Self-Configurable Receiver for Underwater Acoustic Communications using Bayesian Optimization

Cheikh Malainine Cheikh Melainine, François-Xavier Socheleau, Arnaud Jarrot

Self-Configurable Receiver for Underwater Acoustic Communications using Bayesian Optimization

Cheikh M. Cheikh Melainine*†, François-Xavier Socheleau†, Arnaud Jarrot*
* Schlumberger Riboud Product Center, SLB, Clamart, France
†IMT Atlantique, Lab-STICC UMR CNRS 6285, Brest, France

***Abstract*—Underwater communication receivers usually rely on the fine-tuning of numerous hyperparameters to perform optimally. This fine-tuning process is challenging and time-consuming, and must be carried out by domain experts. Using a receiver with a decision-feedback equalizer (DFE), it is shown that finding optimal hyperparameters can be formalized as a black-box optimization problem for which an objective function is minimized. This objective function is designed to address the degenerate states of the DFE, known as "DFE hallucinations", where the DFE produces small apparent mean square errors yet very high bit error rate. We propose a method for automating the tuning of hyperparameters using a tree-structured parzen estimator (TPE) approach, an algorithm belonging to the large family of Bayesian optimization algorithms. Results obtained from synthetic and real channels demonstrate the efficiency of the method, allowing significant reduction of the packet error rate.**

***Index Terms*—Decision-Feedback Equalizer (DFE), Tree-Structured Parzen Estimator (TPE), Underwater Acoustic Communications, Bayesian Optimization**

I. Introduction

Receivers in underwater acoustic communications (UAC) systems encounter a variety of challenges including multi-path reflections and Doppler, which depend on geographical locations and operational conditions. Furthermore, a channel itself undergoes changes during transmission, sometimes in an unpredictable manner [1]. For optimal performance, receivers must be robust to the changes of conditions and channel variability. To achieve this, the hyperparameters of the receivers' algorithms must be set with great care. Examples of hyperparameters that have a strong impact on the performance of the UAC receiver are the length of the feedforward diversity combining filters, the adaptation rate of the adaptive equalizer and the bandwidth of the time and phase recovery loop. Setting these parameters can be a complex task given that they can be coupled [2] [3]. This can be made even more more difficult depending on the choice of the adaptive algorithm, which might require additional sets of hyperparameters [4] [5]. The tuning of these values is commonly determined empirically by domain experts to ensure that the chosen parameters provide the best average performance over a given range of operating conditions. Once selected, they are consequently hardcoded. This process of gathering examples and testing sets of parameters can be costly and time-consuming, and it does not guarantee optimal performance in the event of changes in operational conditions.

Without precise and extensive prior knowledge of the operating conditions, the task of identifying optimal hyperparameters is akin to searching for a needle in a haystack. To the best of our knowledge, there is no general consensus on techniques that allow for the selection of optimal values. This paper presents a method for automating the process of tuning the hyperparameters of a UAC receiver without any prior knowledge. Bayesian optimization is employed for this purpose, specifically the tree-structured parzen estimator approach [6]. This work focuses on the receiver component of an UAC, where we examine the impact of automated hyperparameter tuning on the performance of a DFE implemented within the receiver.

This paper is organized as follows. Section II provides a brief introduction to Bayesian optimization and the TPE approach. Section III presents the DFE architecture and an objective function giving access to the optimal hyperparameters of a DFE. Finally, Section IV presents the performance of the self-configurable receiver. The emphasis is on simulated channels using the UAC parametric MAXENT model [7] as well as a real channel recorded in the freshwater Loch Ness, in Scotland.

II. Bayesian Optimization

A. Objective model

Bayesian optimization (BO) algorithms are widely used to tackle the problem of finding an optimal solution to objective functions that are expensive to compute, do not provide an efficient mechanism for computing their gradient, and for which there is no useful analytic expression [8]. BO was crafted for black-box optimization, which can be formalized as follows:

$$\boldsymbol{\theta}^* \triangleq \underset{\boldsymbol{\theta}\in\Theta}{\arg\min}(f(\boldsymbol{\theta})) \tag{1}$$

where f is the objective function, Θ is the domain of the hyperparameters $\boldsymbol{\theta}$, and $\boldsymbol{\theta}^*$ is the target optimal hyperparameter. Given a budget of N iterations, the main steps of a BO algorithm are

- First, a warm-up session in which a data set with n– usually random–hyperparameters and associated observations is initialized ($D_0 \triangleq \{(\boldsymbol{\theta}_i, f(\boldsymbol{\theta}_i))\}_{i=0}^{n}$).
- In the remaining $N-n$ iterations, a careful decision is made about the next observation point $\boldsymbol{\theta}_j$, given D_{j-1}, using a policy π, and D_j is updated with $\{(\boldsymbol{\theta}_j, f(\boldsymbol{\theta}_j))\}$, where j is the iteration index.

In BO, the unknown quantities are treated as random. f is thus modeled as an infinite collection of Gaussian random variables $f_{\boldsymbol{\theta}}$ representing a score associated with $\boldsymbol{\theta}$.

B. Tree-Structured Parzen Estimator

The heart of a BO algorithm is its policy [9]. Based on previous observations, a policy suggests the next hyperparameter $\boldsymbol{\theta}_j$ at iteration j–theoretically better than those observed in D_{j-1}—thus approaching the optimal hyperparameters. The policy maximizes an acquisition function such as the probability of improvement (PI):

$$\alpha_{PI}(\boldsymbol{\theta}, D_{j-1}) \triangleq \Pr(f_{\boldsymbol{\theta}} \leq y^*) \tag{2}$$

where y^* is the best score observed in D_{j-1}. This acquisition function gives higher scores to hyperparameters $\boldsymbol{\theta}$ that are likely to yield scores better than or equal to y^*.
The resulting policy is the following:

$$\pi_{PI}(D_{j-1}) \triangleq \arg\max_{\boldsymbol{\theta}\in\Theta}(\alpha(\boldsymbol{\theta}, D_{j-1})) \tag{3}$$

The tree-structured parzen estimator (TPE) is probably the most popular BO algorithm [6]. The TPE approach essentially segments the dataset D_{j-1} into a good population D^g_{j-1} and a bad population D^b_{j-1} using a quantile γ. These two populations respectively contain the $\gamma \times L$ best elements of D_{j-1} in terms of $f(\boldsymbol{\theta})$ and the remaining $(1-\gamma)\times L$ elements. The random variable S, corresponding to the next observation hyperparameter $\boldsymbol{\theta}$, is then introduced, allowing the estimation of the following probability density functions [10]:

$$p(S \in D^g_{j-1}) = \frac{1}{N_g} \sum_{\boldsymbol{\theta}\in D^g_{j-1}} \mathbf{k}(S, \boldsymbol{\theta}) \tag{4}$$

where $\mathbf{k}(S, \boldsymbol{\theta}_i)$ is usually a Gaussian kernel. $p(S \in D^b_{j-1})$ is defined analogously to (4). It has been shown in [11] [12] that optimizing PI is equivalent to optimizing $r(S, D_{j-1}) = \frac{p(S\in D^g_{j-1})}{p(S\in D^b_{j-1})}$. The TPE policy is thus expressed as follows:

$$\boldsymbol{\theta}_j = \pi_r(D) \triangleq \arg\max_{\boldsymbol{\theta}\in\Theta} (r(S, D_{j-1})) \tag{5}$$

This policy has the characteristic of being less computationally expensive than the original PI (2)In practice, $\boldsymbol{\theta}_j$ can be determined by avoiding an optimization. In this case, n_c candidates are drawn from $p(S \in D^g_{j-1})$, and the choice of $\boldsymbol{\theta}_j$ corresponds to the candidate with the highest $r(S, D_{j-1})$. For this study, $n_c = 24$, as in [10].

III. DFE Tuning

A. Hyperparameters

In this work, we apply the TPE algorithm for the tuning of equalization hyperparameters. Equalization is a crucial component of underwater receivers, compensating for distortions caused by the propagation channel. We use a DFE whose architecture is illustrated in Fig. 1. The transversal filter $\mathbf{A}$ is used for precursor inter-symbol interference (ISI) mitigation, supported by a feedback equalizer $\mathbf{B}$ to deal with residual postcursor ISI.

To track channel variability, the filter taps are updated for each decided symbol using a variant of the least minimum square (LMS) algorithm. One of the most used algorithms is the improved proportionate normalized LMS (IPNLMS) [5], which incorporates both the rapid convergence of the proportionate normalized LMS in sparse channels [4] and the robustness of the normalized LMS in dispersive channels.

The filter update is performed as follows:

$$\mathbf{A}(k) = \mathbf{A}(k-1) + \mu_A \frac{\mathbf{G}_A(k-1)\mathbf{x}(k)e(k)^*}{\mathbf{x}(k)^\dagger \mathbf{G}_A(k-1)\mathbf{x}(k) + \delta_A} e^{j\phi(k-1)} \tag{6}$$

$$\mathbf{B}(k) = \mathbf{B}(k-1) - \mu_B \frac{\mathbf{G}_B(k-1)\hat{\mathbf{d}}(k)e(k)^*}{\hat{\mathbf{d}}(k)^\dagger \mathbf{G}_B(k-1)\hat{\mathbf{d}}(k) + \delta_B} \tag{7}$$

where μ_A and μ_B are the step sizes of the feedforward and feedback filters of $\mathbf{A}$ and $\mathbf{B}$ and k is the symbol index.
L_A and L_B are the filter lengths of $\mathbf{A}$ and $\mathbf{B}$. $\mathbf{x}(k) = [x(k), x(k-1), \ldots, x(k-L_A+1)]^T$ are the input samples, $\hat{\mathbf{d}}(k) = [\hat{d}(k-1), \hat{d}(k-2), \ldots, \hat{d}(k-L_B)]^T$ are the decided symbols, and $e(k) = \hat{d}(k) - z(k)$ the error signal, $z(k)$ being the soft symbol.
$\mathbf{G}_A$ is a diagonal matrix, the elements of which are

$$g^{(i)}_A(k) \triangleq \frac{1-\beta}{2L_A} + (1+\beta)\frac{|\ \mathbf{A}_i(k)\ |}{2\ ||\ \mathbf{A}(k)\ ||_1 + \epsilon} \tag{8}$$

$\mathbf{A}_i$ is the ith tap and ϵ is a regularization term. $\mathbf{G}_B$ is defined similarly for the filter $\mathbf{B}$.
These matrices give estimates of the contribution of each tap in a filter based on its magnitude relative to other taps. Tuning β thus makes the update more or less dependent on these magnitudes. In particular, $\beta = -1$ corresponds to the NLMS case, while $\beta = 1$ updates each tap purely on its magnitude. These update equations have several hyperparameters that are optimized:

- $\beta \in [-1, 1]$: channel sparseness;
- $\mu_A \in [10^{-5}, 1]$: update step of the feedforward filter;
- $\mu_B \in [10^{-5}, 1]$: update step of the feedback filter;
- $L_A \in [10, 3000]$: length of the feedforward filter;
- $L_B \in [0, 1000]$: length of the feedback filter.

The vector of the DFE hyperparameters is therefore denoted as $\boldsymbol{\theta} \triangleq [L_A, L_B, \beta, \mu_A, \mu_B]^T$. Note that μ_A, μ_B, and β are real numbers, while the filter lengths are integers numbers.

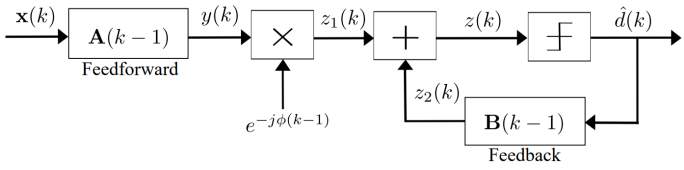

Fig. 1. Decision-feedback equalizer structure.

B. Objective function

Finding the optimal hyperparameters $\boldsymbol{\theta} \in \Theta$ is equivalent to finding the global optimum of an objective function that

measures the performance of the equalizer when it operates under $\boldsymbol{\theta}$. The objective is to identify the optimal $\boldsymbol{\theta}$ for a specific packet. A simple objective function is $f : \Theta \to \Re$:

$$f(\boldsymbol{\theta}) \triangleq \frac{1}{N_s}\sum_{k=1}^{N_s}|e(k)|^2 \triangleq \frac{1}{N_s}\sum_{k=1}^{N_s}|\hat{d}(k) - z(k)|^2 \qquad (9)$$

f is the average mean squared error (MSE) computed for a packet of size N_s.

A major challenge with the use of an MSE error term is the degenerate DFE state coined as "DFE hallucination" illustrated in Fig. 2. In this state, the norm of the feedforward output norms decreases toward zero, while that of the feedback filter output increases toward the norm of the constellation.

The hyperparameters $\boldsymbol{\theta}$ that produce hallucinations are those with the best $f(\boldsymbol{\theta})$ in decision-directed mode. In fact, the decision device converges to a state where decisions are made only by the feedback filter and no longer by the feedforward filter. DFE hallucinations need to be detected and penalized by the objective function. The detection is done by setting a threshold τ such that if $\mu_m(\tau) = \sum_{k=N_s-m}^{N_s}|z_1(k)|^2 \leq \tau$, the objective function is strongly penalized. m is the number of the feedforward outputs used to test this hypothesis. Detecting hallucinations in long packets can be effectively achieved by setting a small τ, as suggested by the trends in Fig. 2. Selecting the appropriate threshold τ involves balancing the risk of false positives, which increases with higher threshold values, against the risk of missed detections, especially for short packets, which is more likely with lower threshold values. For of our simulation, we opted for a threshold of $\tau = 0.1$, which was chosen experimentally for our large packet size (3375 symbols).

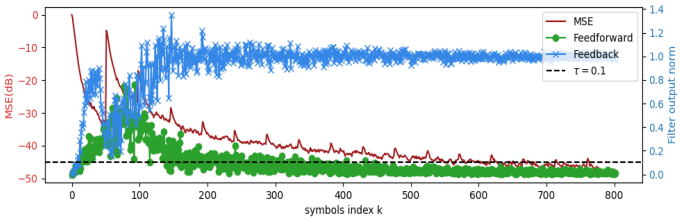

Fig. 2. DFE hallucination example. The curve marked with –o– represents the feedforward L2 output norm, whereas the curve marked with –*– corresponds to the feedback output L2 norm. The red curve — shows the MSE evolution corresponding to this hallucination.

IV. Numerical results

This study focuses on optimizing the lengths of equalizer filters and the parameters governing the adaptive filter updates. Addressing additional critical hyperparameters, including time and clock recovery loops, is not being investigated here but will be part of future work.

Furthermore, the algorithmic complexity of the TPE procedure is $O(N \times \max(L_A, L_B))$, making it challenging to execute on every packet. This complexity is N times higher than that of the legacy algorithm, posing significant difficulties in low-complexity real-time applications. In practical applications, we will consider deploying the TPE procedure at regular intervals or using it to recover high-importance data packets in an offline decoding scenario, by finding $\boldsymbol{\theta}$ best suited for a given packet, maximizing the objective function presented in subsection III-B. Similarly, the execution scenarios of TPE for real-time application remains outside of the scope of this study and will be considered separately. As a first attempt to explore the potential of BO for UAC receivers, we focus here only on evaluating the performance gain in terms of bit-error or packet-error rate.

A. Simulation setup

We consider a single-input single-output communication architecture, where QPSK symbols are generated and encoded with a convolutional error correction code. The transmitted signals consist of packets of a preamble signal followed by the payload symbols.

The default DFE hyperparameters are set to values $\boldsymbol{\theta}_d = [\hat{L}_A, \hat{L}_B, 0.25, 0.05, 0.01]^T$. These default values have been chosen by domain experts and are integrated into the modem code, used in numerous sea trials. These values are used as a benchmark for the default DFE and is referred to as DFE$_d$ in what follows. The delay τ_m of the last detected arrival is used to set the values of $\hat{L}_A$ and $\hat{L}_B$. A uniformly most powerful invariant based detector is employed to identify the channel arrivals using the preamble. A threshold derived from a false alarm probability of $P_{FA} = 10^{-8}$ is set. Once the arrivals have been detected, the feedforward length is selected to be $\hat{L}_A = \max(80, 2\tau_m n_s s_r + 1)$. Similarly, $\hat{L}_B = \max(20, \tau_m s_r + 1)$, where s_r is the symbol rate and n_s is the number of samples per symbol at the DFE input.

B. Illustration of the objective function

Fig. 3 shows the 2D section of the objective function at L_A and L_B calculated for a realization of one channel power-delay profile and noise. The figure on the right shows the MSE error of the DFE in decibels in color, while the grey areas indicates regions for which the DFE hallucinates. It is noticeable that the value of L_B is crucial, which can be explained by the important delay of the second most important echo.

The plots in Fig. 4 further illustrate how the step sizes μ_A and μ_B depend on β. It is evident that larger values of β perform better, highlighting the effect of the channel sparsity. Fig. 3 and Fig. 4, show several grey regions corresponding to hallucinations detected with $m = 200$.It should be noted that the regions of interest are non-intuitive and cover large areas prone to hallucinations. The roughness of this objective function highlights the fact that most gradient-based approaches would perform poorly for the hyperparameter tuning task.

C. Performance

Fig. 5. shows an estimate of the cumulative distribution function (CDF) of the bit-error rate (BER) for DFE$_d$ vs the CDF of the DFE with optimized hyperparameters. To generate

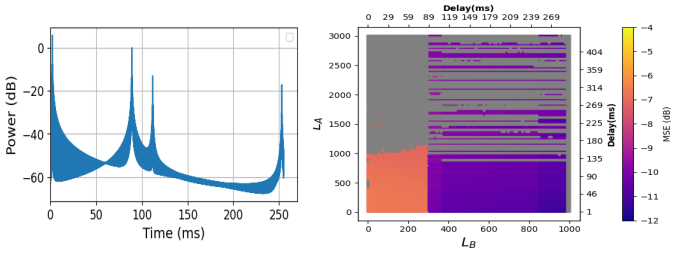

Fig. 3. MSE sensitivity for filter lengths. On the left, the power-delay profile of the considered channel. On the right, the average output MSE (dB) for pairs of filter lengths L_A and L_B. μ_A, μ_B, and β are set to $\boldsymbol{\theta}_d$.

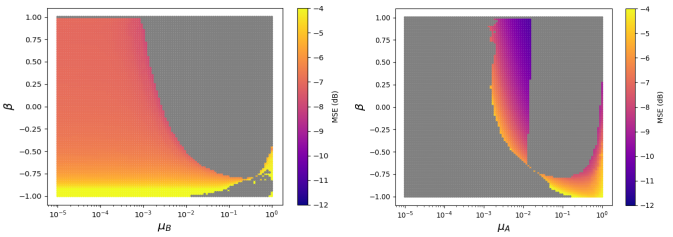

Fig. 4. MSE sensitivity for μ_A, μ_B, and β. On the left, average output DFE MSE(dB) for pairs of μ_B and β. On the right shows MSE for pairs of μ_A and β. L_A and L_B are set to $\boldsymbol{\theta}_d$.

this graph, 12 channels were created using the MAXENT simulator [7] with the conditions specified in Table I. A total of 12,901 packets were generated independently and distributed uniformly across the channels. Gaussian noise was added independently to each transmitted packet. In the event that the default DFE$_d$ failed to fully recover a packet, an optimization was triggered to identify the optimal $\boldsymbol{\theta}$ for this packet. The signal-to-noise ratio (SNR) is within the range of $[10, 15]$ dB. The budget of iterations is limited to $n = 5$ for warm-up and $N - n = 45$ per optimization, II-A. The overall results demonstrate that 70% of the packets are successfully recovered. The average MSE improvement is 3 dB. One can see in Fig. 5 that the self-configurable receiver yields better BER.

Configuration	***Minimum***	***Maximum***
Delay spread	6.6 ms	180 ms
Doppler spread	0.46 Hz	1.87 Hz
Water depth	100 m	200 m
Transmission ranges	500 m	1500 m
Rice factor	5	75

TABLE I
SIMULATED CHANNEL CONFIGURATIONS. THE DOPPLER AND DELAY SPREADS ARE FOUND USING A THRESHOLD OF -20 dB ON THE DOPPLER POWER SPECTRUM AND THE POWER-DELAY PROFILE, RESPECTIVELY.

D. Real channel

A packet error rate (PER) evolution with respect to SNR was computed for 300 packets of 3375 symbols transmitted through this channel for every SNR value. The resulting curves are presented in Fig. 7. It can be observed that our approach yields a significant gain in terms of packet recovery for high

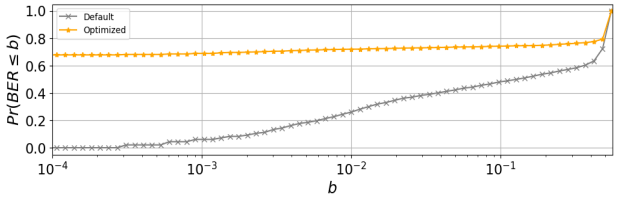

Fig. 5. The –*– curve shows the CDF for DFE$_d$. The -x- curve shows the CDF for hyperparameters obtained from Bayesian optimization.

SNR values, with approximately 20% more packets recovered. The power profile depicted in Fig. 6 was measured in the freshwater Loch Ness, Scotland, in November 2023. However, the channel variability over time was not fully grasped. To address the lack of temporal variability, the power-delay profile and estimates of Rice factors for each tap in Fig. 6 were fed into the MAXENT simulator. A constant Doppler spread of 0.8 Hz was then added, yielding a hybrid channel.

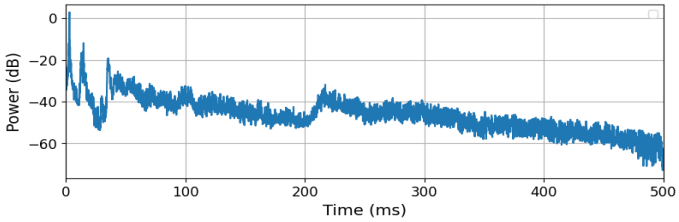

Fig. 6. Channel power-delay profile, measured in Loch Ness, Scotland.

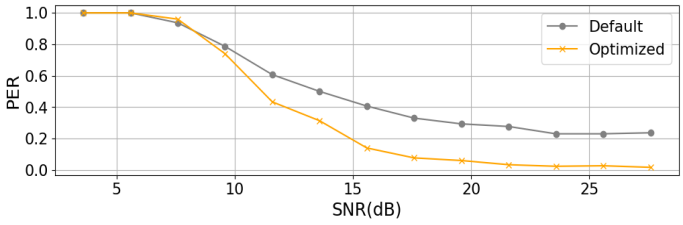

Fig. 7. PER vs SNR. The –*– curve shows the PER vs SNR for $\boldsymbol{\theta}_d$. The curve in -x- corresponds to the optimization performance.

V. CONCLUSION

In this study, we investigated the task of optimizing hyperparameters for an underwater receiver. Unlike traditional methods that rely on domain experts for optimization, we propose to formalize this problem as a black-box optimization task. Our Bayesian optimization approach identifies hyperparameters that minimize the MSE error while addressing the challenges of DFE hallucinations. We conducted a performance study on various simulated channels with a limited number of iterations, demonstrating the relevance of Bayesian optimization for tuning the hyperparameters of an underwater receiver. Although this initial proof of concept shows promise, integrating it efficiently into real-time applications remains an active area of research we are currently exploring.

REFERENCES

[1] P. A. van Walree, "Channel sounding for acoustic communications : techniques and shallow-water examples, ffi-rapport 2011/00007," 2011. [Online]. Available: https://ffi-publikasjoner.archive.knowledgearc.net/bitstream/handle/20.500.12242/2456/11-00007.pdf

[2] K. T. Bae, W. G. Jeon, and Y. S. Cho, "Convergence analyses of timing and carrier recovery loops for a digital communication channel with an adaptive equalizer," in *Proceedings of 40th Midwest Symposium on Circuits and Systems. Dedicated to the Memory of Professor Mac Van Valkenburg*, vol. 2, 1997, pp. 1366–1369 vol.2.

[3] D. S. Oh, W. G. Jeon, Y. S. Cho, H. W. Park, and K. H. Kim, "Convergence analysis of a pll for a digital recording channel with an adaptive partial response equalizer," in *Proceedings of GLOBECOM'96. 1996 IEEE Global Telecommunications Conference*, vol. 2, 1996, pp. 979–983 vol.2.

[4] D. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, 2000.

[5] K. Pelekanakis and M. Chitre, "Comparison of sparse adaptive filters for underwater acoustic channel equalization/estimation," in *2010 IEEE International Conference on Communication Systems*, 2010, pp. 395–399.

[6] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, ser. NIPS'11. Red Hook, NY, USA: Curran Associates Inc., 2011, p. 2546–2554.

[7] F.-X. Socheleau, C. Laot, and J.-M. Passerieux, "A maximum entropy framework for statistical modeling of underwater acoustic communication channels," in *OCEANS'10 IEEE SYDNEY*, 2010, pp. 1–7.

[8] R. Garnett, *Introduction*. Cambridge University Press, 2023, p. 1–14.

[9] ——, *Common Bayesian Optimization Policies*. Cambridge: Cambridge University Press, 2023, pp. 123–156.

[10] S. Watanabe, "Tree-structured Parzen estimator: Understanding its algorithm components and their roles for better empirical performance," *arXiv preprint arXiv:2304.11127*, 2023.

[11] R. Garnett, *Computing Policies with Gaussian Processes*. Cambridge: Cambridge University Press, 2023, pp. 157–200.

[12] J. Song, L. Yu, W. Neiswanger, and S. Ermon, "A general recipe for likelihood-free Bayesian optimization," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 20 384–20 404. [Online]. Available: https://proceedings.mlr.press/v162/song22b.html