EINSTEIN TELESCOPE

INFN CAGLIARI

ET ITALY

Status of phase camera simulations

21/05/2025

STVDIORVM VNIVERSITAS CARALITANA

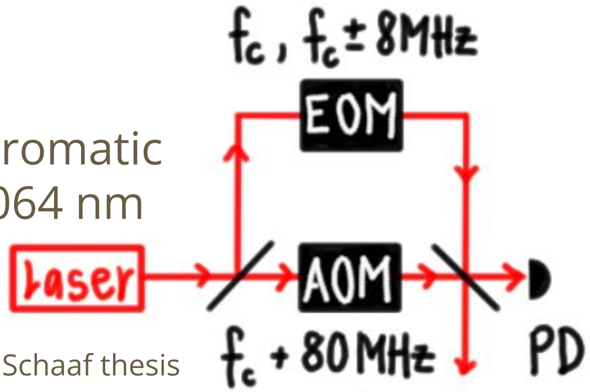
Quick recap

$$E_{\text{in}} = E'_{\text{in}} \cdot e^{i(2\pi f_c t + \beta\cos(2\pi f_{\text{sb}} t))}$$

β = 0.025

$$= E'_{\text{in}} \cdot e^{i2\pi f_c t} \cdot \left[J_0(\beta) + \sum_{n=1}^{\infty} i^n J_{\text{n}}(\beta) e^{in2\pi f_{\text{sb}} t} + i^n J_{\text{n}}(\beta) e^{-in2\pi f_{\text{sb}} t} \right]$$

Monochromatic laser 1064 nm



from van der Schaaf thesis

Phase camera detecting beatings

Laser frequency shifted by 80 MHz

Quick background

Carrier — Sidebands — Reference beam

Total intensity circulating in the system

$$
\begin{aligned}
I(x,y,t) &= |E_c(x,y,t) + E_{usb}(x,y,t) + E_{lsb}(x,y,t) + E_{ref}(x,y,t)|^2 \\
&= \text{DC-term} + \text{PDH-term} + 2f_{sb}\text{-term} + \text{PC-term}, \\
\text{DC-term} &= |E_c(x,y)|^2 + |E_{usb}(x,y)|^2 + |E_{lsb}(x,y)|^2 + |E_{ref}(x,y)|^2, \\
\text{PDH-term} &= E_c(x,y,t)E^*_{usb}(x,y,t) + E^*_c(x,y,t)E_{usb}(x,y,t) + \\
&\quad + E_c(x,y,t)E^*_{lsb}(x,y,t) + E^*_c(x,y,t)E_{lsb}(x,y,t) \\
&= 2|E_c(x,y)||E_{usb}(x,y)|\cos(2\pi f_{sb}t + (\phi_{usb} - \phi_c)) + \\
&\quad + 2|E_c(x,y)||E_{lsb}(x,y)|\cos(2\pi f_{sb}t + (\phi_c - \phi_{lsb})), \\
2f_{sb}\text{-term} &= E_{lsb}(x,y,t)E^*_{usb}(x,y,t) + E^*_{lsb}(x,y,t)E_{usb}(x,y,t) \\
&= 2|E_{lsb}(x,y)||E_{usb}(x,y)|\cos(4\pi f_{sb}t + (\phi_{usb} - \phi_{lsb})), \\
\text{PC-term} &= E_c(x,y,t)E^*_{ref}(x,y,t) + E^*_c(x,y,t)E_{ref}(x,y,t) + \\
&\quad + E_{usb}(x,y,t)E^*_{ref}(x,y,t) + E^*_{usb}(x,y,t)E_{ref}(x,y,t) + \\
&\quad + E_{lsb}(x,y,t)E^*_{ref}(x,y,t) + E^*_{lsb}(x,y,t)E_{ref}(x,y,t) \\
&= 2|E_c(x,y)||E_{ref}(x,y)|\cos(-2\pi f_h t + (\phi_c - \phi_{ref})) + \\
&\quad + 2|E_{usb}(x,y)||E_{ref}(x,y)|\cos(2\pi(f_{sb} - f_h)t + (\phi_{usb} - \phi_{ref})) + \\
&\quad + 2|E_{lsb}(x,y)||E_{ref}(x,y)|\cos(2\pi(-f_{sb} - f_h)t + (\phi_{lsb} - \phi_{ref})),
\end{aligned}
$$

Beatings

"Direct" probe of fields

Field intensity we are interested in

Collecting things

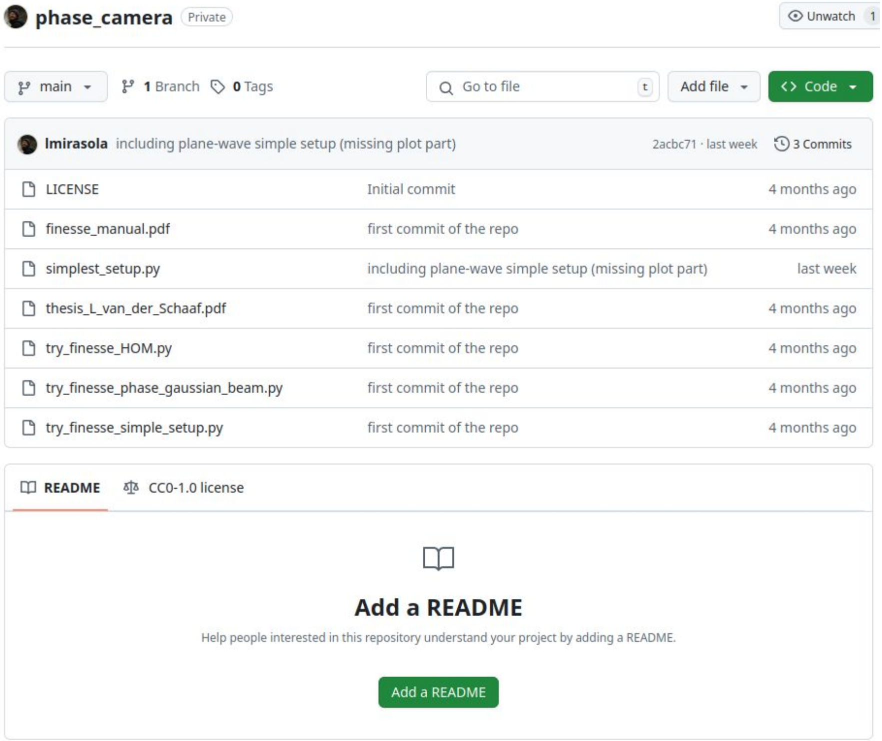
Currently collecting simulation codes in github

Get in touch if you want to be included!

Codes based on finesse package

Finesse 2

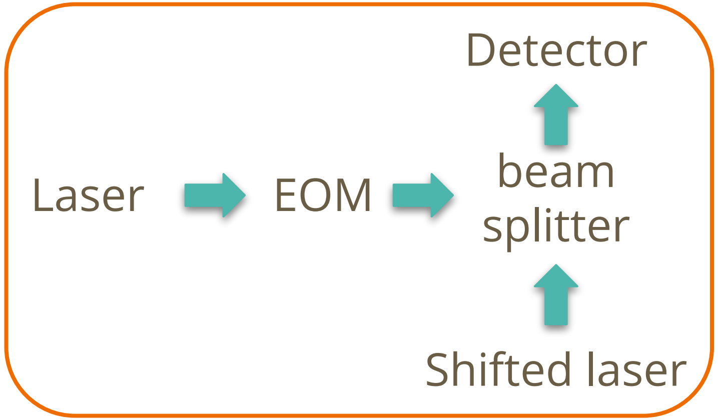
(Frequency domain INterfErometer Simulation SoftwarE)



Note to myself: add instructions for conda env

Plane-wave studies in finesse

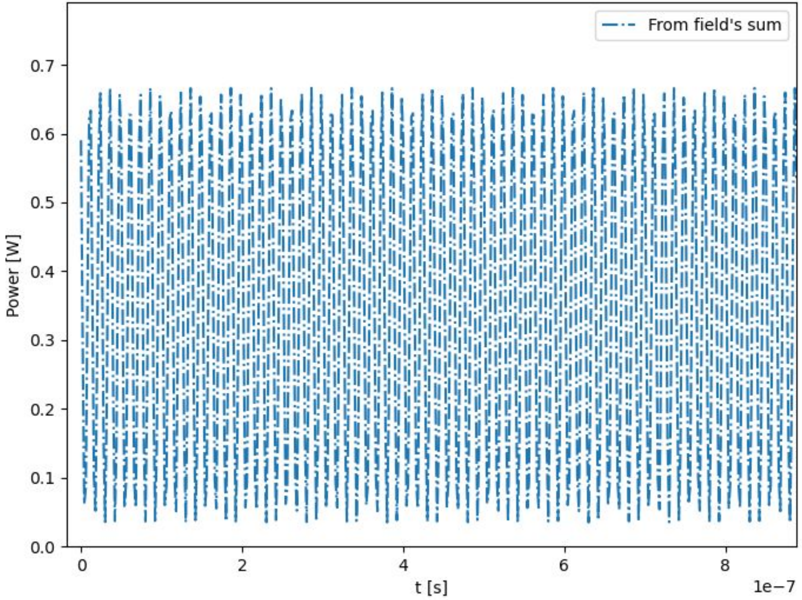
Implemented "simple setup" in try_finesse_simple_setup.py



Quick insight on what a finesse code looks like :)

```
nsb = 3 # number of sidebands
fsb = 8 # MHz --> sideband shift from carrier
fh = 80 # MHz --> frequency shift of primary laser for demodulation
code = f"""
## The optical system ##
l laser1 0.5 0 l1                               # 0.5 W laser with 0 offset wavelength

s l1eom 0.25 l1 eomin

mod eom {fsb}M 0.05 {nsb} pm 0 eomin eomout # EOM (f=8 MHz, m=0.025 as from Eq.5.28 of Schaaf's PhD the
                                              # 5 sideband pairs,
                                              # phase modulation, 0 degree modulation phase)
s eombs 0.25 eomout n1bs

l laser2 0.2 {fh}M l2                           # 0.2 W (0.5 of primary laser reduced by
                                              # 0.4 efficiency from Sec.5.4.1 Schaaf's PhD thesis) laser
                                              # shifted by fh = 80MHz

s l2bs1 0.5 l2 n4bs                            # Space from laser 2 to bs

bs bs1 0.5 0.5 0 45 n1bs outbs dump n4bs       # beam splitter at 45° with R=T=0.5 and no tuning
"""

code = addAD(code, "car", "0", "outbs")  # amplitude detector for carrier

for isb in range(1,nsb+1): # adding amplitude detectors for sidebands
    code = addAD(code, f"up{isb}", f"{fsb*isb}M", "outbs")
    code = addAD(code, f"low{isb}", f"-{fsb*isb}M", "outbs")

code = addAD(code, "adl2", f"{fh}M", "outbs")  # amplitude detector at shifted laser

code += """pd pow outbs # Detecting DC-power\n"""

# carrier demodulation
code = addDeomd(code, "carph", f"{fh}M", 0, "outbs") # Photodiode + mixer + low pass filter
                                                     # (demod. f = 80MHz, demod phase = 0 deg.)
code = addDeomd(code, "carquad", f"{fh}M", 90, "outbs")

for isb in range(1,nsb+1): # adding photodiodes for sidebands
    code = addDeomd(code, f"upsbph{isb}", f"{fh-fsb*isb}M", 0, "outbs")
    code = addDeomd(code, f"upsbquad{isb}", f"{fh-fsb*isb}M", 90, "outbs")

    code = addDeomd(code, f"lowsbph{isb}", f"{fh+fsb*isb}M", 0, "outbs")
    code = addDeomd(code, f"lowsbquad{isb}", f"{fh+fsb*isb}M", 90, "outbs")


code += """
xaxis eom midx lin 0.025 0.025 1        # Varying EOM modulation between 0 and 1
yaxis abs:deg                           # Returns absolute values of detector outputs
"""
```

Plane-wave studies in finesse (2)



Output power at the phase camera

This process can be iterated over carrier and sidebands

Demodulation

$$I = P(t)\cos(2\pi f_{\mathrm{demod}}\, t)$$
$$Q = P(t)\sin(2\pi f_{\mathrm{demod}}\, t)$$

$$P_{\mathrm{demod}} = \sqrt{I^2 + Q^2}$$
$$\phi_{\mathrm{demod}} = \mathrm{atan}\,\frac{Q}{I}$$

Counter-check of finesse's "black box" with "by hand" methods

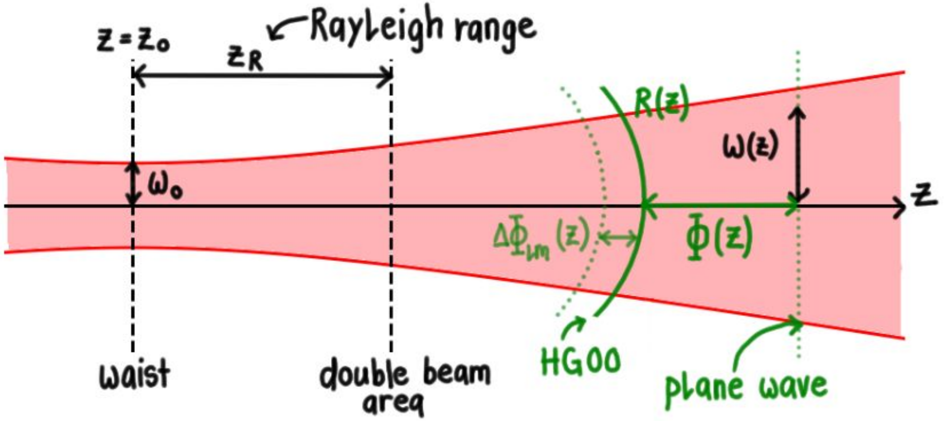
Playing with Gaussian beams

Implemented "simple setup" in try_finesse_phase_gaussian_beam.py

Laser → Detector

```
maxTEM = 3
w0_input = 1e-2 # waist of the beam
z0 = 0 # position of wraist
w_max = 4 # grid around +-3w0 in x-y plane
spacing = 20 # number of bins in the grid

code = f"""
## The optical system ##
l laser1 0.5 0 l1
gauss g1 laser1 l1 {w0_input} {z0}
tem laser1 0 0 0.2 0
tem laser1 1 0 0.2 0
tem laser1 0 1 0.2 0
tem laser1 1 1 0.4 0
```

$$z_R = \frac{\pi w_0^2}{\lambda}$$



$$\vec{E} = \sum_{\mathrm{lm}} A_{\mathrm{lm}} U_{\mathrm{lm}}(\vec{x}) e^{-ikz+i\omega t}\vec{e}$$

$$w(z) = w_0 \sqrt{1 + \left(\frac{z - z_0}{z_R}\right)^2}.$$

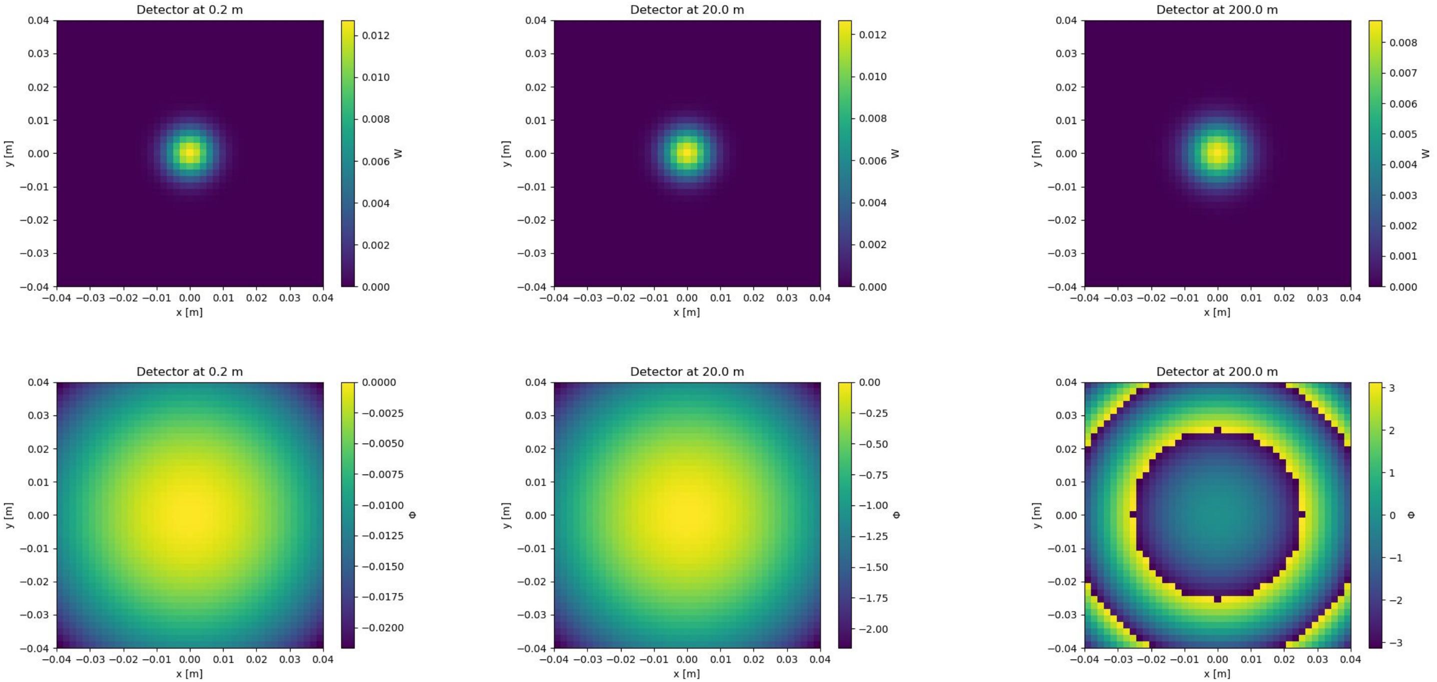
Additional settings for a Gaussian beam

Playing with Gaussian beams (2)

Pure Gaussian beam

Power

Phase

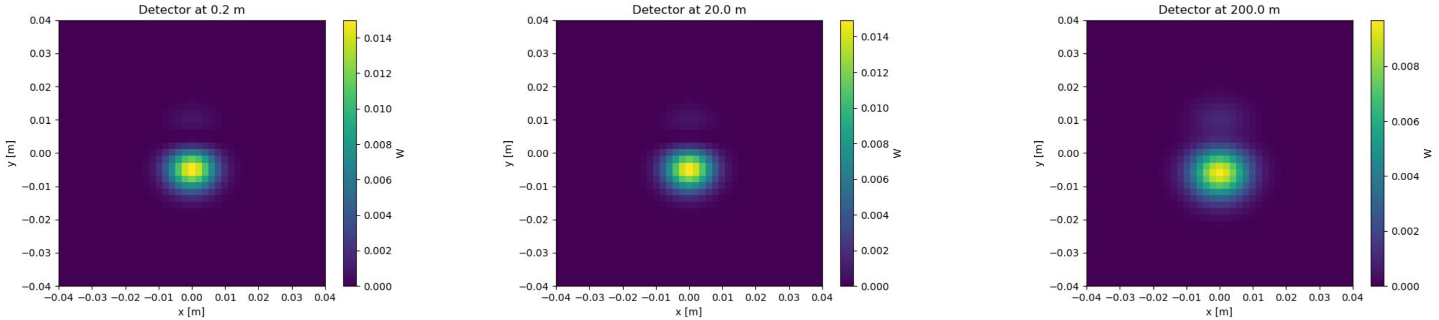


No demodulation

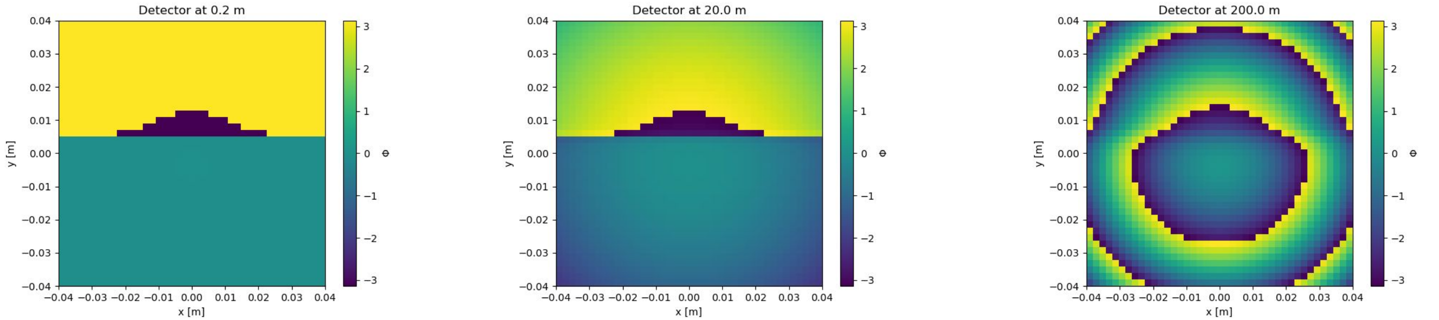
Playing with Gaussian beams (3)

50% (0,0) + 50% (1,0) mode

Power



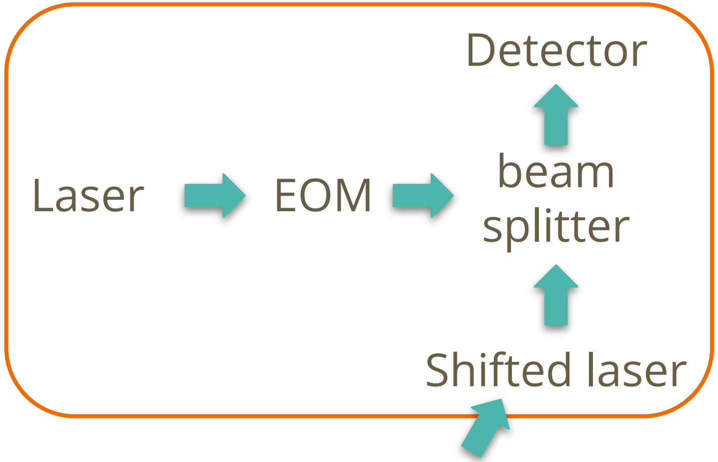
Phase



No demodulation

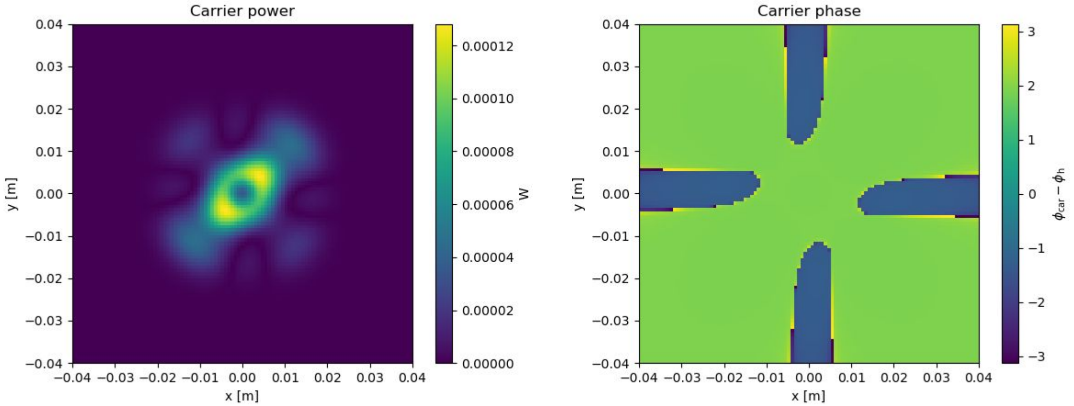
Let’s get a bit more “serious”

Implemented “simple setup” with GB in try_finesse_HOM.py



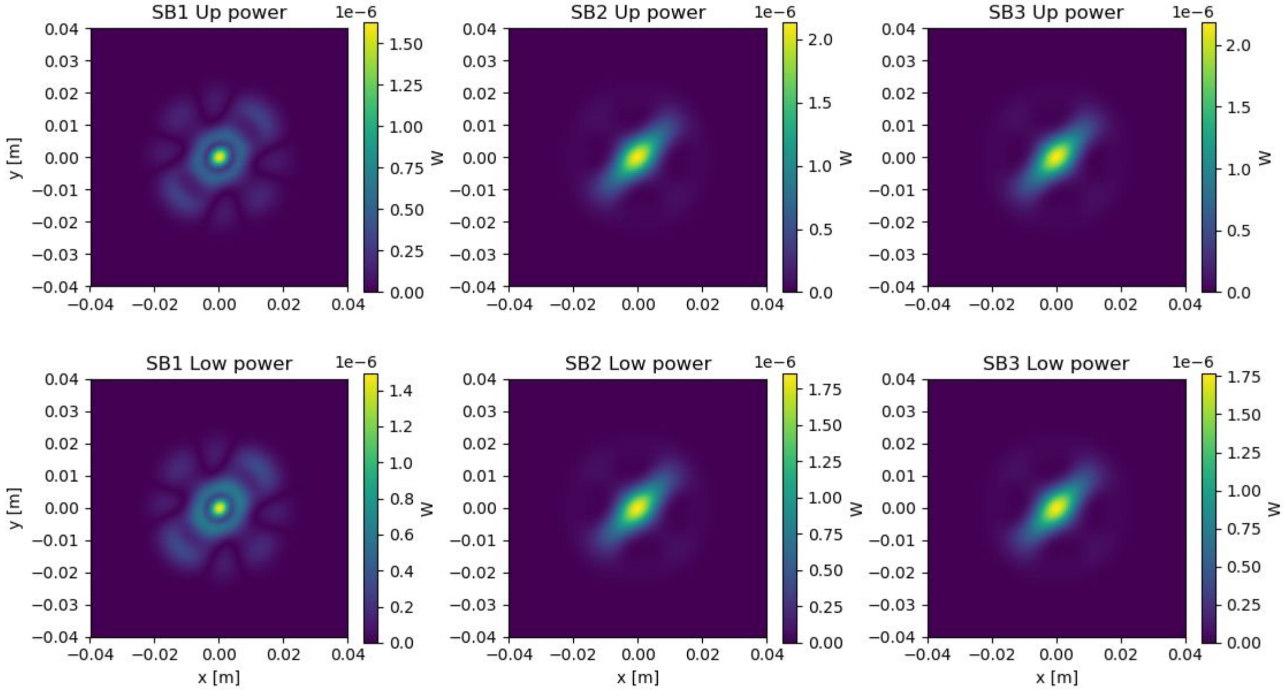
Pure gaussian with $5w_0$

80% (0,0) + 10% (1,1) + 10% (2,2)



Sidebands’ reconstruction in next slides

Let’s get a bit more “serious” (2)



Let’s get a bit more “serious” (3)