Laboratory experiments with photonic lanterns

Aditya Sengupta

1. What do we want to do with a photonic lantern in the lab?

At the moment, I'm interested in

1. Carrying out identification;
2. Taking an interaction matrix and characterizing the linear range; and
3. Testing linear and quadratic phase reconstruction

2. What does it mean to identify a photonic lantern?

We can model lossless optical propagation as a unitary transformation of the input electric field. Photonic lanterns lend themselves to this interpretation well, because their output is an intensity at each single-mode fiber. We can describe the transmission from the pupil plane to the single-mode fiber ports with a matrix $A = UPF$, where U describes the action of the photonic lantern, P is a change of basis into a basis of guided fiber modes (most commonly the LP modes), and F is a pupil-to-focal propagation matrix.

P and F are relatively well understood and independent of the actual lantern, so for our analysis we can mostly look at the propagation matrix U that takes an N-dimensional subspace of the space of focal-plane electric fields and transforms it to a value of the electric field at each of the N ports. We then see the intensities, i.e.

$$p_{\text{out}} = |UE_{\text{in}}|^2$$

where the $|\cdot|^2$ operation applies element-wise. In this way, U completely characterizes the behaviour of a photonic lantern.

It's helpful to identify U, because knowing the exact behaviour of the PL lets us design algorithms for it a lot more precisely: for example, wavefront reconstruction with a PL. Finite-element simulations are computationally expensive and likely to be less accurate to the particular PL we have, so we need some procedure for finding the elements U_{jk} based on the actual behaviour of the lantern.

Identification means calculating the elements of the lantern's propagation matrix based on empirically-collected data.

3. Constraining the matrix as a linear algebra problem

3.1. Problem setup

If we saw $E_{\text{out}} = UE_{\text{in}}$, this would be an easy problem: for some basis of input electric fields, just apply each one, and the output from basis element i would be the i-th column of the propagation matrix. But instead, we only see intensities. Since U and the inputs are complex-valued, this isn't sufficient information to predict the behaviour of the lantern. In addition to this initial set of queries, we'll need to come up with more in order to fully constrain the matrix elements.

Let's just look at one SMF port; this procedure works independently for all of them because we look at the intensity of each port separately. So we can simplify our problem by considering an N-length vector $\vec{s}$ that takes in a vector, say $\vec{v}$, of electric fields and returns the electric field $p = \left|\vec{s}^T \vec{v}\right|^2 = \left|\sum_k s_k v_k\right|^2$ at the SMF output we care about. If we can identify the elements s_k based on the input vectors $\vec{v}$ we choose, we can identify the whole lantern.

3.2. Basis queries

If we sweep over the basis, putting in $\vec{v} = (0, ..., 1, ..., 0)^T$ with the 1 in each position in turn (call these "basis queries"), we'll identify the absolute values of each element:

$$|s_k|^2 = \left| (s_1, ..., s_k, ..., s_N) \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \right|^2$$

We can write $s_k = |s_k| e^{i\varphi_k}$. Since we know $|s_i|$, we just have to identify the phase φ_k. It's impossible to get absolute phases with just intensity measurements – we can see this by noticing that for an arbitrary measurement $U\vec{v}$, we can phase-shift the entire matrix by some fixed offset θ and get the same result:

$$|(e^{i\theta} \odot U)\vec{v}|^2 = \left| \sum_k (e^{i\theta} U_{jk}) v_k \right|^2 = \left| e^{i\theta} \sum_k U_{jk} v_k \right|^2 = e^{-i\theta} (U\vec{v})^* (U\vec{v}) e^{i\theta} = (U\vec{v})^* (U\vec{v}) = |U\vec{v}|^2$$

so there's no real notion of the "true" phase values that we can measure.

3.3. Combined queries

What we can measure are phase *differences* between different s_ks, which we can extract using the cosine formula if we put in sums of basis elements. (Call these "combined queries".) If we put in $\vec{v} = \left(0, ..., \underbrace{1}_{k}, ..., \underbrace{1}_{l}, ..., 0\right)^T$, we'd get $|s_k + s_l|^2$ as our output, which is related to the individual intensities and phases according to

$$|s_k + s_l|^2 = |s_k|^2 + |s_l|^2 + 2|s_k||s_l| \cos(\varphi_k - \varphi_l)$$

$$\cos(\varphi_k - \varphi_l) = \frac{|s_k + s_l|^2 - |s_k|^2 + |s_l|^2}{2|s_k||s_l|}.$$

This is almost sufficient to recover all the phase differences we're interested in, but since cos is even, we're left with a sign degeneracy; we don't know if we've recovered $\varphi_k - \varphi_l$ or $\varphi_l - \varphi_k$. This isn't an issue when $N = 2$ because the difference between the two can be thought of as an overall phase shift, of the type that we've established we can ignore. But for higher N we'll recover the "true" phases by accumulating consecutive differences, so it matters that we get all the signs right.

We can get this information by looking at two combined queries per basis element. Let's say we do combined queries for (k, l), (k, m), and (as part of the next set) (l, m). We can achieve this in practice by saying $l = k + 1$ and $m = k + 2 = l + 1$; in general you look at the difference between each element and the next, and each element and its neighbor two over, since there's no real order on the basis. These queries give us

$$\cos(\varphi_k - \varphi_l), \cos(\varphi_k - \varphi_m), \cos(\varphi_l - \varphi_m).$$

These are related according to the cosine formula:

$$\begin{aligned} \cos(\varphi_k - \varphi_m) &= \cos([\varphi_k - \varphi_l] + [\varphi_l - \varphi_m]) \\ &= \cos(\varphi_k - \varphi_l)\cos(\varphi_l - \varphi_m) - \sin(\varphi_k - \varphi_l)\sin(\varphi_l - \varphi_m) \end{aligned}$$

We know all the cosine terms, and from those we know the sine terms up to a sign, so depending on whether you need to fix it or not, we can tell the sign of $\varphi_l - \varphi_m$ *relative* to $\varphi_k - \varphi_l$. This is enough information to fully determine the matrix as long as we have a first phase difference.

Let's make this more concrete: suppose we had a 3-port lantern and we wanted to recover the phases $\varphi_1, \varphi_2, \varphi_3$ for each matrix element. Without loss of generality, we can say $\varphi_1 = 0$, and our measurements give us $\cos(\varphi_2), \cos(\varphi_3)$, and $\cos(\varphi_2 + \varphi_3)$. If we say φ_2 is positive, then $\sin(\varphi_2) > 0$, so all that's left to determine is the sign of φ_3, or the sign of $\sin(\varphi_3)$. We can get this from

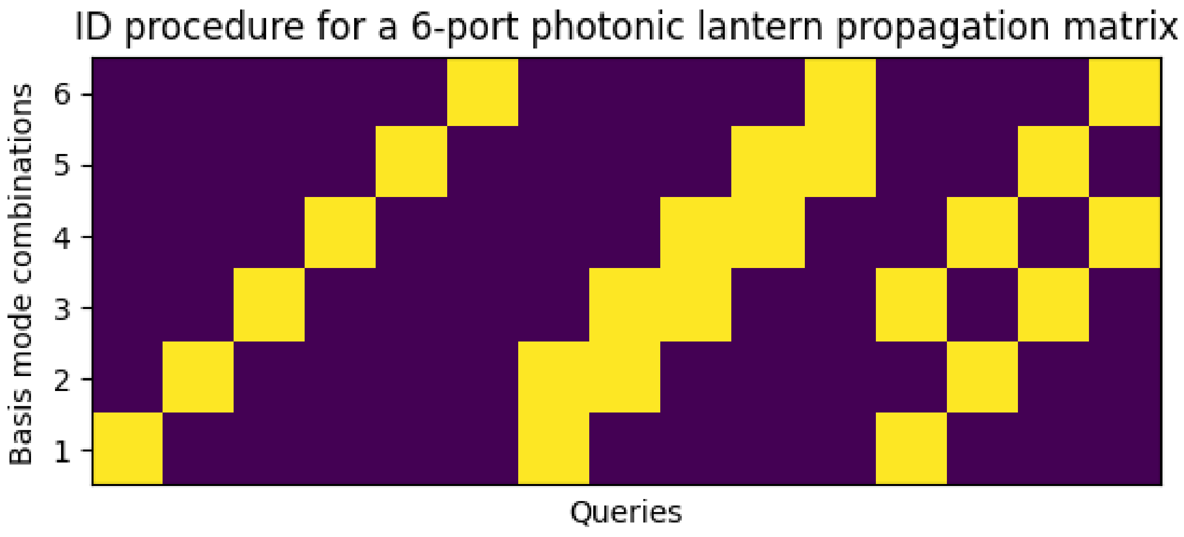
$$\operatorname{sign}(\sin(\varphi_3)) = \operatorname{sign}\left[\frac{\cos(\varphi_2)\cos(\varphi_3) - \cos(\varphi_2 + \varphi_3)}{\sin(\varphi_2)}\right]$$

and if we had a larger number of ports, we'd be able to get the sign of φ_4, φ_5, etc. from the corresponding measurements at higher ports.

Since all of this analysis is for a single row, we'll repeat this N times, or do it once across all the rows as a vector operation.

3.4. Visualizing the queries we need

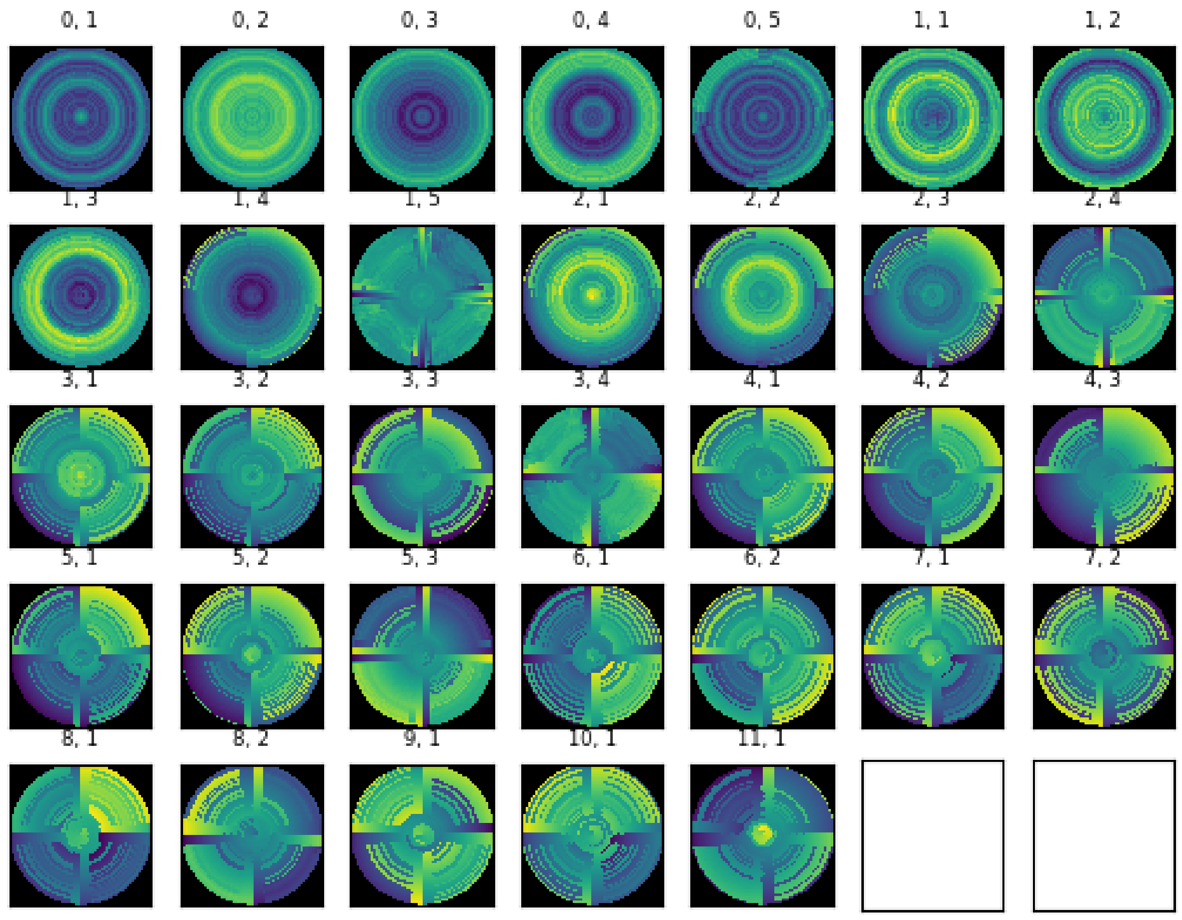
The figure below shows the combinations of basis elements we'll need to produce at the focal plane. On the x axis, we move across different queries, and on the y axis we track combinations of basis elements; each column is a visual representation of the elements we'll need to use. The basis queries are the first 6, and the combined queries are the remaining 9. In general, we'll need $3N - 3$ propagations to constrain an N-port lantern.



4. Choosing bases

So far, we've worked with the assumption that we're able to work in a basis of focal-plane electric field distributions that covers all the aberrations we might be interested in. However, choosing such a basis isn't too easy. The natural basis for a photonic lantern is the LP modes; patterns that describe different solutions to the propagation equation given how large the multi-mode fiber core is relative to the wavelength of light being considered. Unfortunately, it's not too easy to create LP modes at the pupil plane.

LP modes are patterns of light that couple into the fiber, and high-order modes don't always correlate with high-order pupil-plane phase aberrations, so they don't always respond to Zernike modes in a way that's easy for us to track. If we try to back-propagate the LP modes to the pupil plane and phase-unwrap them for visual coherence, we get patterns that look like this:



Note that in order to do this back-propagation, it's necessary to take complex LP modes. The LP modes are usually shown as being real and having odd (sin) or even (cos) angular dependence, but back-propagating the real-valued versions doesn't return anything coherent. It's necessary to take the usual complex combination of these: $\text{LP}_{\text{cos}} + i\text{LP}_{\text{sin}}$.

Many of the LP mode phase screens are likely to be hard to produce with DMs and may need the SLM. Another disadvantage of LP modes is we only know that they're completely accurate if we know the lantern parameters accurately: specifically, we need the input fiber radius and the refractive indices of the core and cladding. We also need to be relatively certain that the diameter of the input beam matches the fiber's diameter well.

Given these drawbacks, it's hard to see why LP modes are a good practical choice. The main reasons are that they're more strongly tied to the physics of the photonic lantern, so we can qualitatively understand their behaviour more easily. Further, they form a vector space, so we can take linear combinations and rely on linearly dependent (and squared) outputs. This makes them easy to use for the identification procedure laid out above.

But it'd be nice if we could get at least the second property from a basis we're more familiar with. If we propagated the Zernike polynomials to the focal plane, we'd get a set of linearly independent electric fields we could work with. Unfortunately, we don't immediately have a vector space, because the mapping from Zernike phases to electric fields is nonlinear.

For example, x-tilt and y-tilt are overall shapes in the pupil plane that cause the PSF to move in the focal plane in the x and y direction respectiely. If we added the resulting electric fields together, we wouldn't get the combined effect of putting on those amounts of x- and y-tilt in the pupil plane, i.e. a PSF at (x, y); instead, we'd get two separate PSFs, at $(x, 0)$ and $(0, y)$, which would be hard to produce at the pupil plane where we can only control phase.

However, we can work around this by applying small aberrations, where the mapping is almost linear. In this example, if we applied sufficiently large tilt levels that a change was detectable but sufficiently small levels so as not to separate where the two PSFs would be, we'd get a single oval-shaped PSF, which can be understood as, e.g. a combination of the initial tilts and some (I think) astigmatism terms – more within what we can represent with Zernikes.

For system identification, applying small pupil-plane aberrations should just give us the exact lantern matrix and not an approximation of it. This is because what we care about are linear combinations at the focal plane, and we only have to make the smallness assumption in order to create the pupil-plane patterns needed to make these linear combinations; once we've successfully made them, linear algebra should apply for propagation through the lantern, meaning the small-aberration assumption is no longer significant. Despite this, I'm worried about cross-talk and not exactly producing the electric fields that simulations claim to and that I'll find the linear algebra approach gives me an inconsistent system. But we've got the advantage of not having to worry about the lantern structure in this case.

I don't think either of these choices are perfect, but since we've got a lot of redundant combined queries, we can endlessly cross-check both of them and hopefully find a reasonable-looking solution.

5. Incorporating an interaction matrix

The interaction matrix B of a photonic lantern is related to the propagation matrix according to equation 5 from Lin et al. 2022,

$$B_{jk} = 2\ \mathrm{Im}\ \left[A_{jk}^{*} \sum_{l} A_{jl}\right]$$

which we can simplify under the assumption that we know A_{j1} (recall that we can safely assume these elements are purely real and have zero phase):

$$B_{j1} = 2\left[-[\mathrm{Im}\ A_{j1}] \sum_{l>1} [\mathrm{Re}\ A_{jl}] + [\mathrm{Re}\ A_{j1}] \sum_{l>1} [\mathrm{Im}\ A_{jl}]\right] = 2[\mathrm{Re}\ A_{j1}] \sum_{l>1} \mathrm{Im}\ A_{jl}$$

so if we take an interaction matrix in the usual way, applying small-amplitude aberrations and looking at the corresponding outputs, we're able to get a measurement of the accumulated imaginary component across modes for each fiber output. Since the measurements we need for an interaction matrix are just the same as the basis queries we described previously, this isn't a new channel of information. Instead, we can use this as a check on the phases we derive from the combined queries.

In general, the k-th column of the interaction matrix will contain the sums of the real and imaginary components of $A_{\dots l}$ for every $l \neq k$, weighted by the negative-imaginary and real components of $A_{\dots k}$. In practice, we should only expect to achieve exact equality for the first column, where the basis queries used to derive these are also the measurements we're using for these checks. For the other columns, we'll be making use of phases derived from the combined queries, which will have slightly different noise terms than the basis queries we use for the checks; checking the sums of imaginary components in this way can therefore be useful for checking the amount of accumulated error in the estimation of the complex components $A_{j(2\dots N)}$.

The effectiveness of linear control using this interaction matrix depends on how large the linear range turns out to be, which should analytically depend on the magnitude of the second-order correction. Defining the second-order interaction matrix, from Lin et al. 2022 equation 11,

$$C_{jk} = 2 \text{ Re}\left[A^*_{jk} \sum_l A_{jl}\right].$$

This is likely to have effects that are too small to directly measure, so we likely can't use this to do similar checks as with B. However, if we derive this matrix from A, we can estimate the linear range. The phase-to-intensity mapping to second order is given by Lin et al. 2022, equation 12:

$$\boldsymbol{p}_{\text{out}} = |A\mathbf{1}|^2 + B\boldsymbol{\Delta\phi} - \frac{1}{2}C\boldsymbol{\Delta\phi}^2 + |A\boldsymbol{\Delta\phi}|^2$$

so a coarse measurement of the linear range is the point at which the effects of the C term become significant. For all three matrices, an element jk describes the impact on intensity at the j-th output due to a change in the k-th mode. So we can look at the linear range in mode k by finding the $\boldsymbol{\Delta\phi}_k$ at which the C term is about equal to the B term, i.e.

$$\boldsymbol{\Delta\phi}_{k,\text{cross}} = \frac{2B_{jk}}{C_{jk}}.$$

When we cross this range of error in mode k for all ports j, the assumption of linearity breaks down because we can't sense the first-order effect without being drowned out by the second-order effect.

6. Practical issues

6.1. The order of lantern ports

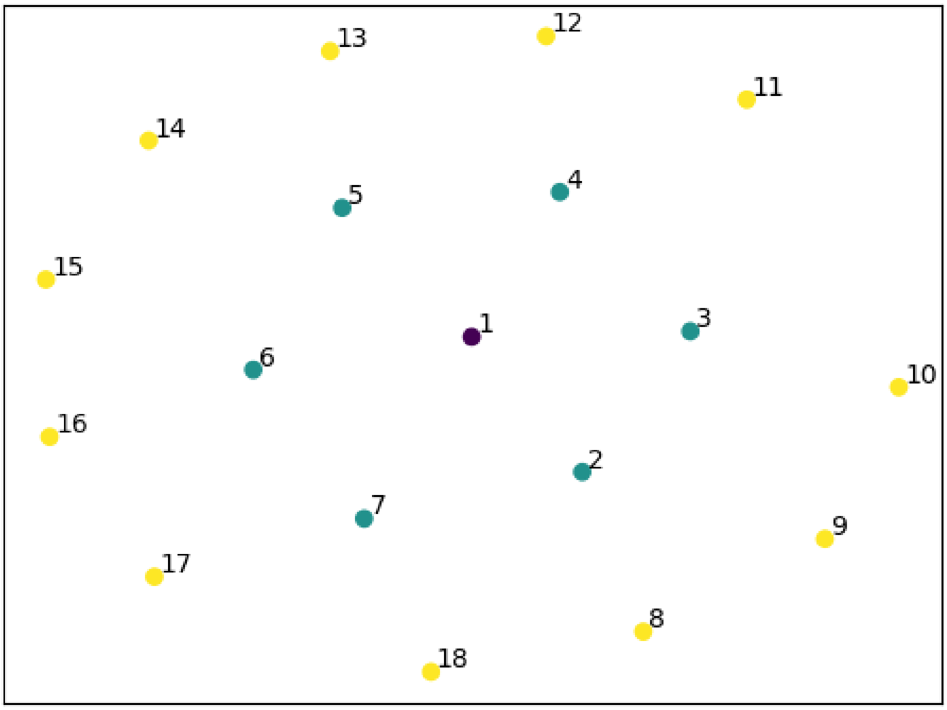
It's possible that the exact positions of the single-mode fiber ports move on the detector over time, with installation changes, etc, so we can't uniquely refer to the ports with their (x, y) coordinates. Instead, I'm adopting a specific convention for port numbering:

1. Ports are numbered from the inside out; the central port is 1, the ports on the circle surrounding the central port are 2 through k (in our case, 7), and so on.
2. Ports within a circle are ordered starting from the lowest one and moving around counterclockwise (in the $+\theta$ direction.)

This matches Lin et al. 2022's convention in Figure 3. To find the correct port ordering on a set of (x, y) centroid locations, do the following:

1. Repeatedly take convex hulls of the set of points (e.g. with *scipy.spatial.ConvexHull*) until only the central port remains, to get sub-arrays consisting of each concentric circle from the outside in.
2. Sort the sub-arrays according to the *arctan2* value of the sub-arrays minus their mean, plus $\frac{\pi}{2}$ modulo 2π.
3. Concatenate the sub-arrays in reverse order, so the central port is first and the circles are stacked from the inside out.

It'll probably never be necessary for anyone to write this functionality again, but I had fun coming up with this algorithm!



6.2. Centroiding and taking port intensity values

Centroid-finding should be carried out before analyzing each new set of data taken on a different day/after a bench adjustment, in case of any drift. We probably want to create a lantern calibration file format and save an instance of it along with each batch of data. So far I've had reasonable success using *photutils.detection.DAOStarFinder*, but I'm guessing that success is going to be very dependent on the parameters we use, like the camera gain, exposure time, and so on. I'm still looking for a more robust way to do this.

We also need to find radii in pixel space. For *photutils*, this is a parameter we put in. This should be fine under ideal circumstances but I suspect they're not all identically sized in reality, so we may want some better tuning/discovery.

With both of these, a simple way of finding an intensity value is to take the average of the masked-out port, i.e. aperture photometry. It's more reliable to take an average than a sum, because even when all the radii are the same, differences in the centroids at the sub-pixel level mean not every port will see the same number of pixels in its mask, so summed values may be skewed by this.

A more robust way of finding intensity values is to weight the average by the intensity pattern of the fundamental mode LP_{01}. This is analogous to PSF photometry, and since we can empirically find the SMF port sizes, it's likely to be accurate.

6.3. SEAL-specific information

- The BlackFly camera we're using is at 128.114.22.1.
- In order to see lantern output, we need to close the loop on the MEMS DM or on both.

6.4. Saturation

There's a small but nonzero chance that we'll saturate the detector on at least one port on one test, so we should have a check for it in data analysis.

Bibliography

Lin, Jonathan, et al. "Focal-Plane Wavefront Sensing with Photonic Lanterns: Theoretical Framework." *Journal of the Optical Society of America B Optical Physics*, vol. 39, no. 10, doi:10.1364/JOSAB.466227.