6% drops for the clocking–rotation cross term across Gaussian, MEM, and NMF models. In short, real detectors underperform the ideal by a few percent.

For the main LPS offset–rotation coupling k_{LPS}, the difference depends on the integration method: with GSL, the real coupling is very close to the ideal (about $+0.67\%$ relative to the ideal, i.e. slightly smaller in magnitude), while DCUHRE yields a stronger coupling (about -5.7% vs. ideal, i.e. larger in magnitude). The LPS cross terms are consistently close, only ~ 1.6–1.9% lower in real devices, and are nearly gap-independent.

Differences DC and RF data For the DWS primary coupling, the RF values are higher than the DC values by about $1.4-1.6\%$ ($+1.39\%$ at 3 MHz and $+1.65\%$ at 30 MHz compared to 2308 rad/rad). The DWS cross-coupling does not show any measurable DC–RF difference at the 10^{-4} rad level. The LPS coupling k_{LPS} also remains stable, with no DC–RF difference larger than ± 0.003 on a mean of ~ 0.831, corresponding to a relative change below 0.4%.

Differences RF frequency For DWS primary coupling, the RF–to–RF change between 3 MHz and 30 MHz is $+0.26\%$ (Gaussian, GSL, zero clocking). For tophat beams, the 30 MHz result differs from the average of 3 MHz and 15 MHz by about 0.25% with no systematic direction. DWS residuals do not show any measurable frequency dependence at the 10^{-4} rad level. The LPS coupling k_{LPS} is also independent of frequency within its reported uncertainty. For DWS coupling, there is a small difference between RF frequencies, about $0.25-0.3\%$. For LPS coupling, No measurable difference between RF frequencies was found.

Across all results, DWS is influenced by both beam jitter and detector misalignment, with offset and rotation contributing independently but not multiplicatively. LPS remains mostly insensitive to

misalignment alone but reacts strongly when offset and rotation act together in the same plane. The models and fitted coefficients thus capture the dominant geometric dependencies governing QPD behavior, with realistic responsivity data confirming that these couplings persist under practical, non-ideal conditions.

In conclusion, both MEM and NMF yield consistent and physically meaningful results for tophat-like beams, whereas GBDC produced unstable DWS and unusable LPS responses, and should therefore be avoided for precision simulations. Between the two viable models, MEM is constrained by its hard-edged aperture, which limits how smoothly its profile can be tuned. NMF, by contrast, allows greater control over beam shape, supporting both soft-edged and sharp-edged tophats while maintaining numerical stability and reproducibility. Combined with the GSL integration method, NMF provides the most flexible and robust representation of flat-top beams and is therefore recommended for future IfoCAD simulations and for LISA-level interferometric error budgeting.

References

[1] I. Newton. *Philosophiae naturalis principia mathematica*. J. Societatis Regiae ac Typis J. Streater, 1687.

[2] Albert Einstein. The field equations of gravitation. *Sitzungsber. Preuss. Akad. Wiss. Berlin (Math. Phys.)*, 1915:844–847, 1915.

[3] Karl Schwarzschild. Über das gravitationsfeld eines massenpunktes nach der einsteinschen theorie. *Sitzungsberichte der königlich preussischen Akademie der Wissenschaften*, pages 189–196, 1916.

[4] David Finkelstein. Past-future asymmetry of the gravitational field of a point particle. *Physical Review*, 110(4):965, 1958.

[5] Dennis Walsh, Robert F Carswell, and Ray J Weymann. 0957+ 561 a, b: twin quasistellar objects or gravitational lens? *Nature*, 279(5712):381–384, 1979.

[6] Tilman Sauer. A brief history of gravitational lensing. *Einstein Online*, 04:1005, 2010. Accessed: 2024-11-18.

[7] Albert Einstein. Über gravitationswellen. *Sitzungsber. Preuss. Akad. Wiss. Berlin*, 154:1918, 1918.

[8] Gudrun Wanner. Complex optical systems in space: numerical modelling of the heterodyne interferometry of lisa pathfinder and lisa. *Ph. D. Thesis*, 2010.

[9] American Scientist Editorial. Gravitational waves and the effort to detect them. *American Scientist*, 92(4):350, 2004. Accessed: 2024-11-18.

[10] Russell A Hulse and Joseph H Taylor. Discovery of a pulsar in a binary system. *Astrophysical Journal, vol. 195, Jan. 15, 1975, pt. 2, p. L51-L53.*, 195:L51–L53, 1975.

[11] Peter AR Ade, Randol W Aikin, Denis Barkats, SJ Benton, Colin A Bischoff, JJ Bock, JA Brevik, I Buder, E Bullock, CD Dowell, et al. Detection of b-mode polarization at degree angular scales by bicep2. *Physical Review Letters*, 112(24):241101, 2014.

[12] L Conti, Auriga Collaboration, et al. The auriga second scientific run and the dual detector of gravitational waves. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 518(1-2):236–239, 2004.

[13] Joseph Weber. Detection and generation of gravitational waves. *Physical Review*, 117(1):306, 1960.

[14] GH Janssen, BW Stappers, M Kramer, M Purver, A Jessner, and I Cognard. European pulsar timing array. In *AIP Conference Proceedings*, volume 983, pages 633–635. American Institute of Physics, 2008.

[15] Albert Abraham Michelson and Edward Williams Morley. On the relative motion of the earth and the luminiferous ether. *American journal of science*, 3(203):333–345, 1887.

[16] Benjamin P Abbott, Richard Abbott, TDe Abbott, MR Abernathy, Fausto Acernese, Kendall Ackley, Carl Adams, Thomas Adams, Paolo Addesso, Rana X Adhikari, et al. Observation of gravitational waves from a binary black hole merger. *Physical review letters*, 116(6):061102, 2016.

[17] Benjamin P Abbott, Richard Abbott, TD Abbott, F Acernese, K Ackley, C Adams, T Adams, P Addesso, Rana X Adhikari, Vaishali B Adya, et al. Gw170814: a three-detector observation of gravitational waves from a binary black hole coalescence. *Physical review letters*, 119(14):141101, 2017.

[18] LIGO Scientific Collaboration, Virgo Collaboration, KAGRA Collaboration, R Abbott, H Abe, F Acernese, K Ackley, N Adhikari, RX Adhikari, VK Adkins, et al. First joint observation by the underground gravitational-wave detector kagra with geo 600. *Progress of Theoretical and Experimental Physics*, 2022(6):063F01, 2022.

[19] Pau Amaro-Seoane, Heather Audley, Stanislav Babak, John Baker, Enrico Barausse, Peter Bender, Emanuele Berti, Pierre Binetruy, Michael Born, Daniele Bortoluzzi, et al. Laser interferometer space antenna. *arXiv preprint arXiv:1702.00786*, 2017.

[20] Benjamin Lange. The drag-free satellite. *AIAA journal*, 2(9):1590–1606, 1964.

[21] eLISA Science Team. Drag-free operation. https://web.archive.org/web/20131205072116/https://www.elisascience.org/articles/elisa-mission/drag-free-operation, 2013. Accessed: 2024-11-08.

[22] European Space Agency. Lisa factsheet. https://www.esa.int/Science_Exploration/Space_Science/LISA/LISA_factsheet, 2024. Accessed: 2025-08-28.

[23] Monica Colpi, Karsten Danzmann, Martin Hewitson, Kelly Holley-Bockelmann, Philippe Jetzer, Gijs Nelemans, Antoine Petiteau, David Shoemaker, Carlos Sopuerta, Robin Stebbins, et al. Lisa definition study report. *arXiv preprint arXiv:2402.07571*, 2024.

[24] Nils Frederik Hasselmann, Beata Kunicka, Dirk Papendorf, Harald Kögel, Ulrich Johann, Dennis Weise, and Alexander Sell. Siderostat pointing: validation of an advanced line-of-sight pointing concept beyond lisa. *Classical and Quantum Gravity*, 42(12):125005, jun 2025.

[25] Gerhard Heinzel, Javier Alvarez-Vizoso, Miguel Dovale-Alvarez, and Karsten Wiesner. Frequency planning for lisa. *Physical Review D*, 110(4):042002, 2024.

[26] Chiara Caprini, Anna Heffernan, Richard Brito, Gabriele Franciolini, Germano Nardini, Nicola Tamanini, and Danièle Steer. Science of the lisa mission: A summary for the european strategy for particle physics. *arXiv preprint arXiv:2507.05130*, 2025.

[27] Karsten Danzmann, LISA Study Team, et al. Lisa: laser interferometer space antenna for gravitational wave measurements. *Classical and Quantum Gravity*, 13(11A):A247, 1996.

[28] Samuel Banks, Katharine Lee, Nazanin Azimi, Kendall Scarborough, Nikolai Stefanov, Indra Periwal, Nianyi Chen, Colin DeGraf, and Tiziana Di Matteo. On the detectability of massive black hole merger events by laser interferometry space antenna. *Monthly Notices of the Royal Astronomical Society*, 512(4):6007–6020, 2022.

[29] NASA. LISA Preparatory Science (LPS) Program. https://lisa.nasa.gov/LPSprogram.html, 2025. Accessed: 2025-08-30.

[30] Hajime Kobayashi, Shinji Mukohyama, Johannes Noller, Sergi Sirera, Kazufumi Takahashi, and Vicharit Yingcharoenrat. Inverting no-hair theorems: How requiring general relativity solutions restricts scalar-tensor theories. *Physical Review D*, 111(12):124022, 2025.

[31] Mauro Pieroni and Enrico Barausse. Foreground cleaning and template-free stochastic background extraction for lisa. *Journal of Cosmology and Astroparticle Physics*, 2020(07):021, 2020.

[32] Scott A Hughes. Lisa sources and science. *arXiv preprint arXiv:0711.0188*, 2007.

[33] LISA Consortium. What is time delay interferometry (tdi) and how does it work? https://www.lisamission.org/faq/what-is-time-delay-interferometry-tdi-and-how-does-it-work/, 2025. Accessed: 2025-10-06.

[34] Massimo Tinto and Sanjeev V Dhurandhar. Time-delay interferometry. *Living Reviews in Relativity*, 24(1):1, 2021.

[35] Dam Quang Nam, Joseph Martino, Yves Lemiere, Antoine Petiteau, Jean-Baptiste Bayle, Olaf Hartwig, and Martin Staab. Time-delay interferometry noise transfer functions for lisa. *Physical Review D*, 108(8):082004, 2023.

[36] Simon Barke, Nils Brause, Iouri Bykov, Juan Jose Esteban Delgado, Anders Enggaard, Oliver Gerberding, Gerhard Heinzel, Joachim Kullmann, Søren Møller Pedersen, and Torben Rasmussen. Lisa metrology system-final report. 2014.

[37] Michael Chwalla, Karsten Danzmann, G Fernández Barranco, Ewan Fitzsimons, Oliver Gerberding, Gerhard Heinzel, Christian J Killow, Maike Lieser, Michael Perreur-Lloyd, David I Robertson, et al. Design and construction of an optical test bed for lisa imaging systems and tilt-to-length coupling. *Classical and Quantum Gravity*, 33(24):245015, 2016.

[38] Shubhashish Datta, Abhay Joshi, and Jim Rue. Crosstalk analysis in large-area low-capacitance ingaas quad photodiodes. In *Infrared Technology and Applications XXXIX*, volume 8704, pages 46–56. SPIE, 2013.

[39] Gudrun Wanner, Gerhard Heinzel, Evgenia Kochkina, Christoph Mahrdt, Benjamin S Sheard, Sönke Schuster, and Karsten Danzmann. Methods for simulating the readout of lengths and angles in laser interferometers with gaussian beams. *Optics communications*, 285(24):4831–4839, 2012.

[40] Shu An, HyunJung Park, and Munho Kim. Recent advances in single crystal narrow band-gap semiconductor nanomembranes and their flexible optoelectronic device applications: Ge, gesn, ingaas, and 2d materials. *Journal of Materials Chemistry C*, 11(7):2430–2448, 2023.

[41] Bob Taylor and Dennis Coyne. Material qualification rga test results: High quantum efficiency ingaas photodiode. Technical Report LIGO-E1600086-v1, LIGO Laboratory, March 2016. LIGO Laboratory internal technical report.

[42] Marie-Sophie Hartig. Tilt-to-length coupling in lisa pathfinder: model, data analysis and take-away messages for lisa. 2022.

[43] Xiang Lin, Qi Xia, Peng Qiu, Yurong Liang, and Hao Yan. Advanced pure tilt actuator for testing tilt-to-length coupling in space-based gravitational wave detection. *arXiv preprint arXiv:2408.00509*, 2024.

[44] M Armano, H Audley, J Baird, P Binetruy, M Born, D Bortoluzzi, Eleonora Castelli, A Cavalleri, A Cesarini, AM Cruise, et al. Tilt-to-length coupling in lisa pathfinder: A data analysis. *Physical review D*, 108(10):102003, 2023.

[45] Chantal Pitte. *Fundamental physics with supermassive black holes binaries from the detection of gravitational waves with LISA*. PhD thesis, Université Paris-Saclay, 2024.

[46] Daniel Don George. *Examining the Impact of Tilt-to-Length Noise on the Lisa Mission*. University of Florida, 2024.

[47] Niklas Houba, Simon Delchambre, Tobias Ziegler, Gerald Hechenblaikner, and Walter Fichter. Lisa spacecraft maneuver design to estimate tilt-to-length noise during gravitational wave events. *Physical review D*, 106(2):022004, 2022.

[48] Marie-Sophie Hartig, Sönke Schuster, and Gudrun Wanner. Geometric tilt-to-length coupling in precision interferometry: mechanisms and analytical descriptions. *Journal of Optics*, 24(6):065601, 2022.

[49] Wen Tong Fan, Jie Song, Hong Wen Hai, Si Jun Fang, Kai Zhao, Rui Zhang, Bo Hong Li, Jian Luo, Qi Cheng Sun, Lei Fan, et al. Research on the tilt-to-length coupling noise suppression method inside the gravitational wave detection telescope. *Optics Express*, 32(7):12200–12212, 2024.

[50] Marie-Sophie Hartig, Sönke Schuster, Gerhard Heinzel, and Gudrun Wanner. Non-geometric tilt-to-length coupling in precision interferometry: mechanisms and analytical descriptions. *Journal of Optics*, 25(5):055601, 2023.

[51] Ya Zhao, Jia Shen, Chao Fang, Heshan Liu, Zhi Wang, and Ziren Luo. Tilt-to-length noise coupled by wavefront errors in the interfering beams for the space measurement of gravitational waves. *Optics Express*, 28(17):25545–25561, 2020.

[52] Fei Xie, Xiaodong Peng, Wenlin Tang, Mengyuan Zhao, and Xiaoshan Ma. Multiparameter sensitivity analysis of tilt-to-length coupling noise in taiji test mass interferometer. *Journal of Astronomical Telescopes, Instruments, and Systems*, 11(1):018004–018004, 2025.

[53] M-S Hartig, Sarah Paczkowski, Martin Hewitson, Gerhard Heinzel, and Gudrun Wanner. Postprocessing subtraction of tilt-to-length noise in lisa in the presence of gravitational wave signals. *Physical Review D*, 111(4):043048, 2025.

[54] M Chwalla, K Danzmann, M Dovale Álvarez, JJ Esteban Delgado, G Fernández Barranco, E Fitzsimons, O Gerberding, G Heinzel, CJ Killow, M Lieser, et al. Optical suppression of tilt-to-length coupling in the lisa long-arm interferometer. *Physical Review Applied*, 14(1):014030, 2020.

[55] Sönke Schuster. Tilt-to-length coupling and diffraction aspects in satellite interferometry. 2017.

[56] Ruurd Kuipers. Simulating qpd performance for lisa. Design project, Rijksuniversiteit Groningen, 2024. Supervisors: Dr. Niels van Bakel, Dr. ir. Gert de Lange, Timesh Mistry.

[57] Germán Fernández Barranco, Michael Tröbs, Vitali Müller, Oliver Gerberding, Frank Seifert, and Gerhard Heinzel. Spatially resolved photodiode response for simulating precise interferometers. *Applied Optics*, 55(24):6688–6693, 2016.

[58] Gerhard Heinzel, Tim Kaune, and Gudrun Wanner. Ifocad: Interferometer construction and analysis design library. https://gitlab.aei.uni-hannover.de/ifocad/ifocad, 2025. Version 0.2 (June 2025). Licensed under GNU LGPL v3.0. Accessed: 2025-10-06.

[59] Max Planck Institute for Gravitational Physics (AEI). Ifocad: A software library for simulating precision laser interferometry. https://www.aei.mpg.de/ifocad, 2024. C++ library for design and optimization of laser interferometers, available under LGPLv3.

[60] Mengyuan Zhao, Yazheng Tao, Kevin Weber, Tim Kaune, Sönke Schuster, Zhenxiang Hao, and Gudrun Wanner. Method comparison for simulating non-gaussian beams and diffraction for precision interferometry. *Sensors*, 23(22):9024, 2023.

[61] Paul Edwards, Megha Dave, Alexander Joseph Weaver, Mengyuan Zhao, Paul Fulda, Guido Mueller, and Gudrun Wanner. Impact of phase signal formulations on tilt-to-length coupling noise in the lisa test mass interferometer. *Classical and Quantum Gravity*, 2025.

[62] Ya Zhao, Zhi Wang, Yupeng Li, Chao Fang, Heshan Liu, and Huilong Gao. Method to remove tilt-to-length coupling caused by interference of flat-top beam and gaussian beam. *Applied Sciences*, 9(19):4112, 2019.

[63] Xing-Guang Qian, Zhao Cui, Hao-Qi Shi, Xue Wang, Wei-Lai Yao, Rui-Hong Gao, and Yi-Kun Wang. Analysis of beam walk in inter-satellite laser link: Implications for differential wavefront sensing in gravitational wave detection. *Applied Sciences*, 14(13):5526, 2024.

[64] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, and F. Rossi. *GNU Scientific Library Reference Manual*. GNU Project, 3rd edition, 2021. GNU Scientific Library, Free Software Foundation.

[65] Jarle Berntsen, Terje O Espelid, and Alan Genz. An adaptive algorithm for the approximate calculation of multiple integrals. *ACM Transactions on Mathematical Software (TOMS)*, 17(4):437–451, 1991.

[66] Alan W Greynolds. Propagation of generally astigmatic gaussian beams along skew ray paths. In *Diffraction Phenomena in Optical Engineering Applications*, volume 560, pages 33–51. SPIE, 1986.

[67] Nicolay M Tanushev, Björn Engquist, and Richard Tsai. Gaussian beam decomposition of high frequency wave fields. *Journal of Computational Physics*, 228(23):8856–8871, 2009.

[68] Erdem Şahin and Levent Onural. Scalar diffraction field calculation from curved surfaces via gaussian beam decomposition. *Journal of the Optical Society of America A*, 29(7):1459–1469, 2012.

[69] Che Liu and Yanling Guo. Flat-top line-shaped beam shaping and system design. *Sensors*, 22(11):4199, 2022.

[70] Riccardo Borghi. Elegant laguerre–gauss beams as a new tool for describing axisymmetric flattened gaussian beams. *Journal of the Optical Society of America A*, 18(7):1627–1633, 2001.

[71] Neda Meshksar, Moritz Mehmet, Katharina-Sophie Isleif, and Gerhard Heinzel. Applying differential wave-front sensing and differential power sensing for simultaneous precise and wide-range test-mass rotation measurements. *Sensors*, 21(1):164, 2020.

Appendix

A. Tip & Tilt induced clocking angle difference

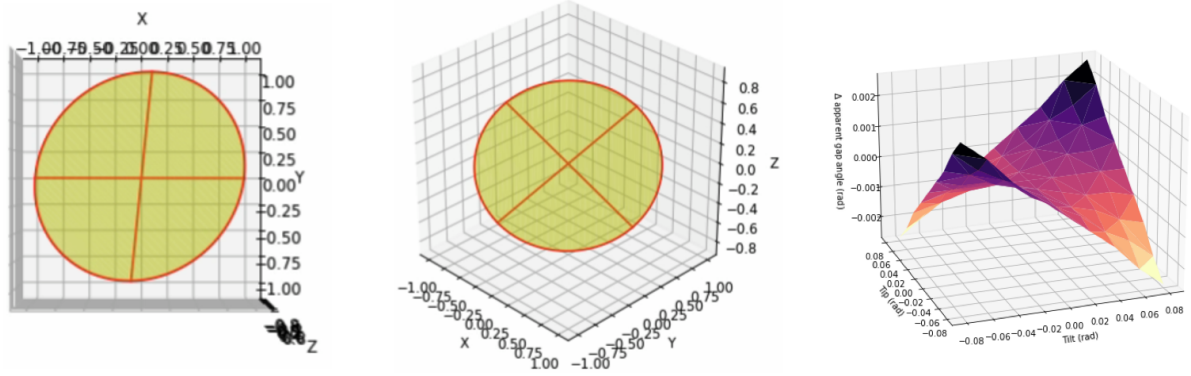


Figure 33: Effect of QPD tip/tilt on gap orientation. Left: ideal projection with orthogonal gaps. Middle: skewed projection after tip/tilt. Right: simulated gap angle difference (DWS_v–DWS_h) as a function of tip and tilt.

When a QPD is ideally aligned, its horizontal and vertical gap lines are orthogonal when viewed along the z-axis. However, introducing tip or tilt — i.e. rotations about the x- or y-axis — distorts this projection. In practice, the detector plane is no longer perpendicular to the line of sight, which skews the apparent orientation of the gap lines. As a result, the horizontal and vertical axes no longer appear exactly orthogonal, and an effective difference in "gap angles" emerges when viewed from the beam direction.

The first two images of Figure 33 illustrates this geometric effect: once the QPD is rotated in tip/tilt, the projection of its quadrants along the z-axis shows a misalignment between the horizontal and vertical divisions. The most right figure quantifies this effect by plotting the induced angular difference between the apparent gap orientations (DWS_v vs. DWS_h) as a function of tip and tilt. Even modest misalignments on the order of a few hundredths of a radian can lead to differences of several milliradians between the two gap orientations, which may contribute to the asymmetry observed in the auto-clocking analysis.

B. Parameter space Roadmap

To further dissect the multidimensional parameter space, two "roadmap" plots were constructed. These plots illustrate how successive filters or constraints can be applied to the data, step by step, to isolate the influence of specific parameters. By narrowing the dataset in stages, the role of each factor becomes visible in a more intuitive manner.

Roadmap 1

Left column (progressive filtering from top to bottom) Top plot: Starts with the same dataset as in Figure 21 (a), showing DWS_h as a function of beam rotation β. Colors encode clocking, with horizontal x-shift indicating tip/tilt variations. Second plot: A filter is applied to select only the case $\beta = -100$ μrad. The wide distribution in the first plot collapses to a narrower band at fixed β, making clocking and tip/tilt differences more prominent. Third plot: From this subset, only the points with QPD clocking $= 3°$ are retained. This removes the color variation, leaving tip/tilt as the main parameter of interest. Bottom plot: Finally, the filter tilt $= -4°$ is applied. The remaining points now represent a single configuration defined by all constraints (fixed β, fixed clocking, fixed tilt), leaving only residual spread due to QPD tip and RF frequency.

Right column (progressive filtering from top to bottom) Top plot: Data subset restricted to QPD clocking $= 0°$, while all other parameters vary. This removes clocking dependence, making beam rotation, beam offset, and QPD tilt more prominent. Second plot: From this, only cases with horizontal beam offset $= 0$ μm are retained. This eliminates offset as a contributing factor. Third plot: Next, the selection fixes beam rotation $\beta = +100$ μrad and channel $= 1$. The spread collapses to a narrow band defined by tip and tilt. Bottom plot: Finally, tip and tilt are both fixed at $4°$. This leaves a single well-defined configuration, where only the variation of RF frequency remains visible.

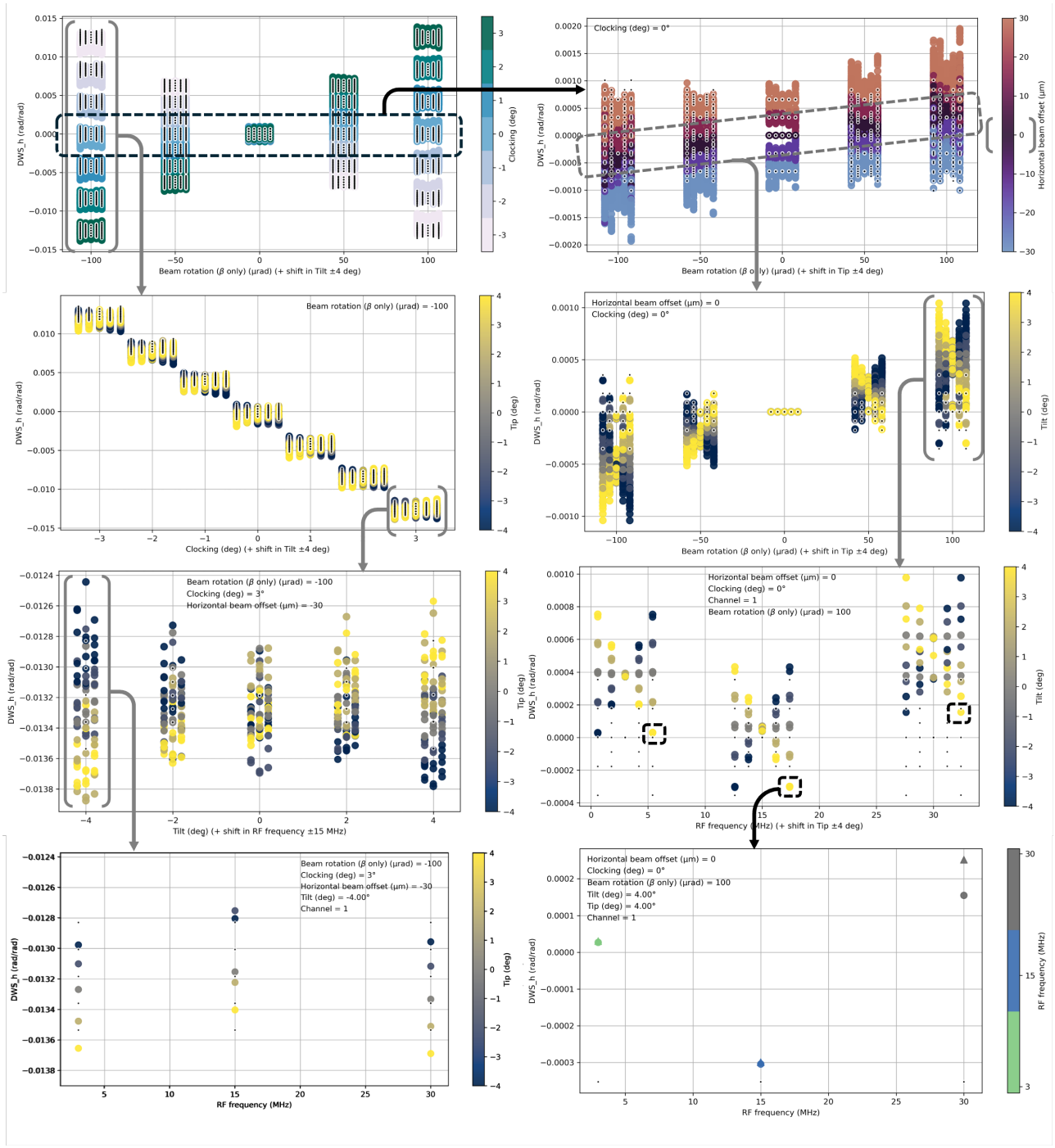


Figure 34: Roadmap 1: stepwise filtering of DWS_h.

Roadmap 2

Left column (progressive filtering from top to bottom) Top plot: Full dataset of Figure 21 (c), showing the longitudinal pathlength signal (LPS) as a function of RF frequency. Color encodes tilt, while horizontal shifts represent beam offset. Second plot: The subset with tilt $= -4°$ and beam

offset $= +30$ μm is selected, corresponding to the extreme bottom-left cluster. Third plot: A further filter fixes beam rotation $\alpha = +100$ μrad. This narrows the spread and highlights residual dependence on tip and RF frequency. Bottom plot: Finally, tip is fixed at $2°$. The resulting configuration produces very large (negative) LPS values, yet still exhibits small systematic variations with RF frequency and differences across measurement repetitions (channels).

Right column (progressive filtering from top to bottom) Top plot: Selection of data with tilt $= 0°$, removing tilt dependence and showing primarily offset and frequency effects. Second plot: A further constraint sets beam offset $= 0$ μm, eliminating offset influence. Third plot: From here, the dataset is reduced to only channel $= 3$, which isolates a single repetition. Bottom plot: Adding tip $= 0°$ fixes the last free degree of freedom. What remains shows residual variations with RF frequency, but also reveals modulation effects from beam rotation and QPD clocking that shape the frequency dependence.

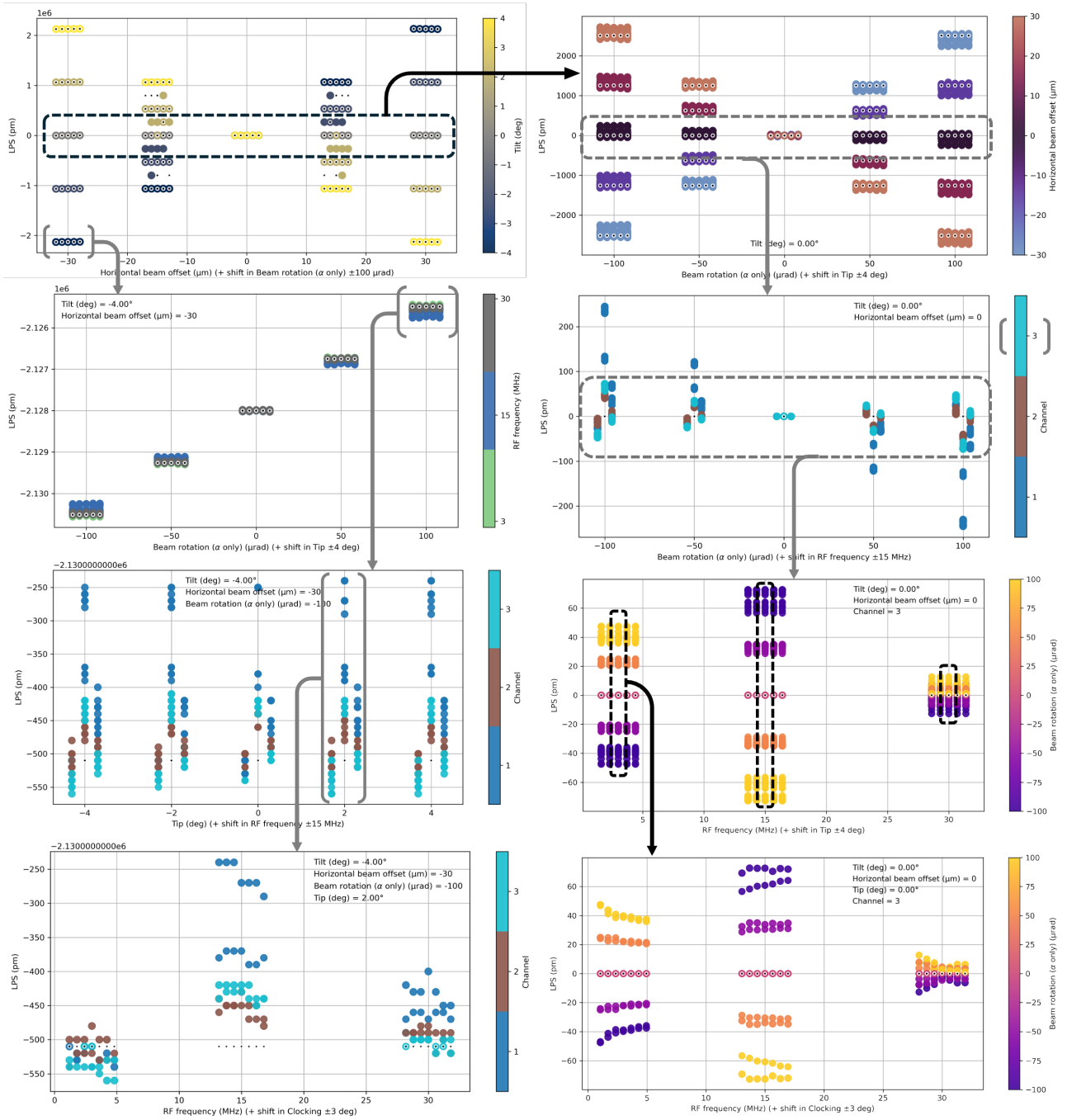


Figure 35: Roadmap 2: stepwise filtering of LPS.

C. Correction of Phase-Wrapped Longitudinal Phase Signal (LPS)

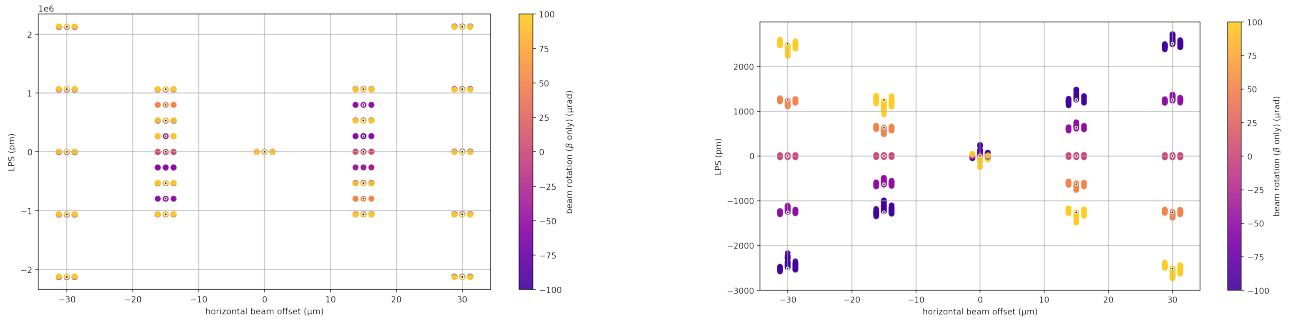
In the raw QPD simulations the longitudinal phase signal (LPS) is expressed in picometres. Since the signal is derived from a phase measurement, it inherits the periodicity of the optical wavelength. For a 1064 nm laser this corresponds to about 1.064×10^6 pm, so the computed LPS values naturally "wrap" whenever the underlying phase exceeds a multiple of 2π. Inspection of the data shows not only plateaus near 0 and $\pm 1.064 \times 10^6$ pm, but also at $\pm 0.27 \times 10^6$ pm, $\pm 0.53 \times 10^6$ pm and $\pm 0.80 \times 10^6$ pm, i.e. at quarter-, half-, and three-quarter-wavelength offsets. These arise from how the interferometric demodulation maps phase to displacement. To obtain a continuous and interpretable LPS, each cluster was shifted by subtracting the nearest multiple of $\lambda/4$, effectively folding all wrapped levels back to a band centred around zero. This removes artificial jumps without altering the physical trends in the signal.

The figures show the longitudinal phase signal (LPS) as a function of horizontal beam offset, with the marker colour indicating the applied beam rotation β (rotation about the y-axis only).

Figure 36 (a)(raw LPS): The data exhibit large discontinuities, with values clustered not only around 0 but also near $\pm 1.0 \times 10^6$ pm and beyond. These jumps are artefacts of the interferometric phase wrapping, where the displacement estimate resets whenever the optical phase passes a multiple of 2π.

Figure 36 (b) (folded LPS): After correcting by subtracting the nearest multiple of a quarter wavelength ($\approx 0.27 \times 10^6$ pm), all plateaus collapse into a continuous band centred around zero. The LPS values now vary smoothly with horizontal offset and beam rotation, without artificial wrapping jumps.

This comparison highlights how the folding procedure transforms the wrapped interferometric output into a physically interpretable displacement signal.



(a) (b)

Figure 36: LPS as a function of horizontal beam offset. Marker colour indicates beam rotation β, x-shift is RF signals for 3 15 and 30 MHz. Values are shown as obtained directly from the simulation output. (a) before phase wrapping, (b) after phase wrapping

D. Offset-to-DWS coupling for ideal QPD

Figure 37 shows the offset–to–DWS coupling computed for ideal QPDs, i.e. without measured responsivity maps. The parameter space is the same as in the main analysis: tip and tilt were varied independently in the range $[-4°, 4°]$, clocking in the range $[-3°, 3°]$, beam rotation angles α and β in $[-100, 100]$ μrad, and horizontal beam offset in $[-30, 30]$ μrad. For each parameter combination, the horizontal DWS signal DWS_h was calculated as a function of beam offset, and a linear fit was applied. The resulting slope represents the coupling strength of horizontal offset into DWS_h, expressed in $[\mathrm{rad/m}]$. Collecting these slopes across the full parameter space produces the coupling map shown here.

On the x-axis the QPD clocking angle is shown, the y-axis gives the offset–to–DWS coupling in [rad/m], tip values are encoded in color, and tilt values appear as small x-shifts of the points. The results show that whenever clocking or a combination of tip and tilt deviate from the perfectly aligned case, the coupling strength decreases slightly. This reduction is expected, since any such misalignment directs part of the beam motion into the wrong quadrants of the detector. However, even at the maximum tested values (clocking $\pm 3°$, tip/tilt $\pm 4°$), the coupling decreases only from about 21.91 to 21.81 rad/m, i.e. less than 0.5%. This small variation explains why in Figure 23 (a) of Section 3.1.6, where the measured QPD responsivity dominates the spread, the systematic dependence on clocking, tip, or tilt is not visible.

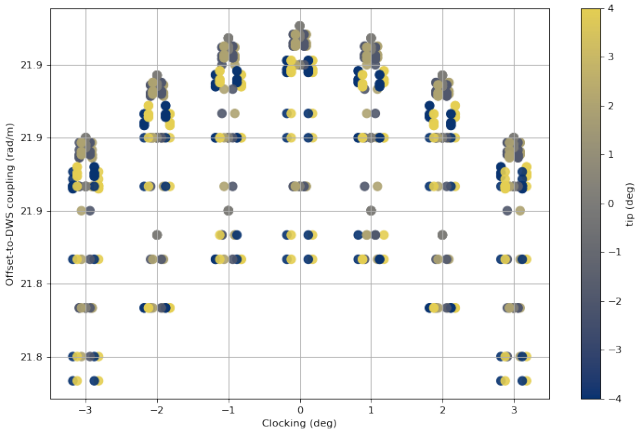


Figure 37: Offset–to–DWS coupling for ideal QPDs. x-axis: QPD clocking angle [deg]. y-axis: coupling slope [rad/m]. Color: tip angle [deg]. Small x-shifts: tilt values. Slopes obtained from linear fits of DWS_h versus horizontal offset across the parameter space.

E. Tip–tilt–induced beam–rotation–to–DWS coupling with raw data and bilinear fit

Figure 38 shows the tip–tilt–induced beam–rotation–to–DWS coupling $C_{(\mathrm{tip,tilt})}$ together with the raw simulation data. The analysis follows directly from the differential signals defined earlier, where for each configuration the perfectly aligned baseline ($x_{\mathrm{off}} = 0,\ \mathrm{tip} = 0,\ \mathrm{tilt} = 0$) was subtracted to isolate the effect of QPD misalignments. To quantify how much beam rotation maps into the DWS channel, the differential vertical signal $\Delta\mathrm{DWS}_v$ was normalized by the applied rotation α (rotation around the x-axis), and equivalently $\Delta\mathrm{DWS}_h$ by β, resulting in the coupling coefficient

Here $C_{(\mathrm{tip,tilt})}$ is expressed in [rad/rad], and the analysis was restricted to $x_{\mathrm{off}} = 0$ to avoid lateral coupling. The resulting coefficient was computed for all $(\mathrm{tip}, \mathrm{tilt})$ combinations in the parameter space. The blue points in the figure represent these computed values, while the colored surface shows the bilinear fit

$$C_{(\mathrm{tip,tilt})} = a\,\mathrm{tip}\cdot\mathrm{tilt}, \tag{54}$$

with fitted coefficient $a \approx 1.592 \times 10^{-3} \pm 1.4 \times 10^{-7}$ [$1/\mathrm{rad}^2$]. The close agreement between the discrete data points and the fitted surface demonstrates that the bilinear form accurately captures the dependence of beam–rotation coupling on tip and tilt misalignments. The fitted surface goodness is represented by R^2=0.9986

$$C_{(\mathrm{tip,tilt})} = \frac{\Delta\mathrm{DWS}_v}{\alpha}, \quad C_{(\mathrm{tip,tilt})} = \frac{\Delta\mathrm{DWS}_h}{\beta}. \tag{55}$$

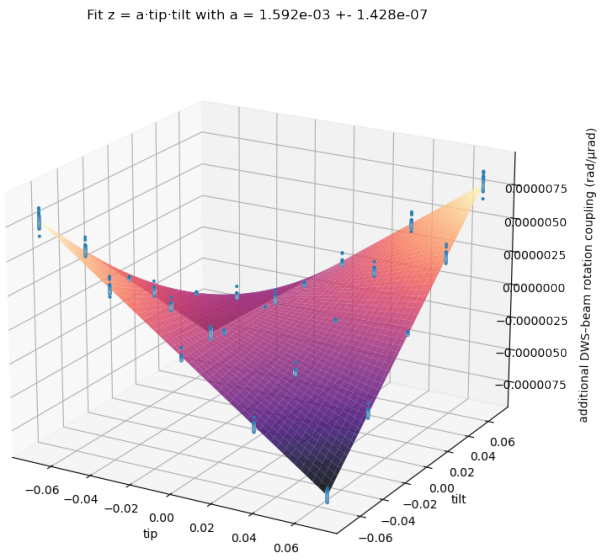


Figure 38: Tip–tilt–induced beam–rotation–to–DWS coupling $C_{(\text{tip,tilt})}$ with raw data (blue points) and bilinear fit surface $C_{(\text{tip,tilt})} = a\,\text{tip}\cdot\text{tilt}$. Analysis performed at $x_{\text{off}} = 0$ using $\Delta\text{DWS}/\alpha$ and $\Delta\text{DWS}/\beta$.

F. QPD response at 0° and 2° Tilt, GSL and DCUHRE

Looking at Figure 39, the GSL panels indicate that setting the QPD at 2° can increase the apparent coupling of DWS to beam offset and to beam rotation, though it is not as systematic as expected. For larger values of beam offset, beam rotation, or both, the error difference increases. However, within our tested ranges, this difference remains well below 0.001 rad for the DWS. In addition, QPD misalignment (clocking, tip, or tilt) does not appear to couple systematically to the difference between the 0° and 2° cases. The ideal QPD follows the same directional tendencies but with tighter clustering; the measured responsivity cases show a broader spread around that trend.

For LPS, the 2° QPD-tilt case shows a stronger response driven mainly by beam rotation; other QPD misalignments remain uncorrelated in this space. The added LPS error is typically 20 pm, with occasional outliers near 40 pm. These values are small in the context of this study. Overall, the 2° tilt does not introduce a new deterministic mapping; rather, it perturbs existing sensitivities that depends on the combination of offset and rotation.

Looking at Figure 40, DCUHRE produces a noticeably broader spread in DWS for both tilt settings. Taking the 2°–0° difference yields more scatter rather than a clear shift in central tendency, making the 2° case harder to interpret. LPS shows no robust, monotonic change attributable solely to the 0°→2° tilt step. Since DCUHRE widens the dispersion, a conservative worst-case reading of the DWS difference would be broader under DCUHRE than under GSL.

Implications First, moving the detector from 0° to 2° can make DWS look more sensitive to offset and rotation in some parts of the parameter space. This is in accordance with the with the what is found in previous sections (Figures 23 a-b).

Second, the integrator choice shapes how clearly weak differences are seen: GSL preserves tighter clusters and makes small changes easier to spot; DCUHRE increases dispersion and thus inflates a worst-case interpretation without revealing new trends.

The outcome here is therefore a robustness check: within the tested ranges, the main message of earlier chapters stands.

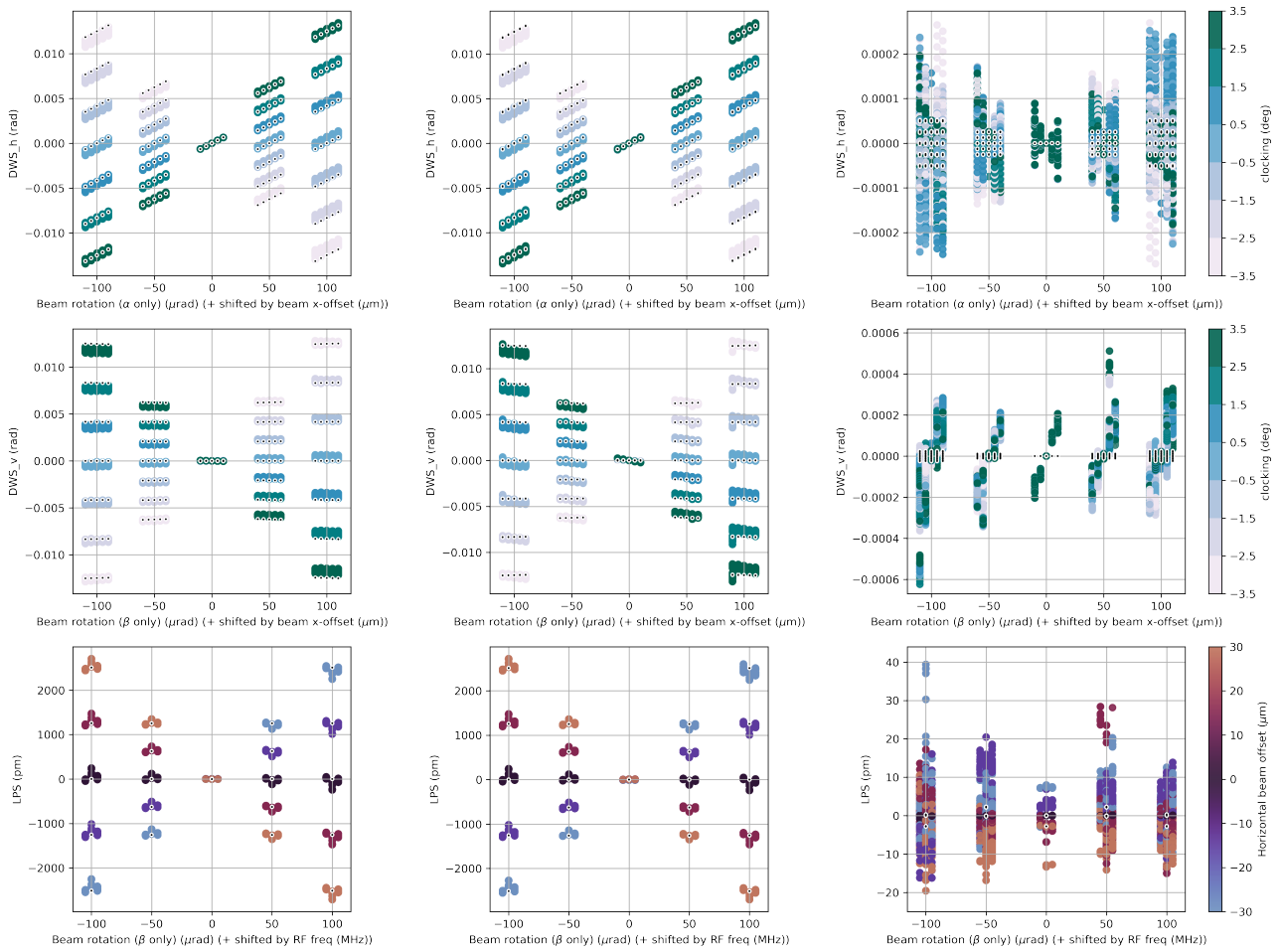


Figure 39: QPD response at tilt $= 0°$ vs. tilt $= 2°$ tilt, GSL integration method. The three columns correspond to the two tilt settings and their difference: the first column shows the $0°$ case, the second column the $2°$ case, and the third column the difference between them. The rows correspond to different readouts. In the first row, the horizontal axis is the beam rotation about α (μrad), shifted horizontally to separate different horizontal beam offsets. The vertical axis is the horizontal DWS signal DWS_h (rad). These panels show how DWS_h depends on α for various beam offsets, and the third panel highlights how this dependence changes when the QPD is tilted by $2°$. In the second row, the horizontal axis is the beam rotation about β (μrad), again shifted horizontally to distinguish different horizontal offsets. The vertical axis is the vertical DWS signal DWS_v (rad). These panels show the response of DWS_v to β and the change induced by moving from $0°$ to $2°$ tilt. In the third row, the horizontal axis is the beam rotation about β (μrad), shifted this time by RF modulation frequency (DC, 3, 15, and 30 MHz). The vertical axis is the longitudinal phase signal (LPS, pm). These panels show how LPS varies with β for the different frequencies and how the $2°$ tilt modifies that relation.

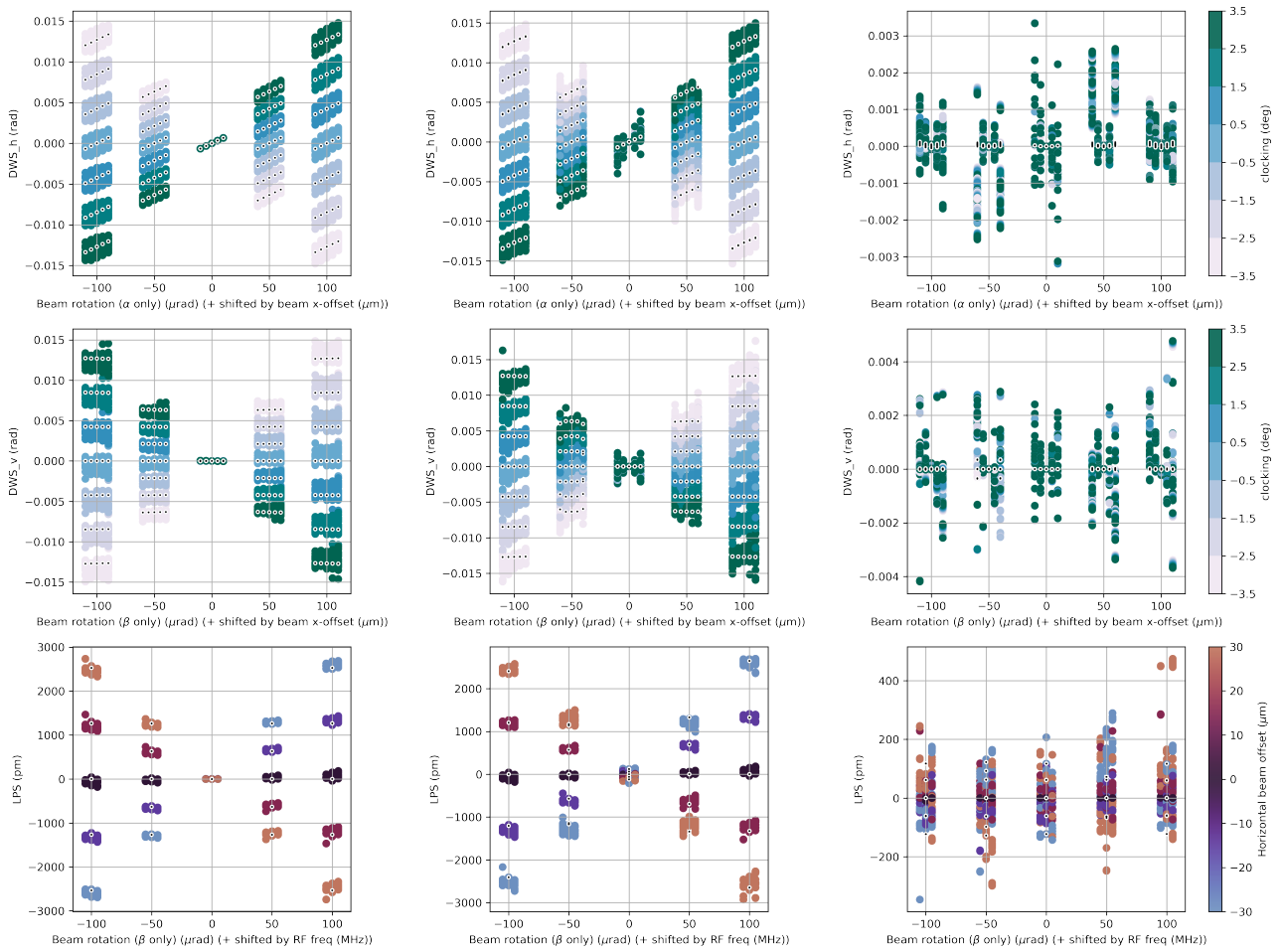


Figure 40: QPD response at tilt $= 0°$ vs. tilt $= 2°$ tilt, DCUHRE integration method. The three columns correspond to the two tilt settings and their difference: the first column shows the $0°$ case, the second column the $2°$ case, and the third column the difference between them. The rows correspond to different readouts. In the first row, the horizontal axis is the beam rotation about α (μrad), shifted horizontally to separate different horizontal beam offsets. The vertical axis is the horizontal DWS signal DWS_h (rad). These panels show how DWS_h depends on α for various beam offsets, and the third panel highlights how this dependence changes when the QPD is tilted by $2°$. In the second row, the horizontal axis is the beam rotation about β (μrad), again shifted horizontally to distinguish different horizontal offsets. The vertical axis is the vertical DWS signal DWS_v (rad). These panels show the response of DWS_v to β and the change induced by moving from $0°$ to $2°$ tilt. In the third row, the horizontal axis is the beam rotation about β (μrad), shifted this time by RF modulation frequency (DC, 3, 15, and 30 MHz). The vertical axis is the longitudinal phase signal (LPS, pm). These panels show how LPS varies with β for the different frequencies and how the $2°$ tilt modifies that relation.

G. DWS Residuals from Parametric Model across RF Frequency and Channels

Residual DWS_h distributions]Residuals of the horizontal differential wavefront sensing signal (DWS_h) after removing the main physics-based contributions. A linear regression model of the form

$$\mathrm{DWS}_h = a \cdot \theta_c \alpha + b \cdot (\mathrm{tip} \cdot \mathrm{tilt} \cdot \alpha) + c \cdot x_{\mathrm{off}}$$

was fitted, where θ_c is the clocking angle, α the beam rotation, and x_{off} the horizontal offset. The plotted residuals correspond to the measured DWS_h minus the model prediction.

On the **x-axis**, the RF demodulation frequency (3, 15, or 30 MHz) is shown. The **y-axis** displays the residual DWS_h in radians. **Color** encodes the numerical integration scheme (green: GSL, orange: DCUHRE). **Columns** separate DC and RF detection modes, while **rows** correspond to measurement channels 1–3.

Statistical analysis of the residuals shows: (i) GSL residuals are centered very close to zero (mean $\sim -8 \times 10^{-5}$ rad) with extremely small scatter ($< 5 \times 10^{-4}$ rad), consistent across RF frequencies, channels, and detection modes; (ii) DCUHRE residuals are also centered near zero but exhibit much larger spread (0.002–0.005 rad), with modest channel-dependent differences. ANOVA confirms that the integration method dominates variance in residuals ($p \ll 10^{-90}$), while effects of RF frequency are formally significant ($p \sim 10^{-3}$) but negligible in magnitude ($\leq 10^{-4}$ rad).

Together, these results show that the physical model captures nearly all systematic structure in the data, with residuals determined primarily by the choice of numerical integration scheme rather than RF frequency, channel, or detection mode.

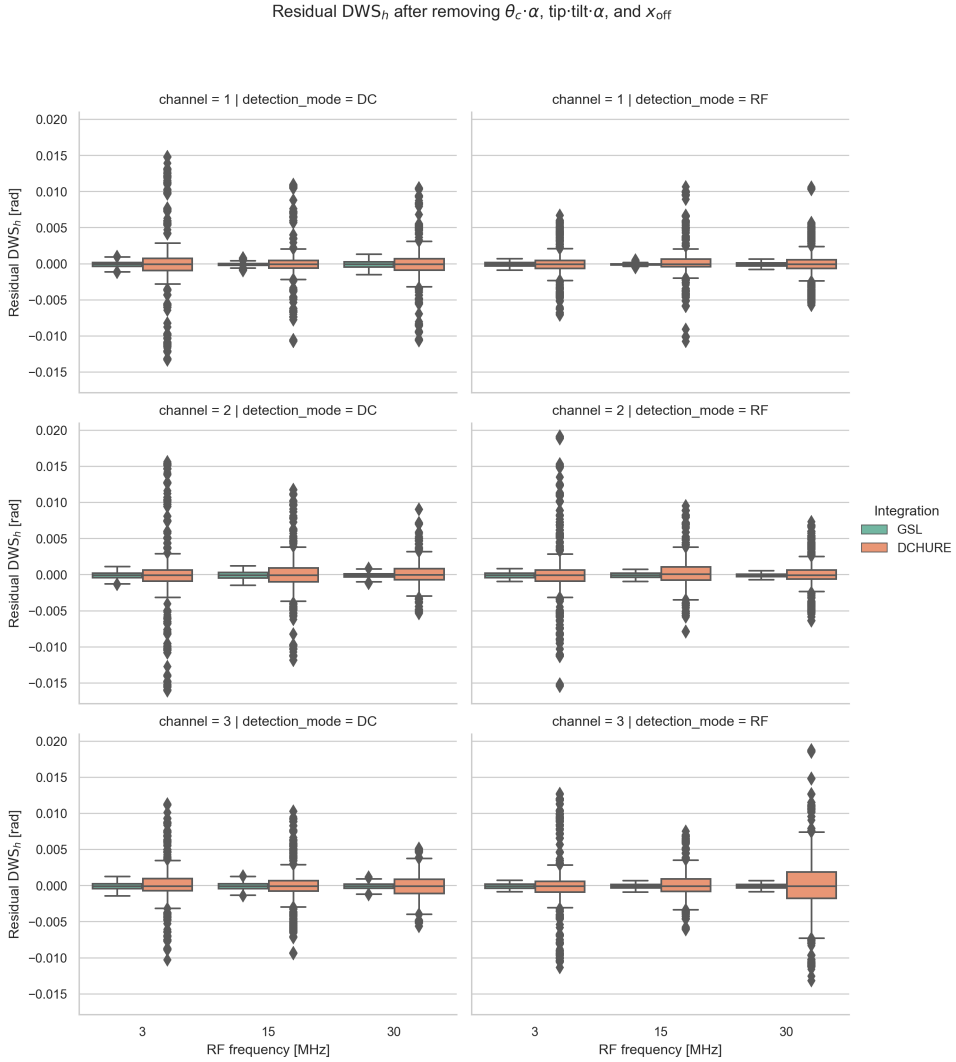


Figure 41: Residuals of the horizontal differential wavefront sensing signal (DWS_h) after subtracting the fitted model $\mathrm{DWS}_h = a\,(\theta_c \alpha) + b\,(\mathrm{tip} \cdot \mathrm{tilt} \cdot \alpha) + c\,x_{\mathrm{off}}$. The x-axis shows RF frequency (3, 15, 30 MHz), the y-axis residual DWS_h [rad]. Colors indicate the integration scheme (GSL, DCUHRE). Columns separate DC/RF detection modes, rows correspond to channels 1–3. Residuals are near zero for both methods, but scatter is much smaller for GSL ($\sim 5 \times 10^{-4}$ rad) than DCUHRE ($\sim$2–5 $\times\, 10^{-3}$ rad).

H. Fitting and Variability of the LPS Coupling Coefficient

To quantify the coupling between longitudinal phase shift (LPS), horizontal beam offset, and horizontal beam rotation (β), a parametric model of the form

$$\mathrm{LPS}\,[\mathrm{pm}] \;=\; k_{\mathrm{LPS}} \cdot x_{\mathrm{off}}\,[\mu\mathrm{m}] \cdot \beta\,[\mu\mathrm{rad}] \tag{56}$$

was fitted to the simulation data. The coefficient k_{LPS} is reported in units of pm/(µm·µrad). For each dataset (integration method, tilt label, ideal vs. non-ideal), the data were grouped by RF frequency (3, 15, 30 MHz), detection mode (DC/RF), channel (experimental repetition), clocking, tip, tilt, and rotation axis. Linear fits within these groups yield k_{LPS} and corresponding R^2 values.

The resulting distributions are summarized in Fig. 42. Mean values show that $k_{\mathrm{LPS}} \approx -0.83 \pm 0.003$ for GSL and $k_{\mathrm{LPS}} \approx -0.88 \pm 0.05$ for DCUHRE. R^2 values were consistently above 0.92 when offset and rotation were taken in the same axis plane.

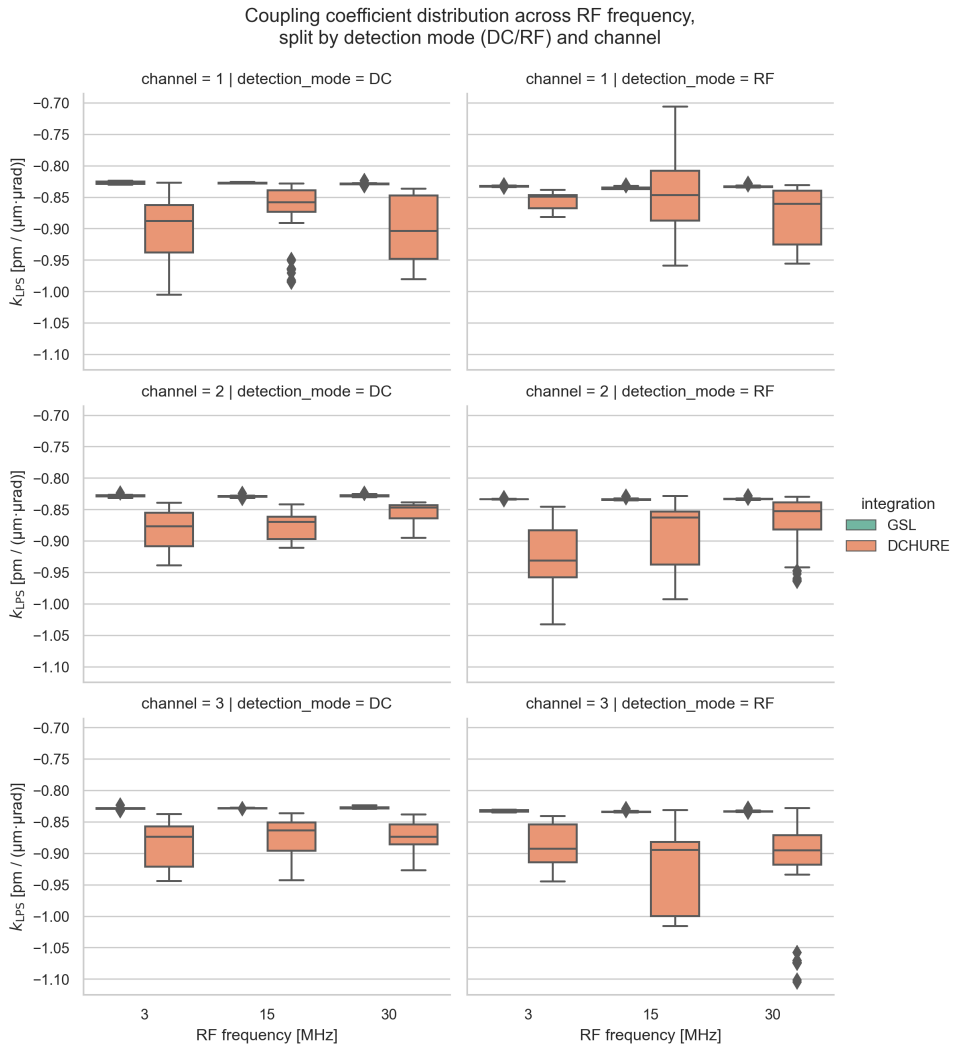


Figure 42: Distribution of the fitted coupling coefficient k_{LPS} across RF frequency (x-axis), detection mode (DC vs. RF; columns), and measurement channel (rows). Colors distinguish numerical integration methods (GSL vs. DCUHRE). Each boxplot represents the distribution of k_{LPS} values from group-wise linear fits of $\mathrm{LPS} = k_{\mathrm{LPS}} \cdot x_{\mathrm{off}} \cdot \beta$. The y-axis shows k_{LPS} in units of pm/(µm·µrad).

I. QPD size and gap data

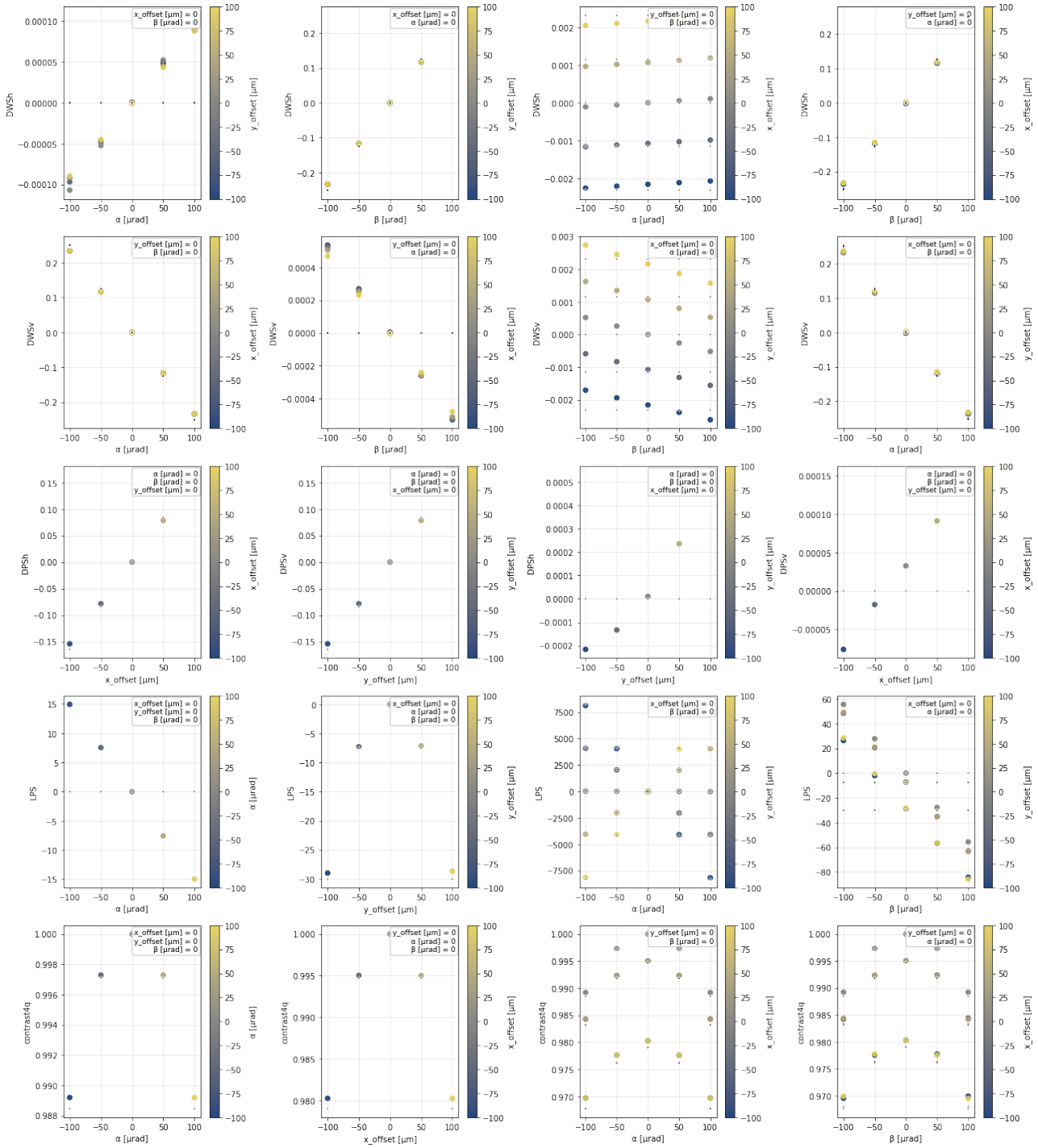


Figure 43: QPD radius = 1000 µm, gap = 40 µm

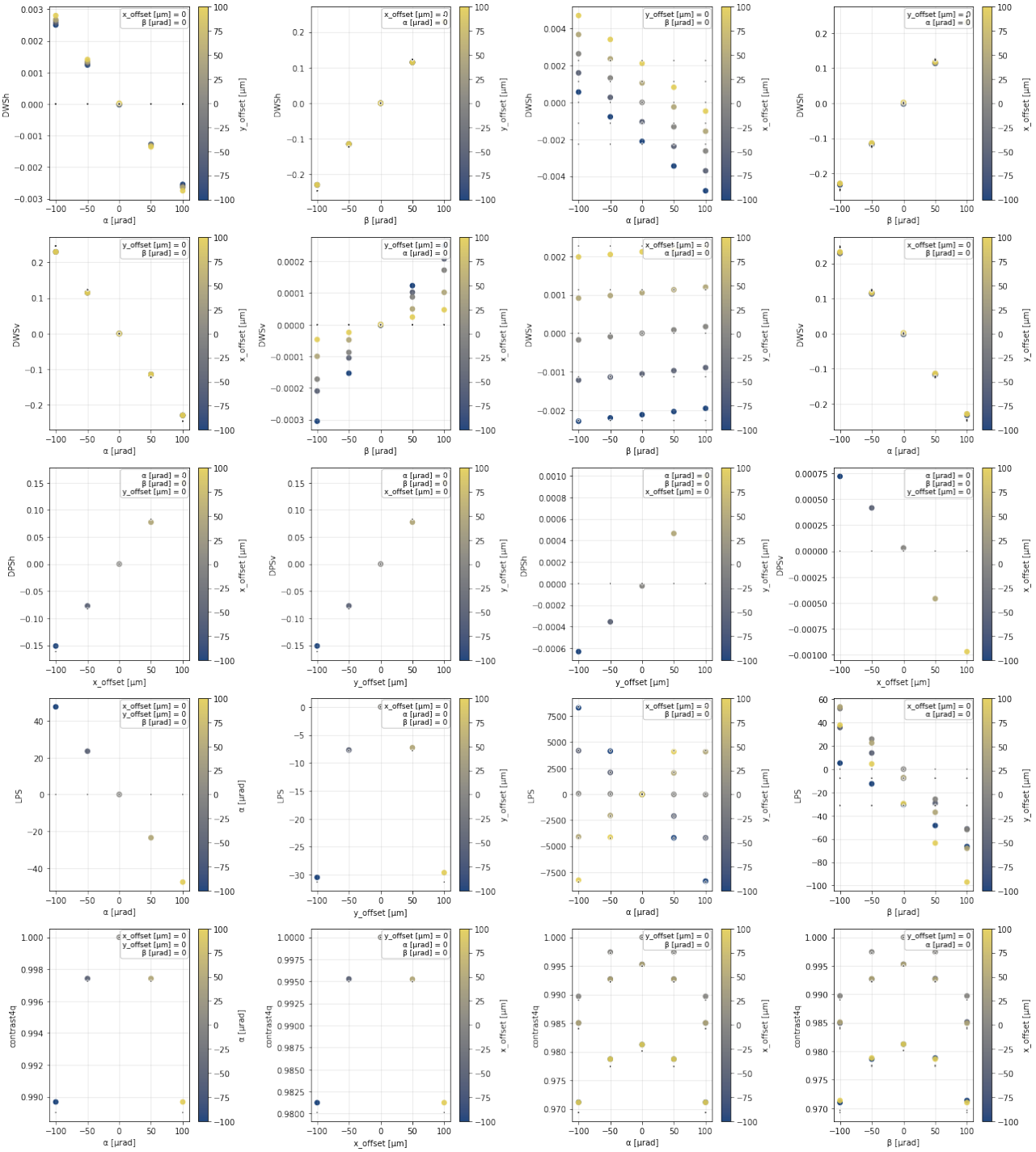


Figure 44: QPD radius = 750 µm, gap = 40 µm

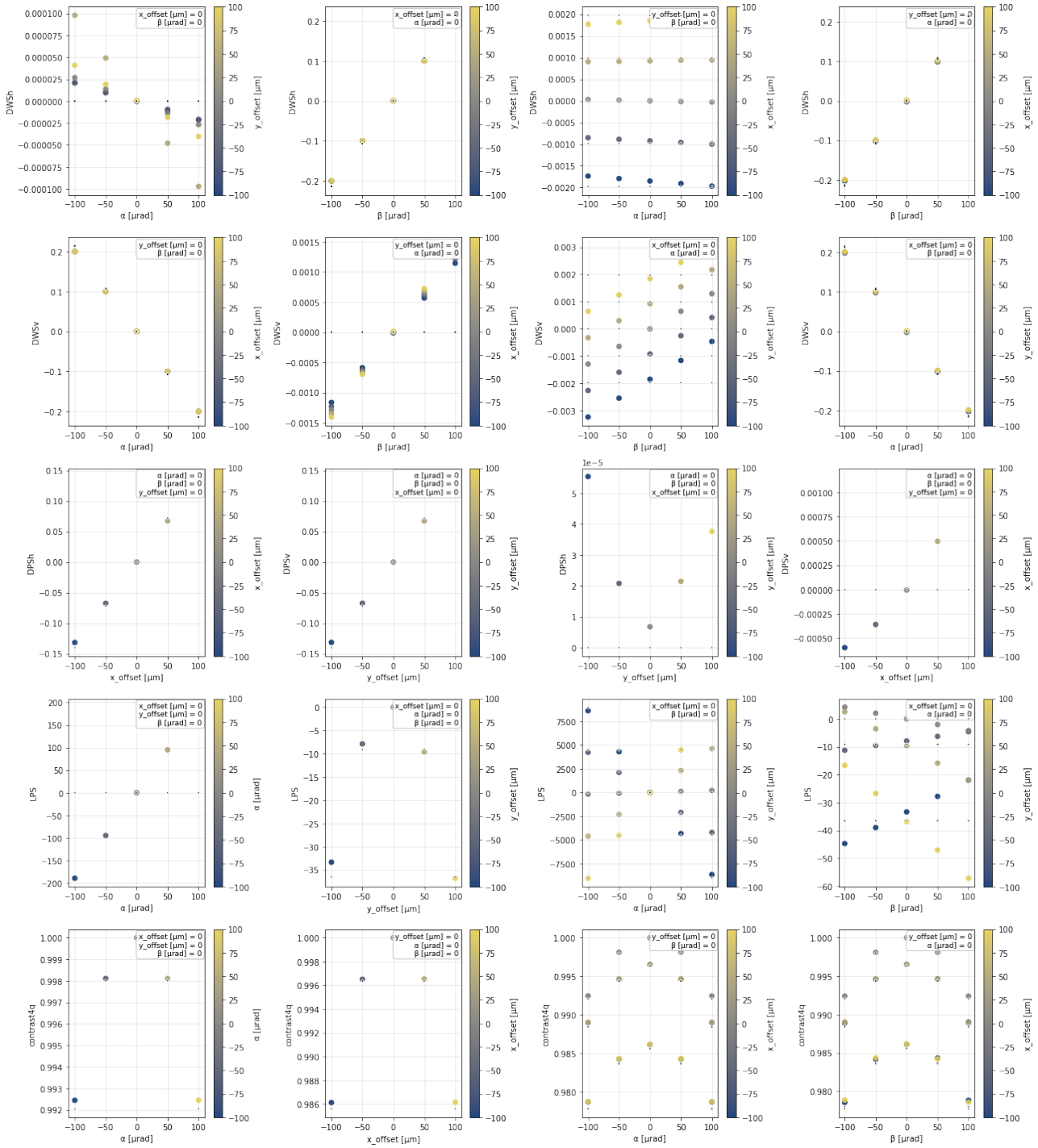


Figure 45: QPD radius = 500 µm, gap = 40 µm

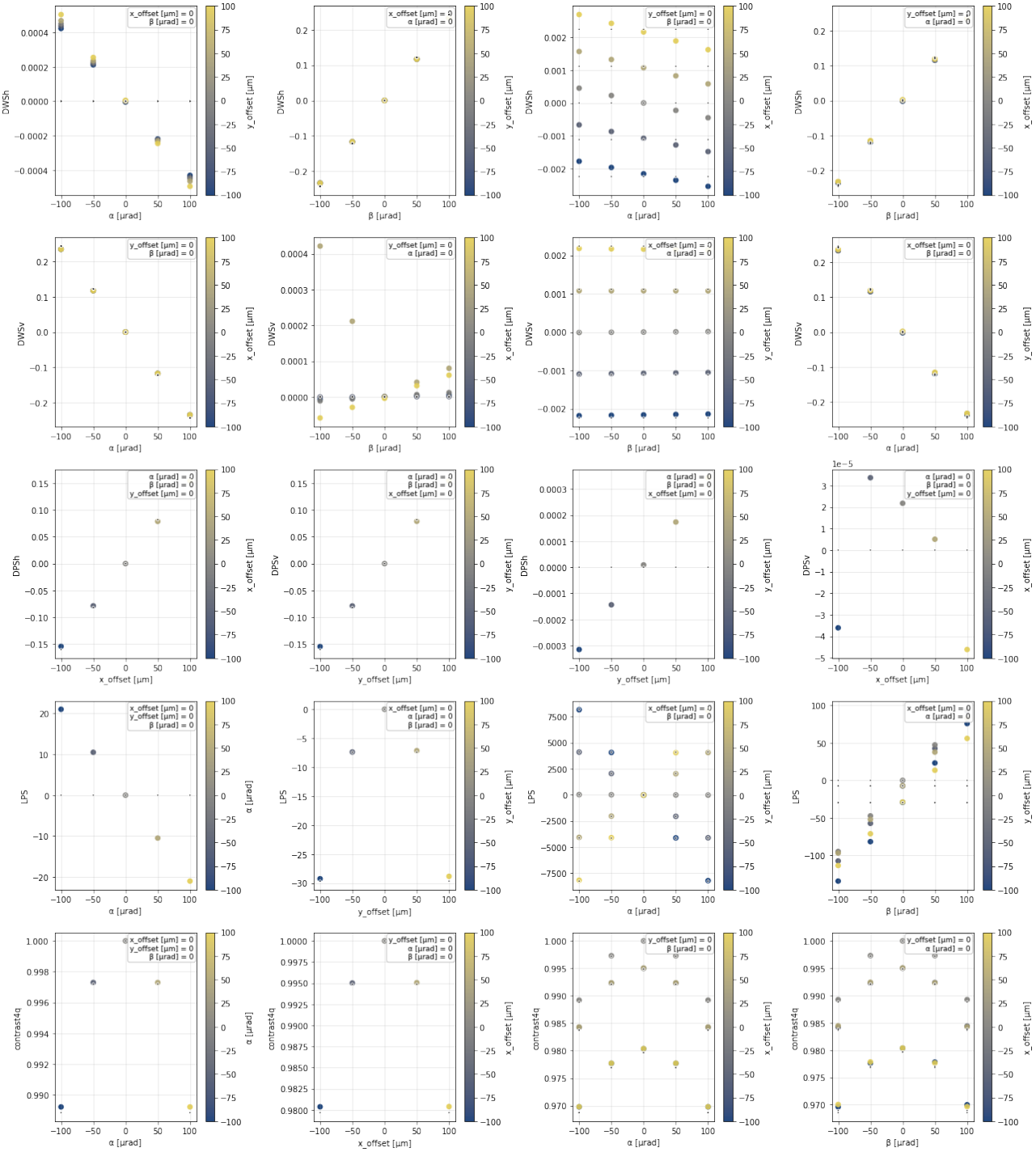


Figure 46: QPD radius = 1000 µm, gap = 20 µm

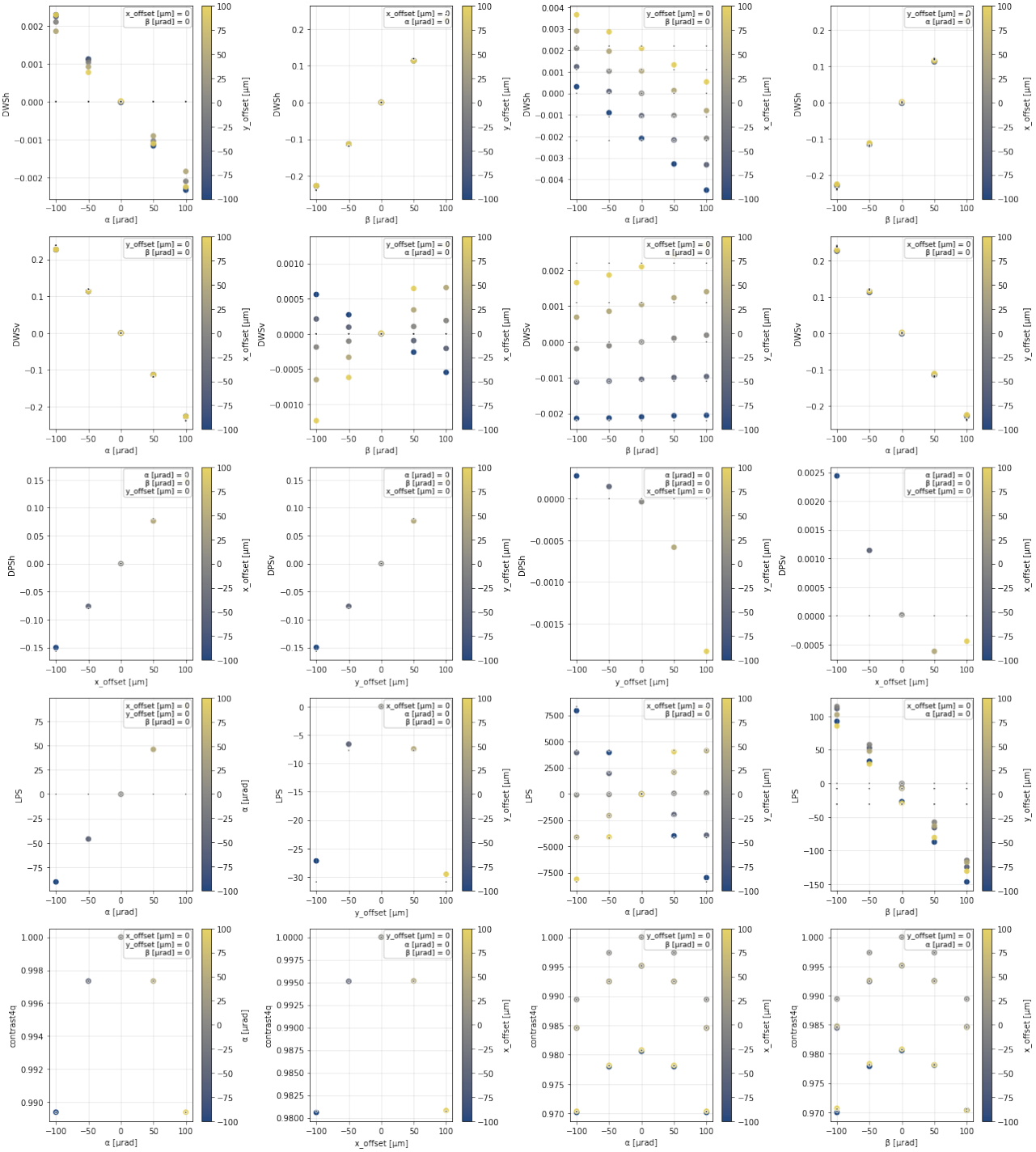


Figure 47: QPD radius = 750 μm, gap = 20 μm

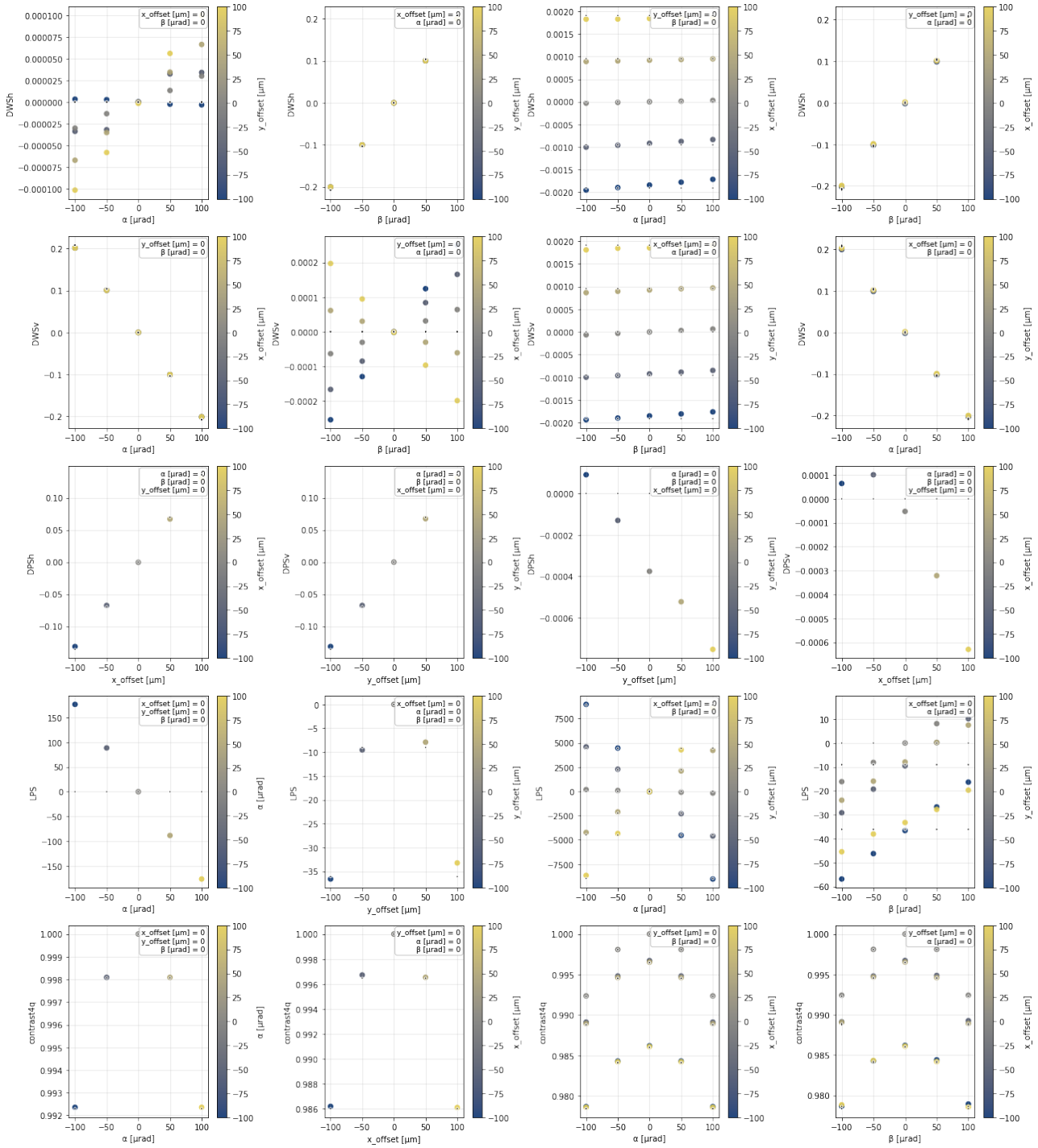


Figure 48: QPD radius = 500 µm, gap = 20 µm

J. Modifications to IfoCAD libraries

Listing 1: gbdc_beam.cpp

```
21
22  static inline Eigen::Vector3d to_eig(const Direction_3d& d){ return {d.x(),d.y(),d.z()}; }
23  static inline Eigen::Vector3d to_eig(const Vector_3d& v){ return {v.x(),v.y(),v.z()}; }
24
25  ...
26
342 const Direction_3d& GBDC_Beam::normal_vector_horizontal() const
343 {
344     return Beam_Properties::normal_vector_horizontal();
345     return m_normal_vector_horizontal;
346 }
347
348 const Direction_3d& GBDC_Beam::normal_vector_vertical() const
349 {
350     return Beam_Properties::normal_vector_vertical();
351     return m_normal_vector_vertical;
352 }
353
354 ...
355
684 const Direction_3d& GBDC_Beam::get_direction() const {
685     return m_ray.direction();
686 }
687 const Point_3d& GBDC_Beam::get_origin() const {
688     return m_ray.origin();
689 }
690 void GBDC_Beam::translate_by(const Vector_3d& delta)
691 {
692     #pragma omp parallel
693     {
694         #pragma omp for schedule(dynamic) nowait
695         for(size_t i=0;i<m_beam_matrix.size();++i)
696             for(size_t j=0;j<m_beam_matrix[i].size();++j)
697                 m_beam_matrix[i][j].set_origin(m_beam_matrix
698     }
699     m_ray.set_origin(m_ray.origin()+delta);
700 }
701 void GBDC_Beam::repoint(const Direction_3d& new_dir,
702                         const Direction_3d& new_nvv,
703                         const Point_3d& pivot)
704 {
705     Eigen::Vector3d z0 = to_eig(get_direction()); z0.normali
706     Eigen::Vector3d y0 = to_eig(m_normal_vector_vertical); y
707     Eigen::Vector3d x0 = z0.cross(y0); x0.normalize();
708     Eigen::Vector3d z1 = to_eig(new_dir); z1.normalize();
709     Eigen::Vector3d y1 = to_eig(new_nvv); y1.normalize();
710     Eigen::Vector3d x1 = z1.cross(y1); x1.normalize();
711     Eigen::Matrix3d R0; R0.col(0)=x0; R0.col(1)=y0; R0.col(2
712     Eigen::Matrix3d R1; R1.col(0)=x1; R1.col(1)=y1; R1.col(2
713     Eigen::Matrix3d R = R1 * R0.transpose();
714     #pragma omp parallel
715     {
716         #pragma omp for schedule(dynamic) nowait
717         for(size_t i=0;i<m_beam_matrix.size();++i)
718             for(size_t j=0;j<m_beam_matrix[i].size();++j)
719                 m_beam_matrix[i][j].rotate_3d(R, pivot);
720     }
721     Base::rotate_3d(R, pivot);
722     m_normal_vector_vertical   = new_nvv;
723     m_normal_vector_horizontal = Direction_3d(x1(0),x1(1),x1
724     // Pin master ray exactly at pivot with desired basis; d
725     m_ray.set_origin(pivot);
726     m_ray.set_direction(new_dir);
727 }
728 end{added}
```

Listing 2: gbdc_beam.h

```
39  class GBDC_Grid_Beam final : public VirtualAssignable<GBDC_Grid_Beam, GA_GBeam>
40  {
41  public:
42  ...
43
40      double gbdc_phase() const { return m_gbdc_phase; }
41
42  ...
43
51  /**
52   * @brief Gaussian beam decomposed wavefront
53   */
54  class GBDC_Beam final : public VirtualAssignable<GBDC_Beam, Beam_Properties>
55  {
56  public:
57
58  ...
59
217
218     const Direction_3d& get_direction() const;
219     const Point_3d& get_origin() const;
220     void translate_by(const Vector_3d& delta);
221     void repoint(const Direction_3d& new_dir,
222                  const Direction_3d& new_nvv,
223                  const Point_3d& pivot);
```

Listing 3: sa_hgbeam.cpp

```cpp
void SA_HGBeam::clip_mode_coefficients(double threshold) {
    for (int k = 0; k < m_coefficient_matrix.outerSize(); ++k) {
        for (SparseComplex::InnerIterator it(m_coefficient_matrix, k); it; ++it) {
            const int row = it.row();
            const int col = it.col();
            const cdouble value = it.value();
            if (std::abs(value) < threshold) {
                m_coefficient_matrix.coeffRef(row, col) = 0.0;
            }
        }
    }
    // Correct pruning using a lambda predicate
    m_coefficient_matrix.prune([threshold](const int&, const int&, const cdouble& val) {
        return std::abs(val) >= threshold;
    });
}
void SA_HGBeam::load_metadata(const std::string& file)
{
    std::ifstream fin(file);
    if (!fin.is_open()) {
        std::stringstream s;
        s << "Warning: Couldn't open metadata file: " << file << std::endl;
        error_handler(s.str().c_str(), MEDIUM_SEVERITY);
        return;
    }

    fin >> std::scientific >> std::setprecision(15);

    double qt_re, qt_im, qs_re, qs_im;
    double origin_x, origin_y, origin_z;
    double dir_x, dir_y, dir_z;
    double waist_rad_t, waist_rad_s;

    fin >> qt_re >> qt_im;
    fin >> qs_re >> qs_im;
    fin >> origin_x >> origin_y >> origin_z;
    fin >> dir_x >> dir_y >> dir_z;
    fin >> waist_rad_t >> waist_rad_s;

    fin.close();

    // Set beam parameters
    set_q(cdouble(qt_re, qt_im), cdouble(qs_re, qs_im));
    set_origin(Point_3d(origin_x, origin_y, origin_z));
    set_direction(Direction_3d(dir_x, dir_y, dir_z));
    set_q_waist_rad(waist_rad_t, 0.0, waist_rad_s, 0.0);  // optional if you want to explicitly re-set waist

    // You might also want to re-calculate constants depending on usage
    // precompute_consts();

...

const Direction_3d& SA_HGBeam::get_direction() const {
    return m_ray.direction();
}

const Point_3d& SA_HGBeam::get_origin() const {
    return m_ray.origin();
}
```

Listing 4: sa_hgbeam.h

```cpp
public:
    /**
     * @brief Clips all mode coefficients in the beam's coefficient matrix that fall below the
     *        specified amplitude threshold.
     *
     *        This method iterates over all non-zero Hermite-Gaussian mode coefficients in the
     *        beam's sparse coefficient matrix. If the magnitude of a coefficient is smaller than
     *        the threshold, it is set to zero. This is useful for cleaning up numerical noise or
     *        eliminating modes that contribute negligibly to the beam.
     *
     *        After zeroing out small coefficients, the method prunes the sparse matrix to
     *        eliminate stored zero entries, ensuring efficient storage and processing.
     *
     * @param threshold The amplitude below which coefficients are considered negligible
     *                  and are removed (set to zero).
     */
    void clip_mode_coefficients(double threshold);

     /**
      * @brief Loads beam parameters (q-parameters, origin, waist radii) from a metadata file.
      *
      * This function restores the internal state of the beam using a plain-text metadata file
      * saved with `save_metadata`. It sets the tangential and sagittal q-parameters, the beam origin,
      * and the tangential and sagittal waist radii. This metadata is used to fully define the geometry
      * and propagation state of the beam, allowing accurate reconstruction and continuation (e.g., propagation).
      *
      * The metadata file must contain the following values in scientific format, each on its own line:
      * 1. Real and imaginary part of q_t (tangential q-parameter)
      * 2. Real and imaginary part of q_s (sagittal q-parameter)
      * 3. Origin coordinates: x y z
      * 4. Waist radii: waist_t waist_s
      *
      * Example contents of the metadata file:
      * @code
      * 1.000000000000000e-03 2.000000000000000e-03
      * 1.000000000000000e-03 2.000000000000000e-03
      * 0.000000000000000e+00 0.000000000000000e+00 0.000000000000000e+00
```

```
256         * 5.000000000000000e-01 5.000000000000000e-01
257         * @endcode
258         *
259         * @param file Full path to the metadata file to load.
260         */
261     void load_metadata(const std::string& file);
262 
263     const Direction_3d& get_direction() const;
264     const Point_3d& get_origin() const;
```

Listing 5: pd_signals.h

```
1671 template <class T1, class T2>
1672 void PD_signals <T1, T2>::phasemeter_complex_integration(size_t segment, T1 ref_beam, T2 meas_beam)
1673 {
1674     double realMix, imagMix;
1675     double result, abserror;
1676     int esc;
1677     Eigen::VectorXd data_resp = m_photodiode.get_data_resp();
1678 
1679     if (data_resp.size() != 0) {
1680         m_integration_method = GSL_SINGLE_DIMENSION_INTEGRATOR;
1681         // m_integration_method = GSL_SINGLE_DIMENSION_INTEGRATOR;
1682     }
```

K. Beam profiles

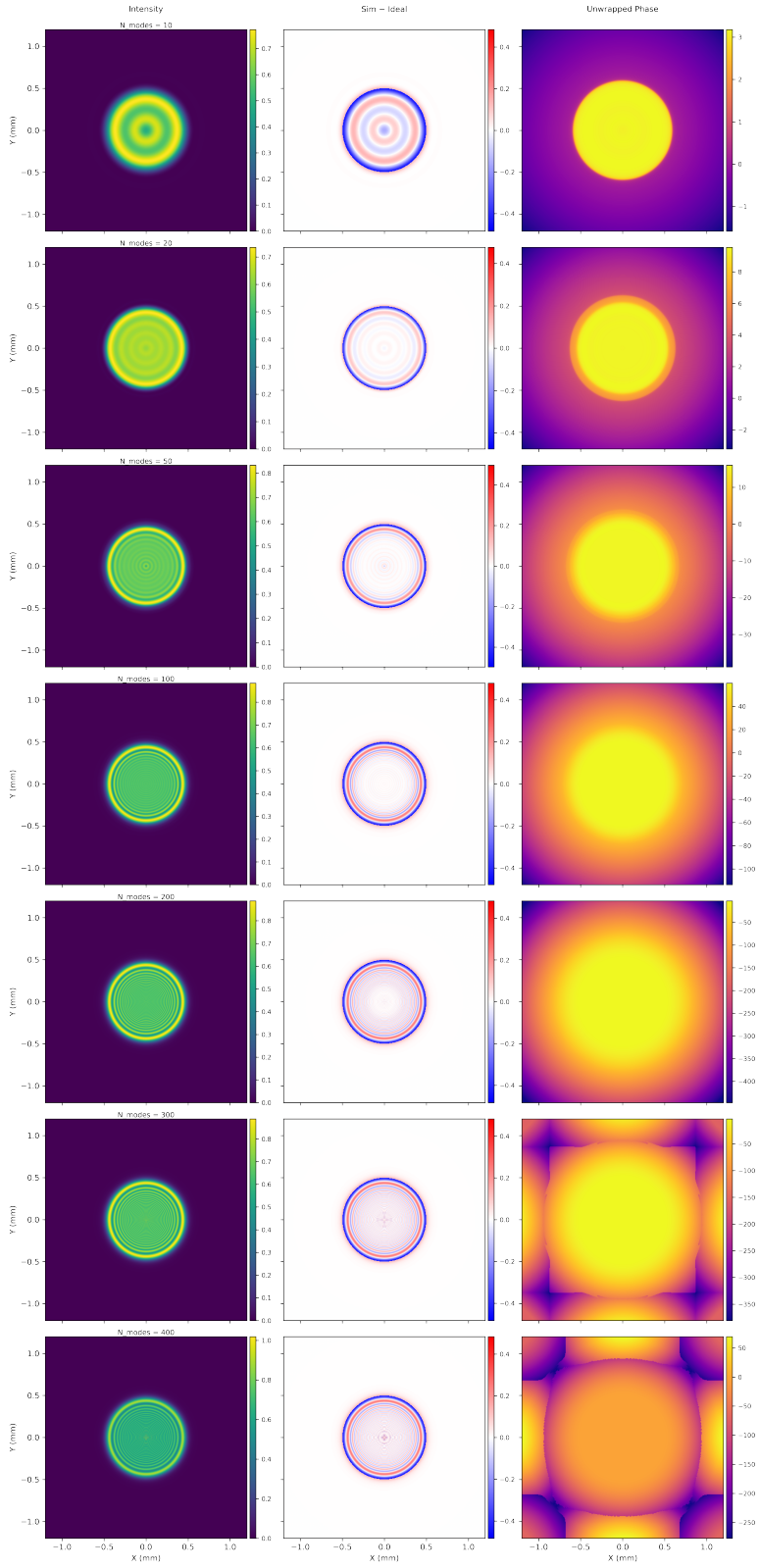
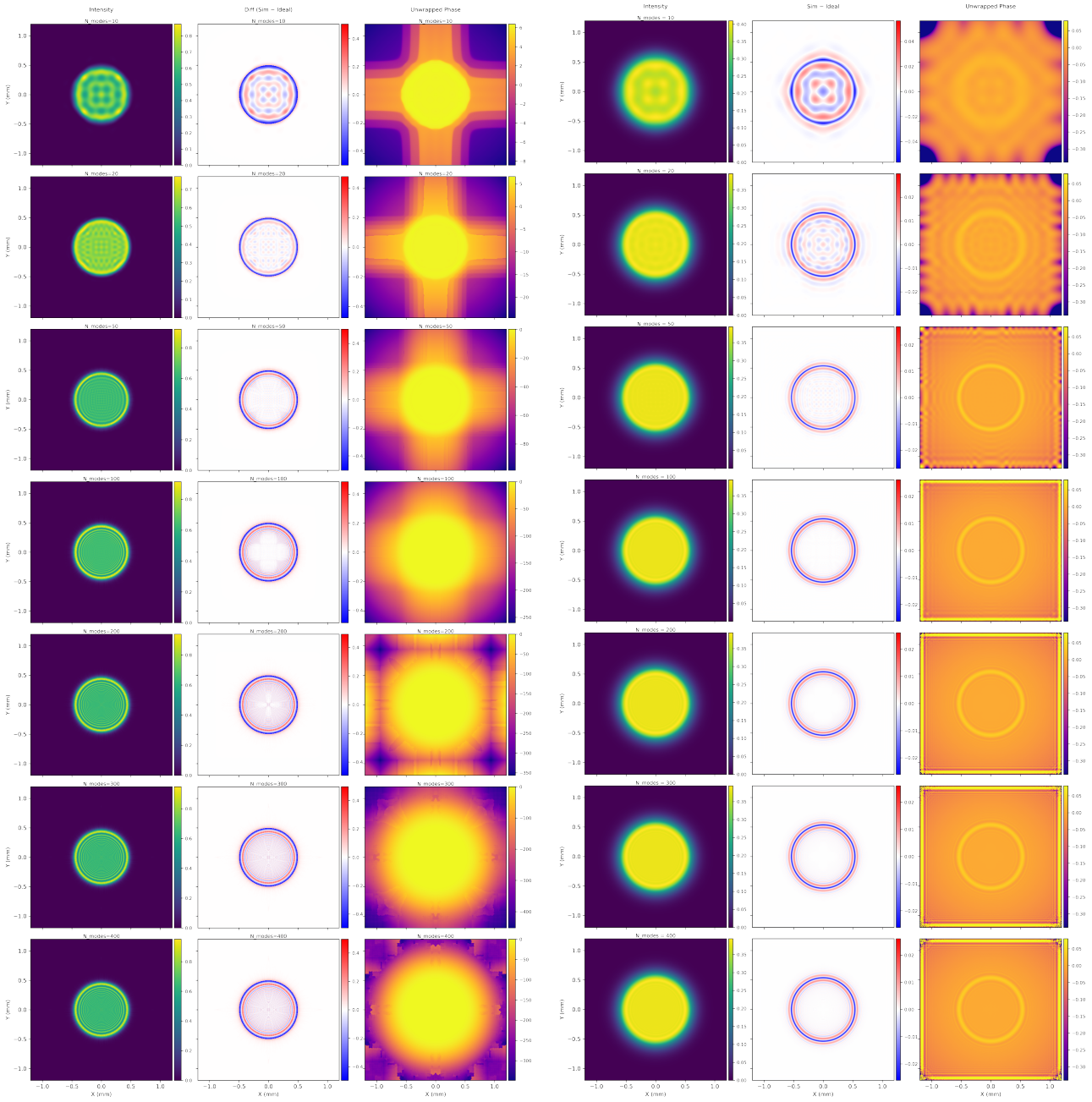
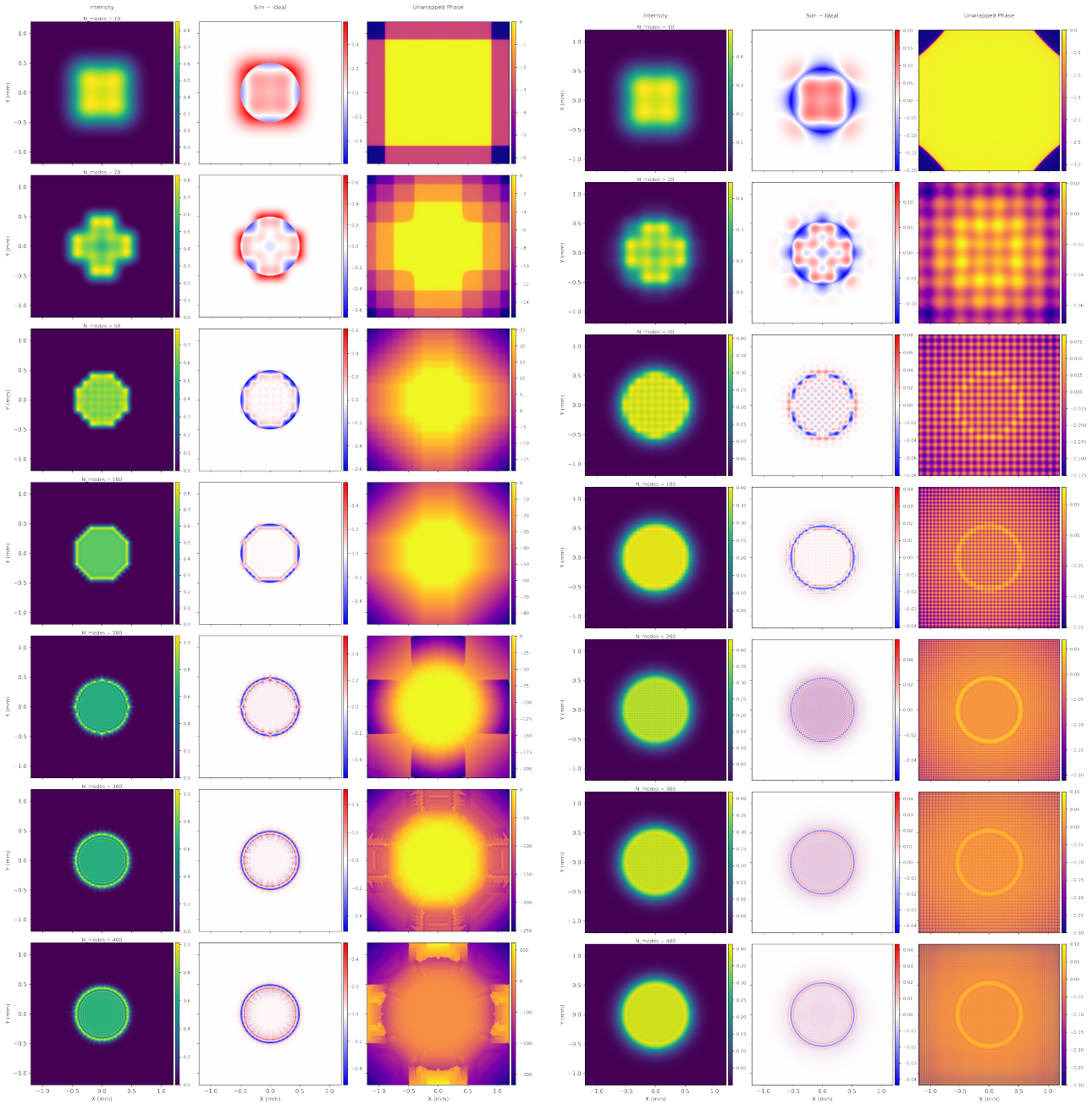


Figure 49: MEM



(a) (b)

Figure 50: NMF



(a) (b)

Figure 51: GBDC

L. All simulation data from Tophat analysis

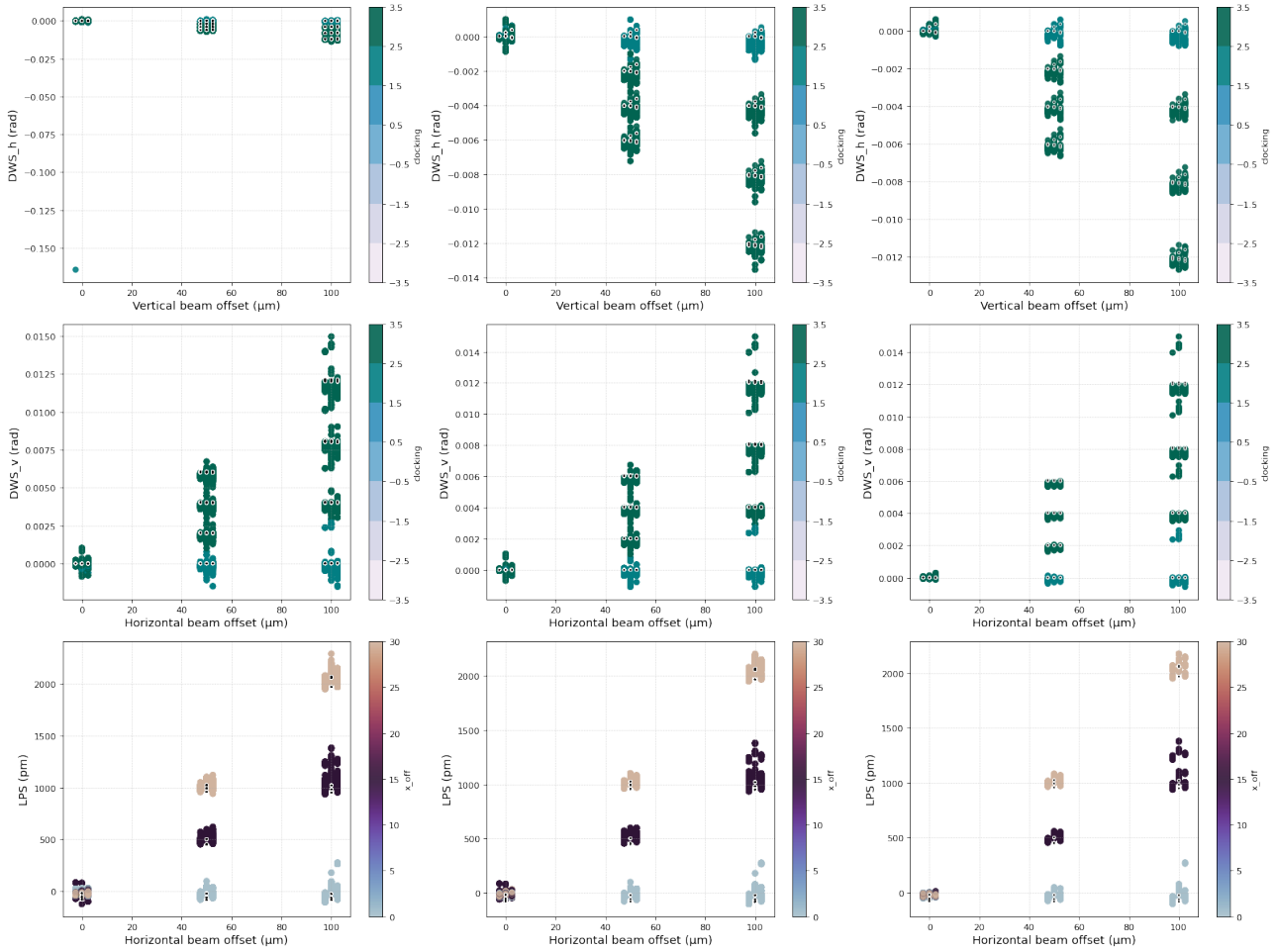


Figure 52: MEM tophat, GSL integration

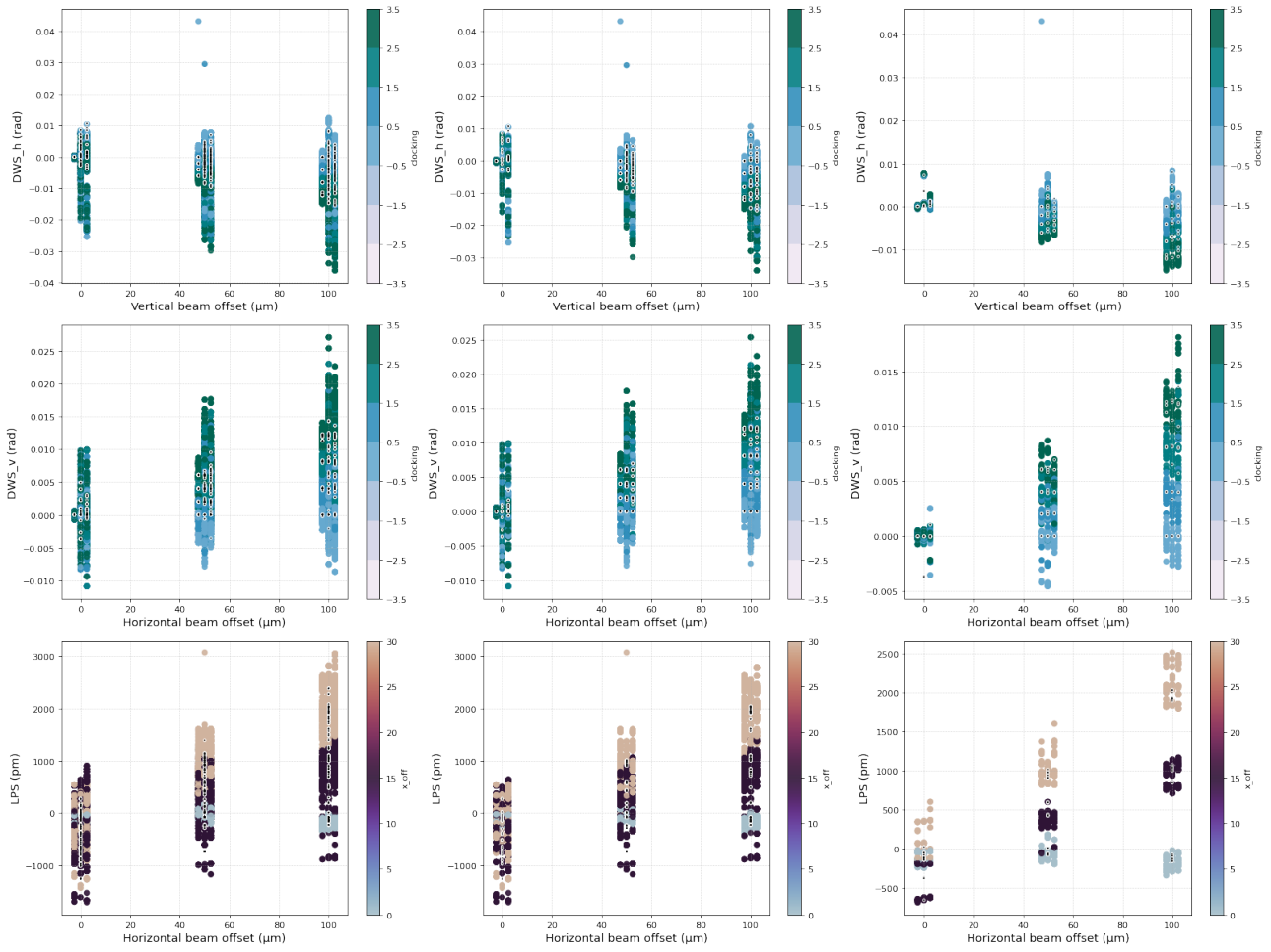


Figure 53: MEM tophat, DCUHRE integration

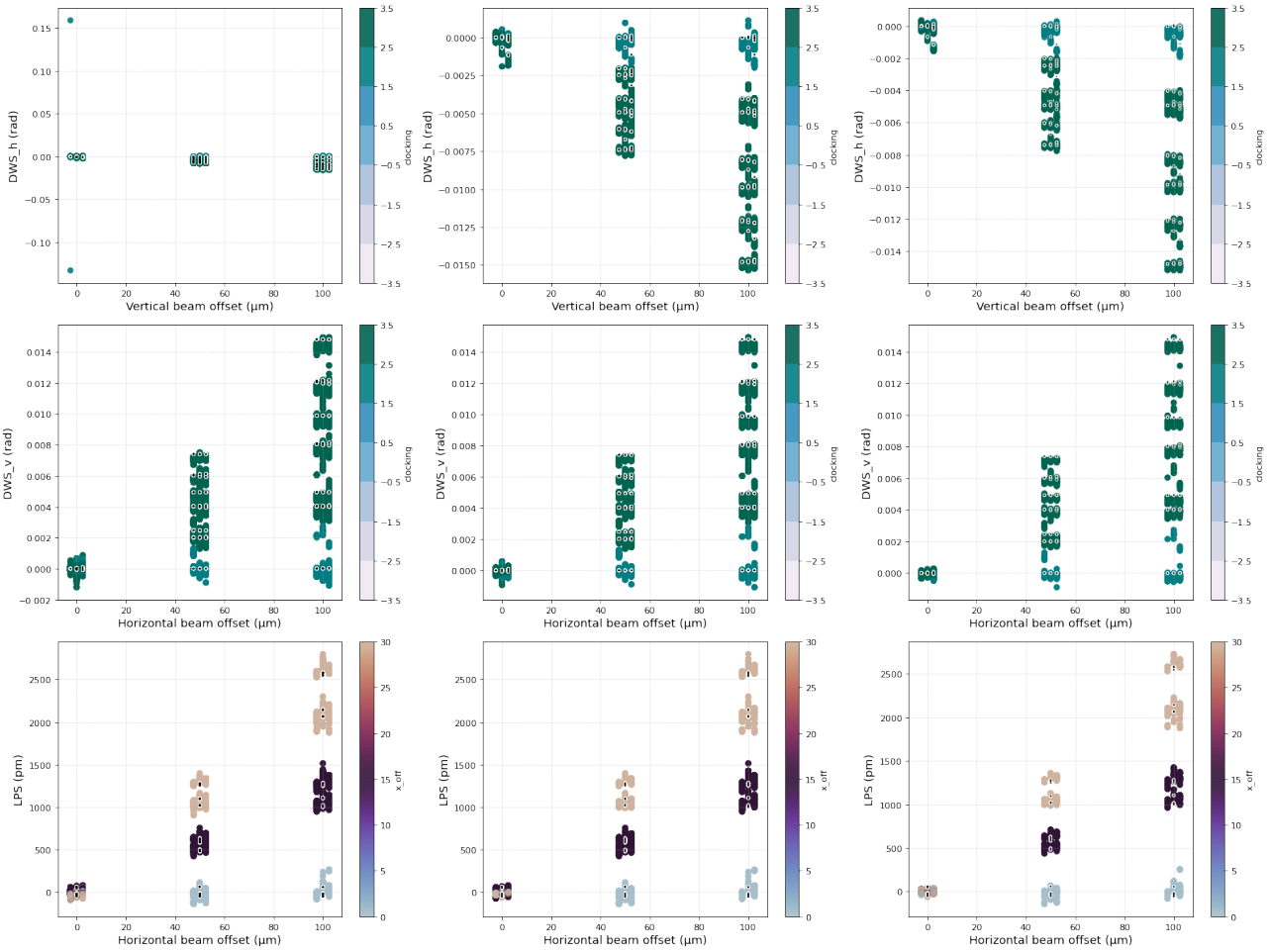


Figure 54: NMF tophat, GSL integration

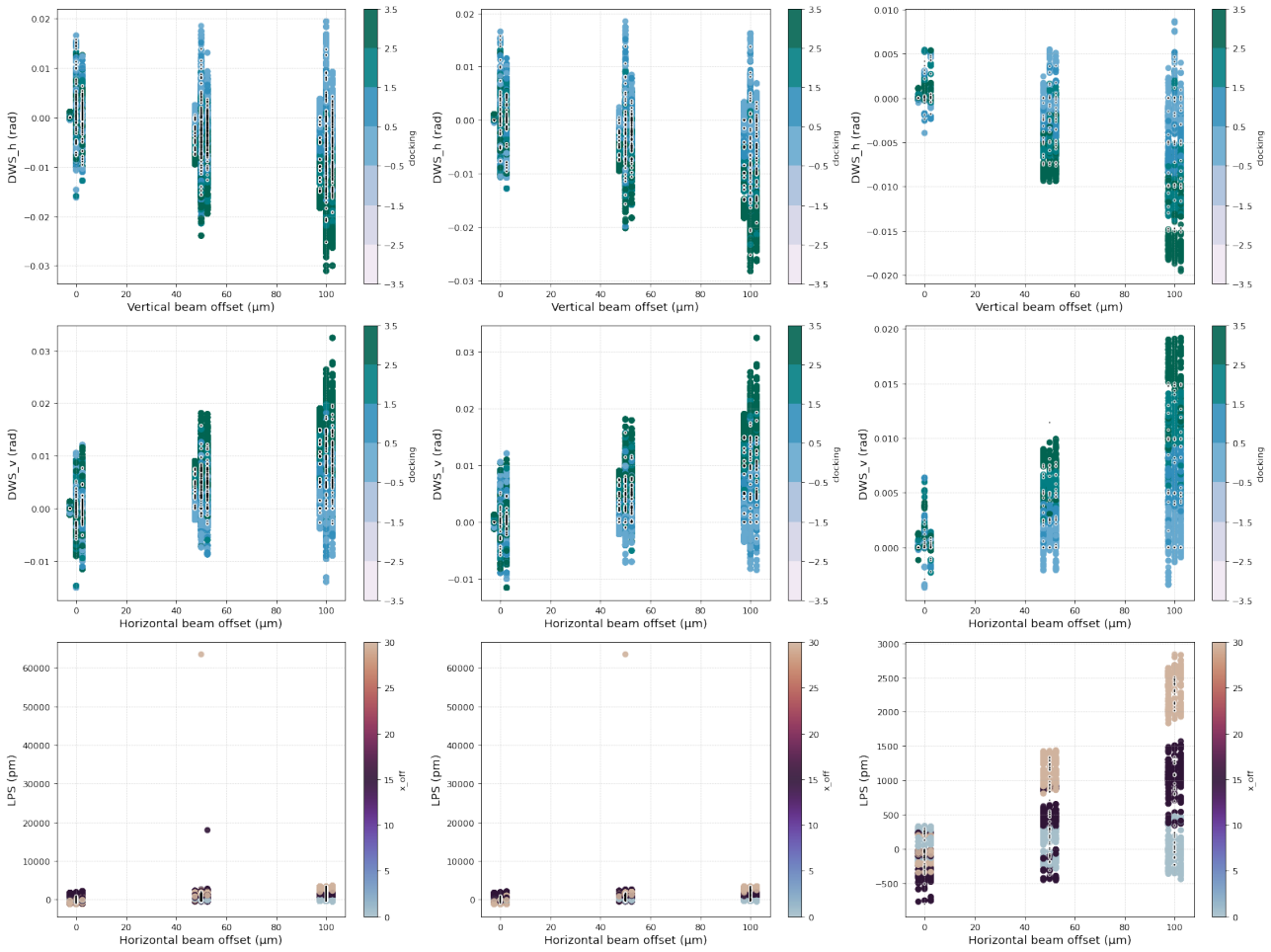


Figure 55: NMF tophat, DCUHRE integration

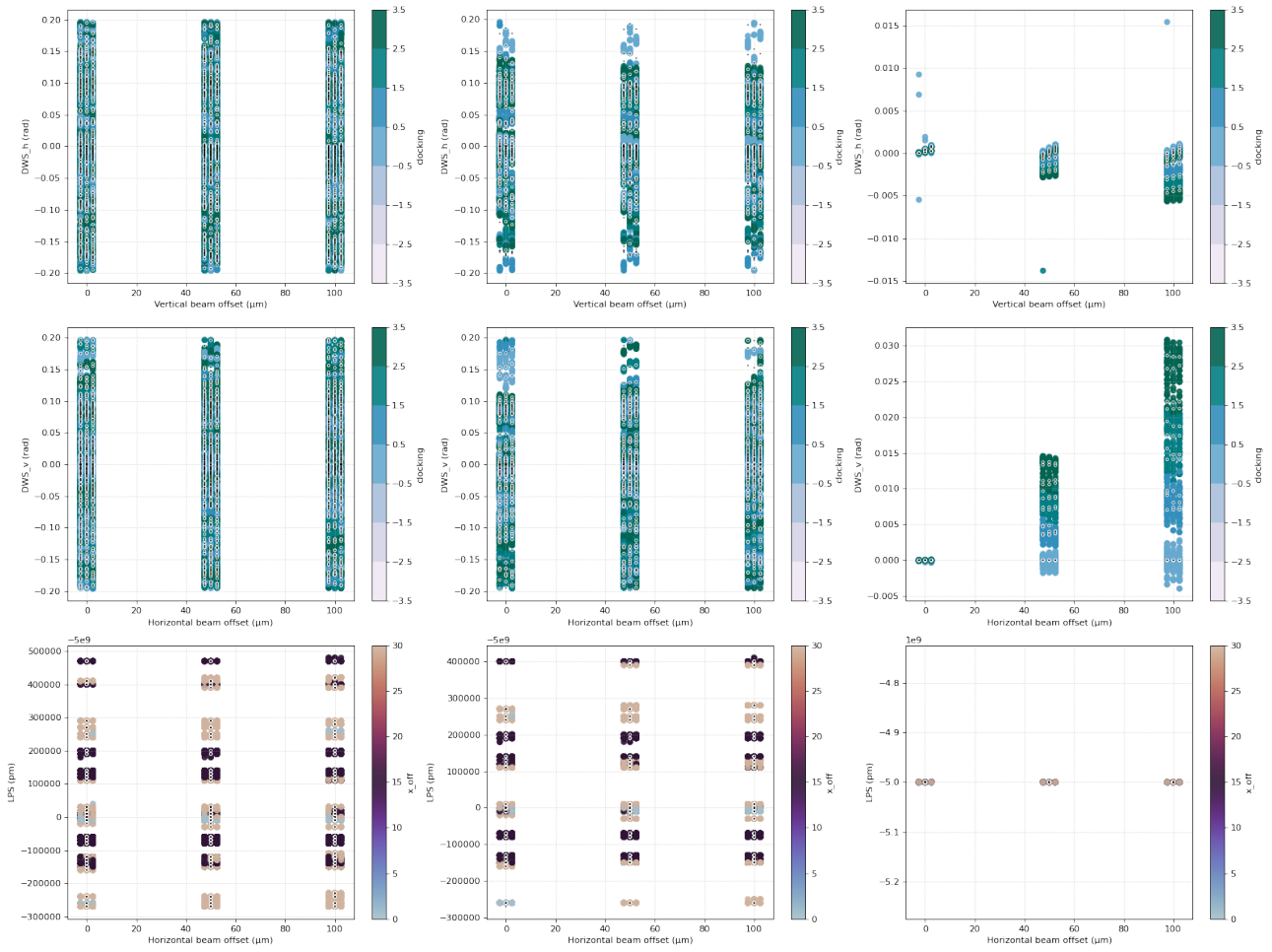


Figure 56: GBDC tophat, DCUHRE integration

M. R2 analysis for clocking and beam offset to DWS fit

We assess how well the bilinear model

$$\mathrm{DWS} = a \times \text{clocking} \times \text{beam-rotation} \tag{57}$$

captures the data. Across all beam types combined, a substantial fraction of cases reach $R^2 > 0.95$, indicating that the bilinear form often describes the data well. For Gaussian beams the agreement is strongest, with about 83% of cases above the 0.95 threshold. NMF and MEM also perform well, with 70% of cases exceeding 0.95. By contrast, GBDC shows much weaker agreement: only 3% of the full dataset surpasses 0.95. However, when restricting to the cleaner subset with tip $= 0$ and tilt $= 0$, the fraction increases to about 42%, indicating a partial recovery but still below the mode-decomposed beams.

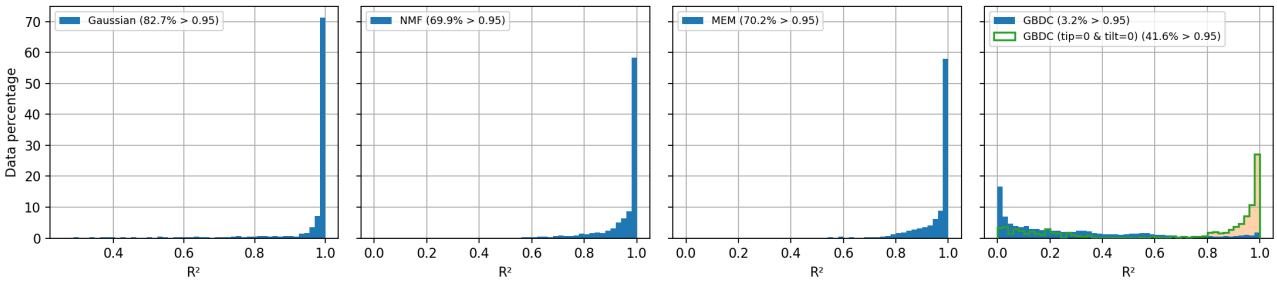


Figure 57: Histograms of R^2 values for the bilinear fit $[\mathrm{DWS} = a \times \text{clocking} \times \text{beam-rotation},]$. From left to right: Gaussian, NMF, MEM, GBDC with all data in blue and the restricted subset (tip$= 0$, tilt$= 0$) overlaid in green. The x-axis shows the R^2 value of the fit; the y-axis shows the fraction of data falling in each bin.

N. LPS coupling coefficients for all beam types

Beam type	Variant	k_{LPS} [pm μm^{-1} μrad^{-1}]
Gaussian	ideal	0.83668 ± 0.00006
Gaussian	real	0.83058 ± 0.00299
MEM (trunc)	ideal	0.69482 ± 0.01489
MEM (trunc)	real	0.69558 ± 0.01957
NMF (notrunc)	ideal	0.86614 ± 0.00128
NMF (notrunc)	real	0.87399 ± 0.01201
NMF (trunc)	ideal	0.69876 ± 0.00280
NMF (trunc)	real	0.70267 ± 0.01400

Beam type	Variant	k_{LPS} [pm μm^{-1} μrad^{-1}]
Gaussian	ideal	0.80675 ± 0.05469
Gaussian	real	0.88794 ± 0.04932
MEM (trunc)	ideal	0.62402 ± 0.09901
MEM (trunc)	real	0.67072 ± 0.13047
NMF (notrunc)	ideal	0.87387 ± 0.06556
NMF (notrunc)	real	0.92569 ± 0.10853
NMF (trunc)	ideal	0.68344 ± 0.07366
NMF (trunc)	real	0.62468 ± 0.11745

Table 2: Mean $\pm$ error of fitted k_{LPS} for each set shown in Fig. 32.

O. C++ code

Listing 6: Beam to QPD simulation, simplest form

```cpp
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <cmath>
#include <list>
#include "signals/pd_signals.h"
#include "beams/ga_gbeam.h"
#include "tracing/beam_tracing.h"

int main() {
    std::string input_file = "/Users/martijn/Documents/Project_Nikhef/data/lab_data/pickle_RF03MHz_1.pkl";
    std::string command = "python Clocking.py " + input_file + " DC";
    system(command.c_str());

    std::ofstream outfile("ideal_output_-z.txt");
    std::ofstream real_out("real_output_-z.txt");

    std::vector<std::pair<double, double>> offsets = {
        {-40,  40},{ 40, -40},{-40, -40},{ 40,  40},{0, 0}};
    GA_GBeam b1, b2;
    for (auto* b : {&b1, &b2}) {
        b->set_power(0.5);
        b->set_circular_waist_rad(0.5, 0);
        b->set_direction_xyz(0, 0, -1);
    }

    for (const auto& [x_offset, y_offset] : offsets) {
        b1.set_origin_xyz(0, 0, 100);
        b2.set_origin_xyz(x_offset*0.001, y_offset*0.001,100);

        Photodiode pd;
        pd.set_nv(Direction_3d(0,0,1),Direction_3d(0,1,0));
        pd.set_center_xyz(0, 0, 0);
        pd.set_active_area_half_width(0.75);
        pd.set_slit_width(0.01);

        PD_signals<GA_GBeam, GA_GBeam> signal;
        signal.set_parameters(b2, b1, pd);

        double A = signal.DC_power_A(), B = signal.DC_power_B();
        double C = signal.DC_power_C(), D = signal.DC_power_D();
        double E = signal.longitudinal_4q_AP();
        double F = signal.DWS_horizontal_AP(), G = signal.DWS_vertical_AP();

        outfile << x_offset << " " << y_offset << " ";
        for (double val : {A, B, C, D, E, F, G}) outfile << val << " ";
        outfile << std::endl;
        // real data
        pd.set_responsivity_on();
        Eigen::VectorXd addParam=pd.get_data_resp();
        pd.set_nv(Direction_3d(0,0,1),Direction_3d(0,1,0));
        signal.set_parameters(b2, b1, pd);

        A = signal.DC_power_A(); B = signal.DC_power_B();
        C = signal.DC_power_C(); D = signal.DC_power_D();
        E = signal.longitudinal_4q_AP();
        F = signal.DWS_horizontal_AP(); G = signal.DWS_vertical_AP();
        real_out << x_offset << " " << y_offset << " ";
        for (double val : {A, B, C, D, E, F, G}) real_out << val << " ";
        real_out << std::endl;
    }
    outfile.close();
    real_out.close();
    return 0;
}
```

Listing 7: Gaussian Beam full simulation

```cpp
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <cmath>
#include <list>
#include <filesystem>



// Include your specific libraries as needed
#include "signals/pd_signals.h"
#include "beams/ga_gbeam.h"
#include "tracing/beam_tracing.h"

// Function to display the loading bar
void displayDualLoadingBars(int progress1, int total1, int progress2, int total2, int barWidth = 50) {
    float percent1 = static_cast<float>(progress1) / total1;
    float percent2 = static_cast<float>(progress2) / total2;

    int filled1 = static_cast<int>(barWidth * percent1);
    int filled2 = static_cast<int>(barWidth * percent2);

    std::cout << "\033[F\r"; // Move up 1 line
    std::cout << "[";
```

```
    for (int i = 0; i < barWidth; ++i) std::cout << (i < filled1 ? "=" : (i == filled1 ? ">" : " "));
    std::cout << "] " << int(percent1 * 100.0) << "%\n";

    std::cout << "\r[";
    for (int i = 0; i < barWidth; ++i) std::cout << (i < filled2 ? "=" : (i == filled2 ? ">" : " "));
    std::cout << "] " << int(percent2 * 100.0) << "%\r";
    std::cout.flush();
}


int main(int argc, char* argv[]) {
    if (argc < 3) {
        std::cerr << "Error: missing input file argument\n";
        return 1;
      }
    std::string file = argv[1];
    std::string mode = argv[2];

    std::cout << "Running with: " << file << " mode: " << mode << std::endl;


    // Iterate over each input file
    for (const auto& f : std::vector<std::string>{file}) {
        bool is_ideal = (file == "ideal");
        // Extract file prefix for naming output files
        std::string prefix = is_ideal ? "reference" : file.substr(0, file.find_last_of('.'));
        std::string output_directory = "/data/detrd/mmoorlag/sim_data_2.0/sim_data_" + std::filesystem::path(__FILE__).stem().
            string() + "/";
        std::filesystem::create_directories(output_directory);

        std::string outfile_name = (is_ideal ? "idealQPD_" : "realQPD_") + prefix + "_" + mode + ".txt";
        std::ofstream outfile(output_directory + outfile_name);
        if (!outfile.is_open()) {
            std::cerr << "Failed to open output file: " << outfile_name << std::endl;
            return 1;}
        std::string header = "DC/RF tip(Rad) tilt(Rad) clocking(Rad) x-rot(muRad) y-rot(muRad) x-off(micron) y-off(micron) A(mA) B
            (mA) C(mA) D(mA) LPS(micron) DWS_h DWS_v ";
        outfile << header << std::endl;

        std::vector<std::string> signal_types = {mode};

        // Run the Python script for the current file
        for (const std::string& signal_type : signal_types) {
            if (!is_ideal) {
                std::string full_path = "/data/detrd/mmoorlag/lab_data/QPD1/" + file;
                std::string python_path = "/user/mmoorlag/myenv/bin/python";
                std::string script_path = "/user/mmoorlag/opt/ifocad_cpp/projects/Martijnstbc/Clocking.py";

            std::string command = python_path + " " + script_path + " " + full_path + " " + signal_type;
            std::cout << "Executing: " << command << std::endl;

                int retcode = system(command.c_str());
                if (retcode != 0) {
                    std::cerr << "ERROR: Python command exited with code " << retcode << std::endl;
                }
                }
                else {
                    std::cout << "Ideal\n" << std::endl;
                }
            // initialize parameters for the simulation
            const double beam_rotation_step = 50;

            const double clock_start = 0.00135;
            const double clock_end = 0.10605;
            const double clocking_step = 0.01745; // 1 degree in radians

            const double tiptilt_start = -0.03490658504*2;
            const double tiptilt_end   = 0.03490658504*2;      // same as start for single value
            const double tiptilt_step  = 0.03490658504;

            const double x_off_start = -80; // in microns
            const double x_off_end = 80; // in microns
            const double x_off_step = 20; // in microns

            // Set up Gaussian beams
            const double L = 5; // Distance beam origin to QPD
            GA_GBeam b1;
            GA_GBeam b2;
            for (auto* b : {&b1, &b2}) {
                b->set_power(0.5);
                b->set_origin_xyz(0, 0, L);
                b->set_circular_waist_rad(0.5, 0);
                b->set_direction_xyz(0, 0, -1);
            }

            for (double beam_x_rot = 0; beam_x_rot <=1.1 ;beam_x_rot+=1) { // leave this one out when clocking> pi
                for (double tip = tiptilt_start; tip <= tiptilt_end+1e-8; tip += tiptilt_step) { //rotation around x-axis
                    for (double tilt = tiptilt_start; tilt <= tiptilt_end+1e-8; tilt += tiptilt_step) { //rotation around y-axis
                        for (double clock = clock_start; clock <= clock_end+1e-8; clock += clocking_step) {
                            // Setup loading bar
                            int tip_index = round((tip - tiptilt_start) / tiptilt_step);
                            int tilt_index = round((tilt - tiptilt_start) / tiptilt_step);
                            int tip_total = round((tiptilt_end - tiptilt_start) / tiptilt_step) + 1;
                            int tilt_total = round((tiptilt_end - tiptilt_start) / tiptilt_step) + 1;

                            int clock_index = round((clock - clock_start) / clocking_step);
                            int clock_total = round((clock_end - clock_start) / clocking_step) + 1;

                            displayDualLoadingBars(tip_index + 1, tip_total, clock_index + 1, clock_total);

                            double roll  = clock;

                            // Create rotation matrices
```

```
                            Eigen::Matrix3d Rx;
                                Rx << 1, 0, 0,
                                    0, cos(tip), -sin(tip),
                                    0, sin(tip), cos(tip);
                                Eigen::Matrix3d Ry;
                                Ry << cos(tilt), 0, sin(tilt),
                                    0, 1, 0,
                                    -sin(tilt), 0, cos(tilt);
                                Eigen::Matrix3d Rz;
                                Rz << cos(roll), -sin(roll), 0,
                                    sin(roll), cos(roll), 0,
                                    0, 0, 1;

                            // Final rotation matrix: apply tip, then tilt, then roll
                            Eigen::Matrix3d R = Rz * Ry * Rx;

                            // Optional pivot point (e.g., center of your QPD)
                            Point_3d pivot(0, 0, 0);  // or actual center if known
                            ///////////
                            double x_beam_off = 0;
                            double y_beam_off = 0;
                            for (x_beam_off = x_off_start; x_beam_off <= x_off_end + 1e-8; x_beam_off += x_off_step) {
                            // for (y_beam_off = -40; y_beam_off <= 40.001; y_beam_off += 20) {
                            for (double i_beam_rot = -100; i_beam_rot <= 100.001; i_beam_rot += beam_rotation_step) {
                                // Round the values to desired decimal places
                                double rounded_tip = std::round(tip * 10000000.0) / 10000000.0;
                                double rounded_tilt = std::round(tilt * 10000000.0) / 10000000.0;
                                double rounded_roll = std::round(roll * 10000000.0) / 10000000.0;

                                if (beam_x_rot == 0) {
                                    outfile << (signal_type == "DC" ? 0 : 1) << " " << rounded_tip << " " << rounded_tilt << " "
                                          << rounded_roll << " " << i_beam_rot << " NaN "<< " " << x_beam_off << " " << y_beam_off
                                           << " ";
                                    double x0 = L * std::sin(1e-6 * i_beam_rot), z0 = L * std::cos(1e-6 * i_beam_rot);
                                    b2.set_origin_xyz(0.001 * x_beam_off + x0, 0.001 * y_beam_off, z0);
                                    Eigen::Vector3d dir = -Eigen::Vector3d(x0, 0, z0).normalized();
                                    b2.set_direction_xyz(dir(0), dir(1), dir(2));
                                } else {
                                    outfile << (signal_type == "DC" ? 0 : 1) << " " << rounded_tip << " " << rounded_tilt << " "
                                          << rounded_roll << " " << "NaN " << i_beam_rot << " " << x_beam_off << " " << y_beam_off
                                           << " ";
                                    double y0 = L * std::sin(1e-6 * i_beam_rot), z0 = L * std::cos(1e-6 * i_beam_rot);
                                    b2.set_origin_xyz(0.001 * x_beam_off, 0.001 * y_beam_off + y0, z0);
                                    Eigen::Vector3d dir = -Eigen::Vector3d(0, y0, z0).normalized();
                                    b2.set_direction_xyz(dir(0), dir(1), dir(2));
                                }

                                Photodiode pd;
                                pd.set_nv(Direction_3d(0, 0, 1), Direction_3d(0, 1, 0)); // Set normal vector and vertical vector
                                pd.set_center_xyz(0, 0, 0.);
                                pd.set_active_area_half_width(0.75);
                                pd.set_slit_width(0.02);

                                PD_signals<GA_GBeam, GA_GBeam> signal;
                                // signal.set_parameters(b2, b1, pd);

                                if (is_ideal) {
                                    // Ideal simulation
                                    pd.rotate_3d(R, pivot);  // Rotate photodiode in 3D
                                    signal.set_parameters(b2, b1, pd);
                                    double A1 = signal.DC_power_A(), B1 = signal.DC_power_B();
                                    double C1 = signal.DC_power_C(), D1 = signal.DC_power_D();
                                    double E1 = signal.longitudinal_4q_AP();
                                    double F1 = signal.DWS_horizontal_AP(), G1 = signal.DWS_vertical_AP();
                                    for (double val : {A1, B1, C1, D1, E1, F1, G1}) outfile << val << " ";
                                    outfile << std::endl;
                                }
                                else {
                                    // Real simulation
                                    pd.set_responsivity_on();
                                    Eigen::VectorXd addParam=pd.get_data_resp();
                                    pd.rotate_3d(R, pivot);  // Rotate photodiode in 3D

                                    signal.set_integration_absolute_error(1e-2);
                                    signal.set_integration_relative_error(1e-8);
                                    signal.set_parameters(b2, b1, pd);

                                    double A2 = signal.DC_power_A(), B2 = signal.DC_power_B();
                                    double C2 = signal.DC_power_C(), D2 = signal.DC_power_D();
                                    double E2 = signal.longitudinal_4q_AP();
                                    double F2 = signal.DWS_horizontal_AP(), G2 = signal.DWS_vertical_AP();
                                    for (double val : {A2, B2, C2, D2, E2, F2, G2}) outfile << val << " ";
                                    outfile << std::endl;
                                }
                                // end loop

                            }//} // x and y beam offset loop
                        }
                        }
                    }
                }
            }
        std::cout << "\n\n";
        }
        // Close output files
    outfile.close();
    }
    return 0;
}
```

Listing 8: QPD gap/size study

```cpp
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <cmath>
#include <limits>
#include <filesystem>
#include <regex>
#include "signals/pd_signals.h"
#include "beams/ga_gbeam.h"
#include "tracing/beam_tracing.h"

static inline void set_beam_pose(GA_GBeam& b,double x_mm,double y_mm,double alpha_urad,double beta_urad,double L){
    const double ax=alpha_urad*1e-6,by=beta_urad*1e-6;
    Eigen::Vector3d dir(-by,ax,-1.0);dir.normalize();
    Eigen::Vector3d p=-L*dir;
    b.set_origin_xyz(0.001*x_mm+p.x(),0.001*y_mm+p.y(),p.z());
    b.set_direction_xyz(dir.x(),dir.y(),dir.z());
}
static inline double step_for_xy(double dps){
    double a=std::abs(dps);
    if(a>2e-2) return dps>0?-10.0:10.0;
    if(a>3e-3) return dps>0?-1.0:1.0;
    if(a>5e-4) return dps>0?-0.1:0.1;
    if(a>4e-5) return dps>0?-0.01:0.01;
    if(a>5e-6) return dps>0?-0.001:0.001;
    if(a>4e-7) return dps>0?-0.0001:0.0001;
    return 0.0;
}
static inline double angle_step_from_dwsh(double a){
    if(a>1e-2) return 0.0523;     // 3
    if(a>3e-3) return 0.01745;    // 1
    if(a>1e-3) return 0.001;      // 0.001 rad
    if(a>1e-4) return 0.0001;     // 1e-4 rad
    return 0.0;
}
static inline Eigen::Matrix3d R_from_tip_tilt_roll(double tip,double tilt,double roll){
    Eigen::Matrix3d Rx;Rx<<1,0,0, 0,std::cos(tip),-std::sin(tip), 0,std::sin(tip),std::cos(tip);
    Eigen::Matrix3d Ry;Ry<<std::cos(tilt),0,std::sin(tilt), 0,1,0, -std::sin(tilt),0,std::cos(tilt);
    Eigen::Matrix3d Rz;Rz<<std::cos(roll),-std::sin(roll),0, std::sin(roll),std::cos(roll),0, 0,0,1;
    return Rz*Ry*Rx; // tip -> tilt -> roll
}

int main(){
    // preprocess responsivity once (Clocking.py writes out what pd_real.reads)
    // std::string input_file="/Users/martijn/Documents/Project_Nikhef/data/lab_data/pickle_RF03MHz_1.pkl";

// "/Users/martijn/Documents/Project_Nikhef/Data_import_to_macbook/20250918_093747_Run1_VIGO17_NS089008_QPD_1000_40_AL_004_CC_AmplitudeUniformity_RF30MHz.pkl",
// "/Users/martijn/Documents/Project_Nikhef/Data_import_to_macbook/20250925_164327_Run1_VIGO17_NS089008_QPD_750_40_AP_003_CC_AmplitudeUniformity_RF30MHz.pkl",
// "/Users/martijn/Documents/Project_Nikhef/Data_import_to_macbook/20250909_163050_Run1_VIGO17_NS089008_QPD_0500_40_AL_005_BL_AmplitudeUniformity_RF30MHz.pkl",

// "/Users/martijn/Documents/Project_Nikhef/Data_import_to_macbook/20250923_111452_Run1_VIGO17_NS089008_QPD_1000_20_AS_004_CC_AmplitudeUniformity_RF30MHz.pkl",
// "/Users/martijn/Documents/Project_Nikhef/data/lab_data/data/RF30MHz/20240801_104059_Run1_VIGO17_NS089008_QPD_0750_20_AS_011_CC_AmplitudeUniformity_RF30MHz.pkl",
// "/Users/martijn/Documents/Project_Nikhef/Data_import_to_macbook/20250904_184835_Run1_VIGO17_NS089008_QPD_0500_20_AK_005_CC_AmplitudeUniformity_RF30MHz.pkl"]
    std::string input_file="/Users/martijn/Documents/Project_Nikhef/Data_import_to_macbook/20250904_184835_Run1_VIGO17_NS089008_QPD_0500_20_AK_005_CC_AmplitudeUniformity_RF30MHz.pkl";
    std::string suffix = "DC";
    std::string command="python Clocking_2.py "+input_file+" "+suffix;
    system(command.c_str());


    // declare in outer scope
    std::string name_str; int qpd_size=0, gap_um=0;
    // parse once
    {
        std::regex re(R"((QPD_(\d{3,4})_(\d{2})_[A-Z]{2}_\d{3}))");
        std::smatch m;
        if(std::regex_search(input_file,m,re)){
            name_str=m[1].str();
            qpd_size=std::stoi(m[2].str());
            gap_um=std::stoi(m[3].str());
        }else{
            throw std::runtime_error("Filename didn't match: "+input_file);
        }
    }


    // beams
    GA_GBeam b1,b2; const double L=5.0;
    for(auto* b:{&b1,&b2}){b->set_power(0.5);b->set_circular_waist_rad(0.5,0);b->set_direction_xyz(0,0,-1);}
    std::string angle_file="angle.txt";
    float angle=0.f;{std::ifstream angle_in(angle_file);if(angle_in) angle_in>>angle;}
    std::cout<<"Angle read from file: "<<angle<<"\n";
    angle=-angle;
    if (input_file == "/Users/martijn/Documents/Project_Nikhef/Data_import_to_macbook/20250918_093747_Run1_VIGO17_NS089008_QPD_1000_40_AL_004_CC_AmplitudeUniformity_RF30MHz.pkl" || input_file == "/Users/martijn/Documents/Project_Nikhef/Data_import_to_macbook/20250925_164327_Run1_VIGO17_NS089008_QPD_750_40_AP_003_CC_AmplitudeUniformity_RF30MHz.pkl" ||
     input_file == "/Users/martijn/Documents/Project_Nikhef/Data_import_to_macbook/20250909_163050_Run1_VIGO17_NS089008_QPD_0500_40_AL_005_BL_AmplitudeUniformity_RF30MHz.pkl" || input_file == "/Users/martijn/Documents/Project_Nikhef/Data_import_to_macbook/20250904_184835_Run1_VIGO17_NS089008_QPD_0500_20_AK_005_CC_AmplitudeUniformity_RF30MHz.pkl")
    {angle = -0.7;}
    if (input_file == "/Users/martijn/Documents/Project_Nikhef/data/lab_data/data/RF30MHz/20240801_104059_Run1_VIGO17_NS089008_QPD_0750_20_AS_011_CC_AmplitudeUniformity_RF30MHz.pkl")
    {angle = 1.6536353;}
    if (input_file == "/Users/martijn/Documents/Project_Nikhef/Data_import_to_macbook/20250923
```

```
              _111452_Run1_VIGO17_NS089008_QPD_1000_20_AS_004_CC_AmplitudeUniformity_RF30MHz.pkl")
        {angle = -1.6;}
        std::cout<<"Angle now: "<<angle<<"\n";


        float nvvy=std::cos(angle),nvvx=std::sin(angle);

        // QPD: start at origin, nominal orientation; DO NOT MOVE CENTER AT ANY POINT
        Photodiode pd_real;
        pd_real.set_nv(Direction_3d(0,0,-1),Direction_3d(nvvx,nvvy,0));
        pd_real.set_center_xyz(0,0,0);
        pd_real.set_active_area_half_width(0.001*qpd_size);
        pd_real.set_slit_width(0.001*gap_um);
        pd_real.set_responsivity_on();(void)pd_real.get_data_resp();

        PD_signals<GA_GBeam,GA_GBeam> sig;

        // // ===== STEP 1: find center (x0,y0) by moving beams, QPD stays fixed =====
        std::cout<<"# center pass1: x y DPSh DPSv A B C D\n";

        double x0=-0.0,y0=-0.0; //
              <<<<<<<<<<<<<<<<<<<<<<<<
        for(int it=0;it<5000;++it){
            set_beam_pose(b1,x0,y0,0,0,L);
            set_beam_pose(b2,x0,y0,0,0,L);
            sig.set_parameters(b2,b1,pd_real);
            double h=sig.DPS_horizontal(),v=sig.DPS_vertical();
            double A=sig.DC_power_A(),B=sig.DC_power_B(),C=sig.DC_power_C(),D=sig.DC_power_D();
            std::cout<<x0<<" "<<y0<<" "<<h<<" "<<v<<" "<<A<<" "<<B<<" "<<C<<" "<<D<<"\n";
            double dx=step_for_xy(h),dy=step_for_xy(v);
            if(dx==0.0 && dy==0.0) break;
            x0+=dx;y0+=dy;
        }
        std::cout<<"center1_mm "<<x0<<" "<<y0<<"\n";



        // ===== STEP 2: find roll (first zero DWSh, then minimize |DWSh|+|DWSv|) =====
    {
        const double PI=3.14159265358979323846;
        auto norm_angle=[PI](double r){while(r<=-PI)r+=2*PI;while(r>PI)r-=2*PI;return r;};

        double roll=0.0;
        Point_3d pivot(0.001*x0,0.001*y0,0.0);

        auto apply_roll=[&](double target){
            double d=norm_angle(target-roll);
            if(std::abs(d)<1e-16) return;
            Eigen::Matrix3d R=R_from_tip_tilt_roll(0,0,d);
            pd_real.rotate_3d(R,pivot);
            roll=norm_angle(roll+d);
        };
        auto eval_DWSh=[&]()->double{
            set_beam_pose(b1,x0,y0,0,0,L);
            set_beam_pose(b2,x0,y0,100,0,L);
            sig.set_parameters(b2,b1,pd_real);
            std::cout  << (angle+roll) << " " << sig.DWS_horizontal_AP() << " " << sig.DWS_vertical_AP() << "\n";
            return sig.DWS_horizontal_AP();
        };
        auto eval_sum_abs=[&]()->double{
            set_beam_pose(b1,x0,y0,0,0,L);
            set_beam_pose(b2,x0,y0,100,0,L);sig.set_parameters(b2,b1,pd_real);double h=sig.DWS_horizontal_AP();
            set_beam_pose(b2,x0,y0,0,100,L);sig.set_parameters(b2,b1,pd_real);double v=sig.DWS_vertical_AP();
            double sum = std::abs(h)+std::abs(v);
            std::cout << (angle+roll) << " " << h << " " << v << " " << sum << "\n";
            return sum;
        };
        auto DWSh_at=[&](double r){double prev=roll;apply_roll(r);double v=eval_DWSh();apply_roll(prev);return v;};
        auto sum_abs_at=[&](double r){double prev=roll;apply_roll(r);double v=eval_sum_abs();apply_roll(prev);return v;};

        // --- Stage 1: zero DWSh ---
        const double tol_f=1e-6, tol_r=5e-7; const int max_it_root=80;
        double guess=roll, span=0.2;
        double a=norm_angle(guess-span), b=norm_angle(guess+span);
        double fa=DWSh_at(a), fb=DWSh_at(b);

        for(int k=0;k<8 && fa*fb>0;++k){
            span=std::min(span*2.0,PI);
            a=norm_angle(guess-span); b=norm_angle(guess+span);
            fa=DWSh_at(a); fb=DWSh_at(b);
            if(span>=PI) break;
        }
        if(fa*fb>0){
            int N=181; double best_r=guess,best_v=std::numeric_limits<double>::infinity();
            for(int i=0;i<N;++i){double r=-PI+(2*PI*i)/(N-1); double v=std::abs(DWSh_at(r)); if(v<best_v){best_v=v;best_r=r;}}
            guess=best_r; span=0.2;
            a=norm_angle(guess-span); b=norm_angle(guess+span);
            fa=DWSh_at(a); fb=DWSh_at(b);
        }
        double r1=guess;
        if(fa*fb<=0){
            double L=a,R=b; int it=0;
            while(it++<max_it_root && std::abs(R-L)>tol_r){
                double M=norm_angle(0.5*(L+R)); double fM=DWSh_at(M);
                if(std::abs(fM)<=tol_f){r1=M;break;}
                if(fa*fM<=0){R=M; fb=fM;} else {L=M; fa=fM;}
                r1=norm_angle(0.5*(L+R));
            }
        }else{
            double L=norm_angle(guess-span), R=norm_angle(guess+span);
            if(L>R) L-=2*PI;
            int it=0,max_it=100;
```

```
        while(it++<max_it && (R-L)>tol_r*4){
            double l=L+(R-L)/3.0, r=R-(R-L)/3.0;
            double fl=std::abs(DWSh_at(norm_angle(l)));
            double fr=std::abs(DWSh_at(norm_angle(r)));
            if(fl<fr) R=r; else L=l;
        }
        r1=norm_angle(0.5*(L+R));
    }
    apply_roll(r1);
    std::cout<<"zeroed_DWSh_at="<<(angle+roll)<<" rad\n";

    // --- Stage 2: minimize g(r)=|DWSh|+|DWSv| ---
    const int max_it_g=120; const double tol_r2=5e-7; const double phi=(1.0+std::sqrt(5.0))*0.5;
    auto golden_min=[&](double center,double halfspan){
        double L=center-halfspan, R=center+halfspan; if(L>R) std::swap(L,R);
        double c=R-(R-L)/phi, d=L+(R-L)/phi;
        double fc=sum_abs_at(norm_angle(c)), fd=sum_abs_at(norm_angle(d));
        for(int it=0; it<max_it_g && (R-L)>tol_r2; ++it){
            if(fc<fd){R=d; d=c; fd=fc; c=R-(R-L)/phi; fc=sum_abs_at(norm_angle(c));}
            else{L=c; c=d; fc=fd; d=L+(R-L)/phi; fd=sum_abs_at(norm_angle(d));}
        }
        return norm_angle(0.5*(L+R));
    };

    double base=sum_abs_at(roll);
    double s=0.2;
    double left=sum_abs_at(norm_angle(roll-s)), right=sum_abs_at(norm_angle(roll+s));
    if(left<base || right<base) s=std::min(0.35, s*1.5);
    double r2=golden_min(roll,s);
    apply_roll(r2);
    std::cout<<"roll_final="<<angle+roll<<" rad\n";
}



    // ===== STEP 3: sanity center check (QPD still fixed, same pivot) =====
    std::cout<<"# center pass2: x y DPSh DPSv A B C D\n";

    double x1=x0,y1=y0;
    for(int it=0;it<5000;++it){
        set_beam_pose(b1,x1,y1,0,0,L);
        set_beam_pose(b2,x1,y1,0,0,L);
        sig.set_parameters(b2,b1,pd_real);
        double h=sig.DPS_horizontal(),v=sig.DPS_vertical();
        double A=sig.DC_power_A(),B=sig.DC_power_B(),C=sig.DC_power_C(),D=sig.DC_power_D();
        std::cout<<x1<<" "<<y1<<" "<<h<<" "<<v<<" "<<A<<" "<<B<<" "<<C<<" "<<D<<"\n";
        double dx=step_for_xy(h),dy=step_for_xy(v);
        if(dx==0.0 && dy==0.0) break;
        x1+=dx;y1+=dy;
    }
    std::cout<<"center2_mm "<<x1<<" "<<y1<<"\n";

    // ===== STEP 4: data acquisition (QPD fixed; b1 fixed at center2; scan b2) =====
    // build scan lists
    std::vector<std::pair<int,int>> xy_list;
    for(int xi=-100;xi<=100;xi+=20) xy_list.emplace_back(xi,0);
    for(int yi=-100;yi<=100;yi+=20) if(yi!=0) xy_list.emplace_back(0,yi);
    std::vector<std::pair<int,int>> ab_list;
    for(int a=-100;a<=100;a+=50) ab_list.emplace_back(a,0);
    for(int b=-100;b<=100;b+=50) if(b!=0) ab_list.emplace_back(0,b);

    // real + ideal in one file
    std::string out="data_"; out+=name_str;out+="_"+suffix; out+="_file_2.txt";
    std::ofstream grid(out);
    grid<<"# mode size gap x_mm y_mm alpha_urad beta_urad DPSh DPSv contrast4q DWS_h DWS_v LPS\n";
    std::cout<<"# data_grid: mode size gap x y a b DPSh DPSv contrast4q DWS_h DWS_v LPS\n";



    // common signal object
    auto run_for_pd=[&](Photodiode& pd,const char* mode,double cx,double cy){
        for(const auto& [xrel,yrel]:xy_list){
            for(const auto& [alpha,beta]:ab_list){
                set_beam_pose(b2,cx+xrel,cy+yrel,alpha,beta,L);
                sig.set_integration_absolute_error(1e-5);
                sig.set_integration_relative_error(1e-2);
                sig.set_integration_workspace_size(1'000'000); // ~40  50   MB RAM; safe
                sig.set_parameters(b2,b1,pd);
                double DPSh=sig.DPS_horizontal();
                double DPSv=sig.DPS_vertical();
                double C4Q=sig.contrast_4q();
                double DWSh=sig.DWS_horizontal_AP();
                double DWSv=sig.DWS_vertical_AP();
                double LPS =sig.longitudinal_4q_AP();
                grid<<mode<<" "<<qpd_size<<" "<<gap_um<<" "<<xrel<<" "<<yrel<<" "<<alpha<<" "<<beta<<" "
                    <<DPSh<<" "<<DPSv<<" "<<C4Q<<" "<<DWSh<<" "<<DWSv<<" "<<LPS<<"\n";
                std::cout<<mode<<" "<<qpd_size<<" "<<gap_um<<" "<<xrel<<" "<<yrel<<" "<<alpha<<" "<<beta<<" "
                         <<DPSh<<" "<<DPSv<<" "<<C4Q<<" "<<DWSh<<" "<<DWSv<<" "<<LPS<<"\n";
            }
        }
    };


    xy_list.clear(); for(int x=-100;x<=100;x+=50) for(int y=-100;y<=100;y+=50) xy_list.emplace_back(x,y);
    ab_list.clear(); for(int a=-100;a<=100;a+=50) for(int b=-100;b<=100;b+=50) ab_list.emplace_back(a,b);

    // // real PD: b1 fixed at (x1,y1), scan b2 around (x1,y1)
    set_beam_pose(b1,x1,y1,0,0,L);
    run_for_pd(pd_real,"real",x1,y1);

    // // ideal PD: b1 fixed at (0,0), scan b2 around (0,0)

    // ideal PD (center at origin, nominal orientation)
```

```
    Photodiode pd_ideal;
    pd_ideal.set_nv(Direction_3d(0,0,1),Direction_3d(0,1,0));
    pd_ideal.set_center_xyz(0,0,0);
    pd_ideal.set_active_area_half_width(0.001*qpd_size);
    pd_ideal.set_slit_width(0.001*gap_um);
    set_beam_pose(b1,0,0,0,0,L);
    run_for_pd(pd_ideal,"ideal",0,0);

    grid.close();

    return 0;
}
```

Listing 9: Tophat Beam creationg/loading

```
#include "beams/ga_gbeam.h"
#include <Eigen/Dense>
#include <Eigen/SVD>
#include "signals/field_decomposer.h"
#include "beams/sa_hgbeam.h"
#include "utility/functions.h"
#include "beams/gbdc_beam.h"
#include "beams/spectrum_propagator.h"
#include <iomanip>
#include <cstdlib>
#include <fstream>
#include <iostream>
#include <typeinfo>
#include <functional>
#include <complex>
#include <cmath>
#include <ctime>
//#include <bits/stdc++.h>
#include <sys/stat.h>
#include <sys/types.h>
#include "beam_monitor.h"
#include <filesystem>



struct wavefront
{
        std::string name;
        Direction_3d direction;
        Point_3d origin;
        std::function<std::complex<double>(Point_3d)> front;

};
void construct_orthonormal_frame(const Direction_3d& direction,
                                 Direction_3d& vertical,
                                 Direction_3d& horizontal)
{
    Eigen::Vector3d z_axis(direction.x(), direction.y(), direction.z());
    Eigen::Vector3d ref = (std::abs(z_axis.z()) < 0.99)
                          ? Eigen::Vector3d(0, 0, 1)
                          : Eigen::Vector3d(0, 1, 0);

    Eigen::Vector3d h = (ref.cross(z_axis)).normalized();
    Eigen::Vector3d v = (z_axis.cross(h)).normalized();

    horizontal = Direction_3d(h(0), h(1), h(2));
    vertical   = Direction_3d(v(0), v(1), v(2));
}

std::function<std::complex<double>(Point_3d)> truncated_Gaussian_1(bool truncate) {
    double ra = 0.5;
    double target_power = 0.5;
    double distance = 5 * MM;
    double z0 = 0.;

    GA_GBeam beam;
    beam.set_origin_xyz(0, 0, distance);
    beam.set_direction_xyz(0, 0, -1);
    beam.set_circular_waist_rad(ra, z0);

    // Sampling parameters
    double dx = 0.01; // mm
    double range = 1.5 * ra;
    int N = static_cast<int>(2 * range / dx); // total samples per axis
    double power = 0.0;

    // Estimate total power of the artificial field
    for (int i = -N / 2; i < N / 2; ++i) {
        for (int j = -N / 2; j < N / 2; ++j) {
            double x = i * dx;
            double y = j * dx;
            if (truncate && (x * x + y * y) > ra * ra) continue;
            double amplitude, phase;
            beam.electric_field_amplitude_phase(x, y, amplitude, phase);
            amplitude = std::min(amplitude, 0.5);
            power += amplitude * amplitude * dx * dx;
        }
    }

    double scale = std::sqrt(target_power / power);

    // Return the scaled and truncated field as a lambda
    return [=](Point_3d point) -> std::complex<double> {
        double x = point.x(), y = point.y();
        double amplitude, phase;
        beam.electric_field_amplitude_phase(x, y, amplitude, phase);
```

```
        if (truncate) {
            if ((x * x + y * y) > ra * ra) return 0.0;
            amplitude = std::min(amplitude, 0.5);
            amplitude *= scale * 1.0735;//1.1480078276; // hardcoded factor to normalize the final GBDC power to 0.5
        } else {
            amplitude = std::min(amplitude, 0.5);
            amplitude *= scale * 1.055;//1.1109220075; // hardcoded factor to normalize the final GBDC power to 0.5
        }

        return amplitude * std::exp(std::complex<double>(0, phase));
    };
}


int main(int argc, char* argv[]) {
    if (argc < 4) {
        std::cerr << "Usage: ./program [GBDC|MEM|pre|clip] [create|load]" << std::endl;
        return 1;}

    std::string beam_type = argv[1];  // GBDC, MEM, pre, clip
    std::string mode = argv[2];       // create or load
    int N_modes = std::stoi(argv[3]);    // Number of modes as integer
    std::string suffix = "_" + std::to_string(N_modes);

    int task_number = -1;
    if (mode == "create") {
        if (beam_type == "GBDC") task_number = 1;
        else if (beam_type == "MEM") task_number = 2;
        else if (beam_type == "pre") task_number = 3;
        else if (beam_type == "clip") task_number = 4;
    } else if (mode == "load") {task_number = 5;}
    if (task_number == -1) {
        std::cerr << "Invalid input: must specify one of [GBDC|MEM|pre|clip] and [create|load]" << std::endl;
        return 1;}

    bool truncate = (argc > 4 && std::string(argv[4]) == "trunc");

    // === Universal constants and settings ===
    double ra = 0.5; // radius for clipping
    double clipping_threshold = 0.5;  // Max amplitude allowed when sampling the original beam
    double target_power = 0.5; // Target power for normalization
    const int N_grid = 400;
    const double grid_range = 1.2;
    const double dx = 2.0 * grid_range / (N_grid - 1);
    const double distance = 5 * MM; // = 5 * MM = 5
    const double lambda = 1064e-6;
    int radial_points = 3000; //4001;

    // === Beam parameters ===
    float i_beam_rot = .20;
    double x_beam_off =100.0;
    double y_beam_off = 0;
    double L = distance;
    double x0 = L * std::sin( i_beam_rot);
    double y0 = 0;
    double z0 = L * std::cos( i_beam_rot);
    Point_3d origin(0.001 * x_beam_off + x0, 0.001 * y_beam_off + y0, z0);
    Eigen::Vector3d d = -Eigen::Vector3d(x0, y0, z0).normalized();
    Direction_3d dir(d(0), d(1), d(2));
    Direction_3d nvv, nvh;
    construct_orthonormal_frame(dir, nvv, nvh);

    std::string output_directory = "/data/detrd/mmoorlag/sim_data_2.0/beams/" + std::filesystem::path(__FILE__).stem().string() +
          "/";
    std::filesystem::create_directories(output_directory);

    std::vector<double> coords(N_grid);
    for (int i = 0; i < N_grid; ++i) coords[i] = -grid_range + i * dx;

        switch (task_number)
        {
        case 1://GBDC
                {
            int max_order_GBD = N_modes; //100, 120; 2000
            double window_size = 6.0;//window size should be a little larger than waist0
            double waist_scaling_faktor = 1.3;

                        wavefront start1;
                        start1.name = "Clipped Beam_w0=5mm";
                        start1.direction = Direction_3d(0, 0, -1);
                        start1.origin = Point_3d(0, 0, distance);
                        start1.front = truncated_Gaussian_1(truncate);

                        GBDC_Beam GBDCbeam(window_size, waist_scaling_faktor, max_order_GBD);//gbdc_beam.cpp line35
                        GBDCbeam.compute_gbdc(dir,nvv,nvh, start1.origin, start1.front);//gbdc_beam.cpp line102
            GBDCbeam.propagate(distance);

            std::cout << "GBDC beam origin: " << GBDCbeam.get_origin() << std::endl;
            std::cout << "GBDC beam direction: " << GBDCbeam.get_direction() << std::endl;
            GBDCbeam.set_origin(Point_3d(0.5, 0.5, 0.5));
            GBDCbeam.set_direction(Direction_3d(0, -1, 0), Direction_3d(0, 0, 1));
            std::cout << "After reset:" << std::endl;
            std::cout << "GBDC beam origin: " << GBDCbeam.get_origin() << std::endl;
            std::cout << "GBDC beam direction: " << GBDCbeam.get_direction() << std::endl;

                //---------------create file and save in it--------------------------------
            std::ofstream fout(output_directory + "GBDC" + suffix + "_" + std::string(argv[4]) + "_propagated.txt");
            fout << std::scientific << std::setprecision(12);
            for (double x : coords)
                for (double y : coords) {
                    auto E = GBDCbeam.electric_field_oblique(x, y, 0);
                    fout << x << " " << y << " " << std::abs(E) << " " << std::arg(E) << "\n";
                }
```

```
            fout.close();
        break;
            }

        case 2://MEM
            {
            int max_order_MEM = N_modes; // 120; 50; 20
                        Aperture ap_lens = Aperture();
                        ap_lens.set_substrate_radius(1e100); //??

                        ap_lens.set_hole_radius(ra);
                        ap_lens.set_center(Point_3d(0., 0., 0.));
                        ap_lens.set_nv(Direction_3d(0, 0, -1), Direction_3d(0, 1, 0));

                        GA_GBeam gaussian_beam;

            double w0 = 10. * ra;
                        double z0 = 0.;
                        double phase_off = 0;
                        gaussian_beam.set_origin_xyz(0, 0, distance);
            gaussian_beam.set_direction_xyz(0, 0, -1);
                        gaussian_beam.set_circular_waist_rad(w0, z0);
                        gaussian_beam.set_phase_offset_for_electric_field(M_PI + phase_off);

            double dx = 0.01; // mm
            double r_max = ap_lens.hole_radius();
            int N = static_cast<int>(2 * r_max / dx);
            double power_through_aperture = 0.0;

            // Temporarily set beam power to 1 to simplify scaling
            double power = 1.0;
            gaussian_beam.set_power(power);

            for (int i = -N/2; i <= N/2; ++i) {
                for (int j = -N/2; j <= N/2; ++j) {
                    double x = i * dx;
                    double y = j * dx;
                    if ((x*x + y*y) > r_max*r_max) continue;
                    double amp, phase;
                    gaussian_beam.electric_field_amplitude_phase(x, y, amp, phase);
                    power_through_aperture += amp * amp * dx * dx;
                }
            }
            // Now compute the actual required beam power before clipping
            double target_power = 0.5;
            double scale = target_power / power_through_aperture;

            // Set the beam's power to this scaled value
            gaussian_beam.set_power(scale * power);

                        SA_HGBeam MEMbeam;
                        Field_Decomposer<GA_GBeam, SA_HGBeam> fd =
                                        Field_Decomposer<GA_GBeam, SA_HGBeam>(gaussian_beam, ap_lens, max_order_MEM);
                        fd.set_absolute_accuracy(1e-9); // what unit? ; 1e-9; 1e-6
                        fd.set_relative_accuracy(1e-9);

                        MEMbeam = fd.decompose_up_to_order(true, true);
                        // MEMbeam.normalise_coefficients(); // comment out for recent use

            MEMbeam.save_coefficient_matrix(output_directory + "MEM" + suffix + "_" + std::string(argv[4]) + "_coefficients.txt");

            std::ofstream beam_meta_mem(output_directory + "MEM" + suffix + "_" + std::string(argv[4]) + "_metadata.txt");
            beam_meta_mem << std::scientific << std::setprecision(15)
                        << MEMbeam.qt().real() << " " << MEMbeam.qt().imag() << "\n"
                        << MEMbeam.qs().real() << " " << MEMbeam.qs().imag() << "\n"
                        << MEMbeam.origin().x() << " " << MEMbeam.origin().y() << " " << MEMbeam.origin().z() << "\n"
                        << MEMbeam.direction().x() << " " << MEMbeam.direction().y() << " " << MEMbeam.direction().z() << "\n"
                        << MEMbeam.waist_rad_t() << " " << MEMbeam.waist_rad_s() << "\n";
            beam_meta_mem.close();

                        MEMbeam.propagate(distance);

            //---------------create file and save in it--------------------------------
            std::ofstream fout2(output_directory + "MEM" + suffix + "_" + std::string(argv[4]) + "_propagated.txt");
            fout2 << std::scientific << std::setprecision(12);
            for (double x : coords) {
                for (double y : coords) {
                    std::complex<double> E = MEMbeam.electric_field_oblique(x, y, 0);
                    fout2 << x << " " << y << " " << std::abs(E) << " " << std::arg(E) << "\n";
                }
            }
            fout2.close();

                break;
                }
    case 3: // pre-installed ifocad tophat - Generate and evaluate 2D flat-top beam, 4-column output
        {

            // === Beam + propagation settings ===
            double power = 0.5;
            unsigned int max_order = N_modes; // 140?

            SA_HGBeam beam;
            beam.make_tophat(ra, power, max_order);
            beam.set_origin_xyz(0, 0, distance);
            beam.set_direction_xyz(0, 0, -1);
            beam.save_coefficient_matrix(output_directory + "pre" + suffix + "_" + std::string(argv[4]) + "_coefficients.txt");

            std::ofstream beam_meta_th(output_directory + "pre" + suffix + "_" + std::string(argv[4]) + "_metadata.txt");
            beam_meta_th << std::scientific << std::setprecision(15)
                        << beam.qt().real() << " " << beam.qt().imag() << "\n"
                        << beam.qs().real() << " " << beam.qs().imag() << "\n"
                        << beam.origin().x() << " " << beam.origin().y() << " " << beam.origin().z() << "\n"
```

```
                    << beam.waist_rad_t() << " " << beam.waist_rad_s() << "\n";
            beam_meta_th.close();

            beam.propagate(distance);

            // === Grid setup ===
            int N = 201;
            double extent = 2.5 * ra;
            double step = 2.0 * extent / (N - 1);

            std::vector<double> x_vals(N), y_vals(N);
            for (int i = 0; i < N; ++i) {
                x_vals[i] = -extent + i * step;
                y_vals[i] = -extent + i * step;
            }
            // === Unified output: x, y, amplitude, phase ===
            std::ofstream f_out(output_directory + "pre" + suffix + "_" + std::string(argv[4]) + "_propagated.txt");
            for (int i = 0; i < N; ++i) {
                for (int j = 0; j < N; ++j) {
                    double x = x_vals[i];
                    double y = y_vals[j];
                    std::complex<double> E = beam.electric_field(x, y);
                    std::complex<double> E_prop = E * std::exp(std::complex<double>(0, -2.0 * M_PI / lambda * distance));
                    f_out << x << "\t" << y << "\t" << std::abs(E_prop) << "\t" << std::arg(E_prop) << "\n";
                }
                f_out << "\n"; // optional: blank line between rows
                }
                f_out.close();
        break;
        }

    case 4: //Clipped Gaussian/ NMF
        {
            const double clipping_threshold = 0.5;  // Max amplitude allowed when sampling the original beam
            const double mode_clipping_threshold = 1e-6;  // Threshold below which mode coefficients are set to zero
            SA_HGBeam original_beam;  // Create the original Hermite-Gaussian beam object
            original_beam.set_circular_waist_rad(ra, 0);  // Set beam waist radius (1 mm), second argument is the axis (0 =
                horizontal)
            original_beam.set_power(1.0);  // Normalize beam power to 1

            // Create a vector of coordinates from -grid_range to +grid_range
            std::vector<double> coords(N_grid);
            for (int i = 0; i < N_grid; ++i)
                coords[i] = -grid_range + i * dx;  // Fill with evenly spaced coordinates

            // Create a 2D grid to store electric field values (complex-valued)
            Eigen::ArrayXXcd E_grid(N_grid, N_grid);

            // Fill grid with sampled and clipped electric field values from the original beam
            for (int i = 0; i < N_grid; ++i) {
                for (int j = 0; j < N_grid; ++j) {
                    double x = coords[i];  // Horizontal coordinate
                    double y = coords[j];  // Vertical coordinate
                    std::complex<double> E = original_beam.electric_field_oblique(x, y, 0);  // Get field at (x, y, z=0)
                    double r = std::sqrt(x*x + y*y);
                    double amp = std::abs(E);
                    if (truncate) {
                        if ((x*x + y*y) > ra*ra) amp = 0.0;
                        else amp = std::min(amp, clipping_threshold);
                    } else {
                        amp = std::min(amp, clipping_threshold);
                    }
                    double phase = std::arg(E);  // Extract the phase
                    E_grid(j, i) = amp * std::exp(std::complex<double>(0, phase));  // Reconstruct clipped complex field (row=
                        vertical=y, col=horizontal=x)
                }
            }
            // Compute total power of truncated/clipped field
            double total_power = 0.0;
            for (int i = 0; i < N_grid; ++i)
                for (int j = 0; j < N_grid; ++j)
                    total_power += std::norm(E_grid(j, i)); // |E|^2
            total_power *= dx * dx;

            // Normalize to target power
            double target_power = 0.5;
            double scale = std::sqrt(target_power / total_power);
            E_grid *= scale; // Rescale field amplitudes

            // Construct a BeamMonitor from coordinate vectors and the filled electric field grid
            BeamMonitor monitor(coords, coords, E_grid);
            SA_HGBeam Tophat;  // New beam to hold reconstructed field
            int max_order = N_modes;  // (60?) Maximum Hermite-Gaussian mode order to use
            bool estimate_params = true;  // Let the reconstruction routine estimate beam parameters (waist, position)
            bool optimise_waist = true;  // Allow waist size to be optimized during reconstruction

            // Fit the reconstructed beam to the grid values using HG decomposition
            Tophat.mode_amplitudes_from_grid(monitor, max_order, estimate_params, optimise_waist);

            // Remove small mode contributions below the clipping threshold
            Tophat.clip_mode_coefficients(mode_clipping_threshold);
            Tophat.save_coefficient_matrix(output_directory + "clip" + suffix + "_" + std::string(argv[4]) + "_coefficients.txt");

            // Save beam geometry/optical parameters
            std::ofstream beam_meta(output_directory + "clip" + suffix + "_" + std::string(argv[4]) + "_metadata.txt");
            beam_meta << std::scientific << std::setprecision(15)
                      << Tophat.qt().real() << " " << Tophat.qt().imag() << "\n"
                      << Tophat.qs().real() << " " << Tophat.qs().imag() << "\n"
                      << Tophat.origin().x() << " " << Tophat.origin().y() << " " << Tophat.origin().z() << "\n"
                      << Tophat.direction().x() << " " << Tophat.direction().y() << " " << Tophat.direction().z() << "\n"
                      << Tophat.waist_rad_t() << " " << Tophat.waist_rad_s() << "\n";
            beam_meta.close();

```

```
            // propagate beam
            Tophat.propagate(distance);
            // Save reconstructed beam field to a text file for visualization (x y amplitude phase)
            std::ofstream fout(output_directory + "clip" + suffix + "_" + std::string(argv[4]) + "_propagated.txt");
            fout << std::scientific << std::setprecision(12);  // Set output format

            for (int i = 0; i < N_grid; ++i) {
                for (int j = 0; j < N_grid; ++j) {
                    double x = coords[i];  // Horizontal coordinate
                    double y = coords[j];  // Vertical coordinate
                    std::complex<double> E = Tophat.electric_field_oblique(x, y, 0);  // Get field from reconstructed beam
                    fout << x << " " << y << " " << std::abs(E) << " " << std::arg(E) << "\n";  // Save x, y, amplitude, phase
                }
            }
            fout.close();  // Done writing
        break;
        }
        case 5: // Load Beam
        {
            SA_HGBeam beam;
            beam.load_coefficient_matrix(output_directory + beam_type + suffix + "_" + std::string(argv[4]) + "_coefficients.txt")
                 ;
            beam.load_metadata(output_directory + beam_type + suffix + "_" + std::string(argv[4]) + "_metadata.txt");
            beam.propagate(distance);
            std::cout << "qt = " << beam.qt() << "\n"
                        << "qs = " << beam.qs() << "\n"
                        << "Power = " << beam.power() << "\n";
            std::ofstream fout(output_directory + beam_type + suffix + "_" + std::string(argv[4]) + "_afterload.txt");
            fout << std::scientific << std::setprecision(12);
            for (double x : coords)
                for (double y : coords)
                    fout << x << " " << y << " "
                            << std::abs(beam.electric_field_oblique(x, y, 0)) << " "
                            << std::arg(beam.electric_field_oblique(x, y, 0)) << "\n";
            fout.close();

        break;
        }
        }
}
```

Listing 10: Tophat beam QPD simulation

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <cmath>
#include <list>
#include <filesystem>



// Include your specific libraries as needed
#include "signals/pd_signals.h"
#include "beams/ga_gbeam.h"
#include "tracing/beam_tracing.h"

// Function to display the loading bar
void displayDualLoadingBars(int progress1, int total1, int progress2, int total2, int barWidth = 50) {
    float percent1 = static_cast<float>(progress1) / total1;
    float percent2 = static_cast<float>(progress2) / total2;

    int filled1 = static_cast<int>(barWidth * percent1);
    int filled2 = static_cast<int>(barWidth * percent2);

    std::cout << "\033[F\r"; // Move up 1 line
    std::cout << "[";
    for (int i = 0; i < barWidth; ++i) std::cout << (i < filled1 ? "=" : (i == filled1 ? ">" : " "));
    std::cout << "] " << int(percent1 * 100.0) << "%\n";

    std::cout << "\r[";
    for (int i = 0; i < barWidth; ++i) std::cout << (i < filled2 ? "=" : (i == filled2 ? ">" : " "));
    std::cout << "] " << int(percent2 * 100.0) << "%\r";
    std::cout.flush();
}


int main(int argc, char* argv[]) {
    if (argc < 3) {
        std::cerr << "Error: missing input file argument\n";
        return 1;
      }
    std::string file = argv[1];
    std::string mode = argv[2];

    std::cout << "Running with: " << file << " mode: " << mode << std::endl;


    // Iterate over each input file
    for (const auto& f : std::vector<std::string>{file}) {
        bool is_ideal = (file == "ideal");
        // Extract file prefix for naming output files
        std::string prefix = is_ideal ? "reference" : file.substr(0, file.find_last_of('.'));
        std::string output_directory = "/data/detrd/mmoorlag/sim_data_2.0/sim_data_" + std::filesystem::path(__FILE__).stem().
             string() + "/";
        std::filesystem::create_directories(output_directory);

        std::string outfile_name = (is_ideal ? "idealQPD_" : "realQPD_") + prefix + "_" + mode + ".txt";
        std::ofstream outfile(output_directory + outfile_name);
        if (!outfile.is_open()) {
```

```
58              std::cerr << "Failed to open output file: " << outfile_name << std::endl;
59              return 1;}
60          std::string header = "DC/RF tip(Rad) tilt(Rad) clocking(Rad) x-rot(muRad) y-rot(muRad) x-off(micron) y-off(micron) A(mA) B
                (mA) C(mA) D(mA) LPS(micron) DWS_h DWS_v ";
61          outfile << header << std::endl;
62
63          std::vector<std::string> signal_types = {mode};
64
65          // Run the Python script for the current file
66          for (const std::string& signal_type : signal_types) {
67              if (!is_ideal) {
68                  std::string full_path = "/data/detrd/mmoorlag/lab_data/QPD1/" + file;
69                  std::string python_path = "/user/mmoorlag/myenv/bin/python";
70                  std::string script_path = "/user/mmoorlag/opt/ifocad_cpp/projects/Martijnstbc/Clocking.py";
71
72              std::string command = python_path + " " + script_path + " " + full_path + " " + signal_type;
73              std::cout << "Executing: " << command << std::endl;
74
75                  int retcode = system(command.c_str());
76                  if (retcode != 0) {
77                      std::cerr << "ERROR: Python command exited with code " << retcode << std::endl;
78                  }
79                  }
80                  else {
81                      std::cout << "Ideal\n" << std::endl;
82                  }
83              // initialize parameters for the simulation
84              const double beam_rotation_step = 50;
85
86              const double clock_start = -0.05235;
87              const double clock_end = 0.05235;
88              const double clocking_step = 0.01745;
89
90              const double deg=0.01745329252;      // ~1 degrees in rad
91              const double span=deg*0.5;           // keep same total range as before
92              const double tip_step=deg*0.25;
93              const double tilt_step=deg*0.25;
94
95              // tip centered at 0
96              const double tip_start=-span;
97              const double tip_end=span;
98
99              // tilt centered at 2*deg
100             const double tilt_center=0*deg;
101             const double tilt_start=tilt_center-span;
102             const double tilt_end=tilt_center+span;
103
104             const double x_off_start = -30; // in microns
105             const double x_off_end = 30; // in microns
106             const double x_off_step = 15; // in microns
107
108             // Set up Gaussian beams
109             const double L = 5; // Distance beam origin to QPD
110             GA_GBeam b1;
111             GA_GBeam b2;
112             for (auto* b : {&b1, &b2}) {
113                 b->set_power(0.5);
114                 b->set_origin_xyz(0, 0, L);
115                 b->set_circular_waist_rad(0.5, 0);
116                 b->set_direction_xyz(0, 0, -1);
117             }
118
119             for (double beam_x_rot = 0; beam_x_rot <=1.1 ;beam_x_rot+=1) { // leave this one out when clocking> pi
120                 for (double tip = tip_start; tip <= tip_end+1e-8; tip += tip_step) { //rotation around x-axis
121                     for (double tilt = tilt_start; tilt <= tilt_end+1e-8; tilt += tilt_step) { //rotation around y-axis
122                         for (double clock = clock_start; clock <= clock_end+1e-8; clock += clocking_step) {
123                             // Setup loading bar
124                             int tip_index = round((tip - tip_start) / tip_step);
125                             int tilt_index = round((tilt - tilt_start) / tilt_step);
126                             int tip_total = round((tip_end - tip_start) / tip_step) + 1;
127                             int tilt_total = round((tilt_end - tilt_start) / tilt_step) + 1;
128
129                             int clock_index = round((clock - clock_start) / clocking_step);
130                             int clock_total = round((clock_end - clock_start) / clocking_step) + 1;
131
132                             displayDualLoadingBars(tip_index + 1, tip_total, clock_index + 1, clock_total);
133
134                             double roll  = clock;
135
136                             // Create rotation matrices
137                             Eigen::Matrix3d Rx;
138                                 Rx << 1, 0, 0,
139                                     0, cos(tip), -sin(tip),
140                                     0, sin(tip), cos(tip);
141                                 Eigen::Matrix3d Ry;
142                                 Ry << cos(tilt), 0, sin(tilt),
143                                     0, 1, 0,
144                                     -sin(tilt), 0, cos(tilt);
145                                 Eigen::Matrix3d Rz;
146                                 Rz << cos(roll), -sin(roll), 0,
147                                     sin(roll), cos(roll), 0,
148                                     0, 0, 1;
149
150                             Eigen::Matrix3d R = Rx * Ry * Rz; // intrinsic rotation order: roll, tilt, tip
151
152                             // Optional pivot point (e.g., center of your QPD)
153                             Point_3d pivot(0, 0, 0);  // or actual center if known
154                             ///////////
155                             double x_beam_off = 0;
156                             double y_beam_off = 0;
157                             for (x_beam_off = x_off_start; x_beam_off <= x_off_end + 1e-8; x_beam_off += x_off_step) {
158                             // for (y_beam_off = -40; y_beam_off <= 40.001; y_beam_off += 20) {
159                             for (double i_beam_rot = -100; i_beam_rot <= 100.001; i_beam_rot += beam_rotation_step) {
```

```
                                    // Round the values to desired decimal places
                                    double rounded_tip = std::round(tip * 10000000.0) / 10000000.0;
                                    double rounded_tilt = std::round(tilt * 10000000.0) / 10000000.0;
                                    double rounded_roll = std::round(roll * 10000000.0) / 10000000.0;

                                    if (beam_x_rot == 0) {
                                        outfile << (signal_type == "DC" ? 0 : 1) << " " << rounded_tip << " " << rounded_tilt << " "
                                              << rounded_roll << " " << i_beam_rot << " NaN "<< " " << x_beam_off << " " << y_beam_off
                                               << " ";
                                        double x0 = L * std::sin(1e-6 * i_beam_rot), z0 = L * std::cos(1e-6 * i_beam_rot);
                                        b2.set_origin_xyz(0.001 * x_beam_off + x0, 0.001 * y_beam_off, z0);
                                        Eigen::Vector3d dir = -Eigen::Vector3d(x0, 0, z0).normalized();
                                        b2.set_direction_xyz(dir(0), dir(1), dir(2));
                                    } else {
                                        outfile << (signal_type == "DC" ? 0 : 1) << " " << rounded_tip << " " << rounded_tilt << " "
                                              << rounded_roll << " " << "NaN " << i_beam_rot << " " << x_beam_off << " " << y_beam_off
                                               << " ";
                                        double y0 = L * std::sin(1e-6 * i_beam_rot), z0 = L * std::cos(1e-6 * i_beam_rot);
                                        b2.set_origin_xyz(0.001 * x_beam_off, 0.001 * y_beam_off + y0, z0);
                                        Eigen::Vector3d dir = -Eigen::Vector3d(0, y0, z0).normalized();
                                        b2.set_direction_xyz(dir(0), dir(1), dir(2));
                                    }

                                    Photodiode pd;
                                    pd.set_nv(Direction_3d(0, 0, 1), Direction_3d(0, 1, 0)); // Set normal vector and vertical vector
                                    pd.set_center_xyz(0, 0, 0.);
                                    pd.set_active_area_half_width(0.75);
                                    pd.set_slit_width(0.02);

                                    PD_signals<GA_GBeam, GA_GBeam> signal;
                                    // signal.set_parameters(b2, b1, pd);

                                    if (is_ideal) {
                                        // Ideal simulation
                                        pd.rotate_3d(R, pivot);  // Rotate photodiode in 3D
                                        signal.set_integration_method(GSL_SINGLE_DIMENSION_INTEGRATOR);
                                        signal.set_integration_max_subintervals(20000);
                                        signal.set_integration_workspace_size(600000); // DCUHRE/GSL max points
                                        signal.set_integration_absolute_error(1e-2);
                                        signal.set_integration_relative_error(1e-8);
                                        signal.set_parameters(b2, b1, pd);
                                        double A1 = signal.DC_power_A(), B1 = signal.DC_power_B();
                                        double C1 = signal.DC_power_C(), D1 = signal.DC_power_D();
                                        double E1 = signal.longitudinal_4q_AP();
                                        double F1 = signal.DWS_horizontal_AP(), G1 = signal.DWS_vertical_AP();
                                        for (double val : {A1, B1, C1, D1, E1, F1, G1}) outfile << val << " ";
                                        outfile << std::endl;
                                    }
                                    else {
                                        // Real simulation
                                        pd.set_responsivity_on();
                                        Eigen::VectorXd addParam=pd.get_data_resp();
                                        pd.rotate_3d(R, pivot);  // Rotate photodiode in 3D
                                        signal.set_integration_method(GSL_SINGLE_DIMENSION_INTEGRATOR);
                                        signal.set_integration_max_subintervals(20000);
                                        signal.set_integration_workspace_size(600000); // DCUHRE/GSL max points
                                        signal.set_integration_absolute_error(1e-2);
                                        signal.set_integration_relative_error(1e-8);
                                        signal.set_parameters(b2, b1, pd);

                                        double A2 = signal.DC_power_A(), B2 = signal.DC_power_B();
                                        double C2 = signal.DC_power_C(), D2 = signal.DC_power_D();
                                        double E2 = signal.longitudinal_4q_AP();
                                        double F2 = signal.DWS_horizontal_AP(), G2 = signal.DWS_vertical_AP();
                                        for (double val : {A2, B2, C2, D2, E2, F2, G2}) outfile << val << " ";
                                        outfile << std::endl;
                                    }
                                    // end loop

                                }//} // x and y beam offset loop
                            }
                            }
                        }
                    }
                }
            std::cout << "\n\n";
            }
            // Close output files
        outfile.close();
        }
        return 0;
    }
```