Topology and Parameter Optimization of High-Order Δ-Σ Modulators Towards Superior Efficiency and Stability with Multi-Agent Reinforcement Learning

Thinh Quang Do*, Lihong Zhang*, Octavia A. Dobre*, Trang Hoang†, Trung Q. Duong*‡
* Memorial University, Canada, e-mail: {tqdo, lzhang, odobre, tduong}@mun.ca
† Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Vietnam, e-mail: hoangtrang@hcmut.edu.vn
‡ Queen's University Belfast, U.K, e-mail: trung.q.duong@qub.ac.uk)

***Abstract*—The Δ-Σ analog-to-digital converter (ADC), with the modulator as its core component, has posed considerable challenges to its designers, due to the complex topologies and instability problem. Thanks to reinforcement learning (RL), appropriate models can be trained to automatically generate efficient modulator structures without the need for prior datasets. Proximal policy optimization (PPO), one of the latest and most promising branches of RL, can be optimally used by virtue of its simplicity and less hyperparameter tuning. This study focuses on multi-agent PPO (MAPPO) for the design of high-order Δ-Σ modulators, with two agents handling topology and parameter optimization respectively in a cooperative way. We address the challenge of both efficiency and stability through proper mathematical formulation and effective integration of weighted objectives. Through extensive simulations and iterative process of MAPPO, our proposed methodology demonstrates effectiveness in maximizing the efficiency and stability objectives of the desired Δ-Σ modulator in reward form.**

***Index Terms*—Δ-Σ modulator, stability, multi-agent proximal policy optimization, topology synthesis, parameter optimization**

I. Introduction

Δ-Σ analog-to-digital converters (ADCs) usually suffer from instability problems as designers thrust towards performance with higher accuracy (also higher complexity [1], even when they are in the right use for audio processing or medical treatment. Modulator, the core component of each Δ-Σ ADC, for most cases decides the performance and structure of the whole device, since both the signal transfer function (STF) and noise transfer function (NTF) are defined here. Some nominal topology structures have been proposed for a good-performance Δ-Σ modulator such as CIFB, CRFB, CIDF, or CIDIFF [2], [3]. Nevertheless, as Δ-Σ modulators aim for high-accuracy applications, designers often rely on increasing the order of the modulator to enhance further the signal-to-noise ratio (SNR), where noises often come from the quantization process. Consequently, stability for these modulators becomes a challenging problem as the complexity of topology also involves a much larger design space which reduces the efficiency and consumes more time of the synthesis procedure. Many optimization methods and techniques, such as Bayesian optimization, evolutionary algorithm, or even mathematical analysis, have been applied with numerical results to cope with this task [4]–[6]. For high-order Δ-Σ modulators, these methods may struggle to find the maximum profitable structures as they often deal with the topology first, then optimize the sizing parameters afterward, or they would be stuck to find a stable route to update their generative schemes otherwise.

Reinforcement learning (RL) refers to remarkable machine learning models that require no dataset to train but gradually enhance their actions (circuit modification) on the environments (the design) by refining their policies (how they determine actions), and hence can be the solution for effective circuit designs since designers often do not have enough data to form a dataset for training. As the circuit scale becomes larger, design procedures using RL also often require the use of multi-agent, where each agent is responsible for a set of actions that may affect the whole environment [7]. Among various RL algorithms, proximal policy optimization (PPO) appears to be much more efficient for circuit topology designs due to stable policy updates and capability of handling high-dimensional data [8]. Design schemes for Δ-Σ modulator can hence use PPO properly to deal with their high-dimensional search space and sensitivity to setup parameters. However, as the modulator order increases, the design may require the use of PPO in multi-agent scenarios since a slight change in Δ-Σ modulator's topology may affect the whole performance, and any update needs to be stable and less aggressive.

This paper proposes a multi-agent PPO (MAPPO) model that works on high-level and high-order design of Δ-Σ modulator with two objectives of efficiency (using a Figure of Merit criteria) and stability (verifying zeros and poles of transfer functions and constraining by Lee's criterion), using two agents for handling topology and parameter sizing respectively. The topology is represented in ABCD matrix form for better generalization of transfer functions and zero-pole analysis. Two agents also share an observation space so that they can understand the effects of the other agent's action as well as theirs. Objectives would take the Figure of Merit (FoM) and stability constraints as weighted components of the general reward which returns back to the agents for action evaluation. Since multi-agent scenarios often struggle with convergence problems, we tackled this by carefully setting up hyperparameter configurations, and giving more rollout threads to improve action selection accuracy. The simulation

results have proven the effectiveness of our model based on the comparison over performance evaluations of SNR (98.32 dB) and Schreier's FoM (187.9dB).

II. Δ-Σ MODULATOR TOPOLOGY AND SYNTHESIS FORMULATION

A. ABCD matrix

$ABCD$ matrix, the merge of 4 component matrices of A, B, C, and D, can generalize a Δ-Σ modulator structure with relationship mapping between input signals, internal signals from the integrators, and feedback signals from the quantizer [9]. The mathematical formula for ABCD representation corresponding to a modulator is

$$x[n+1] = Ax(n) + B\begin{bmatrix} u[n] \\ v[n] \end{bmatrix}, \tag{1}$$

and

$$y[n] = Cx(n) + D\begin{bmatrix} u[n] \\ v[n] \end{bmatrix}, \tag{2}$$

where $u(n)$, $v(n)$, $x(n)$ and $y(n)$ are the representation of input, output signals, first integrators' and last integrator's output, respectively, in time domain. Their counterparts in z-domain would be $U(z)$, $V(z)$, $X(z)$ and $Y(z)$.

Each integrator has a self-feedback coefficient that lies on the diagonal of A, which is generally 1 if no direct self-feedback connection is formed and the integrator is non-delaying. Specifically, the mathematical expression is

$$x_a(n+1) \cong x_a(n)\frac{1}{z-1}, \tag{3}$$

where x_a refers to the a^{th} integrator in the chain ordering from input signal to quantizer, n denotes the current timestep and $n+1$ represents the following one, and $\frac{1}{z-1}$ is the transfer function of the non-delaying integrator. Solving (3) would result in

$$x_a(n+1) \cong z^{-1}(x_a(n) - x_a(n+1)), \tag{4}$$

which indicates a line of feedback with a gain equal to 1 from the integrator output back to its inputs, while z^{-1} represents a step forward in time between the two integrator's outputs from consecutive timesteps.

In case designers intend to introduce a feedback line with gain g while still using non-delaying integrators, (4) can be rewritten as

$$x_a(n+1) \cong z^{-1}[x_a(n) - (g+1)x_a(n+1)], \tag{5}$$

so that $g+1$ would be filled in the corresponding position in the ABCD matrix.

Consequently, any topology structure of any Δ-Σ modulator can be represented using the ABCD matrix, which then can be processed further to infer the transfer function, where the relationship between the ABCD matrix and the transfer functions is expressed as

$$\begin{bmatrix} L_0(z) \\ L_1(z) \end{bmatrix} = C(z^{-1}I - A)^{-1}B + D, \tag{6}$$

where I represents the identity matrix and "$^{-1}$" denotes the inverse of a matrix. Upon getting loop gains of $L_0(z)$ and $L_1(z)$, zeros and poles of the transfer functions can be calculated.

B. Stability analysis

Lee's criterion is widely approved as a rule-of-thumb method to evaluate the stability of a modulator, although it is not actually necessary (some high-performance modulators do not follow this criterion) and not sufficient (no input signal boundary is announced) [2], [10]. The criterion stated that a binary modulator with $NTF(z)$ is stable if

$$\max_{\omega} |H(e^{j\omega})| < 1.5. \tag{7}$$

This quantity on the left side of (7) represents the maximum gain of NTF over all frequencies, which implies that the zeros and poles of NTF are properly distributed within the interest bandwidth and thereby it affirms the stability of the modulator. Then, if NTF is represented as the multiplication of zeros and poles as

$$H(z) = \prod_{i=1}^{N} \frac{z - z_i}{z - p_i}. \tag{8}$$

Positions of zeros and poles are also important to attenuate noise in-band, where poles of NTF are usually kept close to its zeros and poles of STF.

C. Topology and parameter sizing problem formulation

In this design problem, we define two types of objectives as stated earlier: efficiency in the form of FoM and stability criteria. Each objective is assigned a weight coefficient of α that determines which objective is more important. These coefficients would be adjusted as a tuning hyperparameter as described in the multi-agent reinforcement learning model later in section III. The stability criteria includes a number of several constraints that if violated, negative values would be returned. It can be expressed as

$$\begin{aligned} STAB(\mathbf{C}, \mathbf{G}) = &- \sum_{i,j=1}^{N} \|p_i^{NTF} - p_j^{STF}\|^2 \\ &- \sum_{i,j=1}^{N} \|p_i^{NTF} - z_j^{NTF}\|^2 + \prod_{i=1}^{N} \frac{z_i + 1}{p_i + 1} - 1.5, \end{aligned} \tag{9}$$

where $\mathbf{C}$,$\mathbf{G}$ represents connections and gains within the topology of the desired modulator, and p_i, z_i are poles and zeros of the NTF or STF. The first term of (9) indicates that every zeros of NTF should located within the unit circle. The second and third terms are meant to attenuate the noise in-band, while the last term is based on Lee's criterion as described in (7), at $z = -1$ where the half of the sampling frequency is. Besides zero and pole analysis, the output of each integrator is also important to evaluate the stability of the modulator. In this paper, the maximum value allowed is $0.8 \times V_{ref}$ per integrator output to allow some space for other uncertainty before getting into the quantizer.

For efficiency, the FoM is selected to be the Scheirer FoM [2], whose mathematical formula is

$$FoM = SNR + 10log\left(\frac{BW}{P}\right), \tag{10}$$

where BW and P denote the bandwidth and power of the corresponding modulator, respectively. This widely-used FoM allows designers to balance between various performance properties of the modulator and pursue a specific number for evaluation. The overall design problem, whose all components are clarified, can be formulated as follows

$$\begin{aligned}\max_{\mathbf{G},\mathbf{C}} \quad & \alpha_1(fom(\mathbf{G},\mathbf{C}) - 185) \\ & + \alpha_2(\sum_{i=1}^{N}|out_i(\mathbf{G},\mathbf{C})| - 0.8NV_{ref}) \\ & + \alpha_3(STAB(\mathbf{G},\mathbf{C})),\end{aligned} \tag{11}$$

where $\alpha_1, \alpha_2, \alpha_3$ are the assigned weight coefficients that determine the effect of the three criteria and N is the modulator order.

III. Multi-agent Proximal Policy Optimization for Δ-Σ modulator design

PPO is believed to be less sample efficient compared to other off-policy algorithms, therefore being unsuitable for multi-agent scenarios. However, this belief is proven to be less trustworthy as in [11], multi-agent PPO models can achieve competitive results, even when analyzing sample efficiency. Based on that, our proposed model makes use of two agents: one for topology construction and one for sizing optimization. Topology construction includes a set of 3 actions: disconnection (D), connection (C), and no change (R). Sizing optimization also selects 3 types of action: no change (R), increment by step size (I), and decrement by step size (D). After gathering 2 actions from these agents, the MAPPO model creates a joint action to deliver to the environment. The gain along connections are also set with an upper bound and a lower bound of $[-2.5, 2.5]$ to prevent the agent from creating topologies with impractical connections. Training is divided into batches, where the trajectory τ is recorded into a buffer to provide information for the agents and the environment.

We included step size (s) as an important factor in our model. Instead of using a fixed value, this number is adjusted based on the reward (the performance) of the design. Undesired results will increase this step size and the corresponding action will make a larger impact on the topology structure. Large step sizes also prevent topology from creating topology with few connections due to our definition of updating connection upon disconnection joint action as

$$G_{ij} = G_{ij} + 2^{-G_{ij}} \times s, \tag{12}$$

where G_{ij} represents the connection from i point to j point. Disconnection (zero gain) should not be allowed to avoid corner topology cases, and small gain values should be assigned to those connections instead.

The observation space for a return to the critic is the ABCD matrix corresponding to the current topology. This is a shared space where both agents can access and thereby understand the effect of the actions caused by each other. Reward, however, are assigned with different values of α coefficients, using the expression from (11). The coefficient sets of topology reward and sizing reward are $[0.6, 0.1, 0.3]$ and $[0.8, 0.1, 0.1]$, respectively, as we want stability to be assured more firmly for topology construction.

IV. Experimental Results

A. MAPPO settings

TABLE I: MAPPO hyperparameters.

Parameter	Numerical value and options
Gradient clip norm	10.0
GAE λ	0.95
Discount factor, γ	0.995
Optimizer	Adam
PPO epoch	15
Mini batch number	2
Actor learning rate	7×10^{-4}
Critic learning rate	10^{-4}
Critic and actor number of layers	4
Critic and actor hidden size	1024
Entropy coefficient	0.05
Clip ratio	0.2
Number of rollout threads	50

We considered an MAPPO model modified from [11], where we built our environment upon a 4th-order modulator design. Some hyperparameters have been adjusted to fit our purpose as shown in Table I. The critic learning rate was set to a low value of 10^{-4} as we want to avoid agressive policy updates. The hidden size and number of layers constructing neural networks of the actor and critic were also increased to 1024 and 4, respectively, to enhance their learning ability. The entropy coefficient was set to a high value of 0.05 to encourage the agent exploration by adding an entropy term to the value function since we had already restricted the impact of action to prevent any major environmental change. Discount factor γ was also set to 0.995 compared to an identical value of 0.99, as we want to aim for long-term reward, especially when the environment transforms at a low speed.

Since Matlab simulation on a modulator topology consumes too much time for a reinforcement learning model that processes through millions of steps, we considered using analytical equations for most of the training. Matlab is used as a fine-tuning method at the end of training as well as a verification method.

B. Simulation results

1) Convergence behavior: Instead of using rewards for demonstrating convergence behavior, SNR can imply this since it contributes largely to the reward ($\alpha_1^{topo} = 0.6$ and $\alpha_1^{sizing} = 0.8$). Fig. 1 summarizes the convergence process of SNR, where it increases significantly at the early stages of the simulation. After 20 million steps, the value hardly

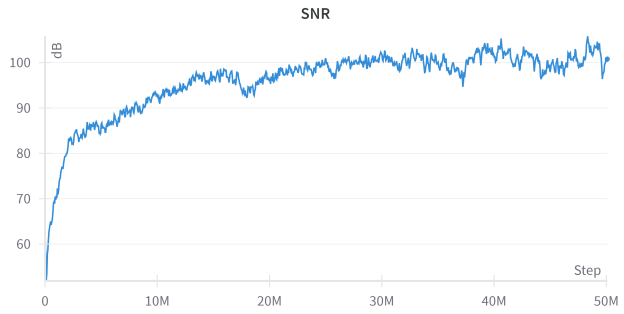

Fig. 1: Convergence behavior of SNR.

increases, although the SNR finally achieves 102dB, which is a fair amount of 10dB margin.

Instead of relying on the decision of actor only, we also created 50 rollout threads to perform parallel runs which select the most suitable action in terms of reward. In theory, this can imply less effectiveness for long-term goals, but as the simulation proceeded, we found out that the actor may get into poor topology structures and create fluctuation in the trends of SNR. As depicted in Fig. 2, for only 1 rollout threat, the model struggled to meet with convergence desire and just fluctuated around the range of 60-80dB. Using higher numbers of rollout threads, our model obtained more steady steps of convergence upon SNR.

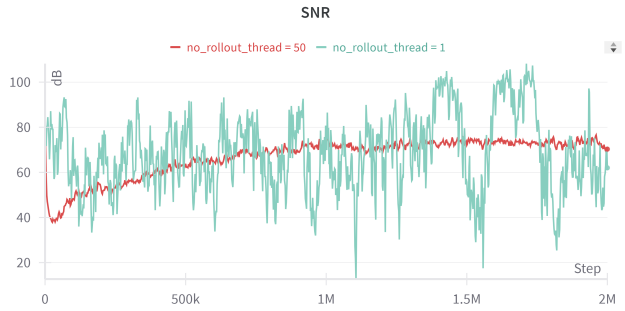

Fig. 2: Impact of rollout thread on convergence of the proposed model.

Moreover, we analyzed the impact of step size and initial starting point for model convergence. Large step size (higher step coefficient), as shown in Fig. 3 , can degrade the overall performance of the training since it creates bad actions that result in large gradients and achieved much lower SNR in comparison to models with smaller step sizes. Initial starting points for the training, as we found out, have little effect on the final convergence. Good initial structures may achieve 80dB earlier than bad ones, however, since the convergence slope hardly increased after this milestone.

2) Δ-Σ modulator performance analysis: We used the settings in Table II for Matlab simulation using topology collected from the MAPPO model. SDToolbox provided us with realistic integrators, DAC within ADC-DAC block, and

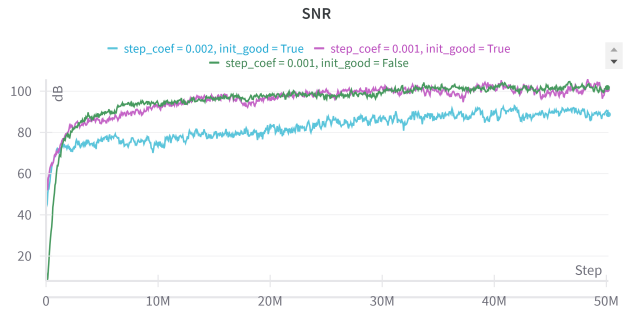

Fig. 3: Impact of step size and initial starting topology on the convergence of proposed model based on SNR.

TABLE II: Matlab simulation settings.

Parameter	Numerical value
Modulator order, N	4
Temperature, T	297K
Oversampling rate, OSR	128
Sampling rate, f_s	2.8 MHz
Input sine wave amplitude	0.9V
Quantizer bit range	15
DAC total capacitance, C_{tot}	2.5pF

a quantizer [12]. ABCD matrix of some novel topologies obtained from the MAPPO model is then converted into Matlab commands that generate a Simulink model of the corresponding modulator. This model obtained an SNR value of 98.32dB, which is marginally lower than its analytical counterpart ($\approx$ 102.4dB) due to high-complex realistic models of integrators and DAC. The system consumes $25.32\mu W$ over a bandwidth of 23kHz, which offers a FoM of 187.9dB. In comparison to other studies, we can prove our efficiency in terms of SNR as shown in Table III.

TABLE III: Performance comparison of proposed method to other design methods.

Design method	Bayesian optimization [4]	Evolutionary algorithm [5]	Mathematical analysis [6]	Proposed
SNR (dB)	98.00	73.7	76	98.32

V. Conclusion

In summary, this paper has investigated the use of multi-agent PPO for handling complex design of high-order Δ-Σ modulators in terms of both efficiency and stability. To face the challenge of convergence due to the large design space, we have carefully set up the configuration parameters, and include boundaries to prevent design topology from undesired change while training. Through comprehensive simulations in different scenarios, our proposed solution has demonstrated its efficiency based on internal and external analyses. Notably, it excels in maximizing Schreier's FoM in comparison to other remarkable studies. This research not only provides a framework for designing a stable Δ-Σ modulator but also makes way for more use of reinforcement learning in the complex environment of analog circuit design.

REFERENCES

[1] R. Baird and T. Fiez, "Stability analysis of high-order delta-sigma modulation for ADC's," *IEEE Trans. Circuits Syst. II*, vol. 41, no. 1, pp. 59–62, Jan. 1994.

[2] R. Schreier and G. C. Temes, *Understanding Delta-Sigma Data Converters*. IEEE, 2004.

[3] J. M. De la Rosa and R. Del Rio, *CMOS sigma-delta converters: Practical design guide*. John Wiley & Sons, 2013.

[4] J. Lu, Y. Li, F. Yang, L. Shang, and X. Zeng, "High-level topology synthesis method for Delta-Sigma modulators via bi-level Bayesian optimization," *IEEE Trans. Circuits Syst. II*, vol. 70, no. 12, pp. 4389–4393, Jul. 2023.

[5] J. L. A. de Melo, N. Pereira, P. V. Leitão, N. Paulino, and J. Goes, "A systematic design methodology for optimization of Sigma-Delta modulators based on an evolutionary algorithm," *IEEE Trans. Circuits Syst. I*, vol. 66, no. 9, pp. 3544–3556, Jul. 2019.

[6] H. Tang and A. Doboli, "High-level synthesis of /spl delta//spl sigma/ modulator topologies optimized for complexity, sensitivity, and power consumption," *IEEE J. Technol. Comput. Aided Design*, vol. 25, no. 3, pp. 597–607, Mar. 2006.

[7] A. Oroojlooy and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *Applied Intelligence*, vol. 53, no. 11, pp. 13 677–13 722, Oct. 2023.

[8] Y. Guan, S. Zou, H. Peng, W. Ni, Y. Sun, and H. Gao, "Cooperative UAV trajectory design for disaster area emergency communications: A multiagent PPO method," *IEEE Internet of Things J.*, vol. 11, no. 5, pp. 8848–8859, Mar. 2024.

[9] P. Kaesser, O. Ismail, J. Wagner, R. F. H. Fischer, and M. Ortmanns, "Frequency-domain analysis of reconfigured incremental Delta-Sigma ADCs on the example of the exponential phase," *IEEE Trans. Circuits Syst. I*, vol. 70, no. 11, pp. 4346–4356, Nov. 2023.

[10] V. Singh, "Lyapunov-based proof of Jury - Lee's criterion: Some appraisals," *IEEE Trans. Circuits Syst.*, vol. 32, no. 4, pp. 396–398, Apr. 1985.

[11] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative multi-agent games," in *Proc. Advances in Neural Inf. Proc. Systs.*, vol. 35, New Orleans, LA, Nov. 2022, pp. 24 611–24 624.

[12] "Mathworks - SDToolbox," accessed: 2024-10-28. [Online]. Available: http://https://www.mathworks.com/matlabcentral/fileexchange/2460-sd-toolbox