

PAPER • OPEN ACCESS

## Mobile robot position estimation using Euler-Maruyama algorithm

To cite this article: Mohamed G Abd Elfatah *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **610** 012074

View the [article online](#) for updates and enhancements.



**ECS** **240th ECS Meeting**  
Digital Meeting, Oct 10-14, 2021  
**We are going fully digital!**  
Attendees register for free!  
**REGISTER NOW**

# Mobile robot position estimation using Euler-Maruyama algorithm

Mohamed G Abd Elfatah<sup>1,3</sup>, Hany Nasry Zaky<sup>1,4</sup> and M Gharib<sup>2,5</sup>

<sup>1</sup>Mathematics Department, Military Technical College, Cairo, Egypt.

<sup>2</sup>Mathematics Department, Faculty of science, Ain shams university, Cairo, Egypt.

<sup>3</sup>mohamedgamal@mtc.edu.eg.

<sup>4</sup>hanynasry@mtc.edu.eg.

<sup>5</sup>mgmahmoud51@gmail.com.

**Abstract.** Mobile robots became very important and more familiar in commercial, industrial and military applications. In many unmanned vehicle applications, sensors and control systems are mounted on mobile robots to identify the surrounding environment for obstacle avoidance. This work aims to estimate the mobile robot location to compensate the time delay problem which appears in teleoperation. According to the stochastic nature of the mobile robot teleoperation, the kinematics equation of mobile robot will be converted into stochastic differential equations (SDEs). Euler-Maruyama algorithm used as it is one of the most popular numerical algorithms to approximate SDEs solution. A simulated results for the solution is produced which seem to be good comparing with mobile robot ideal path simulation.

**Keywords:** Euler-Maruyama method, stochastic differential equation, teleoperation.

## 1. Introduction

All natural events are stochastic and therefore they are highly affected by random events. Environmental phenomenon, such as thunders, earthquakes, storms and hurricanes are all involved with random events and processes. Since all engineering systems work in natural and physical environments, so randomness is an important factor to be taken into consideration in systems design. In engineering, noise is considered as random process in which an unwanted signal that disturbs the performance of the system. Consequently, mathematics which deals with randomness and stochastic processes is of high importance for engineering applications[1]. Unlike deterministic events, random processes are not precisely predictable. However, their determination can be estimated using probability and stochastic analysis. That's why stochastic differential and difference equations become essential for analyzing dynamic systems that have stochastic nature. Hence, in order to predict the future states of a stochastic dynamic system with random processes, noise should be modeled using SDEs.

Stochastic differential equations incorporate random white noise and different types of random processes, within the dynamic systems different mathematical methods exist for formulating stochastic processes and how they affect dynamic systems. Brownian motion is a well-defined and precise model used by researchers to study the effect of noise on dynamic systems. Since SDEs cannot be solved using classic calculus and algebra, closed form solutions are not always possible. As a result, numerical methods (such as Euler) are used to provide an estimation for the behavior of the dynamic systems. Stochastic partial differential equations used for solving stochastic processes is a progressive field for researchers as all the algorithms used the ordinary differential equations do not numerically converge to stochastic processes. Stochastic differential equations may not be solved analytically[2].

This solution depends on the dynamic system analysis and is obtained using n-dimensional Euclidean space theorem determined by m-dimensional Brownian motion. It is of great importance in such algorithm to know the nature of the solution of the SDE associated with the dynamic system in order to understand the behavior of the system and the relation between its future and current states. Hence, this fact is thoroughly analyzed in this work for the mobile robots that are being studied as



stochastic dynamic systems. Moreover, MATLAB is used for numerical computations and simulations.

*1.1. Stochastic process.* One of the main characteristics of SDEs is the stochastic character which can be defined as  $\{X(t), t \in T\}$  that is a group of random variables. Time is usually taken as index  $t$ . Thus, at time,  $X(t)$  will be the state of the process. Where  $t$  represent time,  $T$  represent time interval and  $X(t)$  represent the random variable at time( $t$ )[3].

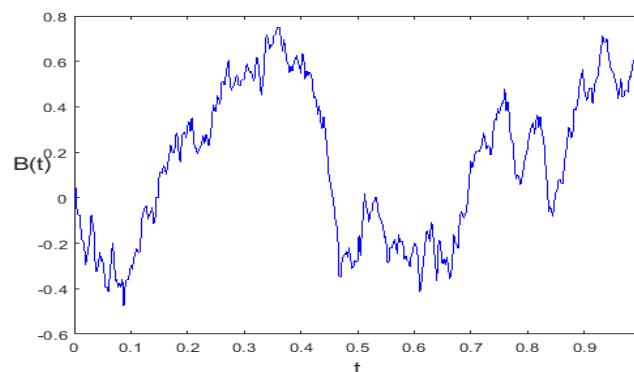
Some examples for the state of the process,  $X(t)$  is the number of customers entered a mall by time  $t$ ; or the number of clients exist in the market at certain time  $t$ ; or the total number of products that have been sold in the market by time  $t$ .

*1.2. Brownian motion.* Brownian motion, also known as Wiener process, over the interval  $[0, T]$  is defined as random variable  $B(t)$  which rely on time at which  $t \in [0, T]$  and fulfills the following conditions:

1.  $B(0) = 0$ , (its probability = 1).
2. For  $0 < v < t < T$  the random variable  $B(t) - B(v)$  has a normal distribution with mean equal to zero and standard deviation  $\sqrt{t - v}$ .
3. Increments Independence. For  $0 < t_1 < t_2 < \dots < T$  the increments  $B_2(t) - B_1(t)$  and  $B_t(t) - B_{t-1}(t)$  are independent.
4.  $B(t)$  Is a continuous process.

There are some features of Wiener processes of great value that should be taken into consideration.[4] In the beginning, we should know that a Wiener process is a Markov process which means that the probability distribution of the process future values depends only on its current value and independent of the past values. Therefore, for predicting the future values the only required value is the current one. In spite, the continuity of the Wiener process, it is not differentiable at any time. Furthermore, a Wiener process does not have bounded variation, i.e., a Wiener process can have any real value no matter how large or how negative this value is[5].

For computational studies, discretized Brownian motion should be taken into consideration, where  $B(t)$  is particular at discrete  $t$  values. So, we take  $\delta t = T/N$  for some positive integer  $N$  and let  $B_j$  denote  $B(t_j)$  with  $(t_j = j\delta t)$ . Condition 1 states that  $B(t_j) = 0$  with probability 1, and conditions 2 and 3 also say that  $B_j = B_{j-1} + dB_j$ ,  $j = 1, 2, \dots, N$  where each  $dB_j$  is an independent



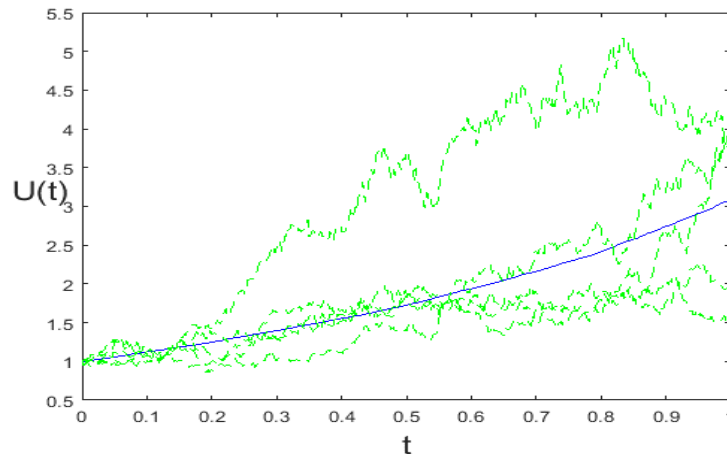
random variable with form  $\sqrt{\delta t}N(0,1)$ ,[4] a simulation of discretized Brownian motion across the interval  $[0,1]$  with number of periods  $N = 600$  is shown in the following figure (1).

**Figure 1. Discretized Brownian motion**

Now, we have sample average stochastic function of the form:

$$u(B(t)) = e^{(t+W(t))} \quad (1)$$

A simulation of that function is placed along 2000 discretized Brownian paths. The mean of  $u(B(t))$  over the whole five paths is figured with a solid blue line and the five paths are represented using a dashed green line.



**Figure 2. Representation of  $u(B(t))$**

In figure (2), we realized that even though  $u(B(t))$  is not straight over the individual paths, its sample mean seems to be straight. The mean value of  $u(B(t))$  found to be  $e^{(1.125t)}$ . It is realized that the maximum difference between the sample mean and the exact expected value over all points  $(t_j)$  equal to 0.0511. And by increasing the number of samples to 6000 the error will decrease to 0.0216.

*1.3. Stochastic Integrals* Let  $a = 0 \leq t_0 \leq t_1 \leq \dots \leq b$  be a collection of points over the interval  $[a, b]$ . we can define the Riemann integral in the limit form:

$$\int_a^b h(x) dx = \lim_{\Delta t \rightarrow 0} \sum_{k=1}^m h(t'_k) \Delta t_k \quad (2)$$

Where  $\Delta t_k = t_k - t_{k-1}$  and  $t_{k-1} \leq t'_k \leq t_k$ . Also, we can define the Ito integral in the limit form[6]:

$$\int_a^b h(t) dB_t = \lim_{\Delta t \rightarrow 0} \sum_{k=1}^m h(t_{k-1}) \Delta B_k \quad (3)$$

Where  $\Delta B_k = B_{t_k} - B_{t_{k-1}}$  represents a Brownian motion steps over the interval  $[a, b]$ .

Pay attention to a main difference here; in the Riemann integral  $t'_k$  could be selected at any point in the interval  $(t_{k-1}, t_k)$ , but it is required to be the left endpoint of that interval in Ito integral, caused by random functions  $f$  and  $B_t$ , so Ito integral can be written as:

$$I = \int_a^b f(t) dB_t \quad (4)$$

The previous integration can be stated in differential form as  $dI = f dB_t$ . The differential  $dB_t$  of Brownian motion  $B_t$  is known as white noise. We found that a mixture of the drift and the diffusion of Brownian motion is a distinctive solution. The chain rule for stochastic differentials (Ito formula) should be known if we intend to have the solution of SDEs analytically[7].

$$S = f(t, X),$$

$$\text{So, } dS = \frac{\partial f}{\partial t}(t, X) dt + \frac{\partial f}{\partial x}(t, X) dx + \frac{1}{2} \frac{\partial^2 f}{\partial x^2}(t, X) dx dx \quad (5)$$

Where the  $dx dx$  expression is deduced by using the identities:

$$\begin{aligned}
 dt dt &= 0 \\
 dt dB_t &= dB_t dt = 0 \\
 dB_t dB_t &= dt
 \end{aligned}
 \tag{6}$$

## 2. Dynamic Systems

Control systems can be subcategorized into dynamic and static control systems. A dynamic system can be defined as a function that predicts future states based on current states. Dynamic systems can frequently be found in nature as well as the industry and technology. They range from simple systems like to difficult ones. Dynamic systems yield a transitory prediction of future states that follow current states as an implicit relation that is either a differential equation, difference equation or other time scale relations. Some dynamic systems can be modelled using ordinary differential equations (ODE's), yielding a unique closed form solution. However, for most of these systems, determining the future states requires iteration using numerical methods, which is referred to as integrating the system in popular texts. For solvable dynamic systems, the future states, also referred to as trajectory or orbit, can be determined using a proper initial point. Since solving or integrating dynamic systems involves sophisticated mathematical operations. Also, numerical methods are applied using computers and electronic platforms to approximate future points. Hence, understanding mathematical iterative solutions using computer aided tools (CAD), such as MATLAB, is important for analyzing dynamic systems[1].

The dynamic systems could be configured into two main types: deterministic and stochastic systems. Deterministic dynamic systems restrict each future state to follow one current state. In other words, they are associated with one-to-one relationships. On the other hand, in stochastic systems, future states are affected by random events and variables. In engineering, noise is the fundamental element behind random and at times unexpected behaviour of systems. Noise is an undesired, disturbing and random signal that is present in most engineering applications and should be carefully studied for precise analysis. Therefore, (SDEs) are used for analyzing and modelling of these systems, such as stock market, mobile robots, and space shuttles.

## 3. Stochastic differential equations

In this section, (SDEs) and their solution are discussed. It should be explained what is meant by (SDEs), SDE is a differential equation that contains some of its terms and its solution are stochastic processes. SDEs became a very vital tool in mathematical modelling which is essential in optimal and control problems, also in financial mathematics[2]. A common form for a SDE is:

$$\begin{aligned}
 dX(t) &= f(X(t))dt + h(X(t))dB(t) \\
 X(0) &= X_0 \\
 0 &\leq t \leq T_0
 \end{aligned}
 \tag{7}$$

Where  $B$  represent a Brownian motion and  $f$  and  $h$  are known functions. We know that the solution is also a stochastic process, which can be deduced as integral equation.

$$\begin{aligned}
 X(t) &= \int_0^t f(X(s))ds + \int_0^t h(X(s))dB(s) \\
 0 &\leq t \leq T
 \end{aligned}
 \tag{8}$$

The first integral ( $ds$  integral) represent an ordinary Riemann Integral while the other one ( $dB$  integral) represent the Ito Integral. It is known from the properties of Brownian motion that the Wiener process  $B(t)$  is not differentiable anywhere, so the classical calculus could not be useful in dealing with the Ito integral, a Stochastic calculus is needed to get the solution of such integral. Ito integral has been explained in section 1.3 in details. Note that, If  $h(X(t))$  is equal to zero and  $X$  is constant, the equation will be reduced to an ordinary differential equation.

$$\frac{dX}{dt} = f(X(t)) \quad , X(0) = X_0
 \tag{9}$$

The Black Scholes diffusion equation (10) is a very important example in finance, which can explain some features of SDEs[8].

$$\begin{aligned}
 dX &= \rho X dt + \sigma X dB_t \\
 X(0) &= X_0, \rho \text{ and } \sigma \text{ are constants.}
 \end{aligned}
 \tag{10}$$

Although the equation is quite simple, it is clear that it can be exactly solved led to its significance, by constructing a closed form formula available for the pricing of simple options (Black and Scholes, 1973). It is found that the solution of the Black-Scholes stochastic is a geometric Brownian motion as in the following equation[9]:

$$X(t) = X_0 e^{(\rho - 0.5\sigma^2)t + \sigma B_t} \quad (11)$$

The main aim of this paper is to get an estimation of mobile robot position numerically by one of the most common algorithms which is Euler-Maruyama Algorithm to compensate the time delay problem in teleoperation.

#### 4. The Euler-Maruyama Method

From the nature of SDEs, it is difficult to solve SDEs analytically or by classical calculus. So, numerical methods and algorithms take place to find the solution of SDEs. The most used and common approach for solution approximation is the simulation of sample paths. The first step in such approach is to discretize the time interval into a large numbers of intervals. An independent pseudo-random path can be generated with the aid of MATLAB program. It is common to represent SDE in differential form as in equation (10).

Note that, we will utilize the SDE in differential form (10) rather than the integral form (8) because it is prohibited to write  $(dB(t)/dt)$ , as Brownian motion is nowhere differentiable with probability equal one. To solve equation (10) by a numerical method over the interval  $[0, T]$ , we must first discretize the interval. Let  $\Delta t = T/N$  where  $N$  is a number of discretization[10]. We let  $B(t)$  denote a value of the Wiener process at  $\tau_k = k\Delta t$  with initial value  $B(0) = 0$ . Then we have

$$B(k) = B(k-1) + dB(k) \quad ; k = 1, 2, 3, \dots, N \quad (12)$$

Where each  $dB(k)$  in equation is normally distributed independent random variable. A MATLAB command "randn" is used to generate  $N$  independent pseudorandom from normal distribution. It can be used in previous equation to get  $B(k)$ . The numerical approximation to  $X(\tau_k)$  will be represented by  $X_k$ [4]. Now, Euler-Maruyama (EM) method will have the form

$$X_k = X_{k-1} + f(X_{k-1})\Delta t + h(X_{k-1})\Delta B \quad ; k = 1, 2, 3, \dots, N \quad (13)$$

Where  $\Delta B = B(k+1) - B(k)$  For expediency, we permanently take the step size  $\Delta t$  for the numerical method to be an integer multiple  $S > 1$  of the increment  $\delta t$  for the Brownian path. This guarantees that the set of all points  $\{t_k\}$  on which the discretized Brownian path is produced contains the points  $\{\tau_k\}$  at which the approximated solution is calculated. In most of applications the Brownian path is identified as part of the problem data. If an analytical path is supplied, then subjectively small  $\Delta t$  can be used[4].

As an important step in robot control system design is to construct a mathematical model of the robot kinematics. The mathematical model transform the velocity of the robot into the generalized coordinate vector as follows[11]:

$$\begin{aligned} x_{t+1} &= x_t + V \Delta t \cos\theta_t \\ y_{t+1} &= y_t + V \Delta t \sin\theta_t \\ \theta_{t+1} &= \theta_t + \Delta t w \end{aligned} \quad (14)$$

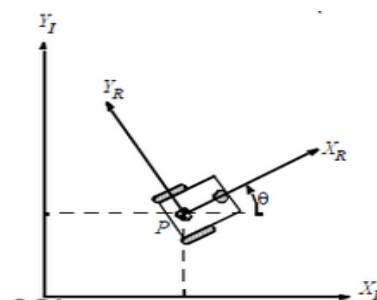


Figure 3. kinematic model of robot

Where the velocity given by the equation:

$$\begin{aligned} V &= r \left( \frac{w_r + w_l}{2} \right) \\ w &= r \left( \frac{w_r - w_l}{L} \right) \end{aligned} \quad (15)$$

The parameters in previous equations (14) and (15) represent:

- $w_r, w_l$  ..... Right and left wheel velocity.
- $x, y$  ..... The position of the robot relative to inertia frame.
- $\theta$  ..... The heading angle of the robot relative to inertia frame.
- $V$  ..... Linear velocity of the robot.
- $w$  ..... Angular velocity of the robot.
- $L$  ..... Distance between left and right wheel.
- $r$  ..... Radius of wheel.

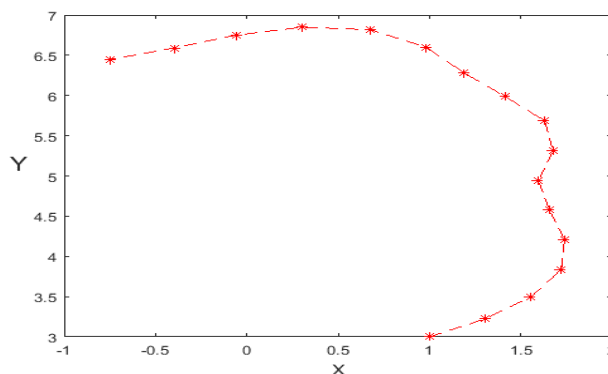
In this paper, the heading of robot will be random which mean that a random  $\theta$  is obtained in each time step. A random term  $B(t)$  weiner process (white noise, random function or Brownian motion) is added to the heading angle equation which is converted into stochastic differential equation. So, the kinematic model can be written as:

$$\begin{aligned} X_{t+1} &= X_t + V \Delta t \cos \theta_t \\ Y_{t+1} &= Y_t + V \Delta t \sin \theta_t \\ \theta_{t+1} &= \theta_t + \Delta t w + B_t \end{aligned} \quad (16)$$

Now, the kinematic model equations will be simulated to estimate the position of robot using Euler-Maruyama method by aid of MATLAB program[12]. In our program we set the following parameters:

$$\begin{aligned} w_r &= 4 \text{ rad/s} , w_l = 2 \text{ rad/s} \\ X_o &= 1 , y_o = 3 \\ \theta_o &= \frac{\pi}{8} \text{ rad} , L = 20 \text{ cm} , r = 20 \text{ cm} \end{aligned} \quad (17)$$

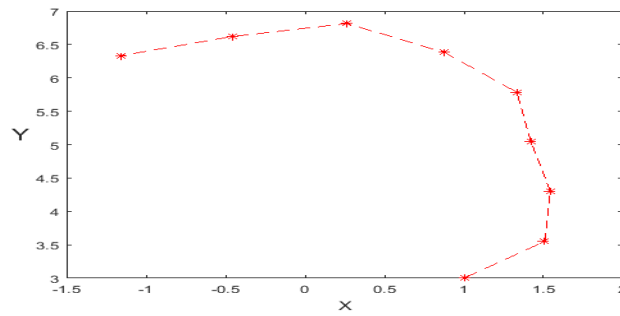
We calculate the discretized Brownian path over the interval  $[0,1]$  with  $\delta t = 2^{-6}$  and get the solution with step size 4 which is shown in



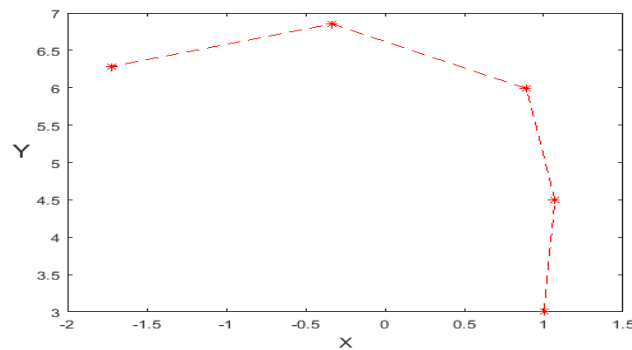
$\Delta t = S \delta t$  ,  $S =$  figure (4).

**Figure 4. Euler-Maruyama with S = 4**

Then we use different step size  $\Delta t$  with different  $S = 8, 16$  which shown in figure (5) and figure (6).



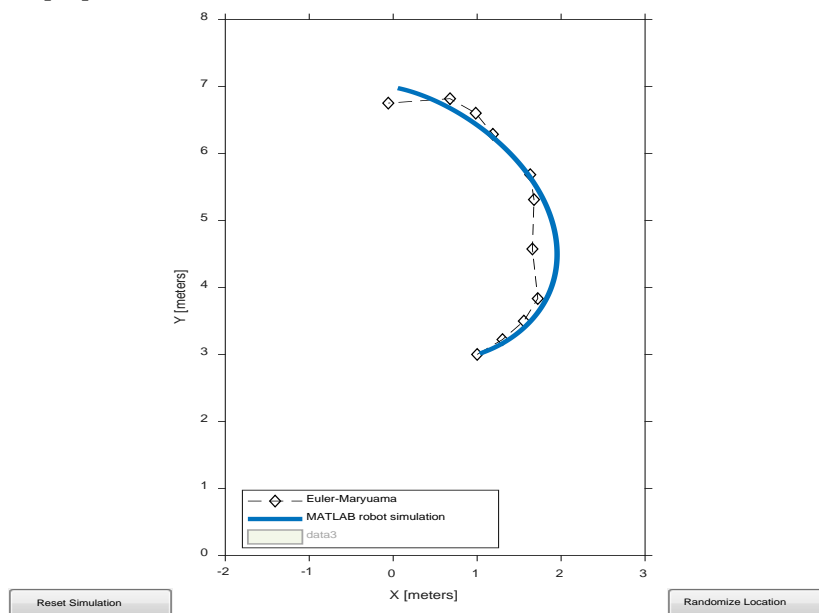
**Figure 5. Euler-Maruyama with S = 8**



**Figure 6. Euler-Maruyama with S = 16**

It is found that the difference between the three figures is the number of points that have been taken. So, the more points used the less error we have in the path.

A comparison between Euler-Maruyama approximated solution and a MATLAB robot simulation[12] along a predetermined path is shown in figure (7). The same parameters are used as in Euler-Maruyama solution. In MATLAB robot simulation, Pure Pursuit path following controller is used to drive a simulated robot along a desired path. It is created and computes the control commands (linear and angular velocities) to follow the given path. The computed control commands are used to drive the simulated robot along the desired trajectory to follow the desired path based on the Pure Pursuit controller[12].



**Figure 7. Euler-Maruyama and MATLAB robot simulation.**



In figure (7), there is an error between the approximated solution by Euler-Maruyama and the MATLAB robot simulation in waypoints and destination point which is caused by the stochastic nature of the heading angle so it has an impact in robot position. Euler Maruyama has a strong convergence of order 0.5, so we can reduce the error in mobile robot estimated position by using another numerical methods with strong convergence order more than 0.5 such as Taylor expansion and Rung Kutta methods[8].

## 5. Conclusion

In this paper, we have discuss the stochastic differential equations with its main definitions and properties. Also, we discuss one of its common numerical methods of solution which is Euler-Maruyama method. Later, we discussed the Dynamic systems and their two categories: deterministic and stochastic and how stochastic process appear in many important application and systems. Euler Maruyama method applied to one of the dynamic systems (robot) to estimate its position with the aid of MATLAB program in calculations and simulation. It is found that an acceptable error between the MATLAB robot simulation (ideal case) and the Euler Maruyama approximated solution which caused by stochastic nature of the robot heading angle. Euler Maruyama give a strong convergence of order 0.5 and the error can be decreased by using another strong convergence numerical methods such as Rung Kutta method and Taylor expansion.

## 6. References

- [1] "Introduction," in *Control of Indefinite Nonlinear Dynamic Systems: Induced Internal Feedback*, S. V. Emel'yanov, I. Burovoi, and F. Levada, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 1-22.
- [2] B. Øksendal, "Stochastic Differential Equations," in *Stochastic Differential Equations: An Introduction with Applications*, B. Øksendal, Ed., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 38-50.
- [3] !!! INVALID CITATION !!!
- [4] D. J. Higham., "An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations," *SIAM Review*, vol. 43, pp. 525-546, 2001.
- [5] P. Todorovic, "Elements of Brownian Motion," in *An Introduction to Stochastic Processes and Their Applications*, P. Todorovic, Ed., ed New York, NY: Springer New York, 1992, pp. 62-91.
- [6] B. Øksendal, "Ito Integrals," in *Stochastic Differential Equations: An Introduction with Applications*, B. Øksendal, Ed., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 14-31.
- [7] B. Øksendal, "Stochastic Integrals and the Ito Formula," in *Stochastic Differential Equations: An Introduction with Applications*, B. Øksendal, Ed., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 32-43.
- [8] E. Platen, "An introduction to numerical methods for stochastic differential equations," *Acta Numerica*, vol. 8, pp. 197-246, 2008.
- [9] T. Sauer, "Numerical Solution of Stochastic Differential Equations in Finance," in *Handbook of Computational Finance*, J.-C. Duan, W. K. Härdle, and J. E. Gentle, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 529-550.
- [10] P. Kloeden and E. Platen, "Numerical methods for stochastic differential equations," *Stochastic Hydrology and Hydraulics*, vol. 5, pp. 172-172, 1991/06/01 1991.
- [11] C. Myint and N. N. Win, "Position and velocity control for two-wheel differential drive mobile robot," *Int J Sci Eng Technol Res (IJSETR)*, vol. 5, 2016.
- [12] H. Gilsing and T. Shardlow, "SDELab: A package for solving stochastic differential equations in MATLAB," *Journal of Computational and Applied Mathematics*, vol. 205, pp. 1002-1018, 2007/08/15/ 2007.