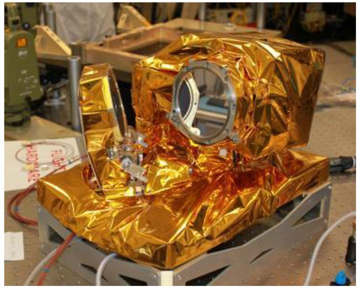
Optical Coms
Intro and Link Budgets

Optical Communications

Reduced spread at optical frequencies:

- Compact, lower power design
- x10 to x100 data throughout at equivalent SWaP
- Tighter pointing requirements



	Deep Space Network	LADEE LLCD
Type	S-Band (~2 GHz)	1550 nm, (~200 THz)
Size	34 m	0.1 m
EIRP	8.3 GW	8.1 GW

Don Boroson, MIT Lincoln Laboratory

2001: SILEX, LEO to GEO, GEO to Ground, 50/2 Mb.s^{-1}, PPM and OOK
2013: LLCD, Moon, 622/20 Mb.s^{-1}, PPM
2016: EDRS, LEO to GEO, 1800 Mb.s^{-1} , Homodyne BPSK
2017: AeroCube-7, 100 Mb.s^{-1} OOK
2021: LRCD, PPM and DPSK
2022: **CLICK-A**, 20 to 50 Mb.s^{-1}, PPM
SDA Tranche 0, 1 (Many vendors), SDA LINCS, PIXL-1 (CubeLCT), ISL Lemur (Spire), Starlink (> 1000?)…
Kuiper, Hedron, ...

Optical Links Components

Laser → Modulator → Optics → (Free space) → Optics → Detector → Converter

Data Handling → Modulator

Pointing → Optics

Pointing → Optics

Converter → Data Handling

Free space

C-band (1550 nm)
Some at 1064 nm, short infrared

- EDFA, YDFA
- Laser diodes
- ND-YaG

Can encode signal on:

- Power
- Frequency
- Phase
- Polarization

Before / after laser

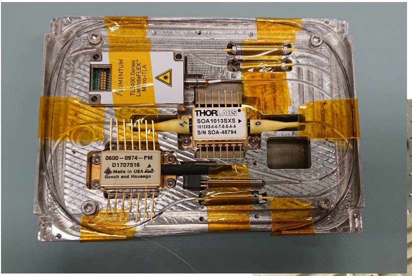
Critical performance driver!

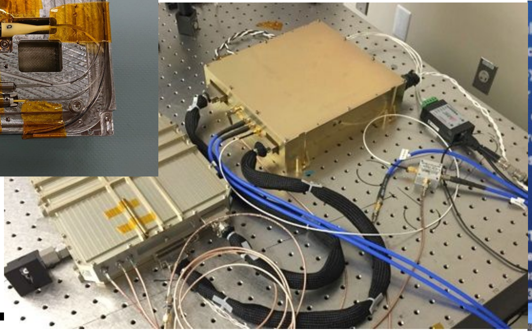
- Gimbals
- Body pointing (ADCS)
- Steerable optics

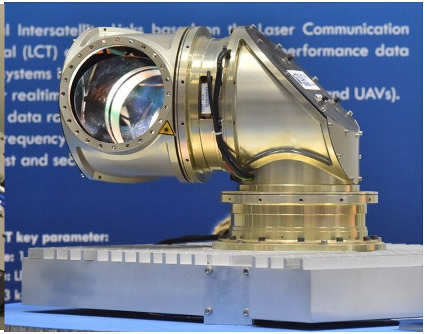
Phased array soon?

Very dependant on modulation method:

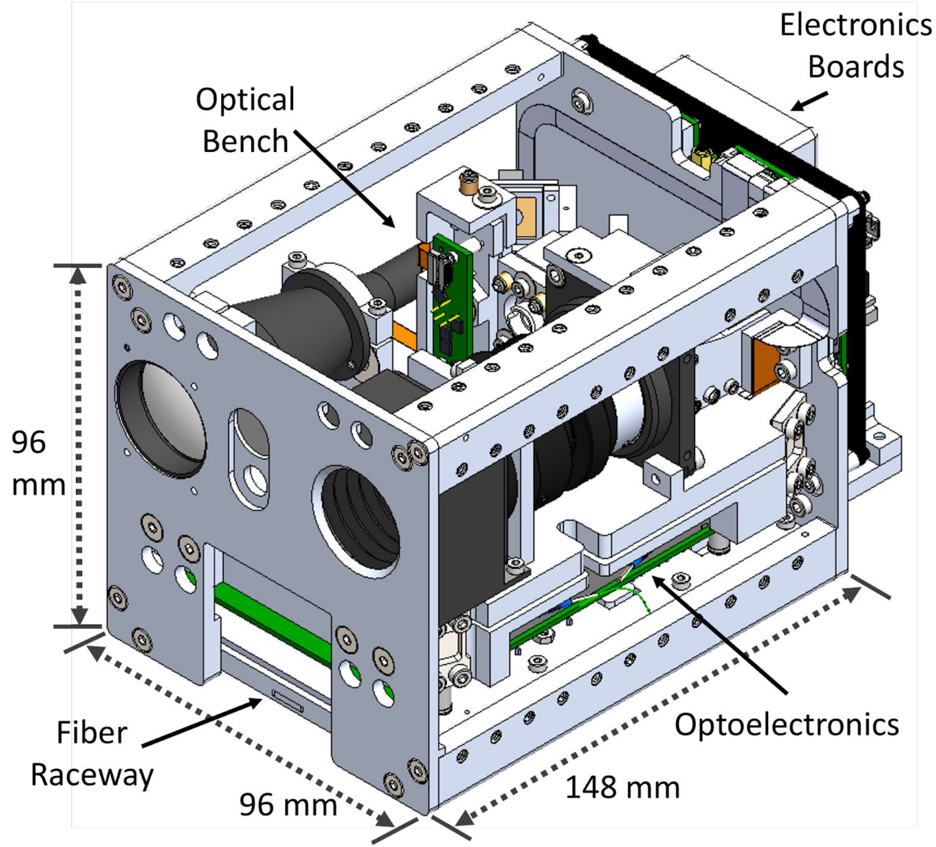
- Direct detection
  - APD (Geiger mode or not)
  - PIN
- Coherent





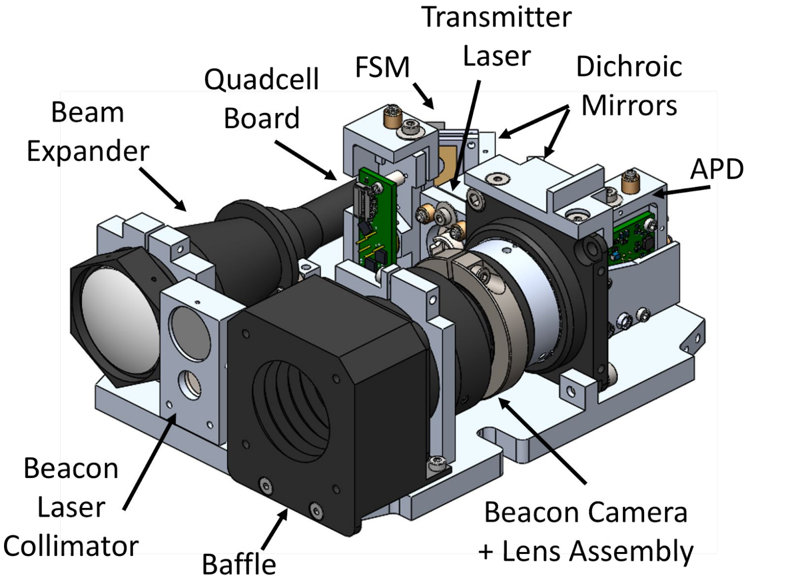


CLICK-B/C Payload Overview



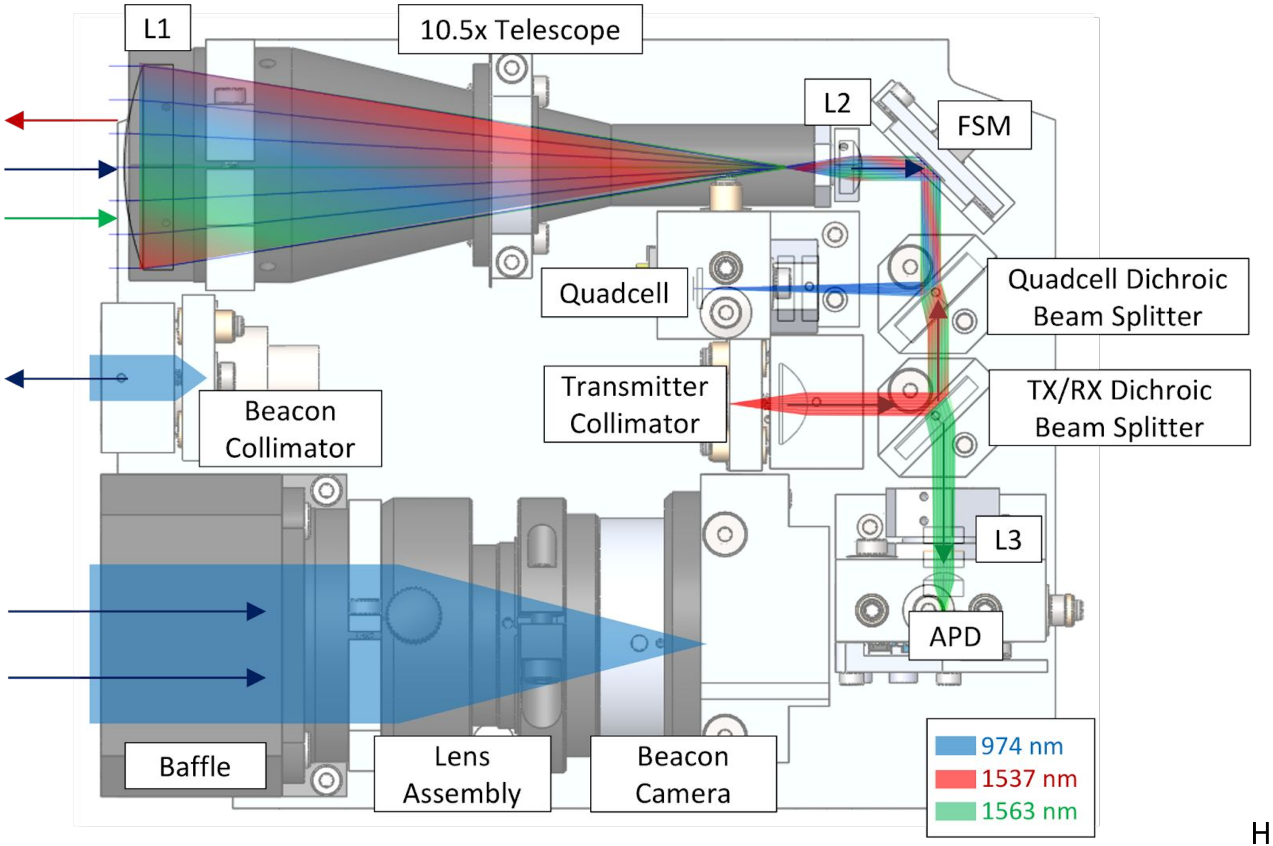
CLICK-B/C Payloads

- Volume – ≤ 1.5U
- Mass – ≤1.7 kg



Hannah Tomio / Will Kamerer, MIT Star lab

CLICK-B/C Optomechanical Design



Beacon Laser
- 974 nm
- 7.15 mrad FWHM
- 250 mW avg.
- Sine-modulated

Transmitter Laser
- 1537 nm for CLICK-B
- 1563 nm for CLICK-C
- 71 urad FWHM
- 200 mW avg.
- 4 PPM – 128 PPM

Hannah Tomio / Will Kamerer, MIT Star lab

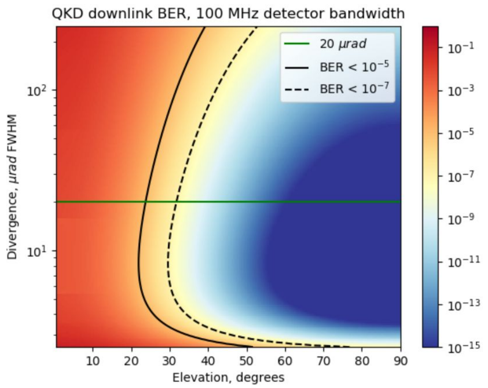
Link Budget Tool

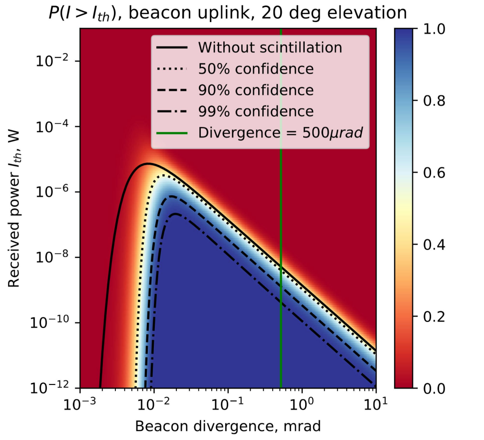
https://github.com/MIT-STARLab/Optical-Link-Budget

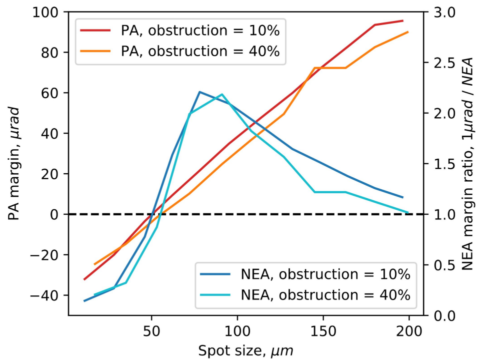
Developed at STAR Lab to help with our cubesat payload design

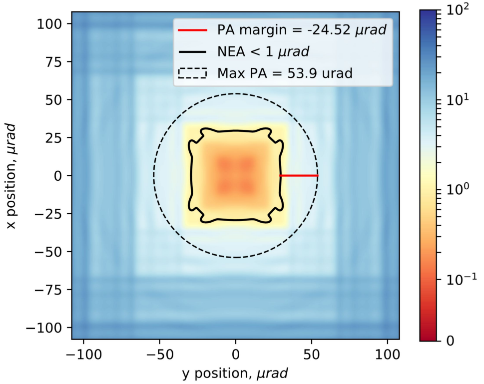
- Python + numpy
- LGPL v3
- Analytical methods, no Monte Carlo
- Optimized for array operations
- Include turbulence effects
- 4 quadrant detector performance analysis

Independent verifications and bug reports welcome!





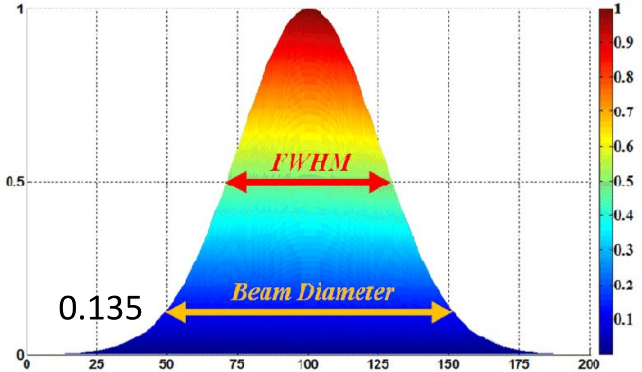




Gaussian Beams

Beam profile:

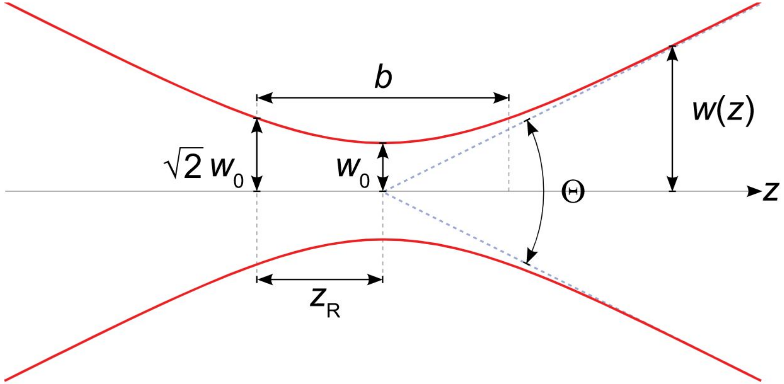
$$I(r,z) = \frac{|E(r,z)|^2}{2\eta} = I_0 \left(\frac{w_0}{w(z)}\right)^2 \exp\left(\frac{-2r^2}{w(z)^2}\right)$$



Wikipedia, CC BY-SA 4.0

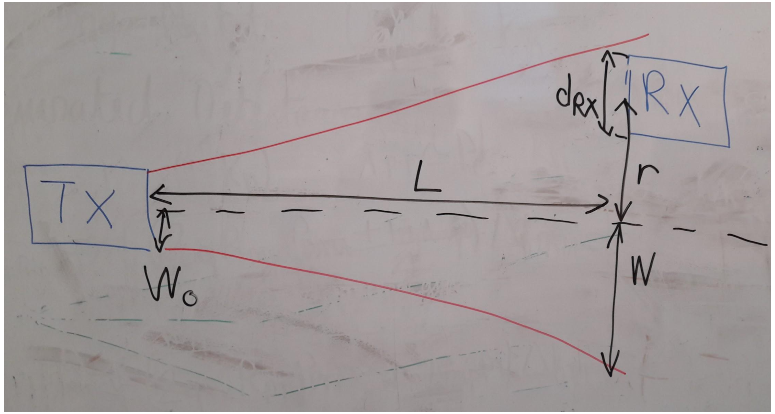
Diffraction:

$$W = W_0 \cdot \sqrt{1 + \left(\frac{\lambda \cdot L}{\pi \cdot W_0^2}\right)^2}$$



Wikipedia, CC BY-SA 3.0

Free-space Loss



$$r = L \cdot \mathbf{tan}(\theta_{PT})$$

$$W = W_0 \cdot \sqrt{1 + \left(\frac{\lambda \cdot L}{\pi \cdot W_0^2}\right)^2}$$

$$G_{range} = \frac{d_{RX}^2}{2 \cdot W^2}\mathbf{exp}\left(\frac{-2r^2}{W^2}\right)$$

L: link distance
θ_{PT}: pointing error
λ: wavelength
W0: 1/e^2 beam radius at transmitter
W: 1/e^2 beam radius at receiver

Example 1: Crosslink

Two satellites in a crosslink, 1000 km apart

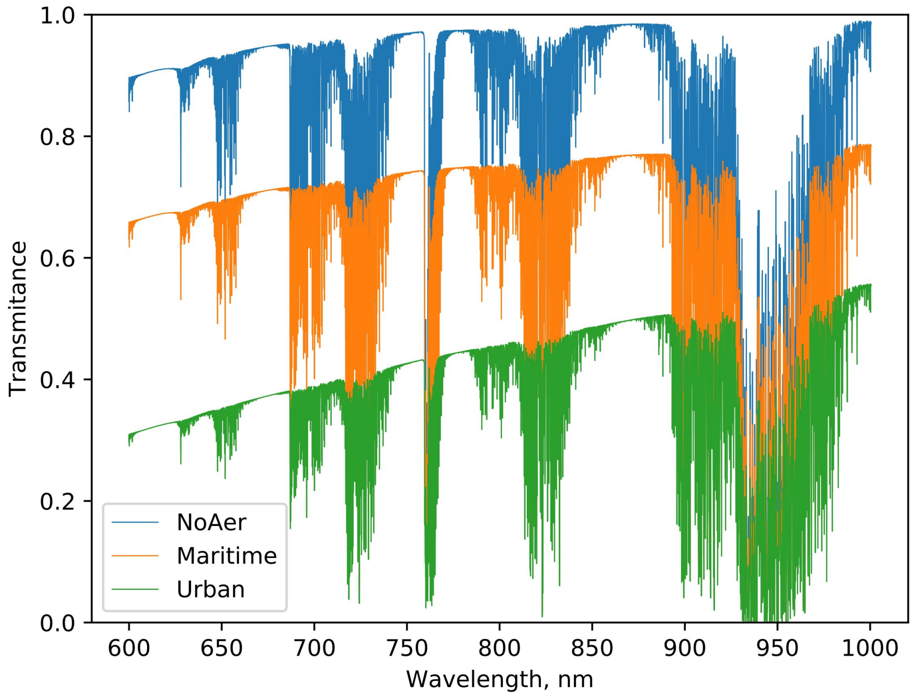
```
import numpy as np
import OLBtools as olb

# Transmit
P tx = 200e-3              # Transmit power laser, W
lambda gl = 1550e-9        # Laser 1 wavelenght, m
beam width = 15e-6         # beam width, FWMH radian
pointing error = 5e-6      # radian
tx_system_loss = 3         # dB

# Receive
apperture = 95e-3          # Apperture diameter, m
rx_system_loss = 3         # dB

link_range = 1000e3        # Link distance, m
```

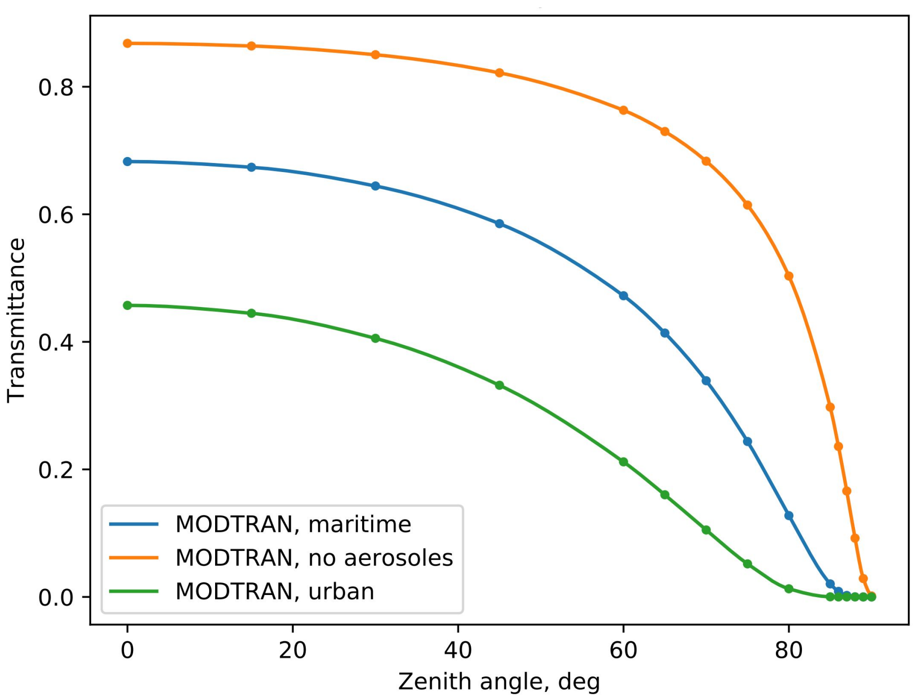
```
# position error at receiver
r = np.tan(pointing_error)*link_range

# beam waist
W_0 = olb.fwhm_to_radius(beam_width,lambda_gl)

# Angular wave number, = 2*pi/lambda
k = olb.angular_wave_number(lambda_gl)

range loss = olb.path loss gaussian(W 0, lambda_gl,
link_range, apperture, pointing_error)

all losses =
range_loss-tx_system_loss-rx_system_loss

P_rx = P_tx*10**(all_losses/10)

print('Received power: %.3f uW' % (P_rx*1e6))
```

Received power: 1.026 uW

Full code

Radiative Transfer





- Absorption and scattering due to aerosoles
- Not within the scope of the script, off-the-shelf solutions exist: MODTRAN

Turbulence

Random refractive index variations due to turbulence in the atmosphere introduce distortions in wavefronts.

The strength of the optical turbulence is noted C_n^2

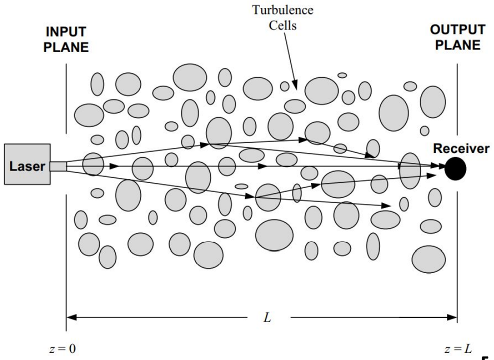
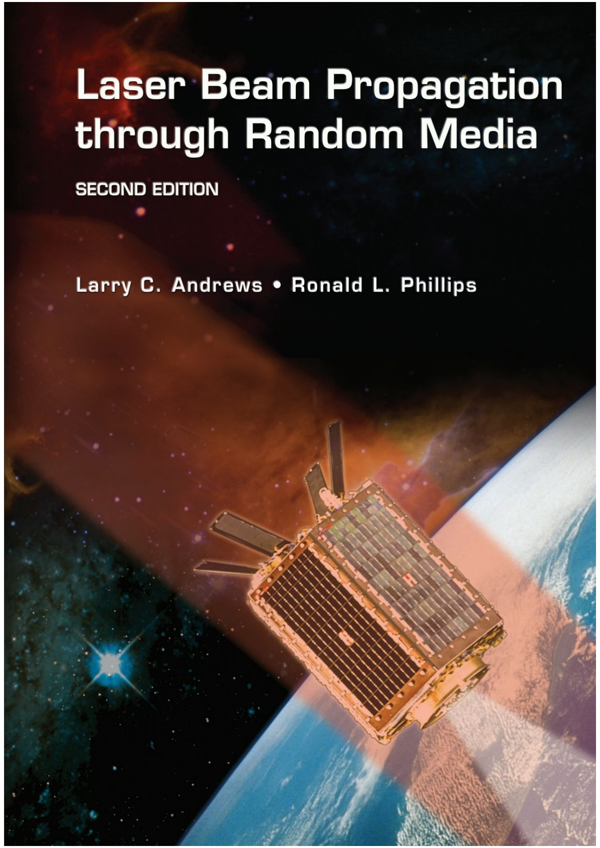


Figure 1.10 Propagation geometry for an extended medium. [1]

Hufnagel-Valley 5/7 model:

$$C_n^2(h) = 3.6 \times 10^{-3}(10^{-5}h)^{10}\exp(-h/1000) \\ + 2.7 \times 10^{-16}\exp(-h/1500) \\ + 1.7 \times 10^{-14}\exp(-h/100)$$



[1] Larry C. Andrews and Ronald L. Phillips. Laser beam propagation through random media. SPIE Press, Bellingham, Wash, 2nd ed edition, 2005. ISBN 9780819459480.

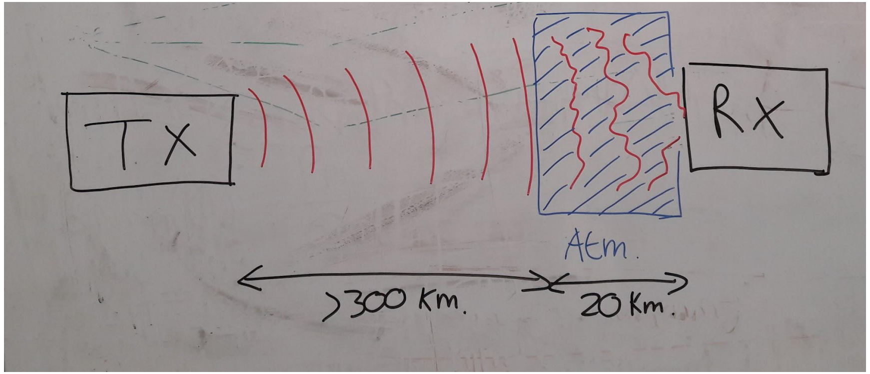
Impact on Uplink and Downlink

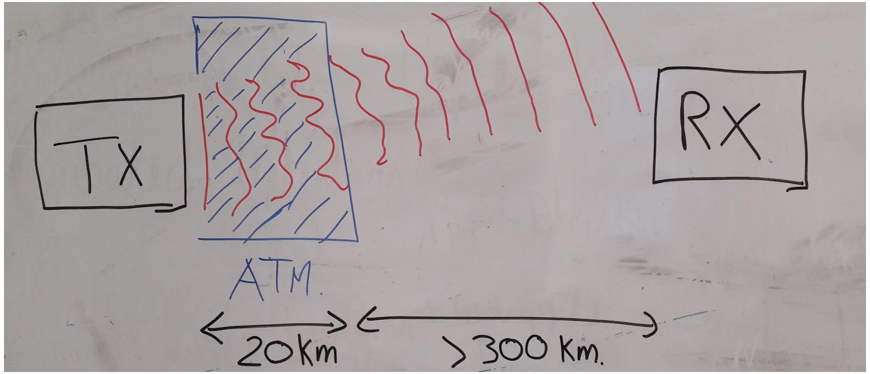
Downlink

- Beam size is driven by diffraction
- No beam wander
- Small correlation scale (= Fried parameter, r_0)
  - Some aperture averaging
  - Adaptive optics useful

Uplink

- Beam spreading due to turbulence
- Beam wander occurs, mitigation depends on *ground* pointing bandwidth
- Large correlation scale, likely >> to a reasonably priced satellite
  - No aperture averaging





Gamma-Gamma Distribution

For both uplink and downlink, a variance due to scintillation can be estimated on-axis and off-axis, based on the C_n^2 and other link parameters, such as zenith angle, altitude, divergence, wavelength. Those variances are used in distributions for the intensity at the receiver. The Gamma-Gamma distribution is a good model, but creates computational challenges.

$$\sigma_x^2 = 1/a \quad \sigma_y^2 = 1/b$$

$$f_{\Gamma-\Gamma}(I) = \frac{2(ab)^{(a+b)/2}}{\Gamma(a)\Gamma(b)I}\left(\frac{I}{I_e(r,L)}\right)^{(a+b)/2} \times \mathrm{K}_{a-b}\left(2\sqrt{\frac{abI}{I_e(r,L)}}\right)$$

Alpha-Mu Distribution

- Similar asymptotic behavior

$$f_{\alpha\mu}(I) = \frac{\alpha\mu^{\mu} I^{\alpha\mu-1}}{(rI_0)^{\alpha\mu}\Gamma(\mu)} \exp\left(-\mu\left(\frac{I}{rI_0}\right)^{\alpha}\right)$$

- CDF much easier to compute
  - Q: regularized upper incomplete gamma function

$$\begin{aligned} P_{\alpha\mu}(I > I_t) =& F_I(I_t) = 1 - \frac{\Gamma\left(\mu, \mu\left(I_t/rI_0\right)^{\alpha}\right)}{\Gamma(\mu)} \\ =& 1 - Q\left(\mu, \mu(I_t/rI_0)^{\alpha}\right) \end{aligned}$$

- Inverse CDF available
  - Both Q and Q^{-1} are fast and available in numpy.

$$\begin{aligned} I(P_{\alpha\mu} > P_t) =& rI_0\left(\frac{1}{\mu}\Gamma^{-1}\left(\mu, (1-P_t)\Gamma(\mu)\right)\right)^{1/\alpha} \\ =& \frac{rI_0}{\mu^{1/\alpha}}\left(Q^{-1}\left(\mu, P_t\right)\right)^{1/\alpha} \end{aligned}$$

Fit

- Distribution can be fitted by setting 3 moments equals
  - n=1: expected value
- Set of two equations, can be converted to logarithmic scale
  - Log Γ easier to use
- The Jacobian can also be expressed in logarithmic scale
  - Ψ is the digamma function

Iterated LSQ solver to find α and μ.

$$\begin{aligned}\mathbb{E}\{X^n\} &= r^n \frac{\Gamma(\mu + n/\alpha)}{\mu^{n/\alpha}\Gamma(\mu)} \\ &= \frac{\Gamma(a+n)\Gamma(b+n)}{\Gamma(a)\Gamma(b)(ab)^n}\end{aligned} \quad \text{for } n \in \mathbb{N}$$

$$\begin{aligned}M_n =& (n-1)\log\Gamma(\mu) + \log\Gamma\left(\mu + \frac{n}{\alpha}\right) \\ & - n\log\Gamma\left(\mu + \frac{1}{\alpha}\right) \\ & - \log\left(\frac{1}{(ab)^{n-1}}\prod_{i=1}^{n-1}(a+i)(b+i)\right) = 0\end{aligned}$$

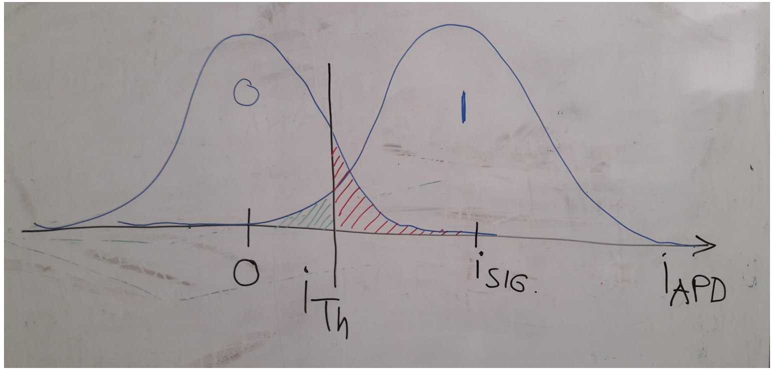
$$\frac{\partial M_n}{\partial \alpha} = \frac{n}{\alpha^2}\left(\psi\left(\mu + \frac{1}{\alpha}\right) - \psi\left(\mu + \frac{n}{\alpha}\right)\right)$$

$$\frac{\partial M_n}{\partial \mu} = (n-1)\psi(\mu) + \psi\left(\mu + \frac{n}{\alpha}\right) - n\psi\left(\mu + \frac{1}{\alpha}\right)$$

Bit Error Ratio

- To get the BER, we need to model a detector, an APD for exemple
  - ΔF: bandwidth
  - R_{APD}: responsivity
  - F_{APD}: excess noise factor
- The detector noise is assumed gaussian
- The BER can be found by adding the probabilities of a:
  - False positive (red)
  - False negative (green)

$$SNR = \frac{R_{APD} P}{\sqrt{2 q_e F_{APD} (R_{APD} P + i_{dark}) \Delta F}}$$



$$BER = \frac{1}{2} \int_0^\infty f_{\alpha\mu}(u) \mathrm{erfc}\left(\frac{SNR\, u}{2\sqrt{2}}\right) du$$

Example 2: Downlink

Downlink from a LEO small satellite:

```
# Objectives
elevation_min = olb.radians(00)  #00 degrees
elevation_max = olb.radians(90)  #90 degrees

# Orbits
altitude    = 500e3 # spacecraft altitude

# Transmit
P_avg =  0.050          # Transmit power laser, W
lambda_gl = 915e-9      # Laser 1 wavelength, m
beam_width = 1000e-6    # beam width, FWMH radian

# Receive
aperture = 600e-3        # Aperture diameter, m
aperture_scaling = 0.60 # Fraction of clear aperture
```

```
# Detector
Responsivity = 50          # A.W-1
Fn_apd = 3.2               # Excess noise factor @M=10
i_dark_apd = 1.5e-9

# Losses
pointing_error = 5e-6      # radian
tx_system_loss = 3.00      # dB (10Log)
rx_system_loss = 3.00      # dB (10Log)

# Atmosphere
Cn2 = olb.Cn2_HV_57        # Hufnagel-valley 5/7 model
```

Full code

Example 2: Downlink

Step 1: Gaussian beam propagation

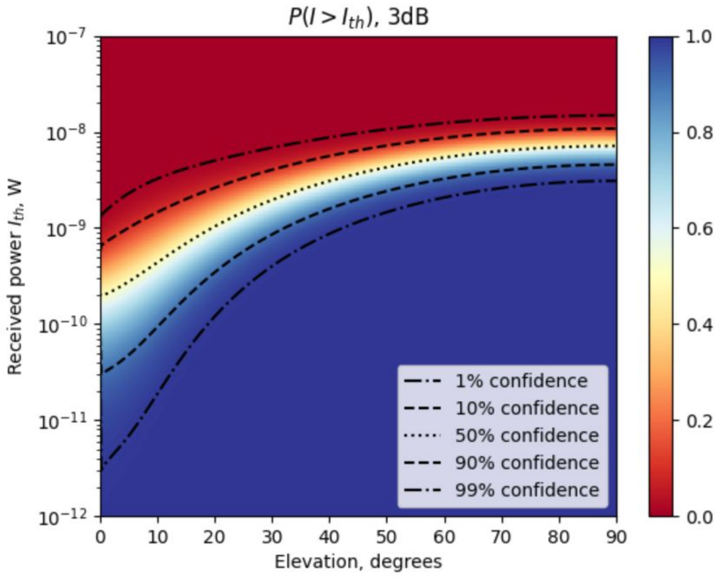
```
H = altitude
h_0 = 0

# Link range for elevation range
link_range = olb.slant_range(h_0,H,zenith,olb.Re)

# Misspointing as distance at receiver
r_s = np.tan(pointing_error)*link_range

# 1/e2 beam radius
W_0 = olb.fwhm_to_radius(beam_width,lambda_gl)

# Wavenumber
k = olb.angular_wave_number(lambda_gl)

range_loss = olb.path_loss_gaussian(W_0, lambda_gl, link_range, aperture, pointing_error)

all_losses = range_loss-tx_system_loss-rx_system_loss

# Expected value of the power at he receiver
Pe = aperture_scaling*P_avg*10**(all_losses/10)
```

Full code

Example 2: Downlink

Step 2: Scintillation

```
# Scintillation
sig2_x, sig2_y = olb.get_scintillation_downlink_xy(h_0,H,zenith,k,W_0,Cn2)

…
# Power distribution coeficients
alpha,mu,r =  olb.gamma_gamma_to_alpha_mu(sig2_x,sig2_y,orders=[2,3])

…
# Cumulative distribution function
cdfs = olb.alpha_mu_cdf(alpha,mu,r,Pe,Psn)
```

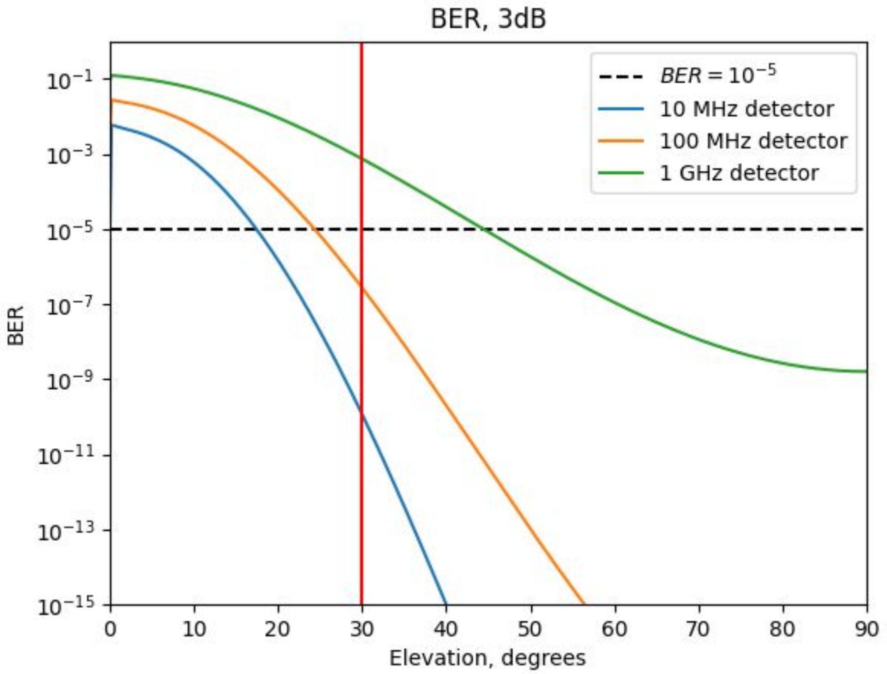


Full code

Example 2: Downlink

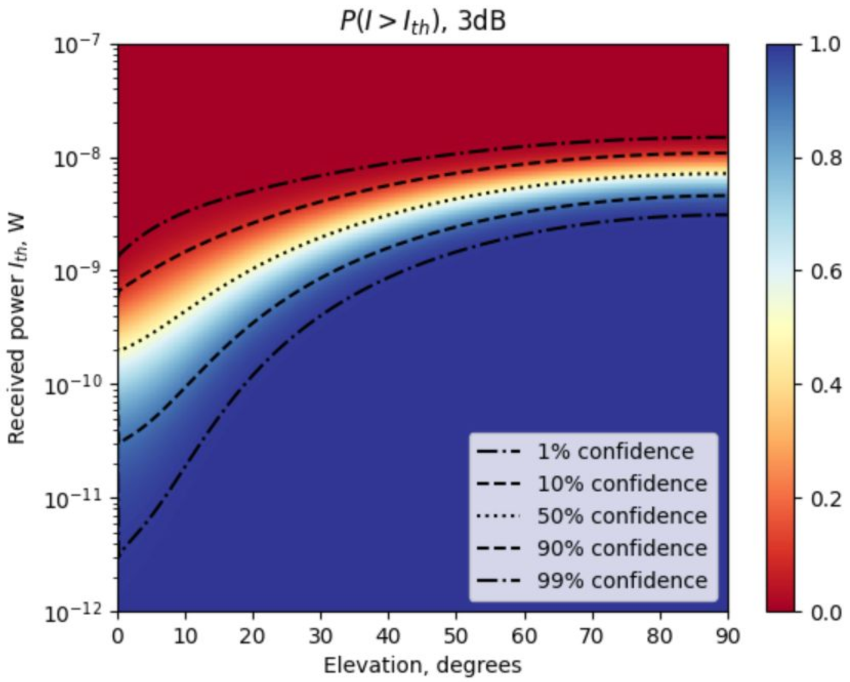
Step 3: Bit Error Ratio

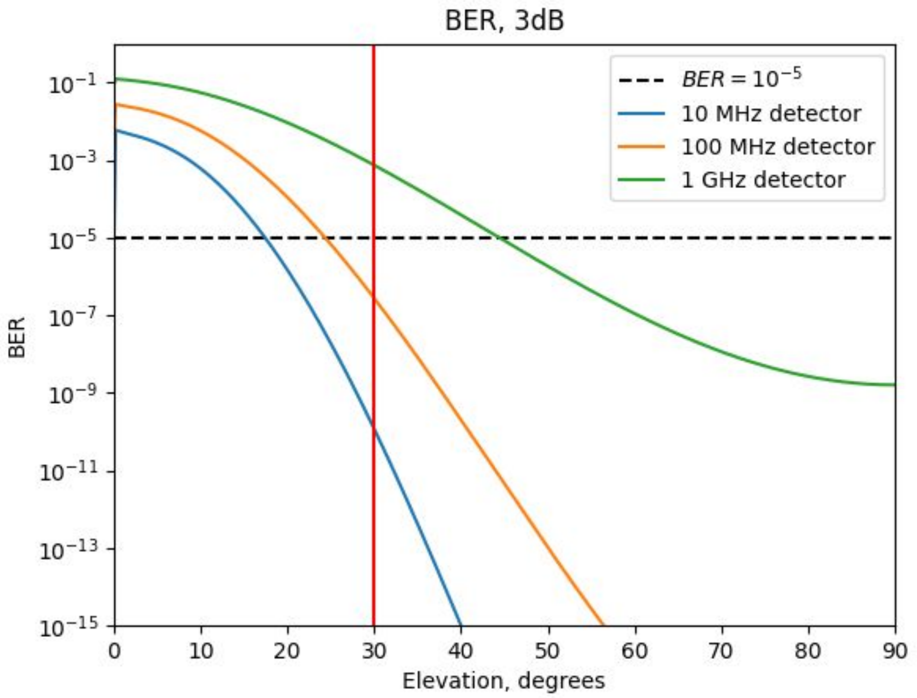
```
pd = old.Photodiode(
    gain=1,
    responsivity=Responsivity,
    bandwidth=10e6,
    excess_noise_factor=Fn_apd,
    dark_current=i_dark_apd)
SNR_10 = pd.SNR(pdfPs)
BER_10 = olb.BER_OOK_integrated(SNR_10,pdfst)

pd = old.Photodiode(
    gain=1,
    responsivity=Responsivity,
    bandwidth=100e6,
    excess_noise_factor=Fn_apd,
    dark_current=i_dark_apd)
SNR_100 = pd.SNR(pdfPs)
BER_100 = olb.BER_OOK_integrated(SNR_10,pdfst)

...
```



Full code

Example 2: Downlink





[Full code]

Example 3: Uplink

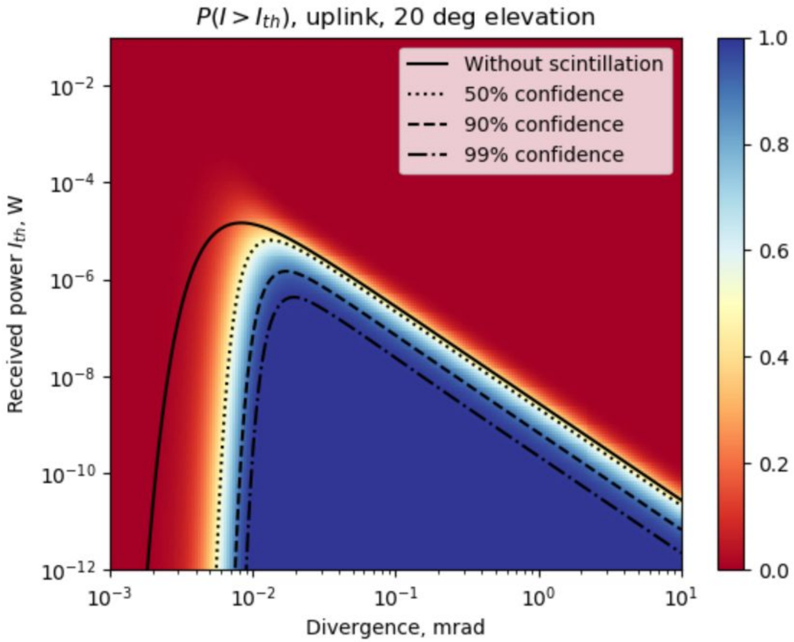
Finding the minimum divergence for an uplink

```
elevation = olb.radians(20)  #20 degrees

# Orbits
altitude    = 500e3 # spacecraft altitude

# Transmit
P_avg =  2.50         # Transmit power laser, W
lambda_gl = 978e-9     # Laser 1 wavelength, m
beam_width_min = 1e-6  # beam width, FWMH radian
beam_width_max = 10e-3

# Receive
aperture = 95e-3    # aperture diameter, m

# Losses
pointing_error    = 5e-6 # radian
tx_system_loss = 3.00   # dB (10Log)
rx_system_loss = 3.00   # dB (10Log)

# Atmosphere
Cn2 = olb.Cn2_HV_57      #Hufnagel-valley 5/7 model
```

Full code

Example 3: Uplink

```
sig2_x, sig2_y = olb.get_scintillation_uplink_untracked_xy(h_0,H,zenith,k,W_0,Cn2,r)
```



Full code