An Investigation of Parameter Scheduling for Image Restoration in Optical Analog Circuits

Taisei Kato*, Ryo Hayakawa†, Soma Furusawa‡, Kazunori Hayashi‡, and Youji Iiguni*
* Graduate School of Engineering Science, The University of Osaka, Osaka, Japan
† Institute of Engineering, Tokyo University of Agriculture and Technology, Tokyo, Japan
‡ Graduate School of Informatics, Kyoto University, Kyoto, Japan

***Abstract*—Optical analog circuits are promising devices for signal processing due to their potential for low latency and low power computations. However, implementing iterative algorithms on such circuits is challenging, especially due to additive noise from optical amplifiers. Our previous work investigated the feasibility of image restoration using total variation regularization under these constraints. In this study, we aim to further improve algorithm performance by learning algorithm parameters through a deep unfolding framework. Simulation results show that the learned parameters achieve comparable performance to that of manually tuned fixed parameters. This finding suggests that, given the implementation cost of dynamically varying parameters, using well-optimized fixed parameters may be more practical for image restoration on optical analog circuits.**

***Index Terms*—Optical analog circuits, image restoration, ADMM, PDS, deep unfolding**

I. Introduction

Optical devices have attracted attention as a technology that can perform vector-matrix products with low latency and low power consumption [1]. Unlike traditional electronic circuits that use electrons as information carriers, optical analog circuits operate using photons. As a result, they are expected to perform operations such as matrix-vector multiplication with lower latency and reduced power consumption compared to their electronic counterparts [2]–[6]. However, signal processing using optical analog circuits involves several implementation constraints. For instance, division by dynamically changing variables at each iteration is difficult to implement, and signal amplification inevitably introduces additive noise.

To address the challenges associated with optical analog circuits, the implementation of iterative signal processing algorithms has been discussed. In compressed sensing, for example, performance nearly equivalent to conventional methods has been achieved by replacing the dynamic division within iterative algorithms with an appropriate fixed value [7]. In addition, when evaluating performance under the effects of amplifier noise, it has been demonstrated that the performance remains comparable to the noise-free case [8]. For image restoration, it is also shown in [9] that images can be effectively denoised, even when noise from signal amplification is taken into account. In [9], iterative algorithms such as the alternating direction method of multipliers (ADMM) [10], [11] and primal dual splitting (PDS) [12], [13] are investigated as promising approaches for image restoration methods tailored to implementation in optical analog circuits.

The performance of many iterative algorithms can be improved by tuning their parameters at each iteration. To tackle this challenge, an approach known as deep unfolding has recently attracted attention [14], [15]. In this framework, each iteration of the algorithm corresponds to one layer of the neural network, and its parameters are treated as learnable weights. For example, it has been shown that parameter learning via deep unfolding approach improves the performance in compressed sensing [16].

In this paper, we investigate parameter scheduling based on the deep unfolding framework for image restoration in optical analog circuits. As the target algorithm, we consider ADMM and PDS with additive circuit noise discussed in [9]. In the parameter learning via deep unfolding, the parameters to be learned are allowed to vary across iterations to improve the restoration accuracy. Although modifying parameters at each iteration may be challenging in current optical hardware, it is still valuable to assess how such scheduling affects restoration accuracy when it is assumed to be feasible. Moreover, the insights obtained from this evaluation can serve as a guideline for the design and development of future optical signal processing devices.

We assess the effectiveness of the proposed approach via computer simulations. As a preliminary experiment, we first trained the algorithm parameters using a single image to verify the applicability of the deep-unfolding framework in our setting. The results demonstrate that the restoration accuracy is improved under the given condition. For the main evaluation, we assessed the restoration accuracy for unknown test images using parameters learned from multiple training images. However, the performance gains obtained through this learning process were limited. These results suggest that the optimal parameter values may vary from image to image. We also found that merely scheduling the parameters targeted in this study—such as step sizes and regularization coefficients for each iteration—is insufficient to achieve consistent improvements in restoration accuracy across different images.

II. Preliminaries and Related Work

A. Optical Analog Circuits and Their Constraints

Optical analog circuits, which exploit photons as information carriers, are currently being developed to execute operations such as matrix-vector multiplication with lower latency and reduced power consumption [1]. The components of

979-8-3315-7206-8/25/$31.00

TABLE I: Estimated additive noise power for amplifiers with different power gains [8]

Power Gain G	Additive Noise Power
8	1.79×10^{-8}
16	3.84×10^{-8}
32	7.94×10^{-8}
64	1.61×10^{-7}
128	3.25×10^{-7}
256	6.53×10^{-7}

these circuits include signal splitters (SSs), adders, subtractors, multipliers, attenuators, amplifiers, and delay elements. Among these, the SS, adder, and subtractor can be realized using a common optical device known as a beam splitter (BS) [8].

A BS has two input ports and two output ports. When a complex signal $\alpha \in \mathbb{C}$ is input to one port, the BS outputs are $\frac{1}{\sqrt{2}}\alpha$ from the same-side port and $\frac{1}{\sqrt{2}}\alpha \cdot e^{-j\pi/2}$ from the opposite-side port, where j is the imaginary unit. Because the amplitude is reduced by $1/\sqrt{2}$, each output carries half of the input power. An SS can be implemented by combining a BS with an appropriate phase shifter. Adders and subtractors are also realized by cascading a BS with phase shifters; in these operations, the output signal power is also halved.

Optical analog circuits are subject to structural constraints that differ from those of their electronic counterparts. A key challenge is the difficulty of implementing division by dynamic variables in iterative algorithms. Furthermore, to compensate for attenuation in components like adders and SSs, optical circuits require signal amplification, which inevitably introduces additive noise.

The amount of noise introduced by amplification can be estimated based on the amplification gain. The power spectral density of amplified spontaneous emission (ASE) noise generated by an erbium doped fiber amplifier (EDFA) is given by

$$G_{\mathrm{ASE}} = F(G-1)h\mu, \tag{1}$$

where F is the noise figure (NF), G is the power gain, h is Planck's constant, and μ is the frequency [17]. Assuming a typical optical system operating at a wavelength of $1550\,\mathrm{nm}$ and a signal bandwidth of $10\,\mathrm{GHz}$, the added ASE noise power [8] is

$$G_{\mathrm{ASE}} \times 10\,\mathrm{GHz} = (G-1) \cdot 2.56 \times 10^{-9}. \tag{2}$$

Table I summarizes the estimated additive noise power for different power gains G.

B. Image Restoration with Total Variation Regularization

Image restoration aims to estimate an original clear image $\boldsymbol{x}^* \in \mathbb{R}^N$ from degraded observation

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{e} \in \mathbb{R}^M, \tag{3}$$

where $\boldsymbol{A}$ is a degradation matrix and $\boldsymbol{e}$ is observation noise. A fundamental approach for image restoration is solving the total variation (TV) regularized optimization problem given by

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \left\{ \frac{1}{2}\left\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\right\|_2^2 + \lambda \left\|\boldsymbol{D}\boldsymbol{x}\right\|_{1,2} \right\}, \tag{4}$$

where the first term is the data fidelity term and the second is the TV regularization term with parameter $\lambda > 0$ [18]. TV is a measure of an image's smoothness and calculated as the sum of the differences between its adjacent pixels. The TV regularization term $\left\|\boldsymbol{D}\boldsymbol{x}\right\|_{1,2}$ is a mixed $\ell_{1,2}$-norm that promotes group sparsity in the image gradient, thereby suppressing noise while preserving sharp edges.

The optimization problem (4) can be solved with several proximal splitting algorithms such as ADMM [10], [11] and PDS [12], [13]. As derived in [9], the update equations of ADMM for (4) are given by

$$\boldsymbol{x}_{k+1} = \left(\boldsymbol{A}^\top\boldsymbol{A} + \frac{1}{\gamma}\boldsymbol{D}^\top\boldsymbol{D}\right)^{-1}\left(\boldsymbol{A}^\top\boldsymbol{y} + \frac{1}{\gamma}\boldsymbol{D}^\top(\boldsymbol{z}_k - \boldsymbol{v}_k)\right) \tag{5}$$

$$\boldsymbol{z}_{k+1} = \mathrm{prox}_{\gamma\lambda\|\cdot\|_{1,2}}(\boldsymbol{D}\boldsymbol{x}_{k+1} + \boldsymbol{v}_k) \tag{6}$$

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k + \boldsymbol{D}\boldsymbol{x}_{k+1} - \boldsymbol{z}_{k+1} \tag{7}$$

while the updates of PDS are expressed as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \gamma_1\left(\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{y}) + \boldsymbol{D}^\top\boldsymbol{v}_k\right) \tag{8}$$

$$\boldsymbol{z}_{k+1} = \boldsymbol{v}_k + \gamma_2\boldsymbol{D}(2\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \tag{9}$$

$$\boldsymbol{v}_{k+1} = \boldsymbol{z}_{k+1} - \mathrm{prox}_{\lambda\|\cdot\|_{1,2}}(\boldsymbol{z}_{k+1}). \tag{10}$$

Here, k denotes the iteration index. For a function g and a scalar $\gamma > 0$, the proximal operator of γg is defined as

$$\mathrm{prox}_{\gamma g}(\boldsymbol{x}) := \underset{\boldsymbol{u}\in\mathbb{R}^N}{\arg\min}\left\{ g(\boldsymbol{u}) + \frac{1}{2\gamma}\|\boldsymbol{x}-\boldsymbol{u}\|_2^2 \right\}. \tag{11}$$

The proximal operator of the mixed $\ell_{1,2}$-norm [19] is a group-wise scaled soft-thresholding function as

$$[\mathrm{prox}_{\gamma\lambda\|\cdot\|_{1,2}}(\boldsymbol{z})]_{\mathfrak{g}} = \max\left\{1 - \frac{\gamma\lambda}{\|\boldsymbol{z}_{\mathfrak{g}}\|_2}, 0\right\}\boldsymbol{z}_{\mathfrak{g}}, \tag{12}$$

where $[\mathrm{prox}_{\gamma\lambda\|\cdot\|_{1,2}}(\boldsymbol{z})]_{\mathfrak{g}}$ is the subvector of $\mathrm{prox}_{\gamma\lambda\|\cdot\|_{1,2}}(\boldsymbol{z})$ corresponding to group $\mathfrak{g}$. Here, the indeces of the vector are partitioned into N non-overlapping groups, which correspond to the horizontal and vertical differences at each pixel. The update equations of ADMM involve a matrix inversion in (5), while the equations of PDS do not. It should be noted that the proximal operators involve divisions that depend on variable values. In this paper, we assume that the proximal operators are computed accurately in electronic circuits and focus on the impact of circuit noise as in [9].

C. Image Restoration in Optical Analog Circuits

The main challenges in implementing image restoration on optical analog circuits are avoiding division by dynamic variables and evaluating the impact of circuit noise introduced by signal amplification. In ADMM, the matrix inversion required in the $\boldsymbol{x}$-update step can be pre-computed, and hence division by dynamic variables in the iterative computation can be avoided. PDS does not involve matrix inversion. Therefore,

Algorithm 1 ADMM considering circuit noise [9]

Require: $\boldsymbol{z}_0, \boldsymbol{v}_0, \gamma > 0, \lambda > 0$

while stopping criterion is not met **do**

$\boldsymbol{x}_{k+1} = (\boldsymbol{A}^\top \boldsymbol{A} + \frac{1}{\gamma}\boldsymbol{D}^\top \boldsymbol{D})^{-1}(\boldsymbol{A}^\top \boldsymbol{y} + \frac{1}{\gamma}\boldsymbol{D}^\top(\boldsymbol{z}_k - \boldsymbol{v}_k))$

$\boldsymbol{z}_{k+1} = \mathrm{prox}_{\gamma\lambda\|\cdot\|_{1,2}}(\boldsymbol{D}\boldsymbol{x}_{k+1} + \boldsymbol{v}_k + \boldsymbol{n}_k^{256})$

$\boldsymbol{v}_{k+1} = \boldsymbol{D}\boldsymbol{x}_{k+1} + \boldsymbol{v}_k + \boldsymbol{n}_k^{256} - \boldsymbol{z}_{k+1}$

$k \leftarrow k + 1$

end while

Ensure: $\boldsymbol{x}_k$

Algorithm 2 PDS considering circuit noise [9]

Require: $\boldsymbol{x}_0, \boldsymbol{v}_0, \gamma_1 > 0, \gamma_2 > 0, \lambda > 0$

while stopping criterion is not met **do**

$\boldsymbol{x}_{k+1} = (\boldsymbol{I} - \gamma_1 \boldsymbol{A}^\top \boldsymbol{A})(\boldsymbol{x}_k + \boldsymbol{n}_k^{32}) + \gamma_1 \boldsymbol{A}^\top \boldsymbol{y} - \gamma_1 \boldsymbol{D}^\top \boldsymbol{v}_k$

$\boldsymbol{z}_{k+1} = \boldsymbol{v}_k + \boldsymbol{n}_k^{16} + \gamma_2 \boldsymbol{D}(2\boldsymbol{x}_{k+1} + \boldsymbol{n}_k^2 - \boldsymbol{x}_k) + \boldsymbol{n}_k^{256}$

$\boldsymbol{v}_{k+1} = \boldsymbol{z}_{k+1} - \mathrm{prox}_{\lambda\|\cdot\|_{1,2}}(\boldsymbol{z}_{k+1})$

$k \leftarrow k + 1$

end while

Ensure: $\boldsymbol{x}_k$

the primary concern is the impact of additive noise introduced by amplifiers within optical analog circuits.

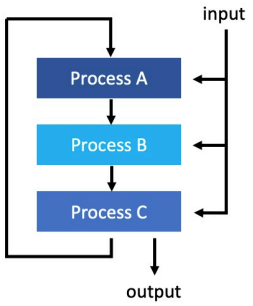
In the ADMM-based optical analog circuit configuration, an optical amplifier with a gain of 256 is required to compensate for signal attenuation (see [9] for details). This amplification adds noise $\boldsymbol{n}_k^{256} \in \mathbb{R}^{2N}$ at each iteration. Here, the superscript on the noise $\boldsymbol{n}_k$ denotes the gain of the amplifier. The signal after passing through the optical amplifier becomes $\boldsymbol{D}\boldsymbol{x}_{k+1} + \boldsymbol{v}_k + \boldsymbol{n}_k^{256}$. The algorithm incorporating this noise model is given as Algorithm 1. It should be noted that the update equation for $\boldsymbol{x}_k$ includes multiplication by $\frac{1}{\gamma}$. If $\frac{1}{\gamma}$ is greater than 1, i.e., $\gamma < 1$, the signal is amplified and, additive noise corresponding to the amplification factor of $\frac{1}{\gamma}$ is added.

In the PDS-based optical analog circuit configuration, multiple optical amplifiers with different gains are used (see [9] for details). Therefore, each time the signal passes through an optical amplifier with a different gain, additive noise vectors $\boldsymbol{n}_k^{256}, \boldsymbol{n}_k^{32}, \boldsymbol{n}_k^{16}, \boldsymbol{n}_k^2 \in \mathbb{R}^{2N}$ are added accordingly. The PDS-based image restoration algorithm that considers such circuit noise is shown in Algorithm 2. Note that the update equations for $\boldsymbol{x}_k$ and $\boldsymbol{z}_k$ include multiplications by γ_1 and γ_2, respectively. If γ_1 or γ_2 is greater than 1, the signal is amplified and the corresponsing noise is added.

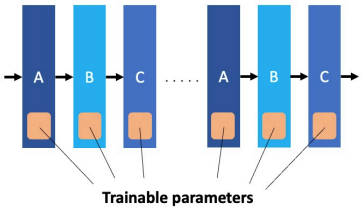
Simulation results in [9] demonstrate that both ADMM and PDS are capable of denoising images even in the presence of additive circuit noise. Furthermore, ADMM achieves better restoration accuracy than PDS, suggesting that ADMM is more robust to noise introduced by optical amplification.

III. Parameter Training via Deep Unfolding for ADMM and PDS in Optical Analog Circuits

The performance of image restoration algorithms heavily depends on the algorithm parameters. In particular, parameters that vary across iterations have the potential to improve restoration accuracy. However, identifying their optimal values remains a significant challenge. This chapter begins with an overview of deep unfolding [14], [15] and incremental training, followed by a detailed description of our study. Specifically we describe the application of deep unfolding to ADMM and PDS, with a focus on their implementation in optical analog circuits.



(a) Signal flow graph of an iterative algorithm.



(b) Signal flow graph unfolded over time.

Fig. 1: Conceptual diagram of signal flow graphs in deep unfolding.

A. Overview of Deep Unfolding

The performance of many iterative algorithms depends on their internal parameters. While tuning these parameters for each iteration can improve accuracy, finding their optimal values is often challenging. Deep unfolding is an approach developed to address this issue by learning these parameters directly from data.

Deep unfolding is an approach that interprets iterative algorithms, such as ADMM or PDS, as a feed-forward neural network by "unfolding" them along the time axis (i.e., the iteration axis). Fig. 1(a) shows the signal flow graph of a generic iterative algorithm, and Fig. 1(b) illustrates its unfolded counterpart. The sub-processes A, B, and C in Fig. 1(a) represent the computational operations within each iteration. In the unfolded signal flow graph in Fig. 1(b), each iteration corresponds to one layer of the network, and the parameters within the algorithm are treated as learnable weights. If the operations within these sub-processes are differentiable, these parameters can be learned using standard deep learning techniques, such as backpropagation and stochastic gradient descent. This enables the acquisition of algorithm parameters that are tailored to specific problems or datasets, thereby improving performance.

B. Incremental Training for Deep Unfolded Models

It is well known that deep neural networks are susceptible to vanishing and exploding gradient problems during training [20]. These issues arise during backpropagation, as gradients are propagated through multiple layers. The repeated multiplication of gradient terms can cause their magnitudes to shrink toward zero or grow uncontrollably, resulting in unstable learning dynamics.

In deep unfolding models, the number of algorithm iterations corresponds to the depth of the network. Consequently, attempting to train a model with a deep structure resulting from a large number of iterations presents a risk of gradient instability. To address this issue, an approach known as incremental training can be employed [16]. In this strategy, the depth of the network is gradually increased as the training progresses, starting from a shallow configuration. This helps to stabilize the learning process in its early stages and improve convergence.

C. Proposed Parameter Learning Method

This study applies deep unfolding to learn the parameters of ADMM and PDS, with a particular emphasis on their implementation on optical analog circuits. Given the physical constraints on optical analog circuits, we need to preserve the fundamental structures of these algorithms and assume circuit configurations that are physically feasible. To balance hardware feasibility with the benefits of deep unfolding, we restrict the learnable parameters to a small set, such as step sizes and regularization coefficients.

While the original algorithms assume fixed values for parameters such as γ and λ across iterations, we allow them to vary at each iteration to potentially improve performance. Specifically, the learnable parameters are given by (γ^k, λ^k) for ADMM and $(\gamma_1^k, \gamma_2^k, \lambda^k)$ for PDS, where k ($k = 0, 1, \dots$) denotes the iteration index. The network parameters are trained by minimizing the mean squared error (MSE) loss function $\mathcal{L}$. This loss is averaged over all images in the dataset and is defined as

$$\mathcal{L} = \frac{1}{SN} \sum_{s=1}^{S} \left\| \hat{\boldsymbol{x}}_s^{(K)} - \boldsymbol{x}_s^* \right\|_2^2 \tag{13}$$

where S is the total number of images in the dataset, N is the number of pixels in a single image, $\hat{\boldsymbol{x}}_s^{(K)}$ is the restored vector for the s-th image after K iterations, and $\boldsymbol{x}_s^*$ is the corresponding ground truth vector. To account for the characteristics of optical circuits, our model incorporates the effects of additive noise from amplifiers during both training and evaluation. The parameters are updated by minimizing the loss function using the Adam optimizer [21], with gradients computed via backpropagation through the unfolded iterations. Furthermore, we adopt the incremental training strategy to achieve effective and stable learning.

IV. Simulation Results

A. Simulation Setup

In this paper, we consider denoising problems for simplicity, where the observation matrix $\boldsymbol{A}$ is an identity matrix. We use 30 grayscale images of size 256×256 pixels as the original images. This dataset was partitioned into a training set of 10 images used to train the model, a validation set of 10 images for hyperparameter tuning (such as the learning rate), and a test set of 10 images for final performance evaluation. In light of current developments in optical analog circuits, which are expected to support computations in the range of tens to hundreds of dimensions, we divide each image into 256 non-overlapping patches of 16×16 pixels. The restoration algorithm is then applied to each patch individually.

The observed images are generated by adding Gaussian noise with a mean of 0 and a standard deviation of $10/255$ to the original images. The initial values for both ADMM and PDS are set to $\boldsymbol{x}_0 = \boldsymbol{y}$ (the observed image), $\boldsymbol{z}_0 = \boldsymbol{0}$, and $\boldsymbol{v}_0 = \boldsymbol{0}$. The additive noise from optical amplifiers is modeled as zero-mean Gaussian noise, corresponding to the gain of each amplifier as in TABLE I. Moreover, considering that the typical signal power in optical fiber transmission is approximately 0.001, the variance used in the simulation is set to be 1000 times larger than the values in TABLE I. The maximum number of iterations for ADMM and PDS is set to 30, and we employ the incremental training approach described in Section III-B for training. At each stage of the incremental training, the epoch size is set to one. The initial values for the learnable parameters were adopted from the fixed parameters that demonstrated the best performance in preliminary simulations. Specifically, we set $\gamma = 20, \lambda = 0.03$ for ADMM, and $\gamma_1 = 0.1, \gamma_2 = 5.0, \lambda = 0.03$ for PDS.

B. Simulation Results and Discussion

First, as a preliminary experiment, we learned the algorithm parameters for a single image and evaluated the restoration accuracy on that same image. In this case, the learned parameters achieved a higher PSNR (e.g., $30.1\,\mathrm{dB}$) compared to the PSNR achieved using the best manually-tuned fixed parameters (e.g., $29.8\,\mathrm{dB}$). This result confirms that specializing the parameters for a specific image can improve its restoration accuracy.

Next, to evaluate the generalization performance, we trained the parameters using multiple training images and evaluated the restoration accuracy on unseen test images. In the simulations, we set the learning rate to 10^{-5}. We observed that using a larger learning rate caused the training process to diverge.

Fig. 2 shows the PSNR values of ADMM both learned and fixed parameters. The figure indicate that the learned parameters achieve a PSNR comparable to that of the manually tuned fixed parameters, which represent the best performance obtained through exhaustive search. Fig. 3 shows the learned parameter values against the number of iterations. Specifically, Fig. 3(a) shows the value of γ^k, while Fig. 3(b) shows the value of λ^k. It is observed that both parameters do not deviate significantly from their initial settings throughout the iterations.

Fig. 4 shows the PSNR values of PDS both learned and fixed parameters. The figure indicates that the final accuracy is almost equivalent to that achieved with the best fixed parameters, and significant performance improvement cannot be obtained through parameter learning. Fig. 5 shows the learned parameters for the PDS algorithm, i.e., γ_1^k, γ_2^k, and λ^k. As with the ADMM parameters, these values also show no significant deviation from their initial settings throughout the iterations.

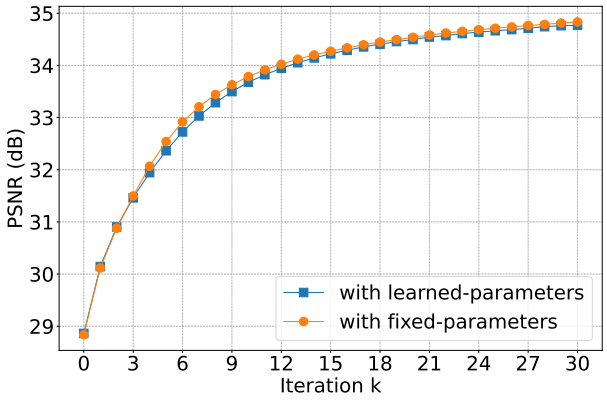
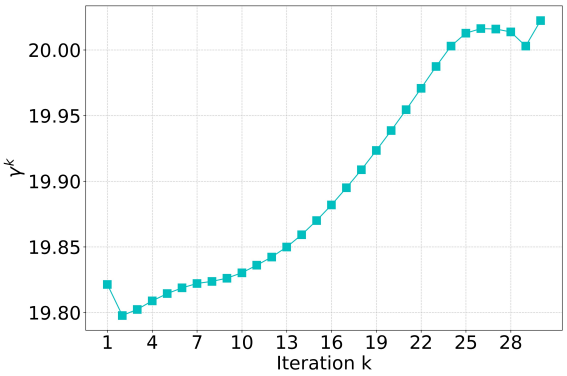
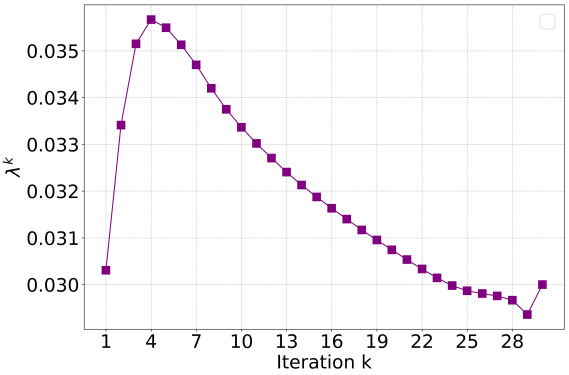


Fig. 2: Restoration accuracy with learned ADMM parameters.



(a) Value of γ^k at each iteration.



(b) Value of λ^k at each iteration.

Fig. 3: Learned values for each parameter in ADMM.

These results indicate that making the parameters variable at each iteration through learning did not improve the performance of ADMM and PDS compared to using the optimal fixed parameters. In other words, considering the implementation cost associated with the increased circuit complexity required for dynamic parameter changes, using good fixed parameters appears to be a more practical approach that offers a better balance between performance and ease of implementation.

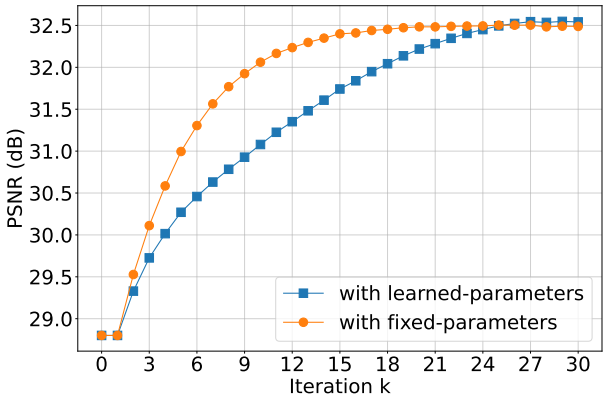
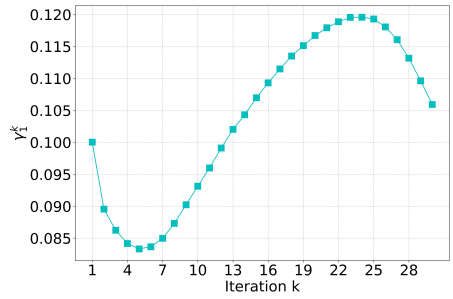
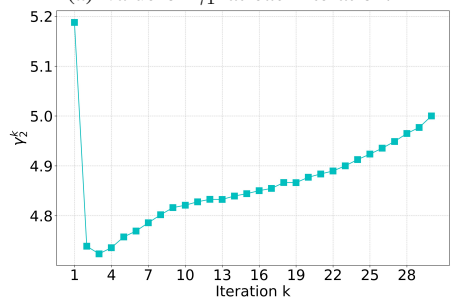


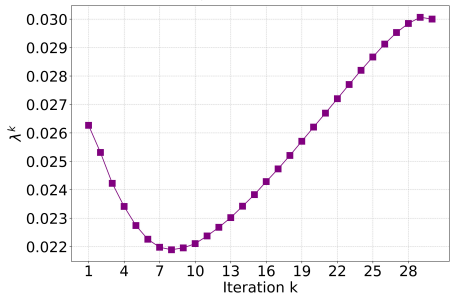
Fig. 4: Restoration accuracy with learned PDS parameters.



(a) Value of γ_1^k at each iteration.



(b) Value of γ_2^k at each iteration.



(c) Value of λ^k at each iteration.

Fig. 5: Learned values for each parameter in PDS.

V. Conclusion

In this study, we investigated parameter scheduling of ADMM and PDS for image restoration, taking into account the characteristics of optical analog circuits, where signal amplification introduces additive noise. Specifically, we employed a

deep unfolding framework to learn iteration-dependent parameters, aiming to understand the impact of dynamic parameter scheduling on image restoration performance. Our simulation results show that the variable parameters obtained via deep unfolding achieved comparable performance to that of the best fixed parameters, which were optimized manually beforehand. These results suggest that, for the denoising problem addressed in this paper, a simpler approach using optimal fixed parameters may be more practical, particularly given the implementation cost associated with the increased circuit complexity required for dynamic parameter changes.

Future work includes applying this approach to other image restoration tasks beyond denoising, such as deblurring and super resolution, and investigating methods for implementing the proximal operators in electronic circuits.

Acknowledgment

This work was partly supported by JST CREST Grant Number JPMJCR21C3 and JSPS KAKENHI Grant Number JP25H01111 and JP24K17277.

References

[1] N. Shlezinger, G. C. Alexandropoulos, M. F. Imani, Y. C. Eldar, and D. R. Smith, "Dynamic metasurface antennas for 6G extreme massive mimo communications," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 106–113, 2021.

[2] Y. Shen, N. C. Harris, S. Skirlo, *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature photonics*, vol. 11, no. 7, pp. 441–446, 2017.

[3] G. Wetzstein, A. Ozcan, S. Gigan, *et al.*, "Inference in artificial intelligence with deep optics and photonics," *Nature*, vol. 588, no. 7836, pp. 39–47, 2020.

[4] T. Zhou, X. Lin, J. Wu, *et al.*, "Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit," *Nature Photonics*, vol. 15, no. 5, pp. 367–373, 2021.

[5] C. Huang, V. J. Sorger, M. Miscuglio, *et al.*, "Prospects and applications of photonic neural networks," *Advances in Physics: X*, vol. 7, no. 1, p. 1 981 155, 2022.

[6] H. Zhou, J. Dong, J. Cheng, *et al.*, "Photonic matrix multiplication lights up photonic accelerator and beyond," *Light: Science & Applications*, vol. 11, no. 1, p. 30, 2022.

[7] K. Kameda, R. Hayakawa, K. Hayashi, and Y. Iiguni, "Performance evaluation of FISTA with constant inertial parameter," in *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2022, pp. 1713–1719.

[8] S. Furusawa, K. Hayashi, K. Kameda, and R. Hayakawa, "Numerical performance evaluation of ℓ_1-ℓ_2 sparse reconstruction using optical analog circuit," in *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2023, pp. 1410–1416.

[9] T. Kato, R. Hayakawa, S. Furusawa, K. Hayashi, and Y. Iiguni, "Optimization-based image restoration under implementation constraints in optical analog circuits," *arXiv [eess.IV]*, Jun. 2025.

[10] J. Eckstein and D. P. Bertsekas, "On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical programming*, vol. 55, pp. 293–318, 1992.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[12] L. Condat, "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *Journal of optimization theory and applications*, vol. 158, no. 2, pp. 460–479, 2013.

[13] B. C. Vũ, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," en, *Adv. Comput. Math.*, vol. 38, no. 3, pp. 667–681, Apr. 2013.

[14] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th international conference on international conference on machine learning*, 2010, pp. 399–406.

[15] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.

[16] D. Ito, S. Takabe, and T. Wadayama, "Trainable ISTA for sparse signal recovery," *IEEE Transactions on Signal Processing*, vol. 67, no. 12, pp. 3113–3125, 2019.

[17] P. Poggiolini, G. Bosco, A. Carena, V. Curri, Y. Jiang, and F. Forghieri, "A detailed analytical derivation of the GN model of non-linear interference in coherent optical transmission systems," *arXiv preprint arXiv:1209.0394*, 2012.

[18] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[19] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 68, no. 1, pp. 49–67, 2006.

[20] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.