SRPH Journal of Fundamental Sciences and Technology
Volume 2, Number 2: (1-8), 2020
Available online at: http://sjfst.srpub.org/

Control And Path Planning Of AUVs Robot Using Krill Herd Optimization Algorithm And Learning Automata

Seyyed Alireza Razavi Asfali
Department of Mechanical Engineering, Sharif University of Technology, Tehran, Iran

Article history:

Received date: 22 December 2019
Review date: 7 March 2020
Accepted date:25 March 2020

Keywords:
optimization, underwater robot, state-space model, LQR optimal controller, learning automata, krill herd algorithm.

Abstract

In this study first, we investigated the equations in movement of an underwater robot and also the state-space model of system is expressed with linearizing existing equations. Then, an energy efficient path is planned using dynamic equations and dynamic planning optimization method. There are moving obstacles in environment in which a robot is moving. It can be seen that the planned path is flat and energy consumption is minimized. The main objective of the study is to present an appropriate controller for the provided state-space model of system. For this purpose, by studying the system controller designing using LQR optimal controller, an appropriate controller for model has been presented. The planning a path to a target for the underwater robot has been presented using combination of optimization algorithm with learning automata and the krill herd optimization algorithm. In other words, the study has applied hybrid algorithm in order to find optimal path for the underwater robot to move in a static environment which is expressed through the map with the nodes and links.

Please cite this article as: Razavi Asfali SA. 2020. Control And Path Planning Of AUVs Robot Using Krill Herd Optimization Algorithm And Learning Automata, SRPH Journal of Medical Sciences and Healthcare Management Volume 2, Number 2: (1-8)

1. Introduction

The undeniable importance of the seas and oceans in commercial, recreational, research and defense applications as well as human hazards in the depths of the seas, have caused unmanned underwater robots to be considered as one of the major types of researches in this area. Among these devices, autonomous underwater vehicles or AUVs due to its considerable advantages on semi-automated underwater devices became an important research area. AUV is a robot which travels underwater without requiring input from an operator, controlled and powered from the surface by an operator/pilot via an umbilical or using remote control. It has many applications in terms of economic, hobby, research and military. The behavior control of these robots is difficult due to the existence of obstacles. There are some theories about obstacles in an environment like the theory of Rief and Sharir (1994) which states that in an environment, obstacles move with a fixed velocity, and Fujimura (2002) which states that all obstacles are slowly moving, and points to a time-minimal path amidst moving obstacles. The first step is to avoid obstacles is to determine the accuracy of sensors including optical and acoustic types, but the use of these sensors will be useful when the performance and efficiency of them are integrated with the navigation and obstacle avoidance algorithms such as vector field histogram (VFH) (Borenstein and Koren,1991), artificial potential field (APF) (Khatib, 1986), and certainty grid. (Toe et al, 2009)

Mobile robot navigation among moving obstacles have been studied by many researchers (e.g. Song et al. 1999; Pratihar et al. 1999; Aranibar and Alsina, 2004; Kanakakis and Valavanis, 2004; Yen and Hickey, 2004; Park et al.2007; Jaradat, 2010). Robot path planning in dynamic environments is a difficult task. Obstacles in the environment cause the complexity of the issue. Path planning is an important primitive for autonomous mobile robots that lets robots find the shortest or optimal path between obstacles. Path planning of a mobile robot need infromation such as starting position, target position, and a 2D map of workplace

This open-access journal is published under the terms of the Creative Commons Attribution-Noncommercial 4.0 International License.

[Downloaded from sjfst.srpub.org on 2026-09-16]

[DOI: 10.29252/sjfst.2.2.1]

including stationary obstacles. Classic path planning approaches include roadmaps method, cell decomposition method and potential field method. A lot of methods have been reported to create an optimal path such as dynamic programming and distance transform method. In dynamic planning we should calculate all possible paths and select one with slowest cost value, while in distance transform method we should find a path from the target location back to the starting location. These two methods need high computation and large number of obstacles.

In this research we try to use krill herd (KH) optimization algorithm and learning automata for path planning of AUVs. KH is one of the most recent heuristic optimization techniques which is proposed based on herding behavior of krill individuals in the nature. Learning automata is a branch of the theory of adaptive control which selects their current action based on past experiences from the environment.

2. Materials and Methods

2.1. Learning automata

A learning automaton is an adaptive decision-making unit situated in a random environment that learns the optimal action through repeated interactions with its environment. The actions are chosen according to a specific probability distribution which is updated based on the environment response the automaton obtains by performing a particular action. It was introduced by Tsypkin (1971) by applying stochastic hill-climbing techniques and developed by Tsetlin (1973) and colleagues. Narendra and Thathachar (1974) were used the term "learning automata" for the first time.

A learning automaton is formed of two main parts: a stochastic automaton with limited amount of actions, and a random environment (see Fig. 1).

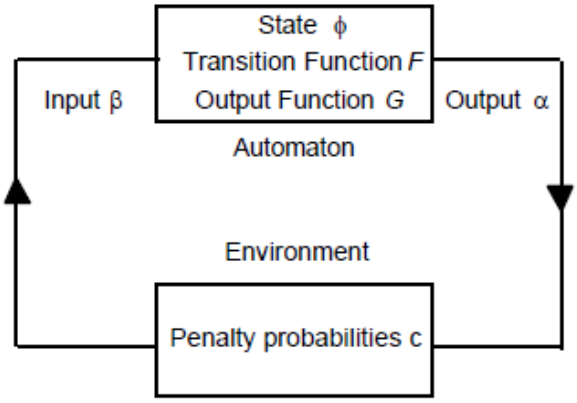

Figure 1. Learning automaton

automaton is presented by a sextuple {α, β, ρ, F, G, ϕ} where ϕ is a set of internal *state*s, α is a set of outputs or actions, β is a set of responses, ρ is the state probability vector ($P(n) \equiv \{p_1(n), p_2(n), \ldots, p_r(n)\}$, n=stage) , F is a function that maps the current state and input into the next state (which can be deterministic or stochastic) generates p(n + 1) from p(n), and G is a function that maps the current state and input into the current output.

The environment is represented by a triple {α, β, c} where α is action/output set, β is input/response set, and c is a set of penalty probabilities. α(*n*) of the automaton belongs to the set α, and is applied to the environment at time *t* = *n*, □(*n*) from the environment is an element of the set □□and $\beta_i(n)$ is between 0 and 1. The c is written as:

$$c_i = \text{Prob}\{\beta(n) = 1 | \alpha(n) = \alpha_i\}, i = \{1,2,\ldots,r\} \quad (1)$$

When the penalty probabilities c_i are constant, the environment is called a *stationary environment*. There are three models defined by the response set of the environment: P-model, Q-model and S-model; if the input from the environment can take only one of two values, 0 or 1, it is referred as P-model. in this case, 1 means an "unfavorable" response, while 0 means the action is "favourable". If the input can take more than two elements, it is called Q-model, and if the input is a continuous random variable with possible values, it is called S-model.

[Downloaded from sjfst.srpub.org on 2026-09-16]

[DOI: 10.29252/sjfst.2.2.1]

2.1.1. Norms of behavior

To measure the performance of stochastic learning automata, some norms of behavior have been defined which allow comparison of different methods of learning. A "pure chance" situation is when all action probabilities are equal. In this case, an *r*-action automaton, the action probability vector *p*(*n*) is given by:

$$p_i(n)=\frac{1}{r} \quad (2)$$

By considering a stationary random environment with penalty probabilities c_i, the average penalty for a given action probability vector is defined as:

$$M(n) = E[\beta(n)|p(n)] = \sum_{i=1}^{r} c_i p_i(n) \quad (3)$$

For the pure-chance automaton, *M*(*n*) is a constant shown by *Mo*

$$M_0 = \frac{1}{r}\sum_{i=1}^{r} c_i p_i(n) \quad (4)$$

Since M(n) is a random variable; its expected value, i.e. E[M(n)] is compared with M_0 , so according to the above definitions:

- A learning automaton is *expedient* if: $\lim_{n\to\infty} E[M(n)] < M_0$ (5)
- A learning automaton is said to be *optimal* if: $\lim_{n\to\infty} E[M(n)] < c_i$ (6)
- where, C_i=min $\{c_i\}$
- A learning automaton is *e-optimal* if: $\lim_{n\to\infty} E[M(n)] < c_i + \varepsilon, \forall \varepsilon > 0$ (7)
- learning automaton is *absolutely expedient* if:

$E\left[M(n+1)|p(n)\right] < M(n), \forall n, \forall p_i(n) \in [0,1]$, for all possible sets $\{c_i\}$ $(i = 1,2,\dots,r)$ (8)

2.2. Krill herd optimization algorithm

KH algorithm is a bio-inspired heuristic optimization technique introduced by Gandomi and Alavi (2012). It imitates the behavior of Krill individuals in krill population. The algorithm use three activities of krill: induced movement (N), foraging activity (F), and random diffusion (D). The KH algorithm start with determination of algorithm parameters; after creating initial population and fitness evaluation, induced movement, foraging motion , and physical diffusion are calculated. Then by updating position and required repetitions, we reach satisfied result.

According to Gandomi and Alavi (2012), algorithms of three activities are as following:

2.2.1. Induced movement

Induced movement (N) is defined as:

$$N_i^{new} = N^{max}\alpha_i + \omega_n N_i^{old} \quad (9)$$

where,

$$\alpha_i = \alpha_i^{local} + \alpha_i^{target} \quad (10)$$

In above equations, N^{max} is the maximum induced speed, ω_n is the inertia weight of the motion induced in the range [0, 1], N_i^{old} is the last motion induced, α_i^{local} is the local effect provided by the neighbors and α_i^{target} is the target direction effect provided by the best krill individual. By considering C^{best} as the effective coefficient of α_i^{target} for the i-th krill individual, it is written as:

$$C^{best} = 2\left(rand\,\frac{I}{I_{\max}}\right) \quad (11)$$

Where *rand* is a random value between 0 and 1, *I* is the actual iteration number and I_{max} is the maximum number of iterations.

2.2.2. Foraging activity

Foraging motion is written as below in terms of food location (F_i) and the previous experience about the food location (β_i):

$$F_i = V_f\beta_i + \omega_f F_i^{old} \quad (12)$$

$$\beta_i = \beta_i^{food} + \beta_i^{best} \quad (13)$$

[Downloaded from sjfst.srpub.org on 2026-09-16]

[DOI: 10.29252/sjfst.2.2.1]

Where V_f is the foraging speed, ω_f is the inertia weight of the foraging motion in the range [0, 1], F_i^{old} is the last foraging motion, β_i^{food} is the food attractive, and β_i^{best} is the effect of the best fitness of the i-th krill so far. According Gandomi and Alavi (2012), the foraging speed is taken 0.02 (ms^{-1}).

2.2.3. Random diffusion

Physical diffusion which is a random activity, is defined as:

$$D_i = D^{max}\left(1-\frac{I}{I_{\max}}\right)\delta \tag{14}$$

where D^{max} is the maximum diffusion speed, and δ is the random directional vector in [-1,1].

2.2.4. Motion Process

The position of each krill is updated during the interval t to $t+\Delta t$ as follow:

$$X_i\left(t+\Delta t\right) = X_i\left(t\right) + \Delta t\ \left(dX_i\right)/dt \tag{15}$$

$$\Delta t = c_t \sum_{i=1}^{NV}\left(UB_j - LB_j\right) \tag{16}$$

where NV is the total number of variables, and LB_j and UB_j are lower and upper bounds of the j-th variables (j = 1,2,. . .,NV).

2.3. Study Vehicle

In this study, we used Remote Environmental Monitoring Unit (REMUS) AUV designed by the Oceanographic Systems Lab as a study model (Prestero, 1994).

2.6. Method

The objective of this study is to plan a path for AUVs in the presence of moving obstacles using a hybrid algorithm combined of KH algorithm and learning automata**,** and based on optimization rules, minimize the objective function. AUVs have energy constraints and often use rechargeable batteries. Hence, the objective function in this study is the amount of energy consumed by robot during mission. Selected optimization method in this study is dynamic programming, and numerical solution for obtained equations is Newton-Raphson method. After path planning, an automatic control system is designed for trajectory tracking of robot. In this paper, the indirect effect of moving obstacles on the dynamics of robot was ignored, and the dynamic equations of sample AUV are fixed during the path planning. In fact, both Path Planning and Trajectory Tracking are employed in present study.

2.6.1. Dynamic programming

Dynamic programming algorithms are used for optimization (for example, finding the shortest path between two points, or the fastest way to multiply many matrices). A dynamic programming algorithm will examine the previously solved subproblems and will combine their solutions to give the best solution for the given problem. Very large computing power is required, especially in environments that have many subproblems. In terms of mathematical optimization, if the start point is *Ps* , goal point is *Pg* and sub-goal point is *Pi* , the path generation method is how to determine a sequence of sub-goals picking out the sub-goals from their set *Pi* (i=1,2,…,g-1). We must calculate all possible paths and select one has the smallest cost value. (Farshchi, 2011)

The present study has investigated the equations in movement of a underwater robot and also the state-space model of system is expressed with linearizing existing equations. The main objective of the study is to present an appropriate controller for the provided state-space model of system. For this purpose, by studying the system controller designing using LQR optimal controller, an appropriate controller for model has been presented. The general path rout planning issue to a target for the underwater robot has been presented using optimization algorithm with the combination of learning automata and the KH algorithm. In other words, the study has applied hybrid algorithm in order to find optimal path for the underwater robot to move in a static environment which is expressed through the map with the nodes and links.

7. Results and discussion

7.1. Simulation

The study has presented state-space model through MATLAB software using defined matrix and also a controller has designed in state-space model. The obtained results from model variables have been illustrated after and before the control in figures as follows:

[Downloaded from sjfst.srpub.org on 2026-09-16]

[DOI: 10.29252/sjfst.2.2.1]

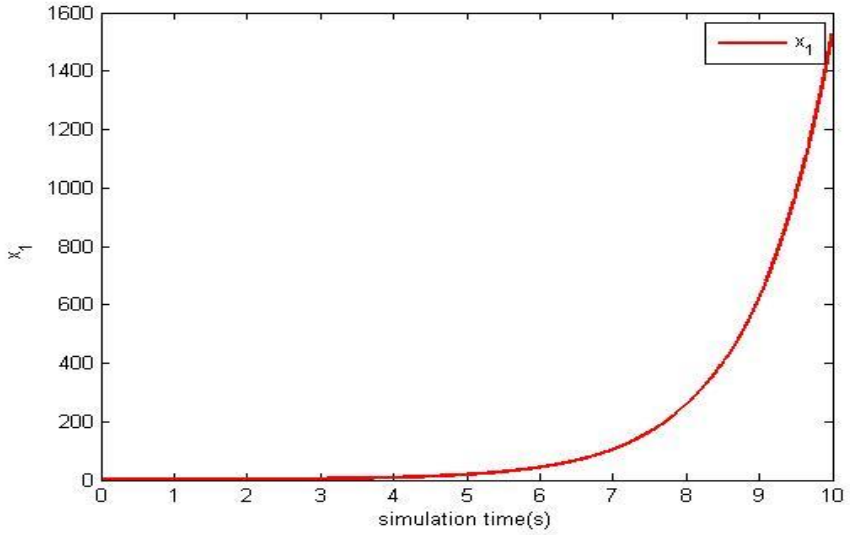

Figure2. x_1 State variable before control

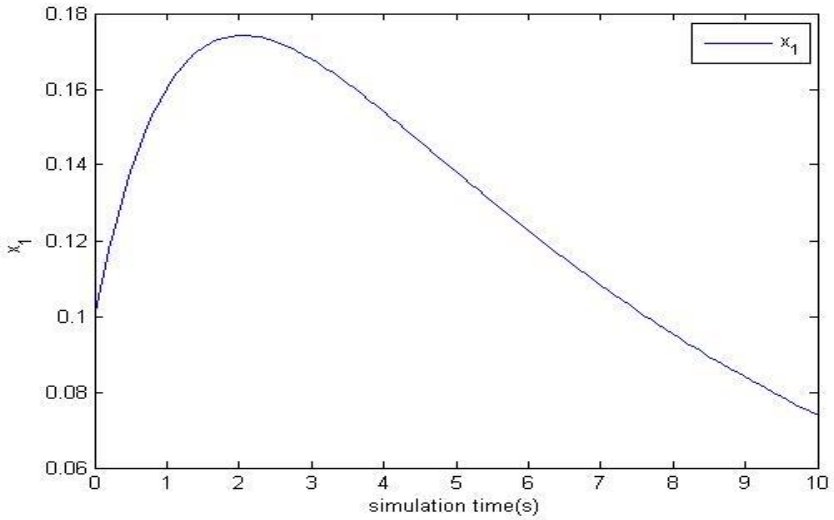

Figure3. x_1 State variable after control

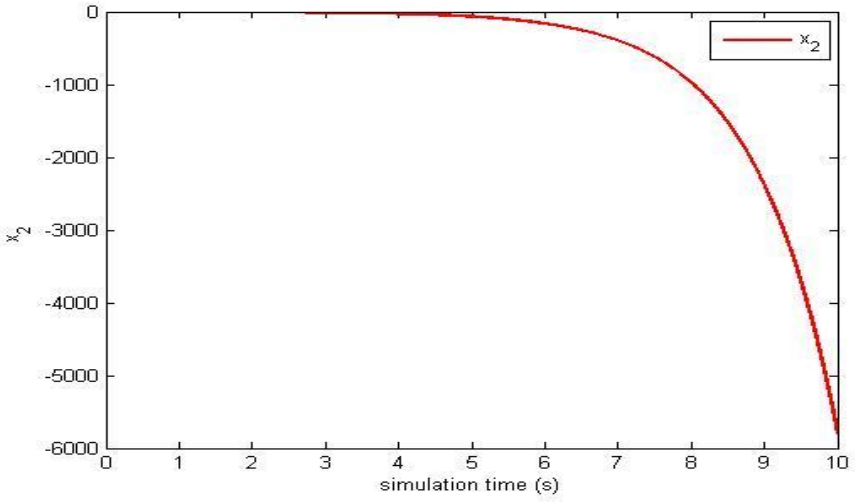

Figure4- x_2 State variable before control

[Downloaded from sjfst.srpub.org on 2026-09-16]

[DOI: 10.29252/sjfst.2.2.1]

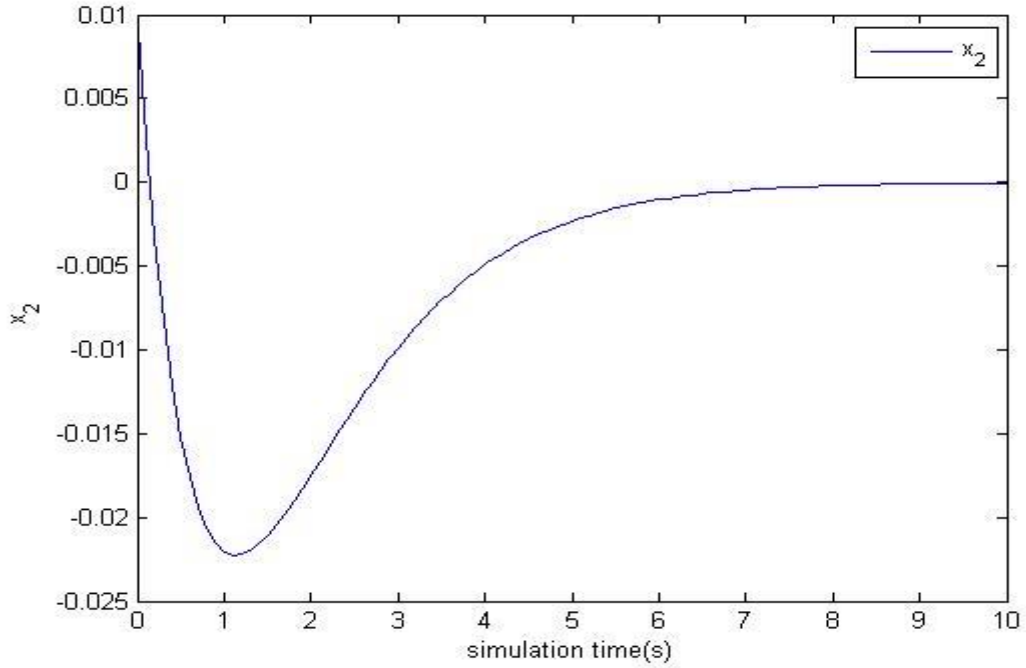

Figure5. x_2 State variable after control

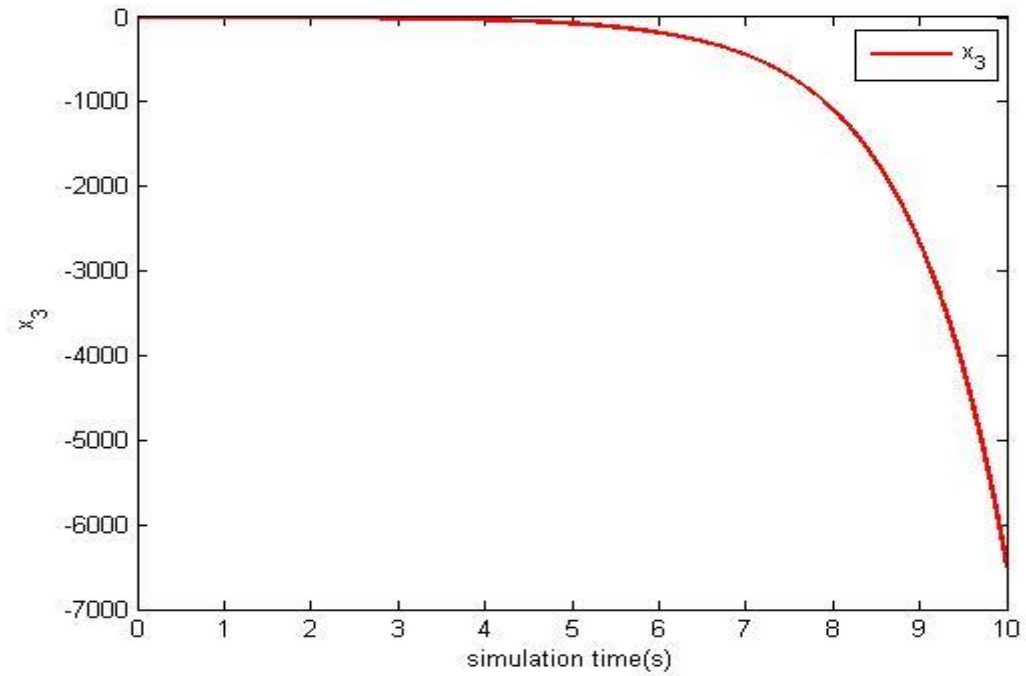

Figure6-x_3 State variable before control

[Downloaded from sjfst.srpub.org on 2026-09-16]

[DOI: 10.29252/sjfst.2.2.1]

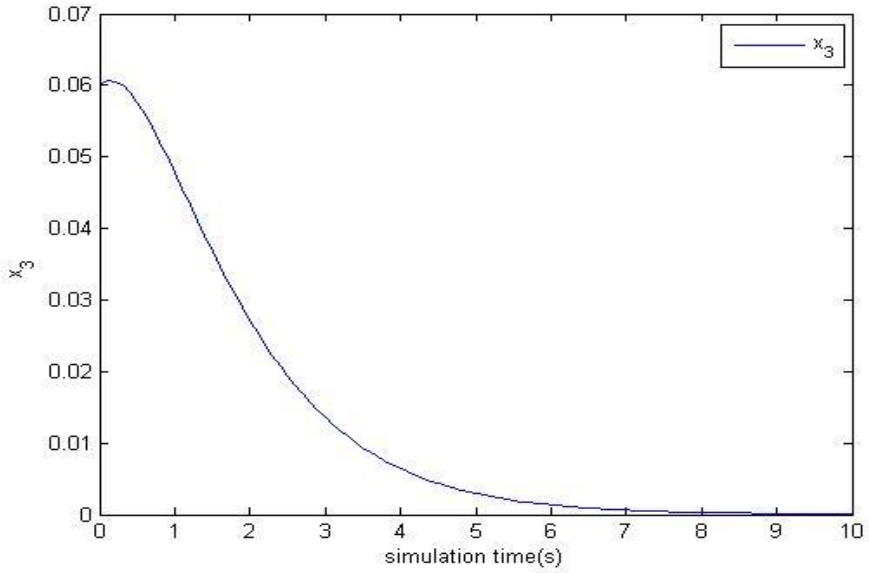

Figure7-x_3 State variable after control

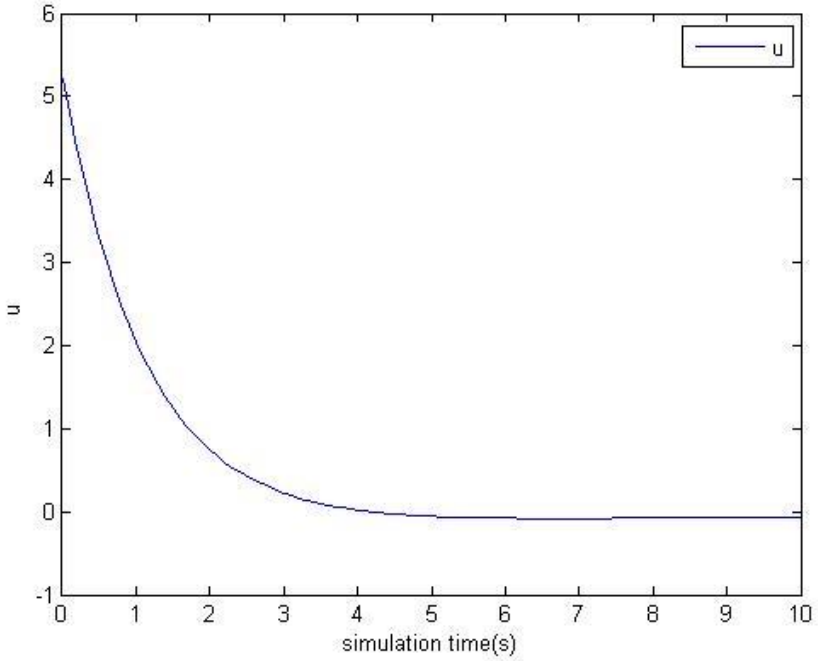

Figure 8- the system control

4. Conclusion

The present study has investigated the equations in movement of a underwater robot and also the state-space model of system is expressed with linearizing existing equations and finally has presented an optimal controller for the provided model. The obtained results and diagrams from simulation have expressed controller accuracy and the method of model variables controlling. The problem of general planning for a underwater robot to achieve an target has been solved using optimization algorithm with combination of learning automata and KH algorithm that include in NP-hard issues.

[Downloaded from sjfst.srpub.org on 2026-09-16]

[DOI: 10.29252/sjfst.2.2.1]

References

Borenstein, J.; Koren, Y. (1991). "The vector field histogram-fast obstacle avoidance for mobilerobots". Robotics and Automation, IEEE Transactions on 7 (3): 278–288. doi:10.1109/70.88137. Retrieved 2008-06-30.

O. Khatib, “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots”, International Journal of Robotics Research, Vol. 5, No. 1, pp.90-99, 1986.

Song K, Chang C. Reactive navigation in dynamic environment using a multisensor predictor. IEEE Transactions on Systems, Man, and Cybernetics 1999;29(6):870–80.

Pratihar DK, Deb K, Chosh A. A genetic-fuzzy approach for mobile robot navigation among moving obstacles. International Journal of Approximate Reasoning 1999;20:145–72.

Aranibar D, Alsina P. Reinforcement learning-based-path planning for autonomous robots ENRI: Encontro Nacional de Robo´ tica Inteligente, 2004.

Park J, Kim J, Song J. Path Planning for a robot manipulator based on probabilistic roadmap and reinforcement learning. International Journal of Control, Automation, and Systems 2007;5:674–80.

Tsypkin, Adaptation and Learning in Automatic Systems. New York: Academic, 1971

Prestero T. 1994. Verification of a Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle. MTS/IEEE OCEANS 2001 Conference, Vol. 1, pp. 450 – 455.

Seyyed Mohammad Reza Farshchi. 2011. A Novel Implementation of G-Fuzzy Logic Controller Algorithm on mobile Robot Motion Planning Problem. Computer and Information Science, Vol. 4, No. 2;102-114.

[Downloaded from sjfst.srpub.org on 2026-09-16]

[DOI: 10.29252/sjfst.2.2.1]