

Computing matrix inversion with optical networks

Kan Wu,¹ Cesare Soci,^{1,*} Perry Ping Shum,¹ and Nikolay I. Zheludev^{1,2}

¹Centre for Disruptive Photonic Technologies, Nanyang Technological University, 21 Nanyang Link, 637371, Singapore

²Optoelectronics Research Centre, University of Southampton, SO17 1BJ, UK

*CSoci@ntu.edu.sg

Abstract: With this paper we bring about a discussion on the computing potential of complex optical networks and provide experimental demonstration that an optical fiber network can be used as an analog processor to calculate matrix inversion. A 3x3 matrix is inverted as a proof-of-concept demonstration using a fiber network containing three nodes and operating at telecom wavelength. For an $N \times N$ matrix, the overall solving time (including setting time of the matrix elements and calculation time of inversion) scales as $O(N^2)$, whereas matrix inversion by most advanced computer algorithms requires $\sim O(N^{2.37})$ computational time. For well-conditioned matrices, the error of the inversion performed optically is found to be around 3%, limited by the accuracy of measurement equipment.

©2016 Optical Society of America

OCIS codes: (200.0200) Optics in computing; (200.4960) Parallel processing; (060.2310) Fiber optics.

References and links

1. H. J. Caulfield and S. Dolev, "Why future supercomputing requires optics," *Nat. Photonics* **4**(5), 261–263 (2010).
2. S. Dolev and H. Fitoussi, "Masking traveling beams: Optical solutions for Np-complete problems, trading space for time," *Theor. Comput. Sci.* **411**(6), 837–853 (2010).
3. M. Oltean and O. Muntean, "Solving Np-complete problems with delayed signals: An overview of current research directions," in *Optical Supercomputing*, S. Dolev, T. Haist, and M. Oltean, eds. (Springer, 2008), pp. 115–127.
4. K. Wu, J. García de Abajo, C. Soci, P. P. Shum, and N. I. Zheludev, "Fiber non-Turing all-optical computer for solving complex decision problems," in *Conference on Lasers and Electro-Optics / Europe (CLEO/Europe)*, (Munich, Germany, 2013), pp. CI-5.2.
5. J. L. O'Brien, "Optical quantum computing," *Science* **318**(5856), 1567–1570 (2007).
6. M. A. Broome, A. Fedrizzi, S. Rahimi-Keshari, J. Dove, S. Aaronson, T. C. Ralph, and A. G. White, "Photonic Boson sampling in a tunable circuit," *Science* **339**(6121), 794–798 (2013).
7. J. B. Spring, B. J. Metcalf, P. C. Humphreys, W. S. Kolthammer, X.-M. Jin, M. Barbieri, A. Datta, N. Thomas-Peter, N. K. Langford, D. Kundys, J. C. Gates, B. J. Smith, P. G. R. Smith, and I. A. Walmsley, "Boson sampling on a photonic chip," *Science* **339**(6121), 798–801 (2013).
8. D. Woods and T. J. Naughton, "Optical computing: photonic neural networks," *Nat. Phys.* **8**(4), 257–259 (2012).
9. L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, "Information processing using a single dynamical node as complex system," *Nat Commun.* **2**, 468 (2011).
10. Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," *Sci. Rep.* **2**, 287 (2012).
11. D. Petrov, Y. Shkuratov, and G. Videen, "Optimized matrix inversion technique for the T-matrix method," *Opt. Lett.* **32**(9), 1168–1170 (2007).
12. O. Arteaga and A. Canillas, "Analytic inversion of the Mueller-Jones polarization matrices for homogeneous media," *Opt. Lett.* **35**(4), 559–561 (2010).
13. V. V. Williams, "Multiplying matrices faster than Coppersmith-Winograd," in *44th Symposium on Theory of Computing* (ACM, 2012), 887–898.
14. A. J. Stothers, *On the Complexity of Matrix Multiplication* (University of Edinburgh, 2010).
15. H. Rajbenbach, Y. Fainman, and S. H. Lee, "Optical implementation of an iterative algorithm for matrix inversion," *Appl. Opt.* **26**(6), 1024–1031 (1987).

16. D. Casasent and J. S. Smokelin, "New algorithm for analog optical matrix inversion," *Appl. Opt.* **30**(23), 3281–3287 (1991).
17. Q. Cao and J. W. Goodman, "Coherent optical techniques for diagonalization and inversion of circulant matrices and circulant approximations to Toeplitz matrices," *Appl. Opt.* **23**(6), 803–811 (1984).
18. E. Barnard and D. Casasent, "Optical neural net for matrix inversion," *Appl. Opt.* **28**(13), 2499–2504 (1989).
19. R. Paschotta, "Noise of mode-locked lasers (part I): Numerical model," *Appl. Phys. B* **79**(2), 153–162 (2004).
20. Z. L. Sámson, P. Horak, K. F. MacDonald, and N. I. Zheludev, "Femtosecond surface plasmon pulse propagation," *Opt. Lett.* **36**(2), 250–252 (2011).

1. Introduction

Optical techniques have shown great potential in various computing areas including NP-complete problems [1–4], quantum [5–7] and reservoir computing [8–10]. The main advantage of optical systems resides in their inherent parallelism, which suggests the possibility to realize integrated high-speed parallel processors within complex optical networks. Here we are interested in a basic but very important mathematical problem: the matrix inversion. Calculation of matrix inversion is required in nearly all computational problems [11,12] and, for a $N \times N$ matrix, requires $\sim O(N^{2.37})$ solving time even with most advanced algorithms on a conventional computer [13,14]. Early work on matrix inversion by optical techniques has been reported with free space optical design and photorefractive amplifiers [15], and some algorithms have been discussed on this platform [16–18]. However, free space optical experiments have very strict requirements on alignment, collimation and detection of the optical signals, and allow very limited integration. All these factors limit experimental calculation accuracy to a level around 10% [15]. Meanwhile, fiber technology enables alignment free and ultra-low loss optical networks with great interconnection and design flexibility. Here we demonstrate the possibility of using an optical fiber network to calculate inverse matrices with error as small as 3%, limited by the accuracy of measurement equipment. The overall solving time scales as $O(N^2)$ including $O(N^2)$ setting time of the matrix elements and $O(N)$ calculation time of inversion. Besides the experimental demonstration, potential and limits of this approach are also discussed, including extension to complex matrix elements, calculation precision and accuracy, and scalability.

2. Experiment and results

A schematic diagram of an optical fiber network with three nodes is shown in Fig. 1(a). This network is built to map a 3×3 transfer matrix. Each node i ($i = 1, 2, 3$) has three inputs, one external (from outside the network), denoted as x_i , and two from other two nodes, denoted as y_j and y_k ($j, k = 1, 2, 3$). Each node i has also three equal outputs, denoted as y_i , one to the external output and two to other two nodes. The actual design of a node is shown in Fig. 1(b). Three 50:50 couplers are used to combine the input signals and split output signals. Attenuators can also be added to the input ports to adjust the transmission coefficients in each branch independently, corresponding to the set values of the input matrix elements. In addition, there are two ports denoted as monitor port and test port used to calibrate the transmission coefficients of the network. Note that, due to this specific configuration, signals experience 50% transmission loss when traveling through the 50:50 coupler. Therefore the external input and external output should be " $4x_i$ " and " $2y_i$ " so that the output to other nodes corresponds to " y_i ". For node i , the output y_i can be expressed by

$$y_i = x_i + m_{ij}y_j + m_{ik}y_k \quad (1)$$

where m_{ij} represents the transmission coefficient from node j to node i . For the ideal case where there is no additional loss in the network, and the couplers have exactly 50:50 coupling ratio, all m_{ij} 's have the same values of 0.125. We then express Eq. (1) in matrix form as

$$Y = X + MY \quad (2)$$

where $Y = \{y_1, y_2, y_3\}^T$, $X = \{x_1, x_2, x_3\}^T$ and

$$M = \begin{pmatrix} 0 & m_{12} & m_{13} \\ m_{21} & 0 & m_{23} \\ m_{31} & m_{32} & 0 \end{pmatrix} \quad (3)$$

Equation (2) can be reorganized as

$$Y = (I - M)^{-1} X \triangleq A^{-1} X \quad (4)$$

If one chooses the input vector X equal to $\{1, 0, 0\}^T$, the output vector Y will then represent the first column of the inverse matrix A^{-1} . Similarly, the other two columns of A^{-1} can be obtained choosing X equal to $\{0, 1, 0\}^T$ and $\{0, 0, 1\}^T$, respectively. Therefore the elements of the inverse matrix A^{-1} can be simply obtained from the network by properly choosing the input signals and measuring their output intensities. Although within this formalism the diagonal elements are zero and non-diagonal matrix elements must be real and satisfy the condition $0 < m_{ij} < 0.125$, any complex arbitrary element can also be implemented by considering the phase of the signals and refining the node design (see discussion in Section 3.2).

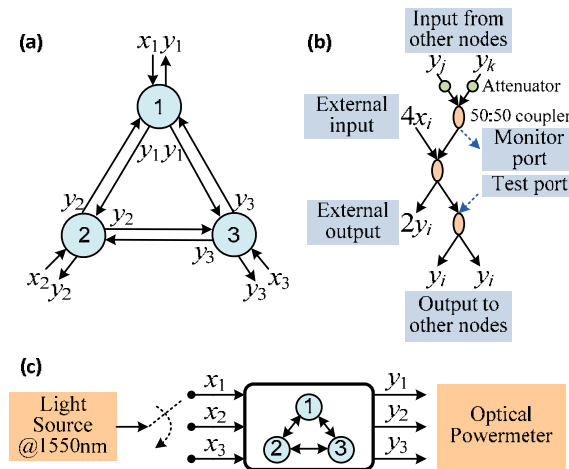


Fig. 1. (a) Schematic of an optical fiber network with three nodes where x_i and y_i ($i = 1, 2, 3$) are the input and output ports of each node; (b) Actual design of a node with optical fiber, couplers and attenuators and (c) Experimental setup of calculating matrix inversion.

This concept is experimentally demonstrated with the setup shown in Fig. 1(c). We first setup the network without any additional attenuation, and estimated the following matrix elements:

$$M = \begin{pmatrix} 0 & 0.121 & 0.122 \\ 0.120 & 0 & 0.121 \\ 0.0979 & 0.0912 & 0 \end{pmatrix} \quad (5)$$

These values are obtained calibrating the network transmission coefficients with the help of monitor and test ports in each node, and using pre-determined coupling ratios of the couplers. Note that m_{ij} are not exactly equal to 0.125 due to the losses and the non-ideal coupling ratio of 50:50 couplers. All matrix elements are expressed with three significant digits due to the accuracy of the power meter used in the measurements, and the uncertainty in the determination of the matrix elements is estimated to be $< 3\%$. In numerical analysis of matrix inversion, the sensitivity of the output values on the error of input matrix elements is

expressed by the so-called condition number, $\kappa = \|A^{-1}\| \cdot \|A\|$, where $\|\cdot\|$ represents the norm of the matrix (here we use $\|\cdot\|_2$). If the condition number is close to one, the matrix is said to be well conditioned and its inverse can be computed with good accuracy. On the other hand, if the condition number is large, then the matrix is said to be ill-conditioned and the computation of its inverse is prone to large numerical errors. The condition number of $A = I - M$ in Eq. (5) is $\kappa(I - M) = 1.45$.

To determine A^{-1} , a light beam near 1550 nm is injected into node 1 via the x_1 input, shown in Fig. 1(c). The light beam is generated by an amplified spontaneous emission (ASE) source. Use of a low coherent source avoids interference between different beams in the network, which would lead to output instability. The output power of the three nodes is then measured and normalized to the input power. With integration time of 100 ms the readings from the power meter did not change during the measurement, indicating effective averaging of the power fluctuations of the ASE source. The measured and calculated results of inverse matrix A^{-1} are:

$$A_{meas}^{-1} = \begin{pmatrix} 1.020 & 0.139 & 0.144 \\ 0.137 & 1.040 & 0.143 \\ 0.113 & 0.109 & 1.030 \end{pmatrix}, \quad A_{calc}^{-1} = \begin{pmatrix} 1.030 & 0.138 & 0.143 \\ 0.138 & 1.030 & 0.142 \\ 0.113 & 0.107 & 1.030 \end{pmatrix} \quad (6)$$

The error on the determination of inverse matrix, ΔA^{-1} , is defined by

$$(A + \Delta A)(A^{-1} + \Delta A^{-1}) = I \quad (7)$$

where ΔA is the error in the definition of matrix A . The relative error of inverse matrix, ε , is then given by

$$\varepsilon \triangleq \frac{\|\Delta A^{-1}\|}{\|A^{-1} + \Delta A^{-1}\|} \leq \|A^{-1}\| \cdot \|A\| \frac{\|\Delta A\|}{\|A\|} = \kappa \frac{\|\Delta A\|}{\|A\|} \quad (8)$$

where $\|\Delta A\|/\|A\|$ is the relative error of A . The inequality in Eq. (8) can be obtained by expanding the terms in Eq. (7) and calculating the matrix norms.

Due to the uncertainty in the determination of A , both measured and calculated inverse matrices are affected by error, and absolute values of inverse matrix A^{-1} are actually unknown. Assuming $\|\Delta A^{-1}\| \approx \|A_{meas}^{-1} - A_{calc}^{-1}\|$, the inverse matrix error ε is 0.96%, and the root mean square (rms) error between corresponding elements in the two matrices is 1.02%.

To further confirm the universal validity of our approach to calculate matrix inversion we modified the matrix elements attenuating the transmission from node 3 to node 1 and from node 2 to node 3. The new corresponding matrix elements m_{13} and m_{32} are given below:

$$M' = \begin{pmatrix} 0 & 0.121 & 0.0763 \\ 0.120 & 0 & 0.121 \\ 0.0979 & 0.0345 & 0 \end{pmatrix} \quad (9)$$

The measured and calculated elements of the inverse of $B = I - M'$ are given by:

$$B_{meas}^{-1} = \begin{pmatrix} 1.020 & 0.128 & 0.092 \\ 0.135 & 1.030 & 0.135 \\ 0.104 & 0.048 & 1.020 \end{pmatrix}, \quad B_{calc}^{-1} = \begin{pmatrix} 1.030 & 0.127 & 0.094 \\ 0.136 & 1.020 & 0.134 \\ 0.105 & 0.048 & 1.010 \end{pmatrix} \quad (10)$$

In this case $\kappa(B) = 1.39$, the inverse matrix error ε is 0.76% and the rms relative error between measured and calculated matrix elements is 0.96%. These values are consistent with

the upper boundary of 3% due to the uncertainty on the determination of matrix elements given by the accuracy of the power meter. Scaling of the error with matrix size will be discussed in the following section.

3. Discussion

3.1 Scalability

To evaluate the accuracy of our optical approach in the case of general matrices, we simulated the inversion of 1000 matrices with random elements a_{ij} . To reproduce the experimental conditions achievable with our optical network, we assumed $-0.125 < a_{ij} < 0$ and $a_{ii} = 1$ for $i, j = 1, 2, 3$. Moreover, to mimic the experimental uncertainty in the determination of the matrix elements ($< 3\%$), a uniformly distributed random error from -3% to 3% was added to each element a_{ij} . The distributions of the original matrix error $\|\Delta A\|/\|A\|$, condition number κ , and inverse matrix error ε are summarized in Fig. 2. With these constraints on the values of the matrix elements, condition numbers are mainly limited between 1 and 1.4, while the inverse matrix errors are mainly below 3%. This is consistent with the experimental results reported previously.

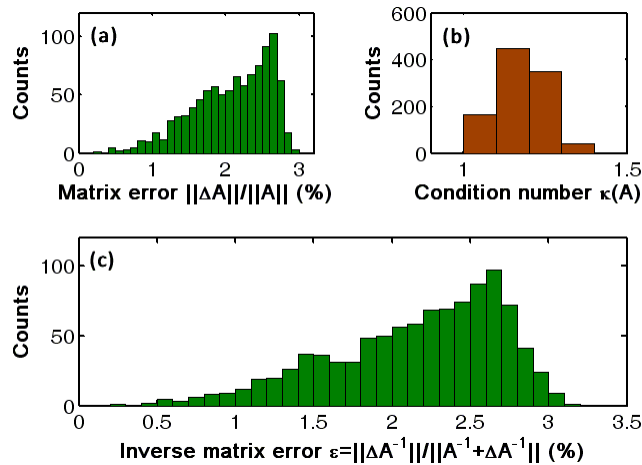


Fig. 2. Simulation results obtained for the inversion of 1000 3×3 matrices $A = I - M$ with random elements ($-0.125 < a_{ij} < 0$ and $a_{ii} = 1$ for $i, j = 1, 2, 3$) and $\sim 3\%$ error in the elements. (a) Matrix error and (b) condition number of matrix A ; (c) Inverse matrix error.

To investigate the scalability of the method, optical inversion of matrices with different sizes ranging from 3×3 to 100×100 was also simulated. 1000 simulations are performed for each matrix size with arbitrary (random) matrix elements ($|a_{ij}| \leq 1$) and 1% rms error in a_{ij} . (i.e., uniform error distribution from $-\sqrt{3} \cdot 1\%$ to $\sqrt{3} \cdot 1\%$) The calculated distributions of condition numbers and inverse matrix errors are shown in Figs. 3(a) and 3(b). With the increase of matrix size, both condition numbers and inverse matrix error increase quickly, and for the large 100×100 matrices considered, 1% initial error already leads to significant errors in matrix inversion. If, for the sake of argument, we define a nominal inverse matrix error ε_0 where most inverse matrix errors distribute within the interval of $\varepsilon_0 \cdot [10^{-0.5}, 10^{0.5}]$, one can estimate the dependence of nominal error on matrix size (Fig. 3(c)). In the case of initial matrix error of 1%, a linear fit to the nominal errors extracted from the simulations gives $\varepsilon_0 \approx 10^{-2.1} N^{0.59}$. Therefore, for $N \times N$ matrices, the nominal error of the inverse matrix scales roughly proportionally to $N^{0.6}$.

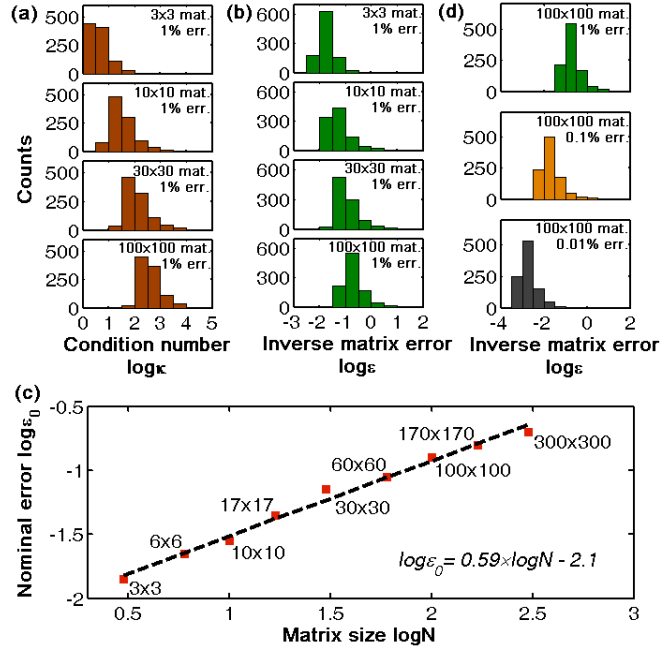


Fig. 3. Calculated distributions of (a) condition numbers and (b) inverse matrix errors for different matrix sizes ranging from 3x3 to 100x100 and 1% initial (rms) error in matrix elements a_{ij} ; (c) nominal error ε_0 with respect to the matrix size. The solid squares are the values extracted from the simulation results in Fig. 3(b) with initial matrix error of 1%. The dashed line is a linear fit to the data according to $\log \varepsilon_0 \approx 0.59 \log N - 2.1$; (d) inverse matrix errors for different initial (rms) errors ranging from 1% to 0.01% in matrix elements of 100x100 matrices. 1000 simulations were performed for each case.

Distributions of inverse matrix errors at different initial errors in the matrix elements a_{ij} are also calculated for large 100x100 matrices, shown in Fig. 3(d). An initial error of 0.1% can already guarantee inverse matrix errors smaller than 10% ($\log \varepsilon < -1$) with likelihood $>90\%$. An initial error of 0.01% can further improve the inverse matrix error to less than 1% ($\log \varepsilon < -2$) with likelihood $>90\%$. This nearly linear relationship between the initial matrix error and inverse matrix error is consistent with the inequality in Eq. (8). Accuracy of $\sim 0.1\%$ error in setting matrix element values by changing the network transmission coefficients is indeed within the capability of current optical technologies.

3.2 Generalization of node design

We provide a general design of the node that allows inversion of matrices with non-zero diagonal elements and complex terms: non-zero diagonal terms are implemented by a self-feedback loop while complex elements are realized by considering amplitude and phase of the propagating signals. This could be done, for instance, on a nanophotonics platform using single-frequency light source and phase sensitive detection.

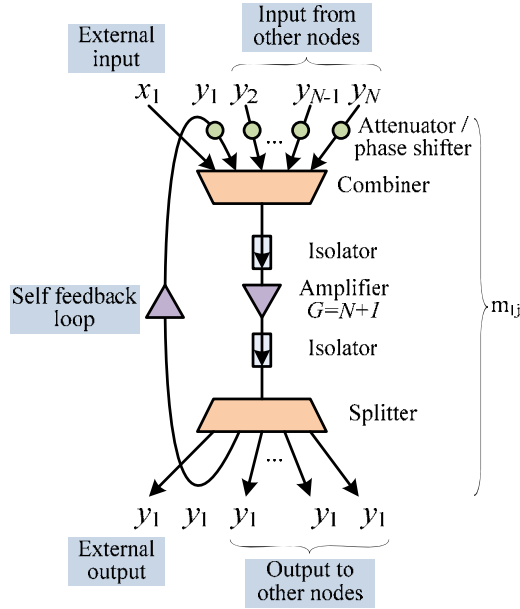


Fig. 4. General node design for $N \times N$ complex arbitrary matrices. Input signals are merged in a combiner after propagating through attenuators and phase shifters. The signals are then amplified to compensate for losses and split to generate the outputs. In the example shown for node 1, one of the outputs is fed back to the input corresponding to the diagonal element m_{11} .

Using node 1 as an example (Fig. 4), the external input to the node and inputs from the other nodes are merged by a combiner after propagating through individual attenuators and phase shifters. The combined signal is then amplified and split to generate one external output, one self feedback output, and $N-1$ outputs to other nodes. The main amplifier has a gain of $N + 1$ to compensate the loss induced by the splitter. For simplicity, we assume that the combiner and isolators are ideal and lossless. The output y_1 can then be expressed by

$$y_1 = x_1 + \sum_{j=1}^N m_{1j} y_j \quad (11)$$

where m_{1j} is the transmission coefficient from node j ($j = 1 \sim N$) to node 1. Thanks to the use of the amplifier and the self feedback loop, m_{1j} can now be any arbitrary number within the dynamic range of the optical setup. Equation (11) holds in the case of any other node i , with proper substitution of subscript 1 by i .

Implementation of complex matrix elements requires use of phase information. In our experimental demonstration based on fiber network, phase information cannot be maintained in the network due to the large length fluctuation of optical fibers, thus a wideband ASE source was used to avoid unstable interference in the network. However, in a platform with much smaller characteristic lengths, such as silicon photonic networks with a coherent and single-frequency light source, optical phase can be well controlled and maintained. To realize a node design such as the one in Fig. 4 in the silicon photonic platform, combiners and splitters can be realized using a Y-shape waveguide and its splitting ratio can be precisely controlled by adding micro-heaters near the waveguide. Tunable attenuators and tunable phase shifters can be realized using micro-electro-mechanical systems (MEMS). Optical amplifier requires semiconductor quantum well design, but this technology is also mature. Free space coupling between isolators, silicon photonic chips and semiconductor optical amplifiers may also be required.

Note that a network design based on the proposed nodes is robust against oscillations. The only condition for the network to oscillate would be with all signals propagating “in phase”, to make the oscillation self-consistent in the time domain. This is fulfilled only if the determinant of the matrix A is zero (A is singular). The convergence property and solving time of our approach are discussed in the following section.

3.3 Convergence and solving time

One key question is whether and when the output of the network will reach a steady state once an input is applied. To understand this, we expand Eq. (4) into the following expression [15]:

$$Y = (I - M)^{-1} X = \sum_{k=0}^{\infty} M^k X \quad (12)$$

Equation (12) converges if $M^k \rightarrow 0$ elementwise when $k \rightarrow +\infty$. This requires $|\lambda| < 1$ where λ 's are the eigenvalues of M . The requirement can always be satisfied by scaling the matrix M with its dominant eigenvalue λ_{\max} . For instance, the matrix can be scaled as $0.9M/\lambda_{\max}$ so that the dominant eigenvalue becomes 0.9.

If a reconfigurable optical network had to be implemented for optical computing of matrix inversion, it would take $O(N^2)$ time to set the N^2 matrix elements to the network transmission coefficients in each of the nodes. Therefore the overall solving time would consist of the network setting time (typical MEMS reconfigurable switches can operate with switching time of ~ 1 ms) and the actual calculation time required for matrix inversion. The calculation time of matrix inversion is given by the calculation time of a column of the inverse matrix times the number of columns (N). To evaluate the calculation time of a single column, one can treat Eq. (12) as an iterative process: when the input X is applied to the network at $t = 0$, the output $Y = X$ if propagation time inside a node is neglected. At time $t = \Delta t$, where Δt equals to the propagation time of an edge in the network (all the edges in the network have the same length), the output Y becomes $X + MX$ (here M is already scaled). Then this $X + MX$ becomes the new “internal” input and at $t = 2\Delta t$, Y becomes $X + M(X + MX) = X + MX + M^2X$. This process is iteratively performed until the contribution from M^k is sufficiently small and below the required precision (e.g. 10^{-6} , if the input power of 1 mW is applied and the minimum measurable output power is 1 nW). When this happens, we consider that the output has reached the steady state. For the particular scaling stated above, this requires $0.9^k < 10^{-6}$, or $k > 120$, which means after time $k\Delta t$ all network outputs are stable. With typical values of Δt between ~ 10 ns (propagation time in 2-meters of optical fiber) and ~ 100 ps (1-cm Si waveguide in a silicon photonic network), the convergence time is of the order of $\sim 10^{-9}$ - 10^{-6} s – much shorter than the setting and measuring time. The solving time of a single column depends only on the required precision and on the signal propagation time along the network edges, and therefore is independent of the size of the matrix. So the calculation time of the inverse matrix (with N columns) scales as $O(N)$. This analysis shows that, in a noiseless optical network, the overall solving time of matrix inversion is limited by the setting time of the network transmission coefficients and scales as $O(N^2)$, whereas even the most advanced computer algorithms currently require a solving time $\sim O(N^{2.37})$ [13,14].

To confirm our analysis, we use optical pulses as the input to the network and monitor the network output dynamics to experimentally investigate the convergence of the output. A pulse with duration of 300 ns is first injected to node 1, shown by the black dashed line in Fig. 5(a). The measured outputs of node 1 and 2 are shown in Figs. 5(b) and 5(c), respectively. All the nodes are nearly equally spaced to each other, that is, are connected by ~ 7 m single mode fiber which corresponds to a propagation delay of ~ 35 ns. In the output waveform of node 1, the step-up after 70 ns from the input (~ 100 ns position in time axis) is contributed by the signal returns from node 2 and node 3 (routes $1 \rightarrow 2 \rightarrow 1$ and $1 \rightarrow 3 \rightarrow 1$). In the output waveform of node 2 two steps-up are visible: the first step-up originates from the signal coming from

node 1 via node 3 (route 1→3→2), corresponding to a time delay of 35 ns with respect to the rising edge of the pulse (travelling along the route of 1→2), while the second step-up derives from the signal traveling along routes 1→2→1→2 and 1→2→3→2, which corresponds to a time delay of 70 ns.

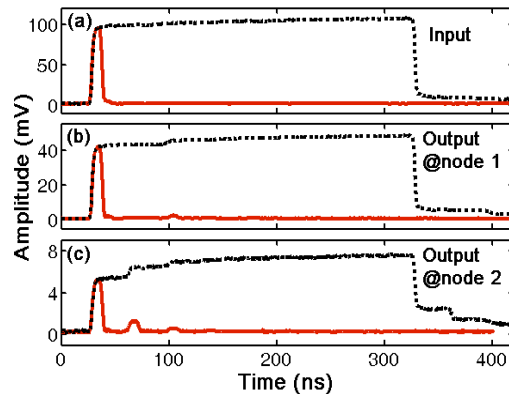


Fig. 5. Time-domain waveform of (a) input pulse, (b) output at node 1 and (c) output at node 2. The black dashed lines represent the waveforms generated by a 300 ns pulse and the red solid lines waveforms generated by an 8 ns pulse.

To further verify the origin of the steps-up, we use a short pulse with duration of 8 ns as the input to node 1, shown by the red solid line in Fig. 5(a). In the outputs measured at node 1 and node 2 shown in Figs. 5(b) and 5(c), one can clearly distinguish different pulses along the time axis. These pulses represent different routes reaching a certain node. Both the timing positions and amplitudes of these pulses match well with the waveform evolution of the node output measured with the long 300 ns pulse input. For example, in the output of node 2 shown in Fig. 5(c) the first pulse (red solid line) travels along the route 1→2. The next pulse travels along the route 1→3→2, and since it travels one more edge is delayed 35 ns with respect to the first pulse (its amplitude is also smaller than the first pulse due to the loss in node 3). This pulse has exactly the same timing position and amplitude of the first step-up in the waveform measured at node 2 output with the 300 ns pulse. The next pulse travels along the routes 1→2→1→2 and 1→2→3→2, accumulating a 70 ns delay with respect to the first pulse, and its amplitude is even smaller due to the losses occurred at each node. Additional pulses would follow at longer delays, but their amplitudes are below the detection limit – consequently the amplitude of the waveform generated by the 300 ns pulse reaches steady-state after approximately 150 ns on the time axis. Therefore the output of a certain node reaches a stable power level when all the upcoming signal amplitudes are smaller than the precision required by the calculation.

Here we shall emphasize that calculation precision refers to the number of significant (decimal) values of the measurement results, whereas calculation accuracy means how close the results are to the analytical values. For example, in the experiment presented in Eq. (6), the precision is 10^{-3} while the accuracy is $1-0.96\% = 99.04\%$. Obviously, longer calculation time is required to achieve higher precision and very careful calibration of the transmission coefficients of the whole network is required to achieve higher accuracy. Main sources of noise in the optical network are the shot noise from the input laser and of the output signals, the noise from photodetector and amplifiers, and the setting error of the transmission coefficients. Shot noise turns out to be negligible: the shot noise power spectral density is given by $2h\nu P$ W²/Hz [19] and for 1 kHz detection bandwidth and 1 mW input power at 1550 nm, the shot noise power of the input laser is ~ 0.5 nW (-63 dB smaller than the input power). Similarly, shot noise of the output signals is ~ 0.5 pW for a minimum measurable optical power of ~ 1 nW (-33 dB lower than the output signal). For the photodetector, typical

noise equivalent power (NEP) is of the order of $1.5 \cdot 10^{-15}$ W/ $\sqrt{\text{Hz}}$. For 1 kHz detection bandwidth (i.e. 1 ms integration time), $\text{NEP} = 1.5 \cdot 10^{-15} \cdot \sqrt{10^3} \text{ W} = 4.7 \cdot 10^{-14} \text{ W}$, which is also much smaller than the minimum measurable output power of 1 nW and thus will not affect the measurement. The setting error in the transmission coefficients (0.1% for state-of-the-art power meters) is the dominant source of noise in the matrix inversion calculation, and the corresponding inverse matrix error has $O(N^{0.6})$ dependence on the matrix size as discussed in Section 3.1. For large matrices, propagation losses and the use of amplifiers will also reduce signal to noise ratio, thus requiring longer integration time to achieve high accuracy.

3.4 Applications

A direct application of the optical method for matrix inversion is to use the network for the solution of linear equations. For example, by injecting certain amplitude values into input X , the output Y will correspond to the solution of the following linear equations

$$(I - M)Y = X \quad (13)$$

Another potential use of the network is for searching the matrix eigenvalues. If we set the amplifier in Fig. 4 to have gain $G = (N + 1)/\lambda$ for all nodes and there is no input, i.e. $X = 0$, Eq. (11) becomes

$$\lambda y_1 = \sum_{j=1}^N m_{1j} y_j; \quad (14)$$

in matrix form, this can be written as

$$\lambda Y = MY \quad (15)$$

or

$$(\lambda I - M)Y = 0. \quad (16)$$

If gain and phase of the amplifier are tuned to make the network oscillate, the determinant of the matrix $\lambda I - M$ must equal to zero, therefore the corresponding value of λ is the actual eigenvalue and the output Y the eigenvector of λ .

4. Conclusions

In conclusion, we proposed an analog optical processor realized with a simple optical fiber network to calculate matrix inversion. An $N \times N$ matrix can be presented by a network with N nodes where the matrix elements correspond to the transmission coefficients through the nodes of the network. The inherent parallelism of optical signals guarantees fast calculation time, which is proven to scale linearly with the size of the matrix (N) whereas the overall solving time is $O(N^2)$ due to the $O(N^2)$ setting time. A proof-of-principle demonstration of inversion of a 3×3 matrix is performed. For well-conditioned matrices, the calculation error can be as small as 3%, limited by the accuracy of measurement equipment. Moreover, it is shown that with 0.1% error in the determination of the initial matrix elements, our optical approach could potentially calculate the inverse of a 100×100 matrix with error smaller than 10% and >90% likelihood. This approach could be further extended to silicon photonics networks, that could be easily scaled to the size of larger matrices and any complex arbitrary matrix element by considering the optical phase of the signals, with immediate application in analogue computing of linear equations and potentially for the solution of eigenvalue problems. By exploiting the slow dispersion of plasmon polariton pulses [20] this strategy can, in-principle, be also deployed on a plasmonics waveguide networks with femtosecond lasers.

Acknowledgments

The authors are grateful to J. García de Abajo for initiation of the idea and Y. Fainman for the comments on condition numbers. This work is supported by the Singapore Ministry of Education Academic Research Fund Tier 3 (Grant MOE2011-T3-1-005), and EPSRC (UK) via the Programme on Nanostructured Photonic Metamaterials.