



Calibrating rectangular interferometer meshes with external photodetectors

CHRISTOPHER ALEXIEV,^{1,2,*}  JASON C. C. MAK,² WESLEY D. SACHER,¹ AND JOYCE K. S. POON^{1,2}

¹Max Planck Institute of Microstructure Physics, Weinberg 2, 06120, Halle, Germany

²Department of Electrical and Computer Engineering, University of Toronto, 10 King's College Road, Toronto, Ontario, M5S 3G4, Canada

*christopher.alexiev@mail.utoronto.ca

Abstract: Multiport interferometer meshes can be used to implement unitary transformations on input vectors of light in both the classical and quantum domain. In practice, the phase-shifters in a mesh photonic circuit must be calibrated to compensate for phase errors due to fabrication variations. Calibration using photodetectors external to the mesh has been demonstrated for triangular meshes, but not rectangular meshes. Here, we propose an algorithm for the calibration of rectangular meshes using only external photodetectors and simulate it to evaluate its feasibility.

© 2021 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

1. Introduction

Multiport interferometer mesh integrated photonic circuits, consisting of 2×2 Mach-Zehnder Interferometers (MZIs) as their “nodes”, are a promising tool for performing unitary transformations on a vector of light, or non-unitary transformations when combined with amplitude modulation [1]. These meshes have potential applications in neural network matrix computation, linear information processing, quantum computing, and mode division multiplexing [1,2]. Two common categories of these meshes are the triangular “Reck” mesh [3] and the rectangular “Clements” mesh [4]. The rectangular mesh is more compact, has less optical loss, and is significantly more tolerant to fabrication-caused phase errors and unbalanced beamsplitter losses [4,5], but, so far, methods to calibrate rectangular meshes have used photodetectors (PDs) in each node, increasing the size, electrical control complexity, and optical loss of the photonic circuit.

Some mesh architectures [6], such as triangular meshes but not rectangular meshes, have been calibrated to be a specific unitary matrix via “self-configuration,” the use of simple feedback loops between phase control inputs and optical outputs to align the mesh with a target transfer function [6]. Progressive algorithms that rely on a sequence of simple single-parameter power maximizations or minimizations, instead of one global optimization, to perform self-configuration have been proposed and demonstrated [7–12]. These algorithms tend to be more flexible with PDs on each node, but can also be performed with only external PDs at the mesh’s output ports. Progressive algorithms for configuring a triangular mesh to implement a unitary transformation, for having all nodes in bar or cross states (where light goes straight through or crosses over, respectively), and for tuning nodes’ beamsplitters to their 50:50 states are detailed in [13].

Rectangular meshes have not been shown to use self-configuration. They can be calibrated with progressive algorithms such as “RELLIM” [2] and “PRELLIM” [14], which calibrate one mesh layer at a time by maximizing (minimizing) the output power for an input vector that should produce an output vector of 1’s (0’s). These algorithms require PDs on each mesh node and accurate phase control for each input, increasing the size and electrical complexity of the photonic circuit. Alternatively, [15] shows how to theoretically factor phase-shifts through a rectangular mesh to calibrate it. However, each column of nodes is a large, separate, and modular unit to enable calibration of the nodes’ splitting ratios (controlled via their “internal phase shifters”).

Calibration of MZI meshes with external PDs has been reported for MZI switching networks [16–18]. Calibration of a 32×32 switch with 1024 elements was experimentally demonstrated in [18], and could be improved to complete calibration within minutes, showing promise for rapid calibration of large scale MZI meshes for real-world applications. However, Clements meshes' topology differs from typical switch networks, such as that of [18], and also features “external phase-shifters” to set phase relations between nodes, which are irrelevant in switching networks.

Here, we theoretically and numerically demonstrate calibration of Clements meshes with external PDs, calibrating nodes' internal and external phase-shifters. We introduce a “Node Isolation Algorithm” for Clements meshes, which—by similar ideas as [18] for rectangular switching circuits—ensures light in a specific node is entering only one of the node's inputs, and that light at the external PD comes from only one output of the node. This enables us to build on prior work [6,13] on triangular mesh self-configuration to calibrate the nodes' internal phase-shifters and/or tunable beamsplitters. We finish the calibration with an “External Phase-Shifter Calibration Algorithm” for calibrating nodes' external phase-shifters based on [6,15].

This manuscript provides a theoretical framework for calibrating mesh phase shifters. Overcoming inaccuracies arising from, for example, optical loss, PD noise, or fixed beamsplitters, is beyond the scope of this work. However, our algorithm can tune tunable beamsplitters, if they are available.

The article is organized as follows: Section 2 describes the notation; Section 3 outlines the overall calibration procedure; Section 3.1 describes the “Node Isolation Algorithm”; Section 3.2 describes the “External Phase-Shifter Calibration Algorithm”; and Section 4 presents numerical simulations demonstrating the feasibility of this approach.

2. MZI mesh notation

Figure 1(b) shows the assumed mesh node MZI [19], which consists of two directional couplers, an internal θ phase-shifter controlling splitting ratio, and an external ϕ phase-shifter controlling output phase. “Ideal” and “actual” versions of the MZI are shown in Fig. 1(b). The ideal version represents an MZI with no phase or beamsplitter errors, with phase-shifts of θ and ϕ . The actual version represents an MZI with imperfect beamsplitter ratios, phase errors (caused by, for example, waveguide width variations), and phase-shifts as functions of input voltages V_θ and V_ϕ . For simplicity, we assume no optical loss; however, if the losses of the two arms of the MZI are equal, a compensating scaling factor can be applied to the mesh outputs [4]. Incorporating unbalanced losses, such as phase-dependent phase-shifter losses, is subject to further work.

The transfer matrix U_{MZI} of the ideal MZI, given by the product of the transfer matrices of the constituent components in Fig. 1(b), is given by [19]

$$U_{MZI} = R_\phi B R_\theta B = i e^{i\frac{\theta}{2}} \begin{bmatrix} e^{i\phi} \sin\left(\frac{\theta}{2}\right) & e^{i\phi} \cos\left(\frac{\theta}{2}\right) \\ \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \end{bmatrix}, \quad (1)$$

where R_ϕ , B , and R_θ are defined in Fig. 1(b). Each MZI, with an extra external phase-shifter in front, implements an arbitrary unitary transformation [19]. When the MZIs are stacked into an N -input mesh with the “column 0” of external phase-shifters in front, as in Fig. 1(a), one achieves an arbitrary $N \times N$ unitary transformation with the transfer function in [19]. The procedure in [4] can translate any unitary matrix into the required phase-shifts of an ideal rectangular mesh.

The notation used to index phase-shifters, columns, inputs, and mesh outputs is shown in Fig. 1(a). The external phase-shifters on the last column of the mesh in Fig. 1(a) are not required if only amplitudes are measured. External phase-shifters on the bottom row of the mesh do not correspond to MZIs and their phase-shifts should be set to zero. The mesh can be streamlined by removing the phase-shifters on the top row, and moving the MZI external phase-shifters from

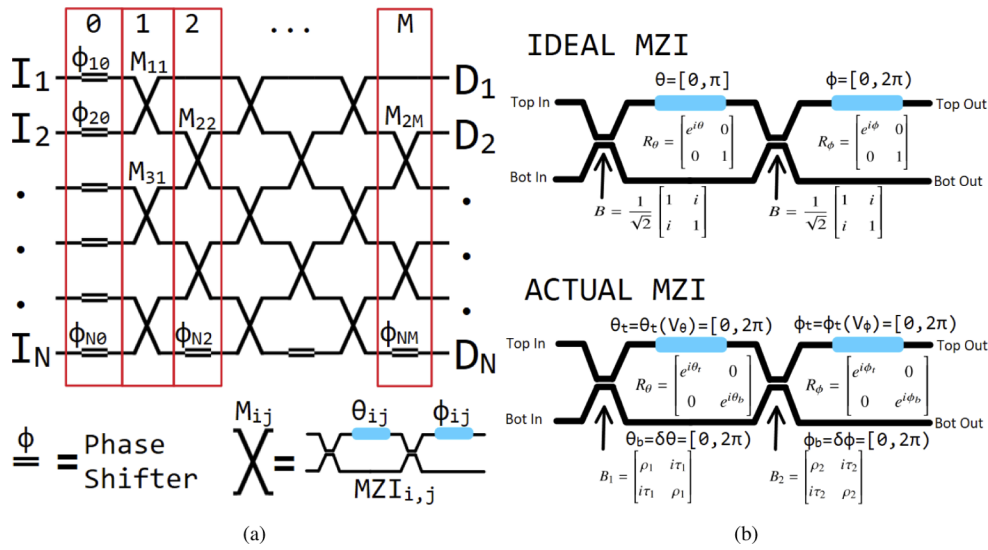


Fig. 1. (a) Schematic of an ideal rectangular mesh node MZI. Black lines represent waveguides in an integrated photonic chip. It has N inputs (labelled I_1, I_2, \dots, I_N), N output detectors (labelled D_1, D_2, \dots, D_N), and M columns. (b) Schematic of the ideal and actual MZI. The transfer matrices for constituent components are shown. θ and ϕ are respectively the internal and external phase-shifts of an ideal MZI; $\theta_t, \theta_b, \phi_t$, and ϕ_b are the top and bottom phase-shifts in an actual MZI; V_θ and V_ϕ are the voltages input to the phase-shifters; τ and ρ are beamsplitter through and cross transmissions; and $\delta\theta$ and $\delta\phi$ are constant phase errors.

the “Top Out” arms to the “Bot In” arms. However, we work with our format throughout for consistency, and algorithms we present can be adapted to other such rectangular mesh designs.

3. Calibration procedure

Mesh calibration finds functions $\theta(V_\theta)$ and $\phi(V_\phi)$ for each node, so an “actual” MZI operates as an “ideal” MZI in Fig. 1(b) with phases θ and ϕ [8]. Once calibrated, the mesh is programmed by calculating the required phase-shifts to implement the desired unitary matrix, then using the $\theta(V_\theta)$ and $\phi(V_\phi)$ relationships to achieve those phases with voltages V_θ and V_ϕ . If the beamsplitter is not 50:50, tunable beamsplitters can be used [7].

The calibration procedure has three steps: 1. Beamsplitter ratios, 2. Internal phase-shifts, and 3. External phase-shifts. Section 3.2 describes step 3. Steps 1 and 2 can use established MZI calibration algorithms, such as “Beamsplitter 50:50 Setup Algorithm” [13] and “Co-sinusoidal Proportional Calibration” [6,8]. However, these algorithms assume light enters only one of the node’s inputs and that only light from one of its outputs is detected. To “isolate” a node for calibration like this using an external PD, Section 3.1 proposes a “Node Isolation Algorithm”.

For the internal phase calibration, after a node is isolated, the voltage to the internal phase-shifter can be swept and its $\theta(V_\theta)$ relation is given by the following equation depending on whether the MZI’s input port is opposite (i.e., top or bottom) from or on the same side as the output detection port:

$$\theta(V_\theta) = \theta_i(V_\theta) - \theta_b = \begin{cases} \arccos\left(\frac{2P(V_\theta) - P_{min} - P_{max}}{P_{max} - P_{min}}\right) & \text{opposite side} \\ \arccos\left(\frac{P_{max} + P_{min} - 2P(V_\theta)}{P_{max} - P_{min}}\right) & \text{same side.} \end{cases} \quad (2)$$

In Eq. (2), P_{max} and P_{min} are the maximum and minimum detected powers of the voltage sweep [6]; $P(V_\theta)$ is the detected power as a function of the voltage; and $\theta_t(V_\theta)$ and θ_b are as shown in Fig. 1(b). Equation 2 is independent of the PD responsivity and the beamsplitting ratios, though beamsplitters should be near ideal for the mesh to work. The reason for the first equality in Eq. (2) is that an MZI with a bottom phase-shift θ_b is mathematically equivalent to an MZI with that phase factored from both internal arms to both output arms, where the phase is absorbed into later calibration steps. So the “actual” MZI can be described as an “ideal” MZI with phase $\theta(V_\theta)$.

In short, the mesh calibration procedure is as follows:

1. Isolate each node in the mesh using the “Node Isolation Algorithm” in Section 3.1. For each isolated node, shine light into the mesh and detect output light at the input and output ports determined by the algorithm. Then do the following in order:
 - (a) Adjust the MZI beamsplitters, if tunable, with “Beamsplitter 50:50 setup algorithm”.
 - (b) Calibrate MZI internal phase with “Co-sinusoidal proportional calibration” (Eq. (2)).
2. Perform the “External Phase-Shifter Calibration Algorithm” in Section 3.2.

3.1. Node isolation algorithm

This algorithm “isolates” each node in the rectangular mesh so light only enters a given node at one of the node’s inputs and only light exiting one of its outputs is detected, allowing simple power maximization calibration algorithms to be used. The algorithm is part of a family of methods that first calibrates internal phase-shifters along the longest diagonal in the mesh, then sets calibrated nodes into bar or cross states to isolate adjacent nodes, similar in principle to the algorithm in [18] developed for a different circuit topology. The algorithm is described in pseudo-code 1 for clarity and proven in Appendix A for $M \leq N$. Select algorithm steps are illustrated in Fig. 2. Its time complexity is $O(NM)$ because each node is isolated once.

First, light is inputted into I_1 and propagates down a main diagonal of uncalibrated nodes $\{\text{MZI}_{i,i} | i \in \mathbb{Z} > 0\}$, reaching a PD that isolates these nodes, enabling their calibration. Then, the top of the mesh can be calibrated one diagonal layer at a time, starting with the layer directly above the main diagonal. Each diagonal is calibrated, using a different PD and input I_1 , by isolating and calibrating one node at a time from bottom to top. To isolate the node being calibrated, already calibrated nodes are set to a bar state, except on the diagonal being currently calibrated and all but one node on the main diagonal, which are set to cross state.

Then, the bottom of the mesh can be calibrated one diagonal layer at a time, starting with the layer directly below the main diagonal and light inputted into I_3 ; for each subsequent diagonal, the input number increases by two. Each diagonal is calibrated by isolating and calibrating one node at a time, from the top down, by setting already calibrated nodes to a bar state, except for those on the current diagonal, which are set to cross state. PD D_{iOut} isolates nodes on row $iOut$.

3.2. External phase-shifter calibration algorithm

This algorithm is similar to the method in the appendix of [15], which describes factoring of external phase errors through a rectangular mesh. We based this algorithm on the triangular mesh calibration methods in [6] and have adapted it to rectangular meshes. All phase-shifts below are taken relative to the phase of the input light to the mesh.

A mesh of ideal phase-shifters has no phase difference between nodes when all external phases are set to zero. To describe an actual mesh as ideal, the phase-shifter on each input of each node is described as $\phi=0$ when the node’s inputs are in phase. This is because an MZI with the same phase, ϵ , at both its inputs is mathematically equivalent to having ϵ on its outputs. The value of ϵ is irrelevant once the next column of phase-shifters is also calibrated; this is shown in Fig. 3.

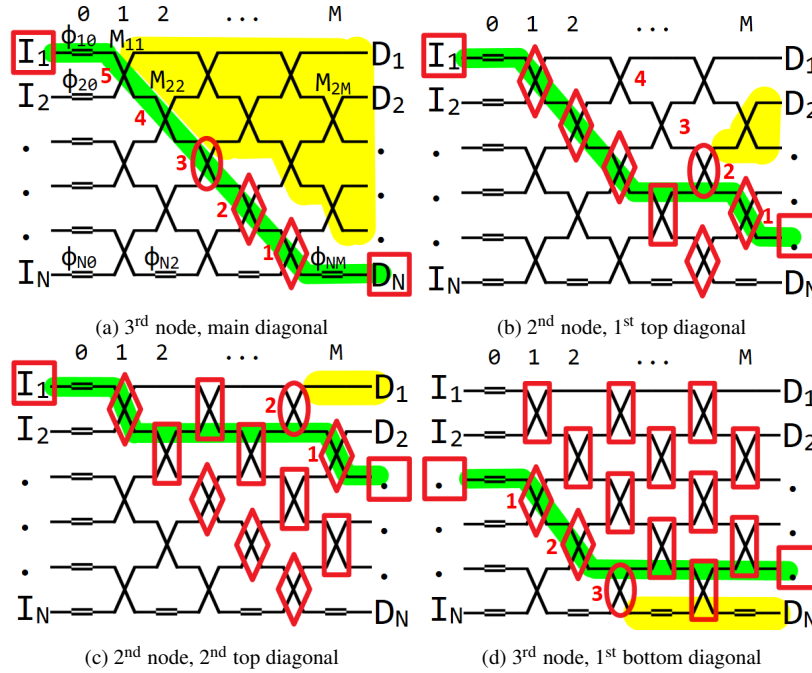


Fig. 2. Calibration of select nodes in select diagonals via the “Node Isolation Algorithm”. Mesh notation is as in Fig. 1; green highlight is the path of light used to calibrate a node; red rectangles encircle already-calibrated nodes in bar states (diamonds in cross states); a red oval shows the node being calibrated; red numbers on nodes indicate the order nodes are calibrated in for a given diagonal; red squares show the required input port and output PD for a given node’s calibration; and yellow regions are where there is an unknown irrelevant amount of light.

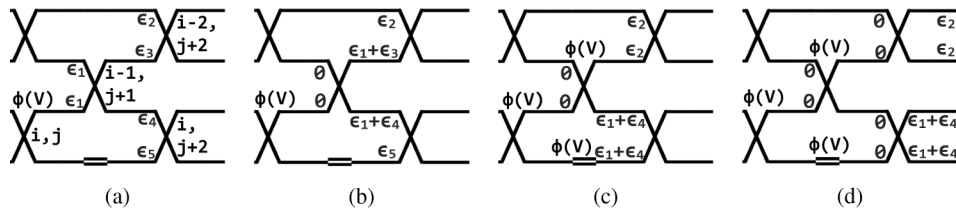


Fig. 3. An explanation of external phase-shifter calibration. Notation is as in Fig. 1. Phases are labelled at MZI inputs. ϵ values are random phase errors. $\phi(V)$ is found such that $\phi = 0$ occurs when the next MZI’s inputs are in phase. Phase-shifters in the top row are irrelevant in this figure. (a) First, find $\phi(V)$ for external phase-shifters in column j . Both input arms of the MZI $_{i-1,j+1}$ now have absolute phase ϵ_1 when $\phi = 0$. (b) Then, describe the mesh equivalently by factoring phases across MZIs in column $j + 1$. Column j is now calibrated. (c) Next, find $\phi(V)$ for external phase-shifters in column $j + 1$. (d) Then, describe the mesh equivalently by factoring phases across MZIs in column $j + 2$. Column $j + 1$ is now calibrated.

The external phase-shifters are calibrated progressively from the input to the output layers of the mesh. For each external phase-shifter $\phi_{i,j}$, one finds $\phi(V_\phi)$ such that $\phi = 0$ occurs when the inputs to the next MZI_{next} are in phase. For $\phi \neq 0$, $\phi(V_\phi)$ is determined by the extra phase created by a voltage. MZI_{next} refers to $MZI_{(i,j+1)}$ for odd i and $j = 0$; $MZI_{(i,j+2)}$ for $i = 1$ and $j \neq 0$; and $MZI_{(i-1,j+1)}$ otherwise (note that $\phi_{N,0}$ can be considered part of $\phi_{N,1}$ for odd N).

To find $\phi(V_\phi)$, we use the external phase-shifter ‘‘Co-sinusoidal Proportional Calibration’’ in [6,8]. First, all nodes are set to their bar states, $\theta = \pi$, so the power at the mesh outputs equals the power exiting the column being calibrated. Then, coherent light, not necessarily of the same amplitude, is shone into all mesh inputs. Then, the internal phase-shifter of MZI_{next} is set to $\theta = \pi/2$, effectively making MZI_{next} into a 50:50 beamsplitter.

Algorithm 1: Node Isolation Algorithm

Data: A rectangular mesh *Mesh* with N inputs and M columns; $M \leq N$
Result: The internal phase-shifters and beamsplitters of *Mesh* will be calibrated.

```

1  $iIn, iOut \leftarrow 1, \text{minimum}(N, M + 1)$ ;
2 for  $row = iOut - 1$  downTo 1 do // isolate main diagonal nodes (downTo includes 1)
3    $\text{isolateMZI}(row, row, iIn, iOut)$ ;
4 if  $M == N$  then  $iOut \leftarrow iOut - 1$ ;
5 else  $iOut \leftarrow iOut - 2$ ;
6 while  $iOut > 1$  do // iterate through each top diagonal
7    $col \leftarrow M$ ;
8   set all  $\theta_{i,j}$  where  $i = j$  to 0, and set all other  $\theta$  to  $\pi$ ;
9   for  $row = iOut - 1$  downTo 1 do // isolate each node on the current diagonal
10    set  $\theta_{row+1, row+1}$  to  $\pi$ ;
11     $\text{isolateMZI}(row, col, iIn, iOut)$ ;
12    set  $\theta_{row, col}$  and  $\theta_{row+1, row+1}$  to 0;
13     $col \leftarrow col - 1$ ;
14   $iOut \leftarrow iOut - 2$ ;
15  $iIn, row \leftarrow 3, 3$ ;
16 while  $iIn < N$  do // iterate through each bottom diagonal
17    $col \leftarrow 1$ ;
18   set all  $\theta$  to  $\pi$ ;
19   while  $row < N$  and  $col \leq M$  do // isolate each node on the current diagonal
20     $iOut \leftarrow row$ ;
21     $\text{isolateMZI}(row, col, iIn, iOut)$ ;
22    set  $\theta_{row, col}$  to 0;
23     $row, col \leftarrow row + 1, col + 1$ ;
24   $iIn, row \leftarrow iIn + 2, iIn + 2$ ;
// isolate a node to calibrate its internal phase and beamsplitters
25 Function  $\text{isolateMZI}(row, column, iIn, iOut)$  is
26   if  $MZI_{row, column}$  has adjustable beamsplitters then
27    Adjust the beamsplitters of  $MZI_{row, column}$  using the ‘‘Beamsplitter 50:50 setup algorithm’’ in Section 3.
    Light is shone into  $I_{iIn}$  and detected at  $D_{iOut}$ ;
28   Calibrate internal phase-shifter  $\theta_{row, column}$  using the ‘‘Co-sinusoidal proportional calibration’’ in Section 3.
    Light is shone into  $I_{iIn}$  and detected at  $D_{iOut}$ ;
29   Set  $MZI_{row, column}$  to its cross state ( $\theta = 0$ );

```

Now, one derives $\Delta\alpha$, the difference between the top and bottom input phases of MZI_{next} (α_t and α_b), as a function of voltage, by sweeping the phase $\phi_{i,j}$ and monitoring power in the PD on the top output row of MZI_{next} . The formula for this is derived in [6], with the P optical power variables with subscripts as in Eq. (2):

$$\Delta\alpha(V_\phi) = \arccos\left(\frac{2P(V_\phi) - P_{max} - P_{min}}{P_{max} - P_{min}}\right). \quad (3)$$

If $\phi_{i,j}$ leads to the bottom of MZI_{next} , then $\phi(V_\phi) = \alpha_b(V_\phi) - \alpha_t = -\Delta\alpha(V_\phi)$. If it is on the top, then the phase relation is $\phi(V_\phi) = \alpha_t(V_\phi) - \alpha_b = \Delta\alpha(V_\phi)$. The constants are phase errors on the arm with no phase-shifter. $\phi(V_\phi)$ describes “actual” $\text{MZI}_{i,j}$ ’s mathematically equivalent “ideal” MZI by factoring out the constant phase errors to the end of MZI_{next} , where they are absorbed into the next column’s calibration. Since ϕ phase-shifters have range $[0, 2\pi)$ but arccos outputs $[0, \pi]$, two voltage sweeps are needed for each external phase-shifter [6].

In short, the summary of external phase calibration, as shown in Fig. 4, is:

- (1) After calibrating the internal phase-shifters, set all internal phase-shifts to bar state ($\theta = \pi$). Shine coherent light into each mesh input, not necessarily identical amplitudes or phases.
- (2) For each column $j = 1$ through M :
 - (a) Set all internal phase-shifts in column j to $\theta = \pi/2$.
 - (b) For each row $i = 1$ through N , find $\phi(V_\phi)$, as described above, for the phase-shifter on the bottom input of $\text{MZI}_{(i,j)}$, if such MZI and phase-shifter exist. Use PD D_i .
 - (c) Set all the phase-shifters calibrated in Step 2b to $\phi = 0$.
 - (d) For each row $i = 1$ through N , find $\phi(V_\phi)$, as described above, for the phase-shifter on the top input of $\text{MZI}_{(i,j)}$, if such MZI and phase-shifter exist. Use PD D_i .
 - (e) Set all the phase-shifters calibrated in Step 2d to $\phi = 0$.
 - (f) Reset all internal phase-shifts to bar state ($\theta = \pi$).

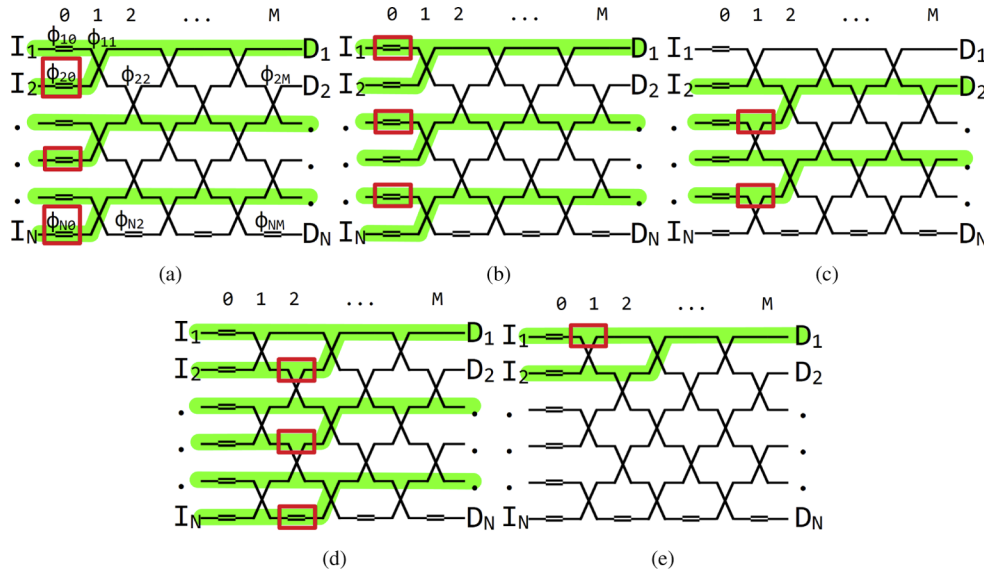


Fig. 4. The first external phase calibration steps: (a) is step 2b for column 1, (b) is step 2d for column 1, (c) is step 2b for column 2, (d) is step 2b for column 3, and (e) is step 2d for column 3. Notation is as in Fig. 1. Green highlight shows the light path and red squares encircle external phase-shifters being calibrated.

In our mesh architecture, external phase-shifters on row N (except in column 0) are set to $\phi = 0$ after calibration. Once calibrated, the unitary transformation of the mesh can be programmed, up to unknown output phases. If output phases are important, features would have to be added to the circuit to enable their measurement and calibration. The algorithm time complexity is $O(NM)$ if calibrating phase-shifters sequentially or $O(M)$ if calibrating each row of a column in parallel.

4. Simulations

Simulations are performed to demonstrate application of this calibration approach using only external PDs. The open-source mesh simulation Python module *Neurophox* [19] is used, with our modified version available in [20]. Simulations in Fig. 5 show a 10×10 mesh being calibrated. The mesh has phases initialized in a Haar-random manner [19], implementing a target matrix, T ; then, uniform random constant phase errors in $[0, 2\pi)$ are added to all external waveguides and internal phase-shifters. Then the calibration is simulated to find “correction constants” that are added to the phase-shifter phases to correct for the phase errors.

When phase errors and calibrated correction constants are added to the original Haar-random phases, the resulting mesh transfer matrix is U . The relative error metric for a given row and column x, y of the transfer matrix is

$$\xi_{x,y} = \left| \frac{T_{x,y} - U_{x,y}}{T_{x,y}} \right|. \quad (4)$$

To account for the phase-shifts at the mesh outputs, the rows of U are normalized before calculating error by multiplying each element of row x by $\exp(i(\arg[T_{x,1}] - \arg[U_{x,1}]))$, where x is any row.

For simplicity, the simulations assume ideal beamsplitters and consider constant phase errors instead of nonlinear phase-voltage relationships. Instead of using the co-sinusoidal proportional calibration, a simple single-parameter gradient ascent of light amplitude (with respect to phase) is used to find the optimal constant phase correction at each step of the calibration. Finally, column 0 phase errors are not considered, but adding them is trivial and like any other column.

We find that after one iteration of the loop calibrating internal phase-shifters of the main diagonal nodes $\{\text{MZI}_{i,i} | i \in \mathbb{Z} > 0\}$, the phases do not always converge. This occurs when phase errors start close to a bar state, lowering PD power and slowing the gradient ascent.

We overcome this using two methods, enabling calibration of extreme initial internal phase errors like π in all nodes. First, if the PD power is below a low threshold, we do a coarse gradient ascent over all phase-shifts on the main diagonal for a few iterations until enough power is detected in the PD. This threshold is rarely reached, and does not have a tangible impact on the calibration speed. Second, we calibrate the main diagonal three times. This ensures the diagonal is well calibrated, which is important for later calibration steps dependent on the main diagonal.

For a mesh with $N = M = 10$, 300 iterations per internal phase-shifter gradient ascent (directly adding the gradient to the phase during each iteration) works well. The external phase calibration achieves good convergence with 600 iterations per phase-shifter gradient ascent. After calibration, the average percent relative error across all elements of the calibrated matrix U is quite low at $0.38\% \pm 0.02\%$ (where uncertainty is the standard error) averaged over 20 independent simulations with the same Haar-random target matrix but different uniform random initial phase errors. Figure 5 shows a simulation with average error of 0.39%. Figure 6 demonstrates light propagation in the mesh during calibration.

Additional simulations varying the mesh sizes and initial mesh state were also carried out to verify the algorithm. All code, data, and simulations are available at [20]. We tested simulations for meshes with different permutations of even and odd for M and N , for M and N up to 32, all of which converged well. We also tested a very suboptimal initialization with all internal phase shifters having maximal π phase errors. With gradient ascent iterations and the target matrix kept constant, this initialization barely increased percent error to $0.43\% \pm 0.02\%$ for $N = M = 10$ (over 20 simulations). For $N = M = 32$, such initialization increased percent error just slightly from $1.3\% \pm 0.2\%$ to $1.6\% \pm 0.1\%$ (over 4 simulations). These results indicate that the algorithm is robust against large errors and scales to large meshes.

Although there is no limit to how finely one can characterize each MZI, we terminate the gradient ascent with these accuracy levels because the remaining phase errors are on the order of

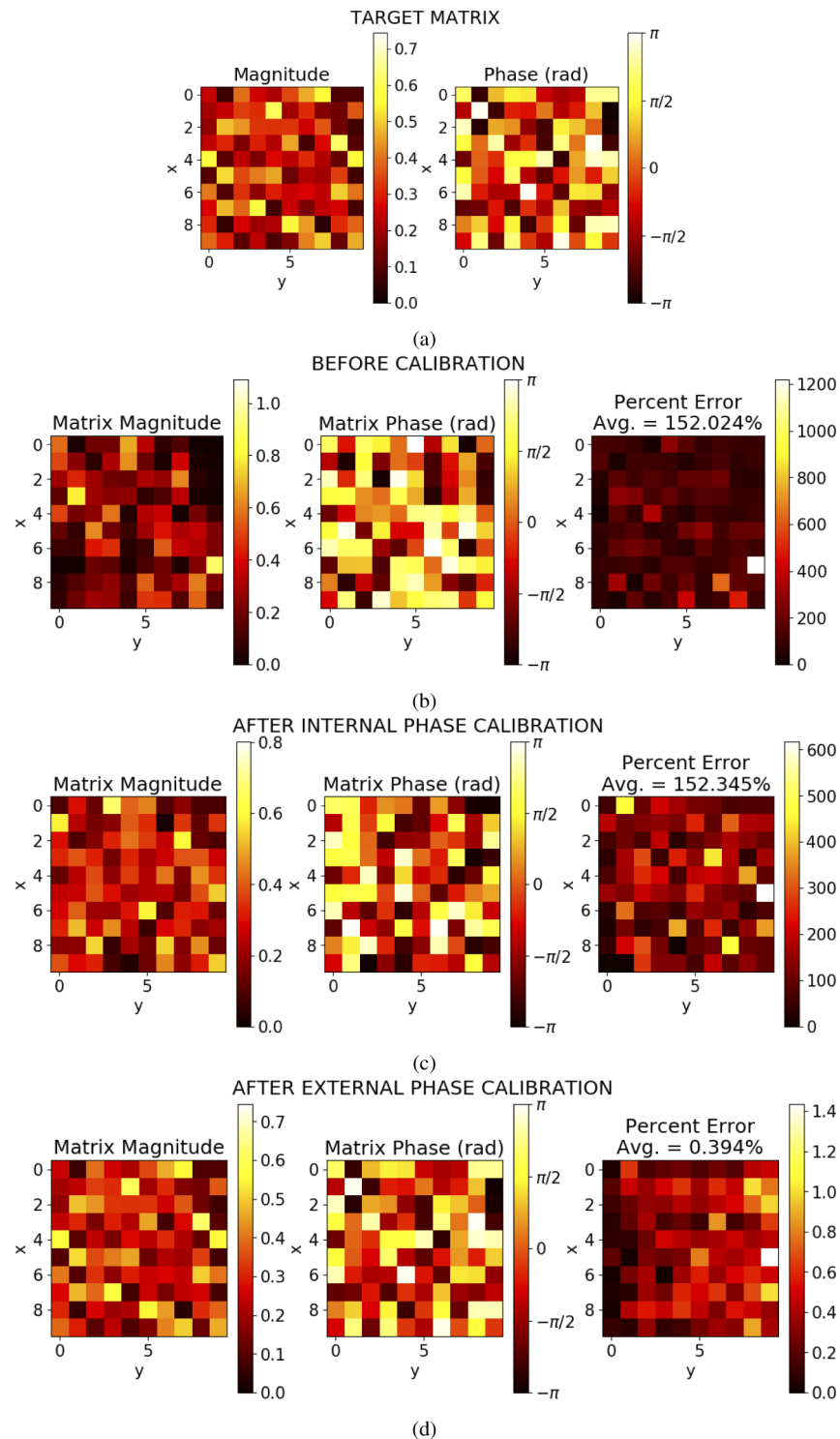


Fig. 5. (a) The target transfer matrix T (amplitude in arbitrary units). Then, matrix and percent relative error (from Eq. (4)) (b) before calibration, (c) after internal phase calibration, and (d) after external phase calibration. Phases are normalized as in Section 4., and x and y are as described in that section.

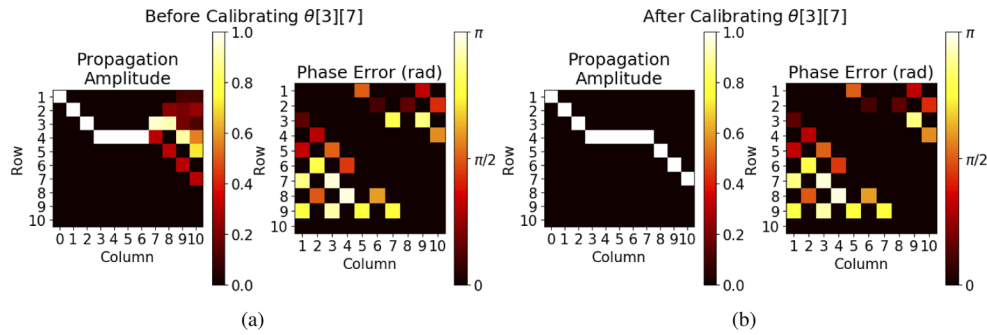


Fig. 6. Calibrating internal phase-shifter $\theta_{3,7}$. (a) shows the light propagation amplitudes (left) and phase errors (right) before calibration. (b) shows these data after calibration. Amplitudes (in arbitrary units) are measured at the output of the given column on the given row. Phase errors are differences between the corrected θ and the desired θ (i.e., cross state where $\theta = 0, 2\pi$, etc.). After calibration, $\theta_{3,7}$'s phase error is near 0 and light is maximized at output port 7. Other images can be found with the rest of our simulation code.

10^{-3} rad, which would have little impact on the accuracy of a neural network running on the mesh [5]. Furthermore, these percent errors are less than what would be expected from other inaccurate circuit components like beamsplitters [21], and are less than percent errors calculated from final error magnitudes in the mesh-optimizing simulations of [19].

5. Discussion

We have shown how to calibrate meshes with $M \leq N$, which can perform $N \times N$ unitary transformations; however, if $M > N$, one can add a column of PDs every N columns of the mesh to calibrate N columns at a time. Multi-mesh circuits like the universal linear network in [1], which implements general non-unitary matrices, can have PDs after each section for calibration. To ensure nodes are well isolated, the inputs can also be monitored so that light only enters a single input port for a given calibration step.

In simulations, two challenges we find with the algorithm are (1) the detected light levels at the PDs during calibration of the main diagonal can be low if the uncalibrated nodes are not close enough to the cross state, and (2) the calibration of nodes is dependent on how well earlier nodes were calibrated.

Point (1) is a bigger concern if gradient ascent is used instead of “co-proportional sinusoidal calibration,” because low detected optical powers slow convergence. As described in Section 4., we overcome this by doing a coarse gradient ascent over the entire main diagonal to increase PD power before calibrating the nodes, and by calibrating the main diagonal more than once. Even if MZI characterization algorithms other than gradient ascent are used, the coarse gradient ascent and repeating the calibration may generally improve the algorithm robustness. Point (1) is usually not an issue on other diagonals, where the transmission depends on already characterized nodes.

In our simulations, Point (2) limits the calibration accuracy, as we only run a finite number of gradient ascent steps. In a real mesh, when “co-proportional sinusoidal calibration” is used, the step size in the power sweep should be minimized to characterize the MZI as well as possible. Inaccurate beamsplitter ratios also limit how well MZIs can achieve perfect cross states [19]; thus, tunable beamsplitters could be used if accurate beamsplitters cannot be manufactured. Other error sources, such as PD noise and analog-digital conversion limitations, can also impact MZI characterization accuracy. Finally, error caused by phase-dependent phase-shifter losses can be minimized by using phase-shifters with less loss, such as MEMS, heaters, or novel designs [1].

Algorithms similar to the Node Isolation Algorithm have successfully calibrated large 32×32 switching meshes [18], showing that our procedure has the promise to overcome these challenges and scale up to large Clements meshes. Further analyzing and minimizing the effects of the above error sources is a focus of future research. For example, having PDs at intermediate columns of very large meshes, or at nodes along the main diagonal, could improve algorithm robustness by making the isolation of a given node less dependent on poorly calibrated nodes.

6. Conclusion

We have presented a method to calibrate rectangular Clements meshes with MZIs as nodes using only $O(N)$ external photodetectors and without precise phase control of input light. First, nodes' splitting ratios (set by MZIs' internal phase-shifters) are progressively calibrated by creating paths through the mesh that isolate nodes such that light only enters the node at one of its inputs and only light exiting one of the node's outputs is detected. Then, phase relations between nodes (set by MZIs' external phase-shifters) are calibrated by sequentially forming interferometers within the mesh. The work here enables practical control of the compact rectangular Clements mesh using minimal peripheral hardware, making large-scale MZI meshes more viable.

Appendix A: proof of node isolation algorithm

Here we prove the Node Isolation Algorithm works with external PDs for $M \leq N$.

First, we define an MZI to be "isolated" when light can only enter one of its ports (Condition 1) and the detector being used only measures light coming out of one of its ports (Condition 2).

We define diagonals as follows: diagonal 0 refers to $\{\text{MZI}_{q,q}\}$ (the main diagonal), diagonal x refers to $\{\text{MZI}_{q,q+2x}\}$ (top diagonals), and diagonal $-x$ refers to $\{\text{MZI}_{q+2x,q}\}$ (bottom diagonals), for $x, q \in \mathbb{Z} > 0$. $q = 1$ for the first MZI in a diagonal.

We now show that every MZI in each diagonal of the mesh, and thus every MZI, is calibrated: **MAIN DIAGONAL:** Every MZI on diagonal 0 can be isolated and internal phase-shifts can be calibrated. Condition 1 is met since light from I_1 only reaches the top input ports of the diagonal's MZIs. Condition 2 is met since light reaching PD $D_{\text{minimum}(N,M+1)}$, connected to the diagonal's last MZI (since $M \leq N$), must go through the bottom output port of each MZI in the diagonal.

TOP DIAGONALS: Loop invariant proof based on the first while loop in pseudo-code 1, which iterates through all top diagonals x starting from diagonal 1:

Loop Invariant: At the start of the loop, all MZIs of all diagonals from 0 through $x - 1$ have internal phase-shifts calibrated.

Initialization: Diagonal 0 internal phase-shifts are calibrated before the first loop iteration.

Maintenance: In each iteration, one loops through from the last through the first MZIs in diagonal x , and calibrates them all. This is shown by the following loop invariant proof:

Loop invariant: MZIs in diagonal x with $q > q_A$ have calibrated internal phase-shifters, where q_A is the value of q for MZI_A , the MZI being calibrated in a given step of the loop.

Initialization: There are no MZIs with $q > q_A$ when MZI_A is the last MZI of diagonal x .

Maintenance: MZI_A can be isolated and its internal phase-shifter calibrated, so the loop invariant holds for the next iteration. Condition 1 is met by shining light into input I_1 and programming diagonals 0 through $x - 1$ into bar and cross states to guide all light into one input of MZI_A . Condition 2 is met by setting MZIs on diagonal x with $q > q_A$ to cross state, so light reaching the PD at the end of diagonal x comes from the bottom of MZI_A .

Termination: The last iteration is with $q = 1$. By the maintenance property, all MZIs ($q \geq 1$) thus get calibrated internal phase-shifters.

Termination: The loop terminates after the topmost top diagonal is calibrated. All MZIs in each diagonal have calibrated internal phase-shifts, as per the maintenance property.

BOTTOM DIAGONALS: Loop invariant proof based on the second while loop in pseudo-code 1, which iterates through all bottom diagonals $-x$ starting from diagonal -1 :

Loop Invariant: At the start of the loop, all MZIs of all diagonals greater than $-x$ have internal phase-shifts calibrated.

Initialization: Internal phase-shifts of diagonals > -1 are calibrated before the first loop iteration.

Maintenance: In each iteration, one loops through from the first through the last MZIs in diagonal $-x$, and calibrates them all. This is shown by the following loop invariant proof:

Loop invariant: MZIs in diagonal x with $q < q_A$ have calibrated internal phase-shifters, where q_A is the value of q for MZI_A , the MZI being calibrated in a given step of the loop.

Initialization: There are no MZIs with $q < q_A$ when MZI_A is the first MZI of diagonal x .

Maintenance: MZI_A can be isolated and its internal phase-shifter calibrated, so the loop invariant holds for the next iteration. Condition 1 is met by setting MZIs on diagonal $-x$ with $q < q_A$ to cross state, so light put in the top of diagonal $-x$ goes into MZI_A 's top input. Condition 2 is met by programming already calibrated diagonals greater than $-x$ into bar states so light at a certain PD comes from only MZI_A 's top output.

Termination: The last iteration has $q_A = z$, where z is the number of MZIs in diagonal $-x$. By the maintenance property, all MZIs ($q \leq z$) thus get calibrated internal phase-shifters.

Termination: The loop terminates after the lowest bottom diagonal is calibrated, and all MZIs in each diagonal have calibrated internal phase-shifts, as per the maintenance property.

Acknowledgments. We thank Ankita Sharma and Zhengbang Zhou for insightful discussions.

Disclosures. The authors declare no conflicts of interest.

Data availability. Simulation code and output images and data are available at <https://edmond.mpg.de/imeji/collection/F0mLePnOuEYeTDb> [20].

References

1. N. C. Harris, J. Carolan, D. Bunandar, M. Prabhu, M. Hochberg, T. Baehr-Jones, M. L. Fanto, A. M. Smith, C. C. Tison, P. M. Alsing, and D. Englund, "Linear programmable nanophotonic processors," *Optica* **5**(12), 1623–1631 (2018).
2. D. A. B. Miller, "Setting up meshes of interferometers - reversed local light interference method," *Opt. Express* **25**(23), 29233–29248 (2017).
3. M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, "Experimental realization of any discrete unitary operator," *Phys. Rev. Lett.* **73**(1), 58–61 (1994).
4. W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walmsley, "Optimal design for universal multiport interferometers," *Optica* **3**(12), 1460–1465 (2016).
5. F. Shokraneh, S. Geoffroy-Gagnon, and O. Liboiron-Ladouceur, "Towards phase-error- and loss-tolerant programmable MZI-based optical processor for optical neural networks," (2020).
6. D. A. B. Miller, "Supplementary document for Analyzing and generating multimode optical fields using self-configuring networks - 4600959.pdf," (2020).
7. D. A. B. Miller, "Perfect optics with imperfect components," *Optica* **2**(8), 747–750 (2015).
8. D. A. B. Miller, "Analyzing and generating multimode optical fields using self-configuring networks," *Optica* **7**(7), 794–801 (2020).
9. A. Ribeiro, A. Ruocco, L. Vanacker, and W. Bogaerts, "Demonstration of a 4×4 -port universal linear circuit," *Optica* **3**(12), 1348–1357 (2016).
10. C. M. Wilkes, X. Qiang, J. Wang, R. Santagati, S. Paesani, X. Zhou, D. A. B. Miller, G. D. Marshall, M. G. Thompson, and J. L. O'Brien, "60dB high-extinction auto-configured mach-zehnder interferometer," *Opt. Lett.* **41**(22), 5318–5321 (2016).
11. A. Annoni, E. Guglielmi, M. Carminati, G. Ferrari, M. Sampietro, D. A. Miller, A. Melloni, and F. Morichetti, "Unscrambling light—automatically undoing strong mixing between modes," *Light: Sci. Appl.* **6**(12), e17110 (2017).
12. F. Shokraneh, M. S. Nezami, and O. Liboiron-Ladouceur, "Theoretical and experimental analysis of a 4×4 reconfigurable MZI-based linear optical processor," *J. Lightwave Technol.* **38**(6), 1258–1267 (2020).

13. D. A. B. Miller, "Supplement 1: Perfect optics with imperfect components," (2015), <https://doi.org/10.1364/OPTICA.2.000747.s001>.
14. S. Pai, I. A. D. Williamson, T. W. Hughes, M. Minkov, O. Solgaard, S. Fan, and D. A. B. Miller, "Parallel programming of an arbitrary feedforward photonic network," *IEEE J. Sel. Top. Quantum Electron.* **26**(5), 1–13 (2020).
15. R. Burgwal, "Integrated photonics for the realisation of multiport interferometers," Master's thesis, Universiteit Leiden (2018).
16. M. S. Hai, M. M. P. Fard, D. An, F. Gambini, S. Faralli, G. B. Preve, G. W. Roberts, and O. Liboiron-Ladouceur, "Automated characterization of SiP MZI-based switches," in *2015 IEEE Optical Interconnects Conference (OI)*, (2015), pp. 94–95.
17. Y. Huang, Q. Cheng, N. C. Abrams, J. Zhou, S. Rumley, and K. Bergman, "Automated calibration and characterization for scalable integrated optical switch fabrics without built-in power monitors," in *2017 European Conference on Optical Communication (ECOC)*, (2017), pp. 1–3.
18. S. Suda, H. Matsuura, K. Tanizawa, K. Suzuki, K. Ikeda, H. Kawashima, and S. Namiki, "Fast and accurate automatic calibration of a 32×32 silicon photonic strictly-non-blocking switch," in *Advanced Photonics 2017 (IPR, NOMA, Sensors, Networks, SPPCom, PS)*, (Optical Society of America, 2017), p. PTu3C.5.
19. S. Pai, B. Bartlett, O. Solgaard, and D. A. B. Miller, "Matrix optimization on universal unitary photonic devices," *Phys. Rev. Appl.* **11**(6), 064044 (2019).
20. C. Alexiev, *Simulations, data, and graphs for the paper "Calibrating Rectangular Interferometer Meshes with External Photodetectors"* Max Planck Society (2021). <https://doi.org/10.17617/3.70>.
21. R. Burgwal, W. R. Clements, D. H. Smith, J. C. Gates, W. S. Kolthammer, J. J. Renema, and I. A. Walmsley, "Using an imperfect photonic network to implement random unitaries," *Opt. Express* **25**(23), 28236–28245 (2017).