Experimental demonstration of channel order recognition in wireless communications by laser chaos time series and confidence intervals

Mitsuhiko Shimomura1a), Nicolas Chauvet1, Mikio Hasegawa2, and Makoto Naruse1

1 Department of Information Physics and Computing, Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-Ku, Tokyo, 113-8656, Japan
2 Department of Electrical Engineering, Tokyo University of Science, 6-3-1 Shinjuku, Katsushika-ku, Tokyo, 125-8585, Japan

a) shimomura@hapis.k.u-tokyo.ac.jp

Received July 10, 2021; Revised August 30, 2021; Published January 1, 2022

Abstract: High-bandwidth irregular oscillations caused by optical time-delayed feedback subjected to the laser cavity, known as laser chaos, have been investigated for various engineering applications. Recently, a fast decision-making algorithm for a multi-arm bandit problem by utilizing laser chaos time series has been demonstrated. Furthermore, the arms order recognition of the reward expectation for each arm has been successfully developed by incorporating the notion of the confidence interval regarding the reward estimate. However, in previous studies, the verification was limited to numerical experiments; real-world demonstrations were not conducted. This study experimentally demonstrated that the arm-order recognition algorithm is successfully operated in channel order recognition in wireless communications while revising the original strategy to take into account the wireless application requirements. Such accurate arm rank recognition involving non-best arms would be useful for various real-world applications such as channel bonding, among others.

Key Words: laser chaos, multi-armed bandit problem, order recognition, wireless communication, dynamic channel selection, reinforcement learning

1. Introduction
Ultrafast irregular oscillatory time series, known as laser chaos, can be generated by injecting a time-delayed optical feedback into a laser cavity [1]. Since it works at an ultrafast regime beyond the GHz order, past studies have proposed various engineering applications such as secure optical communication [2], fast physical random bit generation [3], secure key distribution using correlated
randomness [4], photonic reservoir computing [5], among others.

Recently, laser chaos has been applied for multi-arm bandit problems (MABs), which is one of the simplest problems in reinforcement learning [6]. In the MAB, there are $K (> 1)$ slot machines or arms where a player can select one of them at each step. Each machine provides a reward based on its own probability distribution, which is initially unknown to the player. The objective of the player is to maximize the accumulated rewards.

Solving MAB problems involves a difficult trade-off between exploration and exploitation. A player needs to try different arms to find the best arm, but this implies low-reward arm selections. On the other hand, to maximize rewards, a player needs to choose a specific arm being considered to be the best. However, if the arm chosen is not the best one, the player actually continues choosing the wrong arm. Therefore, the balance of exploration and exploitation is critical.

In 2018, Naruse et al. proposed an efficient decision-making algorithm using laser chaotic time series to solve bandit problems up to 64 arms, which demonstrated optimal arm recognition to maximize the accumulated reward [7]. They also found that laser chaos time series outperforms quasiperiodic signals, computer-generated pseudorandom numbers, and colored noise. Furthermore, Takeuchi et al. experimentally applied such laser-chaos-based decision-maker to dynamic channel selection problems in wireless networks [8]. Therein, the slot machines in the MAB problem are associated with different wireless channels, while the channel quality, such as throughputs, is associated with rewards.

From the viewpoint of further application in real-life problems, however, it may not always be sufficient to recognize only the best arm. For example, in the channel bonding technology that enables high-bandwidth communication by combining multiple frequency bands, appropriate recognition of electromagnetic environments is needed. That is, recognizing the rank order of all arms, not just the best arm, is required. For such arm-order recognition, Narisawa et al. revised the laser-chaos-based decision-maker by incorporating the notion of the confidence interval, by which rank-order recognition is successfully accomplished while at the same time maintaining accumulated rewards [9].

In the present study, we experimentally applied the laser-chaos-based ranking strategy to the channel selection problem in congested wireless communication environments. The target is to recognize the order of throughputs among multiple channels, which is four in the experiment below, while achieving higher throughput. We revised the algorithm in [9] concerning the characteristics of wireless communications. In Section 2, we will introduce previously studied principles in a detailed manner regarding highly relevant aspects of the present study. Section 3 describes the experimental methods, followed by experimental results and analysis in Section 4. Section 5 concludes the paper.

Fig. 1. Channel selection problem in wireless communication based on MAB algorithm using laser chaos.

2. Previous studies

2.1 MAB algorithm using chaotic time series

The previously proposed MAB algorithm using chaotic laser time series [7], referred to as Algorithm I hereafter, utilized time-domain multiplexing in realizing scalability of arms. Let the number of arms be $K = 2^M$, where $M$ is a natural number. This method can be summarized in the following three steps.
STEP 1  We determine a $M$ bit sequence $S_i (i = 1, \ldots, M)$ by comparing signal values of the laser chaos and the thresholds. The first bit $S_1$ is determined by a comparison between laser chaos signal level at the timing $t_1$, which is represented by $s(t_1)$ and the threshold $TH_1$:

$$S_1 = \begin{cases} 
1, & \text{if } s(t_1) \geq TH_1 \\
0, & \text{if } s(t_1) < TH_1.
\end{cases}$$

Similarly, for the timing $t = t_m (m = 2, \ldots, M)$, a bit $S_m$ is determined by comparing a chaotic signal $s(t_m)$ with the threshold $TH_{m,S_1\cdots S_{m-1}}$. Note that the threshold used for the signal $s(t_m)$ at time $t_m$ depends on the previous bit sequence $S_1\cdots S_{m-1}$ from time $t_1$ to $t_{m-1}$. Thus, there are $2^{m-1}$ kinds of thresholds to be compared at time $t_m$.

STEP 2  The resultant bit sequence $S_1\cdots S_M$ obtained in STEP 1 indicates the identity of the slot machine as a binary number. The player plays the selected slot machine and gets the reward according to the reward probability of the slot machine.

STEP 3  Finally, based on the reward obtained in STEP 2, the thresholds are updated; here, we describe the concept of the update briefly. Details are found in [7]. If the selected slot machine yields a reward, adjust the thresholds in such a way that that slot machine is more highly likely selected. Conversely, if the machine does not yield a reward, then adjust the thresholds so that that slot machine is less likely selected.

Figure 2 illustrates an example of a four-armed bandit problem. Assume that a bit sequence $S_1S_2 = 10$ is derived in STEP 1, then a player chooses and plays the machine numbered 2 = (10)$_2$. In STEP 3, the player will update the thresholds $TH_1$ and $TH_{2,1}$ because these two thresholds are involved in deciding the arm. If the player gets a reward, $TH_1$ is decreased, and $TH_{2,1}$ is increased because of increasing the likelihood of selecting $S_1S_2 = 10$ in the subsequent plays. On the contrary, if the player does not get a reward, $TH_1$ and $TH_{2,1}$ are increased and decreased, respectively, so that the likelihood of choosing $S_1S_2 = 10$ decreases in the subsequent plays.

![Fig. 2. Example of 4-arm bandit problem. The thresholds $TH_1$ and $TH_{2,1}$ are involved in deciding the arm 2 = (10)$_2$.](image)

As mentioned in Section 1, the above-mentioned principle successfully resolved MAB problems up to 64 arms and exhibits faster performances compared with pseudorandom numbers [7]. Furthermore, Takeuchi et al. experimentally implemented such a strategy to the channel selection problem in wireless communications [8].

2.2 Order recognition algorithm using chaotic time series

Narisawa et al. extended Algorithm I so that one can recognize the order or ranking of the slot machines, not just the top-ranked slot machine, by introducing the confidence interval that represents the estimation accuracy of reward expectations, which we call Algorithm II hereafter [9]. The key point of this method is to control the degree of exploration adaptively based on the confidence interval.

Before describing the algorithm in detail, we define several variables.
• $K$: Number of arms, where $K = 2^M$ and $M$ is a natural number.

• $A(n)$: Arm selected at time step $n$ ($a(n)$ is the observed value).

• $X_i(n)$: Obtained reward from arm $i$ at time step $n$ (independent at each time step. $x_i(n)$ is the observed value).

• $T_i(n)$: Number of selections of arm $i$ by the end of time step $n$ ($t_i(n)$ is the observed value).

We estimate the arm order at time step $n$ via the estimated reward probabilities, which are given by

$$\hat{\mu}_i(n) = \frac{R_i(n)}{T_i(n)}, \text{ where } R_i(n) := \sum_{s=1}^{n} X_i(s) \cdot \mathbb{I}[A(s) = i]$$

where $i$ is the index number of slot machines ranging from 0 to $K - 1$. $\mathbb{I}[]$ is a function that returns 1 if the expression in parentheses holds and 0 otherwise. That is, $R_i(n)$ means the number of times that a reward is dispensed until time $n$ from the machine $i$.

**Confidence intervals.** In Algorithm II, each threshold $TH_{j,b_1\cdots b_{j-1}}(j \in \{1, \ldots, M\}, b_1, \ldots, b_{j-1} \in \{0, 1\})$ and $z \in \{0, 1\}$ has the following associated values $\hat{P}_{j,b_1\cdots b_{j-1}}(z;n)$ and $C_{j,b_1\cdots b_{j-1}}(z;n)$. Firstly, $\hat{P}_{j,b_1\cdots b_{j-1}}(z;n)$ is defined by

$$\hat{P}_{j,b_1\cdots b_{j-1}}(z;n) := \frac{\sum_{i \in I_{j,b_1\cdots b_{j-1}}(z)} R_i(n)}{\sum_{i \in I_{j,b_1\cdots b_{j-1}}(z)} T_i(n)}$$

where

$$I_{j,b_1\cdots b_{j-1}}(z) := \{i \mid \text{the first } j \text{ bits of bin}(i) \text{ match } b_1 \cdots b_{j-1}z\}$$

with bin$(i)$ the binary representation of $i$. For example, in the case of a four-armed bandit problem shown in Fig. 3,

$$I_1(0) = \{0, 1\}, \ I_1(1) = \{2, 3\}, \ I_2(0) = \{0\}, \ I_2(1) = \{1\}.$$  

Namely, $\hat{P}_{j,b_1\cdots b_{j-1}}(z;n)$ quantifies the probability of observing rewards in the group of slot machines specified by $I_{j,b_1\cdots b_{j-1}}(z)$.

![Fig. 3. Example of set-valued functions $I_{j,b_1\cdots b_{j-1}}(z)$.](image)

Meanwhile, $C_{j,b_1\cdots b_{j-1}}(z;n)$ is defined by

$$C_{j,b_1\cdots b_{j-1}}(z;n) := \sqrt{\frac{\log n}{2 \sum_{i \in I_{j,b_1\cdots b_{j-1}}(z)} T_i(n)}},$$

which represents the confidence interval of the estimated value of $\hat{P}_{j,b_1\cdots b_{j-1}}(z;n)$.  

104
Coarseness/fineness of exploration adjustments. Algorithm II is the same with Algorithm I until STEP 2. STEP 3 is critically different as the confidence interval plays a crucial role. Consider the following two intervals:

\[
\hat{P}(0; n) - C(0; n), \hat{P}(0; n) + C(0; n) \quad \text{and} \quad \hat{P}(1; n) - C(1; n), \hat{P}(1; n) + C(1; n)
\]

which manifest the expected range of \(\hat{P}(0; n)\) and \(\hat{P}(1; n)\), respectively. Figure 4 schematically illustrates two contrasting situations regarding \(\hat{P}(0; n)\) and \(\hat{P}(1; n)\).

In Fig. 4(a), although \(\hat{P}(1; n)\) is greater than \(\hat{P}(0; n)\), the main aspect is that two intervals are overlapped. That is to say, the estimated relation that \(\hat{P}(1; n) > \hat{P}(0; n)\) can be wrong, suggesting that the exploration must be conducted carefully. Hence, it is necessary to decrease the threshold’s updating amount to execute the exploration process in a finer manner.

By contrast, if the two intervals are not overlapped as shown in Fig. 4(b), the order of \(\hat{P}(0; n)\) and \(\hat{P}(1; n)\) is less likely to be incorrectly estimated. Therefore, it is acceptable to increase the threshold’s updating amount to execute the exploration process more coarsely.

3. Experimental demonstration

3.1 Experimental setup

We applied Algorithm II to experimental wireless communication environments and evaluated the accuracy of channel order recognition and the average communication throughput. More specifically, the experiment examined a four-armed bandit problem using 4 channels (36 ch, 40 ch, 44 ch, and 48 ch) of IEEE 802.11-a-based wireless local network. The primary experimental elements and their specifications are summarized below.

**Host device.** A microprocessor unit (Raspberry PI 3 Model B) served as the access points (AP) for each channel to which the client devices connected.

**Client device.** A personal computer (HP ProBook 430 g6, OS: Ubuntu) served as a client device where the proposed channel order recognition principle was operated. The client device is connected to the host device or AP. In the client device, the average value of Transmission Control Protocol (TCP) throughput during three seconds is measured by using the iperf command [10] in each step.

**Load AP.** A router (Buffalo, AirStation; WCR-1166DS) loads the AP provided by the host device by providing an AP to the same channel. We used four routers to load all the channels provided by the host device.

**Load device.** Personal computers (HP ProBook 430 g6, OS: Ubuntu) were utilized to provide traffic to the Load AP to intentionally cause interferences in the communications between the host AP and the client devices. We adjusted the weight of the load by using the ping command with options about the packet size and the frequency.
If the amount of loads applied to each channel is largely different from each other, the channel order recognition is easy. Hence, in this study, we configured the traffic in such a way that the throughput of each channel was close to each other. The actual traffic conditions in the experiment were summarized in Table I.

| Ground Truth | packet size [byte] | ping interval [sec] |
|--------------|--------------------|---------------------|
| 1st          | 0                  | ∞                   |
| 2nd          | 100                | 0.1                 |
| 3rd          | 100                | 0.01                |
| 4th          | 100                | 0.001               |

3.2 Reward function

In the previous experimental demonstration of the dynamic channel selection in wireless communications based on Algorithm I [8], the reward function was given by

\[ X_i(1) := \begin{cases} 
1 & \text{if } A(1) = i \\
0 & \text{otherwise} 
\end{cases} \]

if \( n = 1 \), and

\[ X_i(n) := \begin{cases} 
1 & \text{if } A(n) = i \text{ and } TP(n) \geq \sum_{s=1}^{n-1} TP(s)/(n-1) \\
0 & \text{otherwise} 
\end{cases} \]

if \( n \geq 2 \) where \( TP(n) \) is the value of TCP throughput at time step \( n \). This reward function returns a non-zero reward only if the throughput value at time step \( n \) is equal to or greater than the average throughput values in the previous steps. With such reward function, Algorithm I and II select the fastest channel most frequently; hence their average throughput values asymptotically approaches the throughput value of the best channel. However, from the objective of Algorithm II, this indicates that a player hardly receives a reward when selecting a channel other than the optimal one, meaning that estimates of reward and confidence intervals do not work effectively.

To resolve this problem, in the present study, we design the reward function as follows:

\[ X_i(n) := \begin{cases} 
TP(n), & \text{if } A(n) = i \\
0, & \text{otherwise} 
\end{cases} \]

Namely, we directly utilize the throughput value as the reward. By such reward function, the order recognition of Algorithm II works efficiently because the reward value is always positive.

3.3 Updating thresholds

Furthermore, we also made some changes in the updating method of thresholds because the original Algorithm II in [9] sometimes showed undesirable behavior in the experiment. We explain such an unwanted situation using an example shown in Fig. 5; here the ground truth of the throughputs is in the order of 36 ch, 48 ch, 40 ch, and 44 ch. Now, suppose that the player selects 48 ch, which is the second-best channel, and the observed throughput value is lower than the average throughput over time. Based on the threshold revision policy and the definition of the reward defined above, \( TH_{2,1} \) is updated so that 48 ch is less likely chosen in the subsequent selections. Note, however, that 48 ch exhibits actually greater throughput than 44 ch. In such a way, the original policy does not work as it is.

Hence, we revised the original Algorithm II as the following. At this time, we reconfigure each threshold \( TH_{j,b_1\cdots b_{j-1}} \) \((j \in \{1, \ldots, M\})\) based on the average reward value \( \hat{P}_{j,b_1\cdots b_{j-2}}(b_{j-1};n) \). If the reward \( r \) is greater than or equal to \( \hat{P}_{j,b_1\cdots b_{j-2}}(b_{j-1};n) \), \( TH_{j,b_1\cdots b_{j-1}} \) is updated so that the player is
likely to select the channel in the subsequent connections. On the contrary, if the reward $r$ is less than $\hat{P}_{j,b_{1} \cdots b_{j-1}}(n)$, $TH_{j,b_{1} \cdots b_{j-1}}$ is revised so that it becomes harder for the player to select the channel in the future. By using such a revised method, in the case of the example above, 48 ch is more easily selected than 44 ch.

### 3.4 Performance comparison

In this study, we compared the performances by the following three methods:

- **Round-Robin**: Select all channels in a cyclic regular order ($36 \rightarrow 40 \rightarrow 44 \rightarrow 48 \rightarrow 36 \rightarrow \cdots$).

- **Chaos-CI (A)**: Proposed method ($\beta = 2$).

- **Chaos-CI (B)**: Proposed method ($\beta = 10$).

Note that CI represents “confidence interval” and $\beta$ is a hyperparameter inherent in Algorithm II [9], which represents the amount of increase or decrease for the updated threshold value. When $\beta$ is small, the change in the updated threshold value is small, whereas when $\beta$ is large, the amount of change is large.

We conducted 100 episodes (each episode consists of 100 steps) of each method. Since the behavior of the Chaos-CI algorithm changes according to the ranks of the channels belonging to each $I_{j,b_{1} \cdots b_{j-1}}$, we tested three patterns shown in Table II. The load for each rank is shown in Table I.

### Table II. The channel corresponding to each arm and the true order (or ground truth) of the channels for each pattern.

| arm   | channel | Pattern 1 | Pattern 2 | Pattern 3 |
|-------|---------|-----------|-----------|-----------|
| (00)$_2$ | 36 ch   | 1st       | 1st       | 1st       |
| (01)$_2$ | 40 ch   | 2nd       | 4th       | 3rd       |
| (10)$_2$ | 44 ch   | 3rd       | 2nd       | 4th       |
| (11)$_2$ | 48 ch   | 4th       | 3rd       | 2nd       |

### 3.5 Laser chaos

In this study, we used the signal dataset generated in the previous study [9], instead of building an optical system. We used the same signal dataset in all pattern experiments, but we used the different data in each episode in the same pattern. We show an overview of the experimental system used in the study. A semiconductor laser (NTT Electronics, KELD1C5GAA) with a center wavelength of 1547.785 nm was coupled with a polarization-maintaining coupler. They generated high-frequency chaotic oscillations of optical intensity by providing delayed optical feedback to the laser via a variable fiber reflector. The fiber length between the laser and reflector was 4.55 m, corresponding to a feedback delay time of 43.8 ns. They used a photodetector (New Focus, 1474-A) to detect the output light and a digital oscilloscope (Tektronics, DPO73304D) to sample at 100 G sample/s.
4. Experimental results

Correct order recognition.

To quantify the order recognition, now we introduce the notion of what we call the correct order recognition. Here, correct order at time $n$ means that when the estimated reward probability $\hat{\mu}_i(n)$ defined by Eq. (1) is arranged in the descending order, it agrees with the ground truth order shown in Table II. Based on this, we say that the correct order recognition is completed by time step $n$ by satisfying the fact that for any time step $m$ which is larger than $n$, $\hat{\mu}_i(m)$ is correctly ordered as the ground truth. In other words, the achievement of the correct order recognition means that the order of all the channels is correctly recognized in the step, and the estimated order does not change until the end of the episode.

The blue, orange, and green curves in Fig. 6(a) show the number of episodes such that the correct order recognition is completed by time step $n$ for Round-Robin, Chaos-CI (A), and Chaos-CI (B), respectively, when the ground truth was configured as Pattern 1 in Table II. We can clearly observe that the correct order recognition is accomplished more quickly by Chaos-CI (A) and (B) compared with Round-Robin, while the difference between Chaos-CI (A) and (B) is negligible. Also, with the ground truth configurations of Patterns 2 and 3, the rate of the correct order recognition evolutions exhibit similar behavior as with Pattern 1. The difference between the results of Round-Robin and proposed methods was caused by whether they use information about obtained reward or not. Round-Robin assures periodic observation of the individual channel traffic, so environmental noise easily prevents the accurate estimation of the throughput of each channel. Therefore, the correct order recognition evolves slowly. Conversely, Chaos-CI (A) and (B) are based on bandit algorithms combined with the notion of confidence intervals. Therefore, Chaos-CI (A) and (B) promptly achieve the accurate estimation of the throughput of each channel; namely, fast correct order recognition is realized.

Accumulated reward per step.

The blue, orange, and green curves in Fig. 6(b) show the time evolution of the accumulated rewards per step by Round-Robin, Chaos-CI (A), and Chaos-CI (B), respectively, when the ground truth was configured as Pattern 1 in Table II. Here, the accumulated reward per step (ARPS) at step $n$, noted $arps(n)$, is defined by the following formula:

$$arps(n) := \frac{\sum_{s=1}^{n} x(s)}{n}$$

where $x(s)$ is the mean of the rewards obtained for each episode at time $s$. In the beginning, when the elapsed time cycle $n$ is less than approximately 30, Round-Robin exhibits higher ARPS compared with Chaos-CI (A) and (B), whereas ARPS with Round-Robin does not increase beyond $n = 10$. By contrast, ARPSs with Chaos-CI (A) and (B) monotonically increase beyond step 30 and are greater than that with Round-Robin. Such behavior stems naturally from the principle of these strategies. In the initial moments, Round-Robin surely can obtain the greatest reward every four-cycle, which is the
Fig. 7. The selection rate of each channel up to time step $n$ with (a) Round-Robin, (b) Chaos-CI (A), and (c) Chaos-CI (B). (Pattern 1)

Fig. 8. At time step $n$, (a) number of successful episodes of order recognition, (b) cumulative reward per one step. (Pattern 2)

Fig. 9. The selection rate of each channel up to time step $n$ with (a) Round-Robin, (b) Chaos-CI (A), and (c) Chaos-CI (B). (Pattern 2)

reason for the oscillatory time evolution of the blue curve in Fig. 6(b). By contrast, once Chaos-CI (A) and (B) successfully find the best selection (as well as recognize the correct order), they exploit that specific selection, which increases ARPS. Also, with the ground truth configurations of Pattern 2 and 3, the ARPS exhibit similar behavior with that of Pattern 1.

Actual channel selection.

Figures 7(a), (b), and (c) summarize the evolution of the ratio of the number of times each channel is actually selected until time step $n$ when the order recognition strategy was Round-Robin, Chaos-CI (A), and Chaos-CI (B), respectively, when the ground truth was configured according to Pattern 1 in Table II. With Round-Robin, each channel is equally selected whereas with Chaos-CI (A) and (B), 36 ch, which is the best channel for Pattern 1, is selected most frequently at the time step of 100. Interestingly, the correct order is already recognized after step 30 according to Fig. 6(a). Hence, the increase of the actual selection rate of 36 ch contributes to increase the rewards while maintaining correct order recognition. Similar trend is also observed when the ground truth is configured in Pattern 2.

With Pattern 3, however, the actual channel selection evolves in a different manner as summarized in Fig. 11. Whereas Fig. 11(a) obtained with Round-Robin is the same with the former cases, both Chaos-CI (A) and Chaos-CI (B) actually select 48 ch most often, which is the second-best channel,
and the first-order channel (36 ch) is selected in rank second. Note, however, that accurate recognition of the channel order, which is the primary focus of the present study, is accomplished for Pattern 3. This is because order recognition and channel selection are independently executed in the proposed method; hence the best channel is not always selected intensively depending on the given traffic. Fixing such a problem is an interesting future topic.

5. Conclusion
We experimentally demonstrated the order recognition strategy in MAB problems that combines chaotic laser time series and confidence interval in the channel order recognition problem for wireless communication systems. Here, we developed a new reward function and new adjustment mechanisms of thresholds to take into account the requirement of the wireless communication under study. The proposed method, Chaos-CI, successfully recognized the order at high speed, faster than the Round-Robin approach, even in an environment where the apparent order may change easily from statistical fluctuations. We also found that the proposed Chaos-CI method has a small parameter susceptibility. It implies that there would be no need to conduct fine parameter tuning individually for each problem. In the meantime, even though the order recognition was successful with Chaos-CI, we observed a case, what we call Pattern 3, where the proposed methods sometimes selected not-the-best channel preferentially, when it should have chosen the best one. This indicates that there is a room for further improvement in the channel selection algorithm. In addition, we kept the load applied to each channel (as shown in Table I) being constant during the experiment. In view of applications, dynamically changing electromagnetic environments have to be examined.

Acknowledgments
The authors appreciate Shungo Takeuchi for his kind and critical supports for experimental system constructions. A part of this study was supported by JST CREST (JPMJCR17N2) and JSPS KAKENHI (20H00233).
References

[1] A. Uchida, *Optical communication with chaotic lasers: applications of nonlinear dynamics and synchronization*. John Wiley & Sons, 2012.

[2] I. Fischer, Y. Liu, and P. Davis, “Synchronization of chaotic semiconductor laser dynamics on subnanosecond time scales and its potential for chaos communication,” *Physical Review A*, vol. 62, no. 1, 011801, 2000.

[3] A. Uchida, K. Amano, M. Inoue, K. Hirano, S. Naito, H. Soneya, I. Oowada, T. Kurashige, M. Shiki, and S. Yoshimori, “Fast physical random bit generation with chaotic semiconductor lasers,” *Nature Photonics*, vol. 2, no. 12, pp. 728–732, 2008.

[4] K. Yoshimura, J. Muramatsu, P. Davis, T. Harayama, H. Okumura, S. Morikatsu, H. Aida, and A. Uchida, “Secure key distribution using correlated randomness in lasers driven by common random light,” *Physical Review Letters*, vol. 108, no. 7, 070602, 2012.

[5] L. Larger, M.C. Soriano, D. Brunner, L. Appeltant, J.M. Gutierrez, L. Pesquera, C.R. Mirasso, and I. Fischer, “Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing,” *Opt. Express*, vol. 20, no. 3, pp. 3241–3249, 2012.

[6] S.R. Sutton and B.G. Andrew. *Reinforcement learning: An introduction*. MIT press, 2018.

[7] M. Naruse, T. Mihana, H. Hori, H. Saigo, K. Okamura, M. Hasegawa, and A. Uchida, “Scalable photonic reinforcement learning by time-division multiplexing of laser chaos,” *Scientific Reports*, vol. 8, 10890, 2018.

[8] S. Takeuchi, M. Hasegawa, K. Kanno, A. Uchida, N. Chauvet, and M. Naruse, “Dynamic channel selection in wireless communications via a multi-armed bandit algorithm using laser chaos time series,” *Scientific Reports*, vol. 10, 1574, 2020.

[9] N. Narisawa, N. Chauvet, M. Hasegawa, and M. Naruse, “Arm order recognition in multi-armed bandit problem with laser chaos time series,” *Scientific Reports*, vol. 11, 4459, 2021.

[10] M. Mortimer, “iperf3 Documentation,” 2018.