Differentiable Programming based Step Size Optimization for LMS and NLMS Algorithms

Kazunori Hayashi*†, Kaede Shiohara*, Tetsuya Sasaki*
* Osaka City University, Osaka, Japan
† RIKEN Center for Advanced Intelligence Project
E-mail: kazunori@eng.osaka-cu.ac.jp, a15tn30i18@st.osaka-cu.ac.jp, a16tng0t17@st.osaka-cu.ac.jp

***Abstract*—We propose TLMS (Trainable Least Mean Squares) and TNLMS (Trainable Normalized LMS) algorithms, which use different step size parameter at each iteration determined by machine learning approach. It has been known that LMS algorithm can achieve fast convergence and small steady-state error simultaneously by dynamically controlling the step size compared as a fix step size, however, in conventional variable step size approaches, the step size parameter has been controlled in rather heuristic manners. In this study, based on the concept of differential programming, we unfold the iterative process of LMS or NLMS algorithms, and obtain a multilayer signal-flow graph similar to a neural network, where each layer has a step size of each iteration of LMS or NLMS algorithm as an independent learnable parameter. Then, we optimize the step size parameters of all iterations by using a machine learning approach, such as the stochastic gradient descent. Numerical experiments demonstrate the performance of the proposed TLMS and TNLMS algorithms under various conditions.**

I. Introduction

In gradient descent based iterative algorithms, such as LMS (Least Mean Squares) algorithm[1], how to choose a step size parameter is one of crucial issues, because the selection has a huge impact on the convergence performance of the algorithm. Generally speaking, faster convergence requires larger step size, while stable convergence and lower steady state error require smaller step size. Thus, faster convergence and lower steady state error are conflicting requirements for iterative algorithms with a fixed step size parameter. It is known that both requirements can be simultaneously achieved by using larger step size for the initial stage of the iteration process and decreasing it as the iterative process goes on[2], however, the profile of the variable step size parameter has been determined in rather heuristic manner in existing work, because the appropriate choice of the step size profile intricately depends on various factors. NLMS (Normalized LMS) algorithm [3] can achieve faster convergence than LMS algorithms by taking advantage of the size of filter input signal, but in practical implementation, the choice of the step size is still of great importance for NLMS as well.

This work was supported in part by the Grants-in-Aid for Scientific Research no. 19H02138 from the Ministry of Education, Science, Sports, and Culture of Japan.

In the field of machine learning, deep learning [4] using DNN (deep neural network) has been drawing much attention due to its promising performance in various applications such as image and voice recognitions. One of breakthroughs made by studies on deep learning is the optimization recipe of large number of parameters in a multilayer network (or graph) via gradient descent method based on backpropagation algorithm[5]. By generalizing this computational recipe, K. Gregor and Y. Lecun have been proposing a theoretical framework called *differentiable programming*, where differentiable parameters in a a computational graph are optimized based on gradient descent based algorithms used in deep learning[6]. Here, one of key issues is that the concept of differentiable programming is applicable not only for neural networks but also any algorithms which can be represented by differentiable computational graphs.

As for a preceding work on the application of the idea of differentiable programming to existing signal processing algorithms other than [6], TISTA (Trainable Iterative Shrinkage Thresholding Algorithm)[7],[8] has been proposed recently. TISTA obtains the computation graph by unfolding ISTA (Iterative Shrinkage Thresholding Algorithm), which solves the $\ell_1 - \ell_2$ optimization problem in compressed sensing, and the coefficient of ℓ_1 regularization term in the optimization problem is treated as an independent parameter for each iteration and is optimized by some learning algorithm based on backpropagation. Here, a learning algorithm named *incremental training* is employed to cope with the vanishing gradient problem[9] by learning the computation graph from the input side with increasing layers to be trained one by one, especially for the case with deep computation graph generated from the iterative algorithm with large number of iterations.

In this paper, we consider to apply the idea of differential programming to LMS and NLMS algorithms, and propose TLMS (Trainable LMS) and TNLMS (Trainable NLMS) algorithms, where the step size parameter of each iteration (step size profile) is optimized based on supervised learning using synthetic training data. In the proposed method, we firstly obtain a multilayer signal flow graph similar to neural network by unfolding conventional LMS and NLMS algorithms. Then, assuming the step size of each iteration to be a learnable independent variable, the step size of each iteration is optimized in the sense of mean squared error between the adaptive filter output and the desired filter output using pairs of the input and the desired output as the training data. Since each layer of the graph is differentiable with respect to the learning parameter, we can employ some SGD (stochastic gradient descent) type algorithm, such as SGD, RMSporp, and Adam, for the optimization with the incremental learning. In order to evaluate the performance of the proposed TLMS and TNLMS algorithms, we have conducted computer simulations. Specifically, the convergence performance of the proposed scheme using Adam[10] is evaluated for the system identification problem, and it has been confirmed that TLMS and TNLMS algorithms achieves both faster convergence and low steady state error, while some instability of TLMS algorithm is observed for the case with large number of filter taps or highly colored input signal.

II. Review of LMS and NLMS algorithms

Here, we briefly review LMS and NLM algorithms[11] considered in this paper.

A. LMS algorithm

Let $x(k)$ and $d(k) \in \mathbb{R}$ denote the input signal and the desired output of the adaptive filter at time k, respectively. If we define the input signal vector as $\boldsymbol{x}(k) = [x(k), x(k-1), \ldots, x(k-N+1)]^T$ and the coefficient vector of the adaptive filter as $\boldsymbol{h}(k) = [h_0(k), h_1(k), \ldots, h_{N-1}(k)]^T \in \mathbb{R}^N$, the filter output signal at time k is given by $y(k) = \boldsymbol{h}(k)^T \boldsymbol{x}(k)$. Then, the update equation of LMS algorithm is given by

$$\boldsymbol{h}(k+1) = \boldsymbol{h}(k) + 2\alpha \boldsymbol{x}(k) e(k), \tag{1}$$

where $e(k) = d(k) - y(k)$ is an error signal.

Then, we review the condition on the step size α that the average of the filter coefficient vector $\boldsymbol{h}(k)$ obtained by LMS algorithm converges to the optimum filter coefficient vector $\boldsymbol{h}_{\mathrm{opt}}$, which minimizes $E[e(k)^2]$. By subtracting $\boldsymbol{h}_{\mathrm{opt}}$ from both sides of (1), we have

$$\boldsymbol{h}(k+1) - \boldsymbol{h}_{\mathrm{opt}} = (\boldsymbol{I} - 2\alpha \boldsymbol{x}(k)\boldsymbol{x}^T(k))(\boldsymbol{h}(k) - \boldsymbol{h}_{\mathrm{opt}}). \tag{2}$$

If we assume so called independent assumption,

$$\begin{aligned} &E\left[(\boldsymbol{I} - 2\alpha \boldsymbol{x}(k)\boldsymbol{x}^T(k))(\boldsymbol{h}(k) - \boldsymbol{h}_{\mathrm{opt}})\right] \\ &= E\left[\boldsymbol{I} - 2\alpha \boldsymbol{x}(k)\boldsymbol{x}^T(k)\right] E\left[\boldsymbol{h}(k) - \boldsymbol{h}_{\mathrm{opt}}\right], \end{aligned}$$

then we can easily show that

$$0 < \alpha < \frac{1}{\lambda_{\max}} \tag{3}$$

is the condition on the step size for the convergence. Here, $\lambda_{\max}$ is the maximum eigenvalue of $E[\boldsymbol{x}(k)\boldsymbol{x}^T(k)]$. Note that this condition does not guarantee the stability of LMS algorithm.

B. NLMS algorithm

It is known that LMS algorithm suffers from slow rate of convergence especially for the colored input signal. NLMS is an improved algorithm by taking advantage of the size (Euclidian norm) of input signal vector $\boldsymbol{x}(k)$ in the update equation.

If we assume variable step size of LMS algorithm as $\alpha(k)$, the error signal between the desired output $d(k)$ and the filter output obtained by using the filter coefficient after the $(k+1)$-th iteration $\boldsymbol{h}(k+1)$ is given by

$$\begin{aligned} e^+(k) &= d(k) - \boldsymbol{h}^T(k+1)\boldsymbol{x}(k) \\ &= [1 - 2\alpha(k)\|\boldsymbol{x}(k)\|_2^2] e(k). \end{aligned} \tag{4}$$

This error signal $e^+(k)$ is called a posterior error, and the update equation of NLMS algorithm can be derived by setting the step size of LMS algorithm so that the posterior error to be zero as

$$\alpha(k) = \frac{1}{2\|\boldsymbol{x}(k)\|_2^2}. \tag{5}$$

Thus, the update equation of NLMS algorithm is given by

$$\boldsymbol{h}(k+1) = \boldsymbol{h}(k) + \frac{\boldsymbol{x}(k)}{\|\boldsymbol{x}(k)\|_2^2} e(k). \tag{6}$$

In actual implementations, however, the update equation of (6) is rarely employed as its form. Instead, by introducing fixed parameters $\alpha, \mu > 0$, more robust update equation of

$$\boldsymbol{h}(k+1) = \boldsymbol{h}(k) + \alpha \frac{\boldsymbol{x}(k)}{\|\boldsymbol{x}(k)\|_2^2 + \mu} e(k) \tag{7}$$

is often employed. Here, α can be regarded as a step size parameter of NLMS algorithm, and hence, we are not free from the problem of step size selection even for the case with NLMS algorithm.

III. Proposed TLMS and TNLMS Algorithms

Here, we propose TLMS and TNLMS algorithms, where step size of each iteration is optimized by supervised learning using the pair of the input signal and the corresponding desired signal as training data. In the proposed algorithms, we treat step size of each iteration (or time) $\alpha(k)$ to be independent learnable parameter, and optimize it with some stochastic gradient descent based algorithm.

A. TLMS algorithm

In TLMS algorithm, we firstly set the step size to be time variant parameter $\alpha(k)$, and consider the update equation of

$$\boldsymbol{h}(k+1) = \boldsymbol{h}(k) + 2\alpha(k)\boldsymbol{x}(k)e(k). \tag{8}$$

We employ mini-batch learning with size N. Namely, we define a set of input signal vectors as $\mathcal{X} = \{\boldsymbol{x}_0(k), \boldsymbol{x}_1(k), \ldots, \boldsymbol{x}_{N-1}(k)\}$, a set of desired outputs $\mathcal{D} = \{d_0(k), d_1(k), \ldots, d_{N-1}(k)\}$, and a set of filter coefficient vectors $\mathcal{H} = \{\boldsymbol{h}_0(k), \boldsymbol{h}_1(k), \ldots, \boldsymbol{h}_{N-1}(k)\}$. And, we assume (8) holds for all elements of the sets as

$$\boldsymbol{h}_i(k+1) = \boldsymbol{h}_i(k) + 2\alpha(k)\boldsymbol{x}_i(k)e_i(k), \ \ (i = 0, 1, \ldots, N-1),$$

where $e_i(k) = d_i(k) - y_i(k)$ and $y_i(k) = \boldsymbol{x}_i^T(k)\boldsymbol{h}_i(k)$.

As for the cost function, we use $J(k)$ defined as

$$J(k) = \frac{1}{N}\|\boldsymbol{d}(k) - \boldsymbol{y}(k)\|_2^2, \tag{9}$$

and optimize the step size parameter of each iteration via incremental training [7] with gradient descent based method. Here, $\boldsymbol{y}(k) = [y_0(k), y_1(k), \ldots, y_{N-1}(k)]^T$ is a vector composed by filter outputs of all batches at time k, and $\boldsymbol{d}(k) = [d_0(k), d_1(k), \ldots, d_{N-1}(k)]^T$ is a desired output signal vector.

Then, we describe the gradient descent based approach in TLMS algorithm assuming the size of batch to be $N = 1$ for simplicity. The partial differentiation of the cost function $J(t)$ at the t-th round of the incremental training with respect to the step size $\alpha(t-\tau)$ is given by

$$\begin{aligned}\frac{\partial J(t)}{\partial \alpha(t-\tau)} =& \frac{\partial J(t)}{\partial y(t)}\frac{\partial y(t)}{\partial y(t-1)} \cdots \frac{\partial y(t-\tau+2)}{\partial y(t-\tau+1)}\frac{\partial y(t-\tau+1)}{\partial \alpha(t-\tau)} \\ =&(-2)^{\tau+1}e(t)e(t-\tau)\boldsymbol{x}^T(t-\tau+1)\boldsymbol{x}(t-\tau) \\ &\cdot \prod_{k=t-\tau+1}^{t-1} [\alpha(k)\boldsymbol{x}^T(k+1)\boldsymbol{x}(k)]. \end{aligned} \tag{10}$$

The step size $\alpha(k)$ of TLMS is updated based on (10). For example, if we employ SGD for the optimization algorithm of the step size, the update equation can be written as

$$\alpha(k) \leftarrow \alpha(k) - \beta\frac{\partial J}{\partial \alpha(k)}, \tag{11}$$

where β is a leaning rate. Also, we employ a hard limitter to keep $\alpha(k) \geq 0$.

B. TNLMS algorithm

By setting $\mu = 0$ and the step sizeα to be time variant parameter in the update equation of NLM algorithm (7), we have

$$\boldsymbol{h}(k+1) = \boldsymbol{h}(k) + \alpha(k)\frac{\boldsymbol{x}(k)}{\|\boldsymbol{x}(k)\|_2^2}e(k). \tag{12}$$

TNLMS employs the same cost function as TLMS of (9). Then, the partial differentiation of the cost function $J(t)$ at the t-th incremental training round with respect to the step size $\alpha(t-\tau)$ is given by

$$\begin{aligned}\frac{\partial J(t)}{\partial \alpha(t-\tau)} =& \frac{\partial J(t)}{\partial y(t)}\frac{\partial y(t)}{\partial y(t-1)} \cdots \frac{\partial y(t-\tau+2)}{\partial y(t-\tau+1)}\frac{\partial y(t-\tau+1)}{\partial \alpha(t-\tau)} \\ =&e(t)e(t-\tau)\boldsymbol{x}^T(t-\tau+1)\frac{\boldsymbol{x}(t-\tau)}{\|\boldsymbol{x}(t-\tau)\|_2^2} \\ &\cdot \prod_{k=t-\tau+1}^{t-1} \left[\alpha(k)\boldsymbol{x}^T(k+1)\frac{\boldsymbol{x}(k)}{\|\boldsymbol{x}(k)\|_2^2}\right]. \end{aligned} \tag{13}$$

The step size of TNLMS also can be updated by SGD as in (11), and the hard limitter will be employed to keep $\alpha(k) \geq 0$.

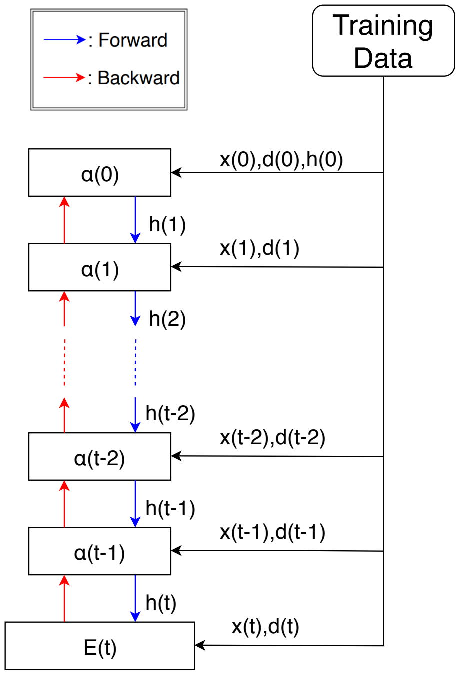

Fig. 1. Flow of proposed step size optimization of TLMS and TNLMS algorithms

At the end of this section, we summarize the flow of the proposed step size optimization based on incremental training of TLMS and TNLMS algorithms in Fig. 1.

IV. Numerical Results

We have conducted numerical experiments of system identification in order to evaluate the performance of the proposed TLMS and TNLMS algorithms in comparison with that of conventional LMS and NLMS algorithms.

A. Experimental Setup

We consider a system identification problem with the desired output of $d(k) = \boldsymbol{w}^T\boldsymbol{x}(k) + v(k)$, where $\boldsymbol{w} = [w_0, w_1, \ldots, w_{L-1}]^T$ ($w_i \in \mathcal{N}(0,1)$) is a true unknown filter coefficient vector, $\boldsymbol{x}(k)$ is a filter input vector, and $v(k) \in \mathcal{N}(0, 0.1)$ is an additive white Gaussian measurement noise. For the generation of the filter input vector, we have generated a sample $\tilde{\boldsymbol{x}} = [\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_{M+L-1}]^T$ from multi dimensional Gaussian distribution having zero-mean and covariance matrix of

$$\boldsymbol{\Sigma} = \begin{bmatrix} r_{0,0} & r_{0,1} & \cdots & r_{0,M+L-1} \\ r_{1,0} & r_{1,1} & \cdots & r_{1,M+L-1} \\ \vdots & \vdots & \ddots & \vdots \\ r_{M+L-1,0} & r_{M+L-1,1} & \cdots & r_{M+L-1,M+L-1} \end{bmatrix},$$

and set $\boldsymbol{x}(k) = [\tilde{x}_{k+L-1}, \tilde{x}_{k+L-2}, \ldots, \tilde{x}_k]^T$. It should be noted that $r_{i,j}$ is the correlation coefficient of $\tilde{x}_i$ and $\tilde{x}_j$, and thus $r_{i,j} = C^{|i-j|}$. The number of samples of $\tilde{\boldsymbol{x}}$ for each incremental training round is set to be $10,000$, and batch size to be $N = 1,000$. Also, the initial value of the step

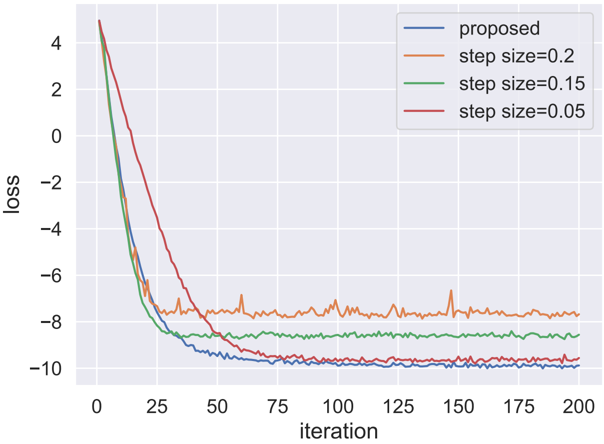

Fig. 2. Learning curve of TLMS ($C = 0,\ L = 3$)

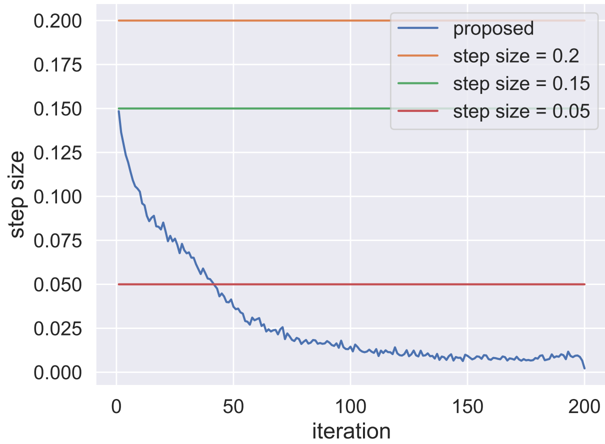

Fig. 3. Step size profile of TLMS ($C = 0,\ L = 3$)

size of time k is set to be $\alpha(k) = \alpha_0 \cdot 0.9^k$, where $\alpha_0 = 0.1$ for TLMS and $\alpha_0 = 1$ for TNLMS. We have employed Adam[10] as the optimization algorithm with the learning rates of 0.001 for TLMS and 0.01 for TNLMS.

B. Performance of TLMS Algorithm

Figures 2 and 3 show the learning curve of the proposed TLMS algorithm and the corresponding step size profile obtained by the proposed learning method with the input correlation coefficient of $C = 0$ and the number of taps of the filter of $L = 3$. For comparison purpose, the performance of the conventional LMS algorithm with fixed step size parameter is also plotted in the same figures. In Fig. 2, the proposed TLMS achieves comparable convergence rate to the conventional LMS with the step size of 0.15 or 0.2 while achieving lower steady state error. In Fig. 3, the learned step size with the proposed method decreases as the iteration goes, which coincides with the conventional heuristic approach qualitatively, while this profile is rather different from the one obtained in [7]. From these results, we can see that the proposed approach is valid for the case with white input signal.

Figures 4 and 5 show the learning curve and the corresponding step size profile for the case with the colored input signal, where we set $C = 0.3$. In Fig. 4, it seems that the step size of 0.15 to be close to the maximum value for convergence of the fixed step size LMS algorithm, while the proposed TLMS simultaneously achieves comparable convergence rate as the case with step size of 0.15 and lower steady state error. From Fig. 5, we can see that the step size profile obtained in this colored input case is similar to that with the case of white input signal.

Figures 6 and 7 show the learning curves and the corresponding step size profiles for the case with much higher input signal correlation ($C = 0.8$). In this case, the learning curve of the proposed approach is rather unstable, and the obtained step size profile has complicated

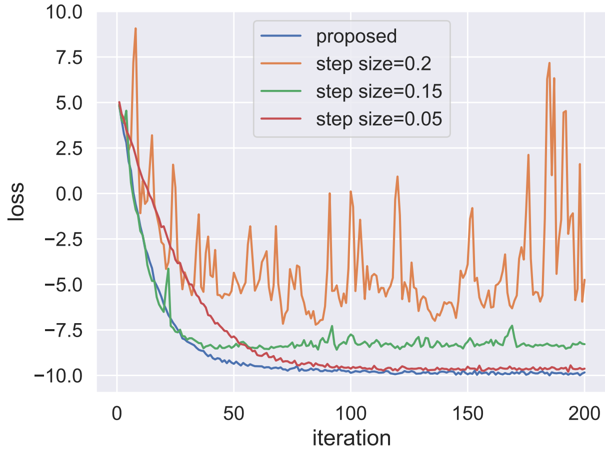

Fig. 4. Learning curve of TLMS ($C = 0.3,\ L = 3$)

form. The reason for the performance degradation will be that, with this highly colored input scenario, the number of iterations of 200 will not be enough for LMS based algorithm, and thus the proposed method fails to learn the appropriate step size profile.

Finally, we have changed the number of filter taps to be $L = 10$. Figures 8 and 9 show the learning curves and the corresponding step size profiles for $L = 10$ and $C = 0.8$. The performance of the proposed approach is unstable in this scenario as well. The reason could be that, because of the increase in the number of unknown parameters, the required number of iterations also increases, which results in poor learning results. Thus, the differentiable programming approach might not be suited for iterative algorithms which requires hundreds of iterations, or some more sophisticated learning algorithm might be required for such iterative algorithms.

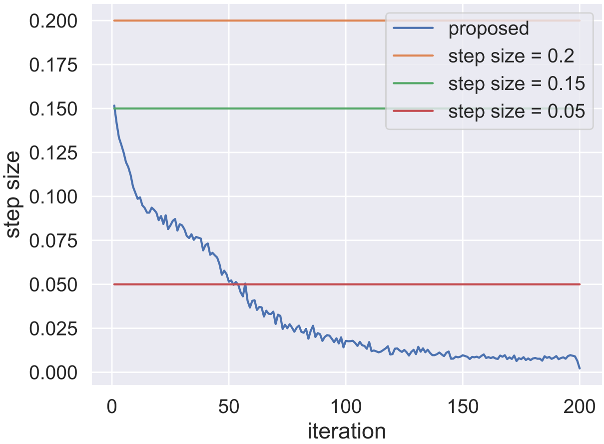

Fig. 5. Step size profile of TLMS ($C = 0.3,\ L = 3$)

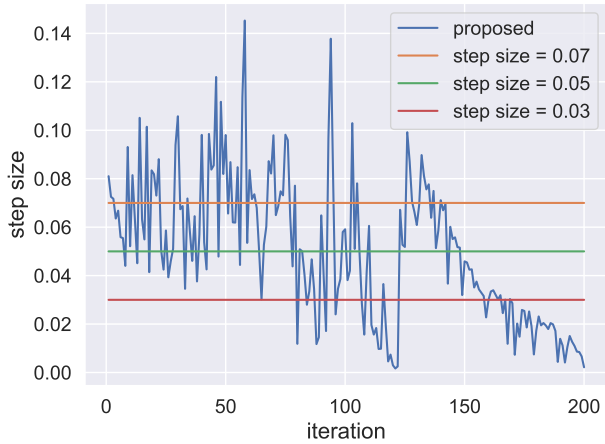

Fig. 7. Step size profile of TLMS ($C = 0.8,\ L = 3$)

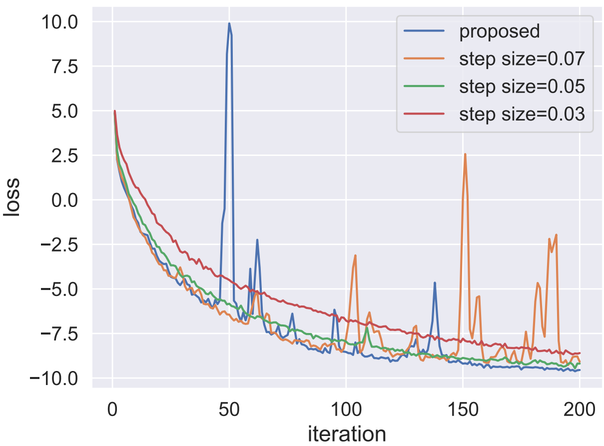

Fig. 6. Learning curve of TLMS ($C = 0.8,\ L = 3$)

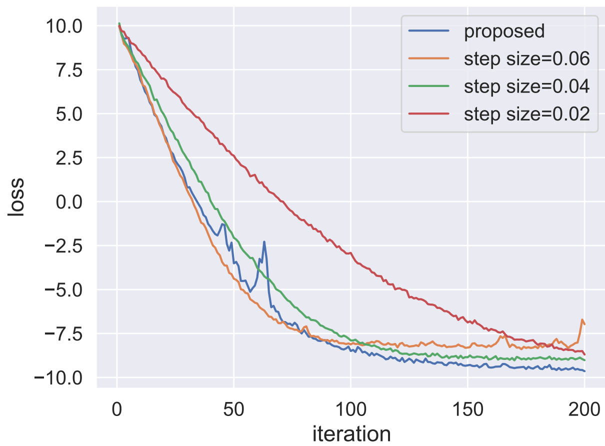

Fig. 8. Learning curve of TLMS ($C = 0.3,\ L = 10$)

C. Performance of TNLMS Algorithm

Figures 10 – 17 show the learning curves of the proposed TNLMS algorithm and the corresponding step size profiles obtained by the proposed learning method with the input correlation coefficient of $C = 0, 0.3, 0.8$ and the number of taps of $L = 3, 10$. For comparison purpose, the performance of the conventional NLMS algorithm with fixed step size parameter is also plotted in the same figures. From Figs. 2, 4, 6, and 8, unlike the case with TLMS algorithm, we can see that the proposed TNLMS achieves stable and fast convergence with the lowest steady state error for all the parameter settings. This will be because of the fact that NLMS requires much smaller number of iterations than that of LMS algorithms, and hence the proposed approach can successfully learn the appropriate step size profiles. In Figs. 3, 5, 7, and 9, the step size profiles of TNLMS is rather different from those of TLMS, while the profiles of TNLMS with different system parameters are quite similar. The reasoning of the shape of the step size profile and the investigation of the performance sensitivity on the profile will be our future work.

V. Conclusion

We have proposed TLMS and TNLMS algorithms, which can optimize step size parameter of each iteration via incremental learning based on the idea of differentiable programming. The performance of the proposed algorithms is evaluated via numerical experiments of the system identification, and we have confirmed that proposed approach can achieve faster convergence rate and lower steady state error simultaneously especially for TNLMS algorithm. Also, we have noticed that the proposed TLMS algorithm suffers from unstable convergence behavior due to failure in learning the step size parameter for the case with highly colored input signals or larger number of unknown filter coefficients. The future work includes the improvement of the performance of TLMS and the

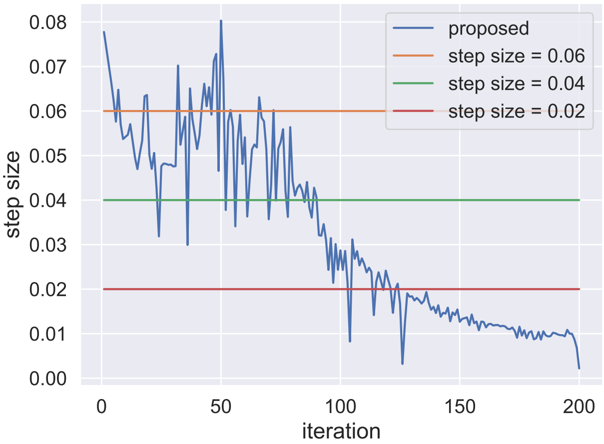

Fig. 9. Step size profile of TLMS ($C = 0.3,\ L = 10$)

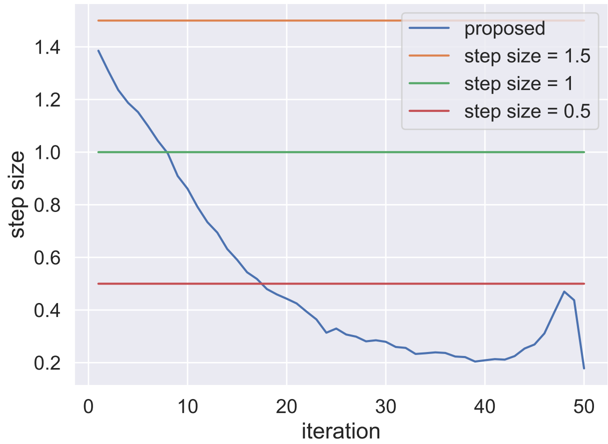

Fig. 11. Step size profile of TNLMS ($C = 0,\ L = 3$)

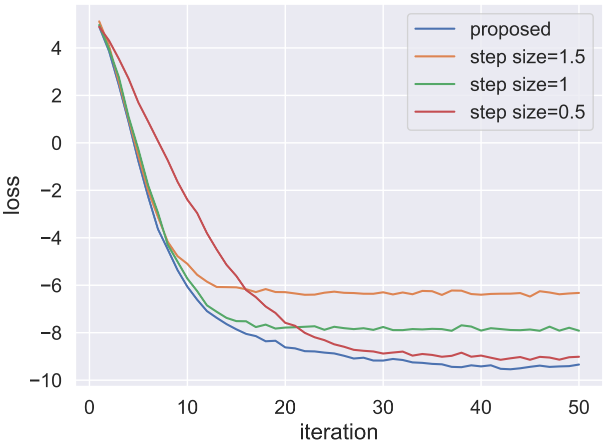

Fig. 10. Learning curve of TNLMS ($C = 0,\ L = 3$)

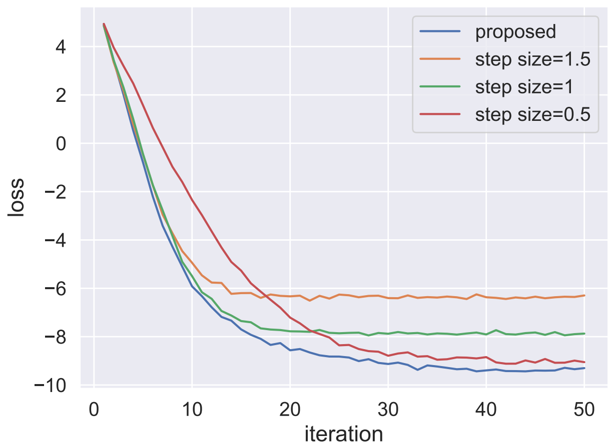

Fig. 12. Learning curve of TNLMS ($C = 0.3,\ L = 3$)

investigation of some other iterative algorithms suited for differentiable programming approach.

References

[1] B. Widrow and M.E. Hoff, "Adaptive switching circuits," 1960 IRE WESCON Convention Record, Part 4, pp.96–104, IRE, New York, 1960.
[2] S. Haykin, Adaptive Filter Theory, 3rd Edition, Prentice Hall, 2015.
[3] G.C. Goodwin and K.S. Sin, Adaptive filtering prediction and control, Prentice-Hall information and system sciences series, Prentice-Hall, 1984.
[4] G.E. Hinton and R.R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol.313, no.5786, pp.504–507, 2006.
[5] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning representations by back-propagating errors," nature, vol.323, pp.533–536, Oct. 1986.
[6] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," Learning Fast Approximations of Sparse Coding, pp.399–406, ICML'10, 2010.
[7] 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, June 2019.
[8] S. Takabe, M. Imanishi, T. Wadayama, and K. Hayashi, "Deep learning-aided projected gradient detector for massive overloaded mimo channels," 2019 IEEE International Conference on Communications, ICC 2019, Shanghai, China, May 20-24, 2019, pp.1–6, 2019.
[9] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, pp.III–1310–III–1318, ICML'13, JMLR.org, 2013.
[10] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Contribution to International Conference on Learning Representations, May 7-9, 2015, San Diego, 2014.
[11] B. Farhang-Boroujeny, Adaptive Filters: Theory and Applications, Wiley, 2013.

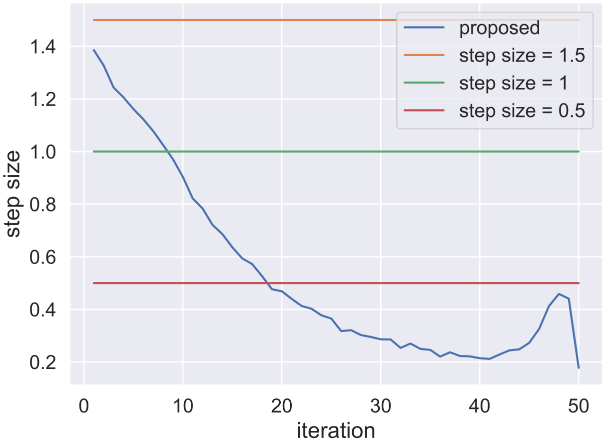

Fig. 13. Step size profile of TNLMS ($C = 0.3,\ L = 3$)

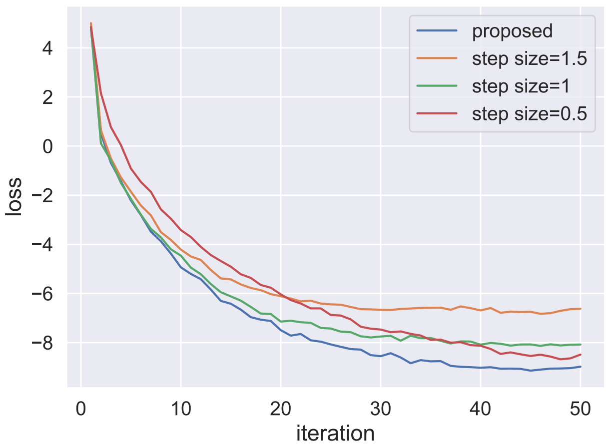

Fig. 14. Learning curve of TNLMS ($C = 0.8,\ L = 3$)

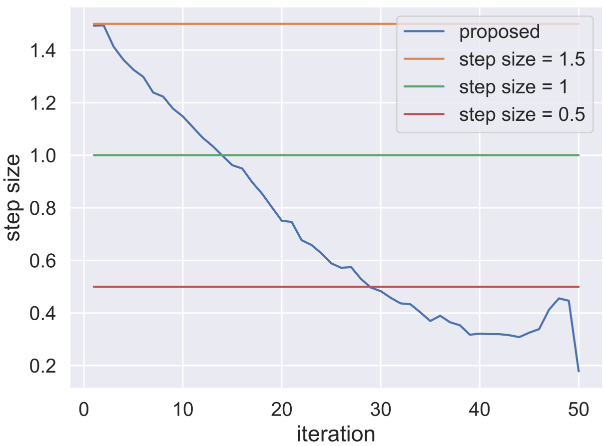

Fig. 15. Step size profile of TNLMS ($C = 0.8,\ L = 3$)

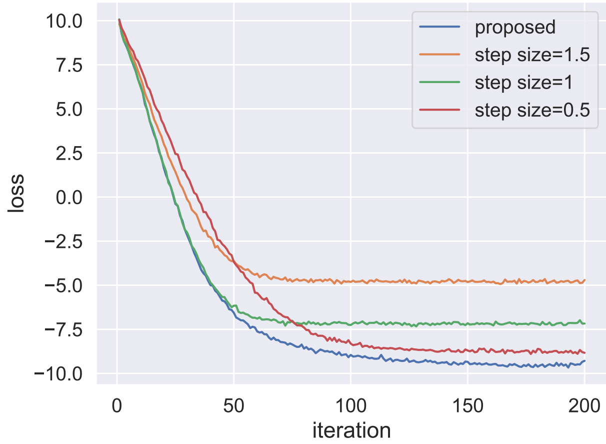

Fig. 16. Learning curve of TNLMS ($C = 0.3,\ L = 10$)

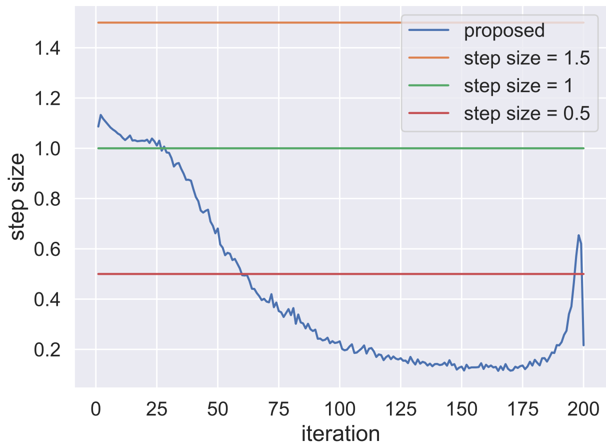

Fig. 17. Step size profile of TNLMS ($C = 0.3,\ L = 10$)