A Pursuit Learning Solution to Underwater Communications with Limited Mobility Agents

Hajar Bennouri
Smart Systems Laboratory (SSL)
ENSIAS-Mohammed V University
Rabat, Morocco

Anis Yazidi
Department of Computer Science
Oslo Metropolitan University
Oslo, Norway
anisy@oslomet.no

Amine Berqia
Smart Systems Laboratory (SSL)
ENSIAS-Mohammed V University
Rabat, Morocco

ABSTRACT

Underwater environments are subject to varying conditions which might degrade the quality of communications. In this paper, we propose an adaptive control mechanism to improve the communication in underwater sensor networks using the theory of Learning Automata (LA). Our LA based solution controls the mobility of thermocline sensors to improve the link stability in underwater networks. The problem is modelled as a variant of the Stochastic Point Location (SPL) problem [14, 20, 25]. The sensor is allowed two directions of movement, either surface or dive, in order to avoid physical phenomena that cause faults. Our proposed scheme constitutes also a contribution to the field of LA and particularly to the SPL problem by resorting to the concept of pursuit LA. In fact, pursuit LA exploits more effectively the information from the environment than traditional LA schemes that are myopic and use merely the last feedback from the environment instead of considering the whole history of the feedback. Experimental results show the performance of our algorithm and its ability to find the optimal sensor position.

CCS CONCEPTS

• **Computing methodologies** → *Computational control theory*; • **Applied computing**; • **Networks** → Network components;

KEYWORDS

Underwater Communications, Learning Automtata, Pursuit Learning, Mobile Thermocline Sensors

ACM Reference Format:
Hajar Bennouri, Anis Yazidi, and Amine Berqia. 2018. A Pursuit Learning Solution to Underwater Communications with Limited Mobility Agents. In *International Conference on Research in Adaptive and Convergent Systems (RACS '18), October 9–12, 2018, Honolulu, HI, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3264746.3264797

1 INTRODUCTION

In the last recent years, several applications of oceanographic monitoring have emerged. However, the feature of underwater environment has many challenges and obstacles in addition to those known in traditional Wireless Sensor Networks (UWSNs). The use of acoustic signals in UWSNs to communicate is considered slower than radio signals used in WSNs which causes a longer propagation delay [7].

In terrestrial WSNs, establishing ad-hoc Mobile Networks (MANETs) constitute an important research topic. Similarly, in underwater environments, MANETs have several potential applications such as naval security and seabed mining operations. Currently, the adoption of MANETs in UWSN is very limited also because of the stochastic nature of the underwater acoustic environments that create difficulties for communication [16]. When it comes to the routing part in underwater communications, much of the existing work in focuses on fixed rule algorithms for transmission that do not leverage the mobility of agents, or utilize the opportunity to alter their rule structure with mobility changes [2]. Another problem that researchers face in submarine communication is the changing acoustic qualities of oceanic channel both seasonally and with local weather phenomena [16, 19]. In addition, the ocean currents can cause the displacement of the sensor nodes, other factors are also present and affect the performance of the network sensors such as water temperature, the noise and also the attenuation of the signal without forgetting the factor of the 3D architecture deployment which makes overall these properties under a strong sensitivity [8].

In spite of the fact that these acoustic properties are inevitable, the purpose of this work is to prove that adaptive learning strategies can be used to modify the depth of deeply anchored Limited Mobility Agents (LMAs) in order to improve the communication of the underwater network. Nowadays, to change the depth of the nodes, the majority of sensor networks that allow a variable depth of anchorage to do this process often resort to human intervention, which makes this process very expensive and causes little movement of the sensor compared to the life of the sensor. Thus, this human intervention can be replaced by programmed engines that can be used for task and communication detection. To achieve this purpose and in order to improve the link stability in a network of underwater acoustic sensors, in this article we propose an approach to take advantage of the acoustic sound speed changes along the thermocline of an acoustic environment under-marine.

An adaptive strategy would not discriminate the cause of faults in the network, thus using a learning strategy will allow LMAs to choose and adapt to the best depth of operation which will avoid fault avoidance and also the collision in UWSN. The high cost

RACS '18, October 9–12, 2018, Honolulu, HI, USA

ACM ISBN 978-1-4503-5885-9/18/10…$15.00
https://doi.org/10.1145/3264746.3264797

of agents in UWSNs gives a prime motivation for using learning strategies to avoid faults in aquatic environment. Implementing marine networks with a system that adapts and evokes failures or link failures in submarine communication is very rare because of the high price that a UWSN agent can cost [18].

Our current work is inspired by the latter work [18] that proposes a Learning Automata (LA) that controls the mobility of thermocline using three actions: surface, dive or keep the same position. In this paper, we only use two actions as we rather map the current problem to the Stochastic Point Location (SPL) problem [14]. In SPL, only two directions are allowed and the SPL does not allow the learner to remain in the same position. Thus, instead of using the action that consists in staying in the same location, similar effect can be obtained by using SPL and oscillating back and forth around the optimal position [14]. Therefore, introducing a third action: staying, for the learning algorithm is not necessary and can be avoided. Furthermore, in contrast to [18], the LA designed in this paper uses the concept of pursuit LA that involves estimating the average reward of an action instead of merely using the feedback from the environment in isolation. In fact, pursuit LA exploits more effectively the information from the environment than traditional LA schemes that are myopic and use merely the last feedback from the environment instead of considering the whole history of the feedback.

According to the description of Partan et al. in [17], shadow areas, multipath interference and bubble cloud regions close to the surface are among several physical limitations of underwater acoustic communication. These physical properties not only cause binding failures in UWSNs, but are clearly characteristic of a varying duration of stochastic environment [3, 4]. In addition, in order to improve communication in terrestrial telephone networks, Narendra et al. in [10, 11, 13] have relied on learning algorithm that takes into account the time changing congestion. In [10, 11, 13], in order to establish an adaptive rule routing in a stochastic demand telephone network, authors use a Mean Action Learning Automaton. The results of this work showed performance improvements over traditional fixed-rule routing. In the same direction, to make good use of the properties of acoustics along the thermocline of a body of water, the use of the automaton was described by Akyildiz et al. [16] in a study of 3D typologies of underwater networks with a possibility to vary the depth of the agent. This solution improves the clandestine monitoring capabilities and could also minimize collision with any kind of sea barriers as the passage of ships. In the same way the work reported in [5, 19] proposes to give anchored nodes in the seabed the power to autonomously modify the operating depth of their locations which allows a network agent the ability to avoid collisions and defects caused by physical phenomena.

The reminder of this paper is organized as follows. In Section 2, we introduce the SPL problem. Section 3 gives the details of our LA based algorithm for controlling the mobility of thermocline sensors. Section 4 gives experimental results that confirm the convergence of the algorithm and shows its behavior. Section 5 concludes the paper.

2 LEGACY STOCHASTIC POINT LOCATION SOLUTIONS

To place our work in the right perspective, we start this section by providing a brief review of the main concepts of the SPL problem as first introduced in [14]. We assume that there is a Learning Mechanism (LM) whose task is to determine the optimal value of some variable (or parameter), λ. We assume that there is an optimal choice for λ – an unknown value, say $\lambda^* \in [0, 1)$. The question which we study here is that of learning λ^*. Although the mechanism does not know the value of λ^*, we assume that it has responses from an intelligent "Environment", Ξ, which is capable of informing it whether any value of λ is too small or too big. To render the problem both meaningful and distinct from its deterministic version, we would like to emphasize that the response from this Environment is assumed "faulty." Thus, Ξ may tell us to increase λ when it should be decreased, and *vice versa*. However, to render the problem tangible, in [14] the probability of receiving an intelligent response was assumed to be $p > 0.5$, in which case Ξ was said to be *Informative*. Note that the quantity "p" reflects on the "effectiveness" of the Environment. Thus, whenever the current $\lambda < \lambda^*$, the Environment correctly suggests that we increase λ with probability p. It simultaneously could have incorrectly recommended that we decrease λ with probability $(1 - p)$. The converse is true for $\lambda \geq \lambda^*$.

Oommen [14] pioneered the study of the SPL when he proposed and analyzed an algorithm that operates on a discretized search space while interacting with an informative Environment (i.e., $p > 0.5$). The space in which the search is conducted is first sliced by subdividing the unit interval into N sub-intervals at the positions $\{0, \frac{1}{N}, \frac{2}{N}, \ldots, \frac{N-1}{N}, 1\}$, where a larger value of N will ultimately imply a more accurate convergence to the unknown λ^*. The algorithm then orchestrated a controlled random walk on this space. Whenever the mechanism was told to go to the right (or left), it obediently moved to the right (or left) by a single step (i.e., by $\frac{1}{N}$) in the discretized space. In spite of the Oracle's erroneous feedback, this discretized solution was proven to be ϵ-optimal.

More formally, the scheme presented in [14] obeyed the following updating rules:

Let $\lambda(t)$ be the value at time step "t". In other words, $\lambda(t)$ is an estimate of the unknown value of λ^* at time step "t". Then,

$\lambda(t+1) := \lambda(t)+1/N$ if Ξ suggests to increase λ and $0 \leq \lambda(t) < 1$;
$\lambda(t+1) := \lambda(t)-1/N$ if Ξ suggests to decrease λ and $0 < \lambda(t) \leq 1$.

At the end states the scheme obeys:

$\lambda(t + 1) := \lambda(t)$ if $\lambda(t) = 1$ and Ξ suggests increasing λ;
$\lambda(t + 1) := \lambda(t)$ If $\lambda(t) = 0$ and Ξ suggests decreasing λ.

The analytical results derived in [14] proved that if the "Oracle" was itself *Informative*, the discretized random walk learning was asymptotically[1] optimal. Thus the mechanism would converge to a point arbitrarily close to the true point with an arbitrarily high probability.

In [20], Yazidi et al. presented a hierarchical solution to solve the SPL problem. The solution can be seen as a stochastic version of the bisection search and is shown to outperform legacy solutions. In this approach, the learner queries the environment each time at three locations: end points of the current interval and the midpoint.

[1] As in the case of the field of LA, all the theoretical results reported here are limiting results, i.e., for example, when $N \to \infty$.

Based on a decision table, a new interval is chosen. Consequently, the current interval might be pruned further or the search might be backtracked to a larger interval containing the previous visited interval.

The SPL has been successfully applied in different domains such as binomial estimation [24], quantile estimation [21, 22], stochastic root finding [25] and solving the non-linear stochastic knapsack problem [6, 23].

3 SOLUTION: LEARNING AUTOMATA CONTROL OF LMA

In this section, we formally present our LA based solution to controlling the mobility of thermocline sensors to improve the link stability in underwater networks. We design a LA with two actions $\alpha_k^i \in \{\alpha_0^i, \alpha_1^i\}$ such that the LA can respond to the environment by telling the Limited Mobile Agent (LMA) to choose one of to control states $\phi_i \in \{\phi_0, \phi_1\}$ corresponding to dive, or surface, respectively such that control maps onto actions as: α_0^i to ϕ_0 corresponding the dive command and α_1^i to ϕ_1 corresponding to a surface command. The informed reader would observe that the problem has analogy to SPL as the LMA is allowed to choose on direction at each time step. At each depth level i, we attach an LA. We suppose that the minimum depth is 0 and the max depth is D. Therefore there are $D+1$ LA attached to the locations $\{0, 1 \ldots D\}$.

In the same manner as in [18], we envisage the use of timeout to to ensure that a learning action occurs at all iterations of the algorithm and in order to allow exploring the different locations. In other words, each action of LA is valid for a certain amount of time and then a new action is chosen.

When diving, we will move to location $min(i+1, D)$ where D is the max depth. When moving to the surface, the new location is $max(i-1, 0)$, where 0 corresponds to the depth at the surface. Please note that this is similar to SPL where the end state are self-loops.

3.1 Construction of the Learning Automata

At each depth i we associate a 2-action S-Model [9, 12] Learning automaton, $(\Sigma^i, \Pi^i, \Gamma^i, \Upsilon^i, \Omega^i)$, where Σ^i is the set of actions, Π^i is the set of action probabilities, Γ^i is the set of feedback inputs from the Environment, and Υ^i is the set of action probability updating rules.

(1) *The set of actions of the automaton*: (Σ^i)
The two actions of the automaton are α_k^i, for $k \in \{0, 1\}$, i,e, α_0^i and α_1^i

(2) *The action probabilities*: (Π^i)
$P_k^i(t)$ represent the probabilities of selecting the action α_k^i, for $k \in \{0, 1\}$, at time step t. Initially, $P_k^i(0) = 0.5$, for $k = 0, 1$.

(3) *The feedback inputs from the Environment to each automaton*: (Γ^i)

Whenever the LMA moves to a location i, the environment will return a continuous value representing the performance at that location which is a noisy measurement. Formally, the response from the Environment at time t and at location i is denoted by $\beta^i(t)$.

We suppose that whenever the i^{th} LA denoted takes action surface, the next LA at position $min(i+1, D)$ will be activated. Similarly, if the action is dive, the next LA at position $max(i-1, 0)$ will be activated. This allows the LMA to find the most stable link in a stochastic environment through adaptation. Let $\bar{\beta}^i(t)$ be the estimated average reward obtained for location i since the first time step. It can be given by:

$$\bar{\beta}^i(t) = \frac{\sum_{l=1}^{t} J(l,i)\beta^i(l)}{\sum_{l=1}^{t} J(l,i)}$$

where $J(l, i) = 1$ if the location i action was deployed at the l^{th} time step.

(4) *The action probability updating rules*: (Υ^i)

If α_k^i for $k \in \{0, 1\}$ was chosen then, for $j \in \{0, 1\}$. The LA update equations are given by:

$$P_j^i(t+1) \leftarrow P_j^i(t+1) + \theta(\delta_{jk} - P_j^i(t)) \quad (1)$$

where $0 < \theta \ll 1$ and:

$$\delta_{jk} = \begin{cases} 1 & \text{if } \bar{\beta}^i(t) > \bar{\beta}^{i^*}(t) \\ 0 & \text{else} \end{cases} \quad (2)$$

Here i^*, denote the locations visited by the LMA at time step $t+1$ as a result of the action taken at time step t. This location can be:

- $Min(i-1, 0)$ whenever α_0^i was taken corresponding to the dive command
- $max(i+1, D)$ whenever α_1^i was taken corresponding to the surface command.

Therefore, in other words, if the newly visited location i^* has better average reward than the location i we will increase the probability of the action leading to this location. However, if if the newly visited location i^* has inferior average reward than the location i we will decrease the probability of the action leading to this location.

In simpler terms, we have two cases.

Whenever $\bar{\beta}_k^i(t) > \bar{\beta}_k^{i^*}(t)$

$$P_k^i(t+1) \leftarrow P_k^i(t) + \theta \times (1 - P_k^i(t))$$
$$P_{1-k}^i(t+1) \leftarrow 1 - P_k^i(t+1).$$

Otherwise (i.e, $\bar{\beta}_k^i(t) \le \bar{\beta}_k^{i^*}(t)$),

$$P_k^i(t+1) \leftarrow P_k^i(t) + \theta \times (0 - P_k^i(t))$$
$$P_{1-k}^i(t+1) \leftarrow 1 - P_k^i(t+1).$$

3.2 Remark

Our algorithm is inspired by the family of pursuit LA algorithm [1, 15, 26]. However, instead of pursuing the action with the highest reward among the offered actions, we pursue the action that leads to an increase in the reward compared to the previously visited state at time instant $t-1$

4 EXPERIMENTAL RESULTS

We test our algorithm for one uni-modal performance function, and for one bi-modal function. In all experiments, the learning parameter $\theta = 0.01$. For obtaining steady probability we run the

Figure 1: Unimodal Performance Function

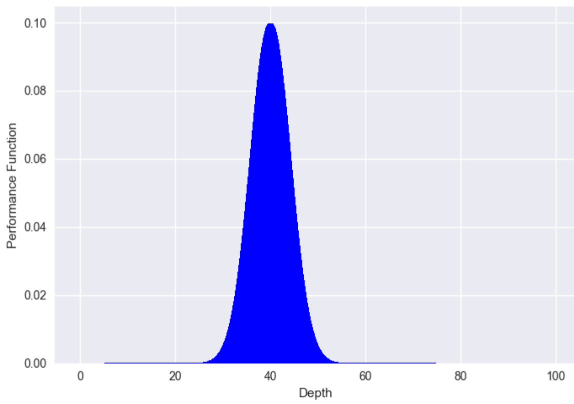
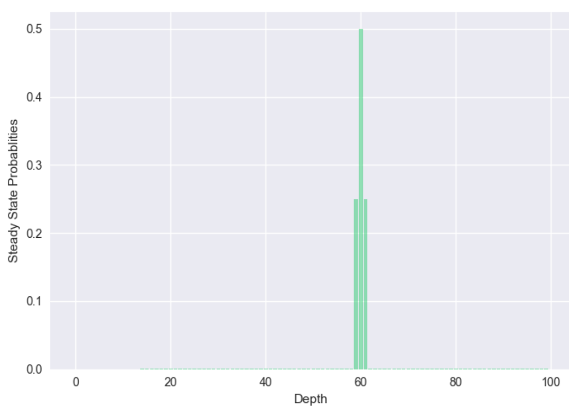


Figure 2: Steady Probability



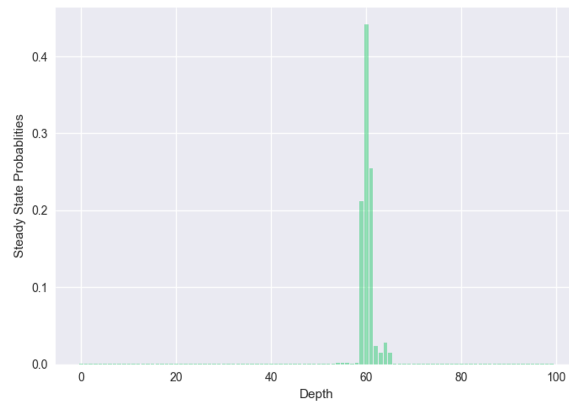
algorithm for 10^6 iterations. We suppose that the maximum depth is 100 called max_D and minimum depth called min_D at the surface is 0. In all the experiments, we observe a noisy version of the performance and therefore we use an additive noise function that follows a normal distribution, i.e, mean 0 and variance 1.

4.1 Noisy Uni-modal Performance Function

We suppose that the initial location for the LMA at time 0 is $\frac{min_D+max_D}{2}$. We used a Gaussian function with mean 60 and standard deviation 4. Figure 1 depicts the latter function.

Figure 2 depicts the steady state probability over the different possible positions that reflects the percentage of time the LMA spends at each location when controlled by our pursuit LA scheme. We observe that most of the probability mass is concentrated around the max of the uni-modal function, namely 60 which corresponds to the max of the uni-modal function.

Figure 3: Steady state probability for uni-modal performance function with larger noise



4.1.1 Testing with Flatter uni-modal function near the optimum. In this experiment, in order to obtain a flatter uni-modal function near the optimum we increase the variance to 30. From Figure 3, we see that the steady probability over the different location forms a distribution that is no longer a peak shape around the extreme but more a flat curve. However, we see clearly a peak around one of the extremes.

4.2 Noisy Bi-Modal Performance Function with equal extreme values

In this experiment, we use a bi-modal which is the superposition of two Gaussian functions: one with mean 20 and standard deviation 4 and one with mean 60 and standard deviation 4. Please note that the function admits two extrema, namely 20 and 60 with the same performance. In other words, the noisy bi-Modal performance function has two equal extreme values. Thus, it is desired that the algorithm converges to one of those two extreme values. Please note that the fact that the two extreme values are equal makes the problem more difficult because the algorithm needs to be designed in a such a manner that we guarantee convergence to one of the two extrema instead of not converging to any of them.

Figure 4 depicts the performance function but without the additive noise.

In the following experiments we will verify that convergence in this case to one of the extrema is depending not only on the initial position but also on how flat is the performance function.

Figure 5 illustrates the steady probability of the LMA over the different positions when the initial depth is 20. As expected, we observe that the LA scheme converges to the position around 20 which is the maximum value of the performance function closest to 20.

A similar results is illustrated in Figure 6 where the initial state is 60. We observe that the scheme concentrates the walk around 60.

Figure 7 depicts the trajectory of the LMA over time when starting at the middle depth, i.e, $\frac{min_D+max_D}{2}$ which corresponds in this

Figure 4: Bi-modal Performance Function with no additive noise

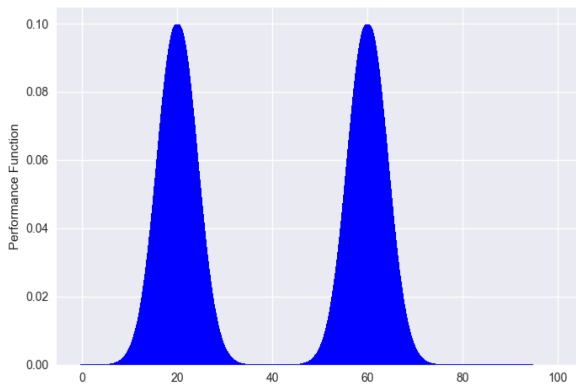
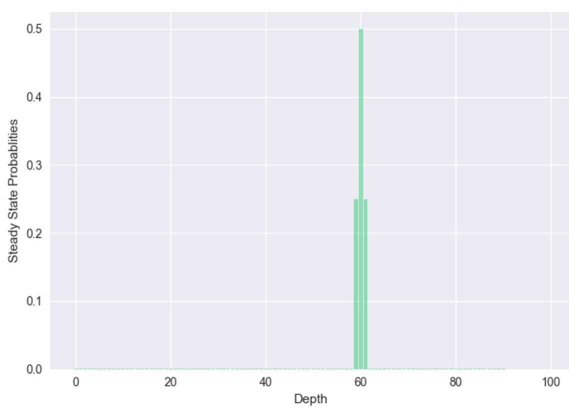


Figure 5: Steady Probability: initial state 20



case to depth 50. We observe that the algorithm quickly converges the location 60 which corresponds to one of the extremes.

4.2.1 Testing with Flatter Bi-Modal Performance Function. We increase the variance of the two Gaussian functions from 4 to 15. In other words, in this experiments we used a bi-modal which is the superposition of two Gaussian functions: one with mean 20 and standard deviation 15 and one with mean 60 and standard deviation 15. By increasing the variance, the function becomes flatter and it becomes more difficult for the LA algorithm to distinguish a maximum performance point from a non-maximum performance point. Figure 8 depicts the non-noisy version of the performance function i.e, with no additive noise which is clearly flatter than the function depicted in Figure 4 where the variance was 4.

In Figure 9, we depict the steady state probability when the initial position is 20. Interestingly, from Figure 9, we see that the LMA

Figure 6: Steady Probability: initial state 60

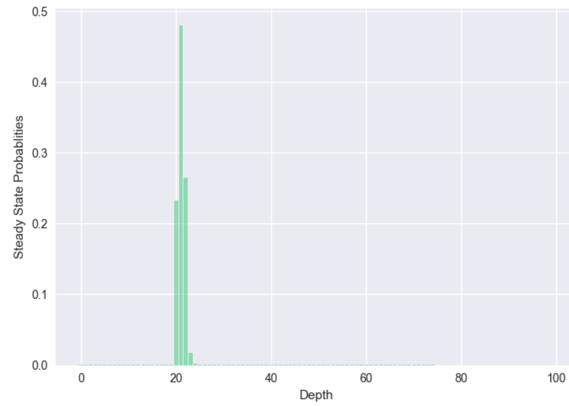
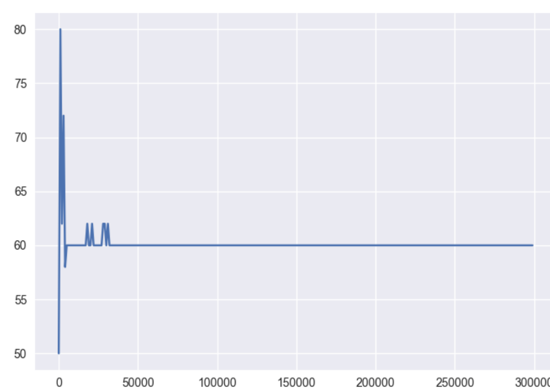


Figure 7: Trajectory of the LMA when starting at the middle depth



converges to the neighborhood of the extrema 60. This is counter intuitive and unexpected as we would expect that since the initial state is 20 which is closer to extrema 40 than 60, the convergence will be around the extrema 40 instead of the far away extrema 60. This can be explained by the fact that the function is flatter in this case which allows the scheme to explore more the solution space and for a longer time.

Similarly, we run an experiments where the initial state is 60, i.e, the initial position juxtaposes with one of the extremes which is 60. Figure 6 depicts the steady state probability distribution where we observe the convergence to the maximum 20, despite we start around 60 which is also a maximum with equal value.

Figure 8: Bi-modal Performance Function with no additive noise and increased variance

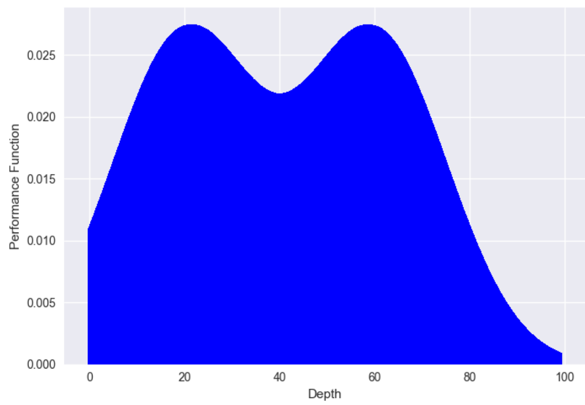
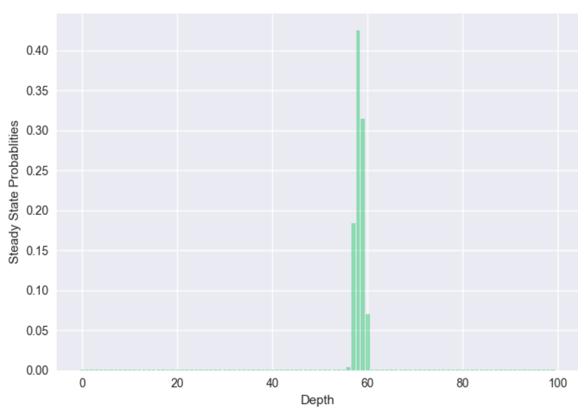


Figure 9: Steady Probability: initial state 20



5 CONCLUSION

In this paper, we introduce an adaptive control mechanism to control the mobility of thermocline sensors to improve the link stability in underwater networks. We model the problem as a variant of the SPL problem [14, 20, 25]. In a similar manner to SPL, the LMA is only allowed to move only into two directions. Our solution has a pursuit LA flavor. In contrast to classical SPL solutions, our pursuit LA exploits more effectively the information from the environment than traditional LA schemes that are myopic and use merely the last feedback from the environment instead of considering the whole history of the feedback. Experimental results show the performance of our algorithm and its ability to find the optimal sensor position.

REFERENCES

[1] Agache, M., and Oommen, B. J. Generalized pursuit learning schemes: new families of continuous and discretized learning automata. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 32*, 6 (2002), 738–749.
[2] Akyildiz, I. F., Pompili, D., and Melodia, T. Challenges for efficient communication in underwater acoustic sensor networks. *ACM Sigbed Review 1*, 2 (2004), 3–8.
[3] Bass, A. H., and Clark, C. W. The physical acoustics of underwater sound communication. In *Acoustic communication*. Springer, 2003, pp. 15–64.
[4] Catipovic, J. A. Performance limitations in underwater acoustic telemetry. *IEEE Journal of Oceanic Engineering 15*, 3 (1990), 205–216.
[5] Etter, P. C. *Underwater acoustic modeling and simulation*. CRC Press, 2018.
[6] Granmo, O.-C., and Oommen, B. J. Solving stochastic nonlinear resource allocation problems using a hierarchy of twofold resource allocation automata. *IEEE Transactions on Computers 59*, 4 (2010), 545–560.
[7] Liu, X., Liu, P., Long, T., Lv, Z., and Tang, R. An efficient depth-based forwarding protocol for underwater wireless sensor networks. In *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)* (2018), IEEE, pp. 467–475.
[8] Luo, J., Fan, L., Wu, S., and Yan, X. Research on localization algorithms based on acoustic communication for underwater sensor networks. *Sensors 18*, 1 (2018), 67.
[9] Mason, L. An optimal learning algorithm for s-model environments. *IEEE Transactions on Automatic Control 18*, 5 (1973), 493–496.
[10] Narendra, K. S., and Mars, P. The use of learning algorithms in telephone traffic routingâĂŤa methodology. *Automatica 19*, 5 (1983), 495–502.
[11] Narendra, K. S., and Thathachar, M. A. On the behavior of a learning automaton in a changing environment with application to telephone traffic routing. *IEEE Transactions on Systems, Man, and Cybernetics 10*, 5 (1980), 262–269.
[12] Narendra, K. S., and Thathachar, M. A. L. *Learning Automata: An Introduction*. Prentice-Hall, New Jersey, 1989.
[13] Narendra, K. S., Wright, E. A., and Mason, L. G. Application of learning automata to telephone traffic routing and control. *IEEE Transactions on Systems, Man, and Cybernetics 7*, 11 (1977), 785–792.
[14] Oommen, B. J. Stochastic searching on the line and its applications to parameter learning in nonlinear optimization. *IEEE Transactions on Systems, Man and Cybernetics SMC-27B* (1997), 733–739.
[15] Oommen, B. J., and Lanctôt, J. K. Discretized pursuit learning automata. *IEEE Transactions on systems, man, and cybernetics 20*, 4 (1990), 931–938.
[16] Otnes, R., and Haavik, S. Duplicate reduction with adaptive backoff for a flooding-based underwater network protocol. In *OCEANS-Bergen, 2013 MTS/IEEE* (2013), IEEE, pp. 1–6.
[17] Partan, J., Kurose, J., and Levine, B. N. A survey of practical issues in underwater networks. *ACM SIGMOBILE Mobile Computing and Communications Review 11*, 4 (2007), 23–33.
[18] Porretta, S., Barbeau, M., Garcia-Alfaro, J., and Kranakis, E. Learning to communicate underwater: An exploration of limited mobility agents. In *Proceedings of the International Conference on Underwater Networks & Systems* (2017), ACM, p. 2.
[19] Robert, J. U. *PRINCIPLES OF UNDERWATER SOUND*. PENINSULA PUBLISHING, 2013.
[20] Yazidi, A., Granmo, O.-C., Oommen, B. J., and Goodwin, M. A novel strategy for solving the stochastic point location problem using a hierarchical searching scheme. *IEEE transactions on cybernetics 44*, 11 (2014), 2202–2220.
[21] Yazidi, A., and Hammer, H. Multiplicative update methods for incremental quantile estimation. *IEEE Transactions on Cybernetics* (2017).
[22] Yazidi, A., Hammer, H., and Oommen, B. J. Higher-fidelity frugal and accurate quantile estimation using a novel incremental discretized paradigm. *IEEE Access 6* (2018), 24362–24374.
[23] Yazidi, A., and Hammer, H. L. Solving stochastic nonlinear resource allocation problems using continuous learning automata. *Applied Intelligence* (2018), 1–20.
[24] Yazidi, A., and Oommen, B. J. Novel discretized weak estimators based on the principles of the stochastic search on the line problem. *IEEE transactions on cybernetics 46*, 12 (2016), 2732–2744.
[25] Yazidi, A., and Oommen, B. J. A novel technique for stochastic root-finding: Enhancing the search with adaptive d-ary search. *Information Sciences 393* (2017), 108–129.
[26] Zhang, X., Oommen, B. J., Granmo, O.-C., and Jiao, L. A formal proof of the ε-optimality of discretized pursuit algorithms. *Applied Intelligence 44*, 2 (2016), 282–294.