IJournals: International Journal of Software & Hardware Research in Engineering
ISSN-2347-4890
Volume 4 Issue 12 December, 2016

A Nonlinear Predictive Control for Mobile Robot Navigation

Author: Mohammadghasem Bahrami Samani

Deptartment of Electrical Engineering, Islamic Azad University of Khomeini Shahr, Khomeini Shahr, 8418148499, Iran.

E-mail: mbsamani68@yahoo.com

Abstract

This paper presents a novel mobile vehicle navigation algorithm based on the stability analysis of the model predictive control approach. The dynamic model of the prediction of movement obstacles is performed with the navigation function to obtain a new virtual vehicle model that generates candidate feasible trajectories for the motion planner. Stability of the nonlinear model predictive control system is obtained by the passivity concept providing a guaranteed task completion. The proposed approach is adapted for a mobile vehicle but the work suggests that the nonlinear model predictive control concept can be adapted for the navigation purposes for a broad range of mobile vehicle models.

Keywords: predictive control, movement obstacles, mobile vehicle, nonlinear model, navigation function.

1. INTRODUCTION

This paper considers developing a new motion and navigation planner that helps the vehicle to move from an initial position to a goal position while avoiding obstacles. A comprehensive overview of motion planning algorithms has been presented in [1] and [2]. Potential field approach for obstacle avoidance was introduced in [3] while its extension to moving obstacles was examined in [4]. The work given in [5] included the general path planning with local minima problem observation using multi d.o.f manipulators and motion planning in a structured environment. The concept of navigation functions was illustrated in [6] and [7] and a new class of those functions appropriate for nonholonomic motion planning were presented in [8].

The research on motion planning evolved by including the vehicle motion dynamics constraints within the well known dynamic window approach [9] and [10]. This subject was extended to the high-speed navigation of a mobile robot in [11] by the global dynamic window approach, as the generalization of the dynamic window approach. A combination of the dynamic window approach with other methods yielded to some improvements in long-term real world applications [12]. Dubowski and Iagnemma extended the dynamic window approach to rough terrains using the vehicle curvature-velocity space. In this space the stability constraints of the vehicle, expressed by limit values of the roll-over and side slip indexes, can be easily described. The given algorithm was also suitable for high speed vehicles and appropriate for real-time implementation [13]–[15]. The RHC/CLF (Receding Horizon Control/Control Lyapunov Function) scheme developed in [16] used the concept of control Lyapunov function to obtain the stability of RHC scheme. The authors have presented the generalization of the RHC/CLF scheme and found the conditions of connectivity between pointwise min-norm as well as optimal controller using the concept of control Lyapunov function. In [17] the authors have implemented this scheme (MPC/CLF, Model Predictive Control-Control Lyapunov Function) for the navigation planning of a unicycle mobile vehicle. Constraints such as obstacles, vehicle velocity and acceleration limits, as well as a constraint which decreases the control Lyapunov function along trajectories of the system, have been implemented within the MPC/CLF optimization procedure. In [17] the authors used the navigation function as a virtual potential term in the control Lyapunov function providing the information on the goal position through the feedback.

Passivity-based nonlinear model predictive control scheme with guaranteed closed loop stability for any prediction horizon has been recently presented in [18]. Similar to [16], the authors have examined the relation of this approach to the optimal control. Inspired by this idea from the control theory concept, we propose a framework for mobile robot navigation using a passivity-based nonlinear model predictive control approach (PB/MPC) in order to stabilize the given goal position.

Globally, the goal position is included into the dynamical model of the vehicle by the navigation function that depends both on the goal position and the presented obstacles. This is done by the energy-shaping technique and by artificially injecting a damping based on the passivity control concept making the goal position to be a stable equilibrium point. A new obtained virtual vehicle model is used to generate candidate feasible trajectories for the motion planner. Locally, the nonlinear model predictive control is used to find the best trajectory along the given time horizon, among all generated candidates satisfying all presented constraints such as obstacles and vehicle dynamics. We show that the PB/MPC navigation framework is a convergent dynamic window approach that guarantees finding the goal position. The MPC/CLF navigation framework is also a dynamic window approach but it requires the knowledge of the control input based on the Lyapunov function that stabilizes the given goal position. In general, this is hard to achieve for a complex system and, in our case, for more complex vehicle models. The main advantage of the PB/MPC navigation framework comparing to the MPC/CLF is that it gives a straightforward procedure to obtain the navigation algorithm that generates feasible trajectories while moving the vehicle toward the goal position. This is possible since the passivity-based constraint, that is included into the MPC optimization scheme, requires some achievable steps assuming that all model signals are available from measurements. For this reason, the PB/MPC could be seen as a generalized navigation technique that can be easily adapted to the variety of vehicle models not only acting in flat but also in rough terrains.

2. APPROACH OF SOLVING THE PROBLEM

In this part, the target is to demonstrate the manner of modeling and approach of solving the problem. For this purpose, the robot model and obstacles for anticipation are explained. Afterward, predictive control will be discussed.

2.1 Modeling of Two-Dimension Mobile Robot

The considered robot is a wheeled robot controlled by the angular velocity steering wheel. The robot moves in two-dimension plate and its condition is determined through four degrees freedom at any moment. Such degrees of freedom are longitudinal coordinate of robot mass center, cross coordinate of robot mass center, and angle position of the robot proportion to horizontal axis as well as the steering wheel angle of the robot [12]. In fact, the robot is considered as an automobile which includes two active wheels in front that thrust power and steering wheel torque import force to it. Also, the two wheels in the rear for preserving of the balance. Such a robot has equations as below:

$$\dot{x} = v\cos\theta$$
$$\dot{y} = v\sin\theta$$
$$\dot{\theta} = v\frac{tg\,\varphi}{l}$$
$$\dot{\varphi} = \omega$$
(1)

Where x is the robot Longitudinal coordinates of the mass center, θ is the angle between the robot axis and horizontal axis and φ is the angle between the wheels of robot and its axis. v is the linear velocity of the robot and ω is the angular velocity of the steering wheel. Also, L is the distance between the active wheels and inactive ones. For this purpose, see fig. 1. Since predictor control is indeed a discrete one and work with the discrete model of the system, the dynamic of the robot in discrete state-space is defined as:

$$x(k+1) = x(k) + v(k)\cos\theta(k)T$$
$$y(k+1) = y(k) + v(k)\sin\theta(k)T$$
$$\theta(k+1) = \theta(k) + v(k)\frac{tg\varphi(k)}{l}T$$
$$\varphi(k+1) = \varphi(k) + \omega(k)T$$
(2)

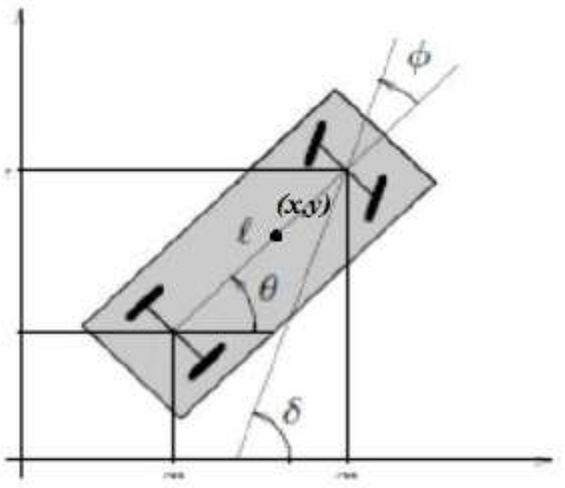

Fig 1: Two-dimension schematic of the robot

Eq. 2 actually is the very eq. 1 with the period of T that breaks into four equations through Euler approximation for derivation. K is the number of sampling. v and ω are control inputs and should be determined in a desirable manner every moment. Through having $v(k)$ and $w(k)$ as well as $x(k), y(k), \theta(k), \varphi(k)$, the value of $x(k+1), y(k+1), \theta(k+1), \varphi(k+1)$, which are related to k+1 moment, are obtained and therefore it is possible to predict the situation of the robot.

3.THE DYNAMIC MODEL OF THE PREDICATION OF OBSTACLES MOVEMENT

In order to robot avoids from obstacles, it should preserve its distance with them. Suppose that obstacles are symmetric meaning that no matter which direction the obstacle is. We cannot consider back and forth for the obstacle. No matter in what direction the obstacle is, the robot should approach it not at all. It is assumed that the sensor which the robot use is the sensor measuring the location of robot and relative distance. Also, the robot has memory and is cable of reminding the last location of the robot. Therefore, it is able to calculate the obstacle movement in a period of sampling. With this in mind, the obstacle model is defined as:

$$\begin{aligned} x_0(k+1) &= x_0(k) + \Delta x_0(k-1) \\ y_0(k+1) &= y_0(k) + \Delta y_0(k-1) \qquad (3) \\ \Delta x_0(i) &= x_0(i) - x_0(i-1) \qquad i \geq 2 \end{aligned}$$

Where $x_0(k)$ and $y_0(k)$ are longitude and latitude of the mobile robot in the moment of k. In fact, this equation declares that it is assumed that the robot goes through the current step consistent to last sample rout.

4.PREDICTABLE CONTROL

In the methods of optimum control, the principle is to obtain control signal values in a way that the given performance criteria is optimized through present dynamic equations according to recognized dynamic from the system. But in some cases in which the model of system or environment is changed as well as there is no adequate accuracy for the system, the need for another approach is obvious in order to be adopted to main system and then the control is done on the adapted control. This method is called adaptive control based on the model [22].

The heart of model predictive control is the process model itself. Models can be classified by various features. Since MPC requires the solution of a model to predict future process outputs, the form of the model selected has a large impact on our ability to implement MPC. Some specific categories of models are discussed below:

• Linear or nonlinear: The response of linear dynamic systems obeys the principle of superposition, i.e., the response of the system to a linear combination of inputs is a linear combination of their responses to each input separately. Many (perhaps most) systems of engineering interest approximate this behavior for small inputs, which accounts for the universal study and application of linear control theory. In almost any control application, linear design techniques are usually the first to be attempted and are completely satisfactory for many engineering applications, especially those involving regulation about a steady-state operating point. Linear models are used extensively in the industrial practice of MPC. Nonlinear models have no specific characteristics except that they don't include the linear case. This makes it difficult to generalize, since nonlinear models can be devised to have almost any characteristic or demonstrate any strange behavior. For many engineering applications, it is usually appropriate to assume that the models are continuous with respect to model parameters and that differential or difference equation models satisfy growth conditions that permit solutions to be computed.

• Continuous-time or discrete-time: Most of the physical laws that are used by engineers to develop models are presented as differential equations with time as the independent variable. Before the widespread availability of digital computers, differential equation models were the central tool of researchers and control engineers for the study of dynamic systems. If simulations were needed, analog computers could be used. Prediction was rarely used as an element of control engineering, since complex models could not be solved using paper-and-pencil methods. With the advent of digital computers, the study of difference equations, which had been relegated to a minor role previously, assumed a new significance. Since MPC is universally implemented via digital computer, difference equations are the model of choice. Differential equation models must be discretized for computer solution. In this chapter, our examples are based on continuous-time models that are discretized using orthogonal collocation.

• Distributed parameter or lumped parameter: A distributed parameter model involves partial differential equations, instead of ordinary differential equations. An example of a distributed parameter model would be that for a plug flow reactor, in which the changes in chemical concentrations in the reactor are subject to both spatial variation and variations in time. Although we do not specifically consider distributed parameter models in this chapter, the basic MPC concept is completely applicable to systems described by distributed parameter models, with a corresponding increase in computations necessary to solve the model.

• Deterministic or stochastic: All physical processes are subject to unpredictable disturbances. Disturbances can affect MPC control design and operation in at least two distinct ways: — In the process of model identification, a model is often selected from some class based on experimental results. The selection process implicitly or explicitly uses assumptions about the disturbances to select and evaluate the best model of the class. These assumptions have a direct impact on the model selected. — After the model identification phase is complete, the assumptions about the disturbances are sometimes discarded, and control design may be based on this "nominal" model. If the model allows us to predict the statistics of process variables based on assumptions about random effects on the models, we say that it is a stochastic model.

In general, better control can be achieved by using explicit stochastic models [16, 17] in the prediction phase of MPC, but with a much higher computational burden. The traditional approach to stochastic MPC is through dynamic programming [2, 3], which suffers from combinatorial growth in the number of optimization variables with increasing horizon length (Bellman's "curse of dimensionality"). Better ways to incorporate stochastic models in MPC are active areas of research.

• Input-output or state-space: As indicated by the name, input-output models provide a relation between the process input and the output, without reference to variables internal to the process. An input-output model of a distillation column, for example, might relate the temperature of a side-product stream to reboiler duty, without consideration of individual tray temperatures or compositions; whereas, a state-space model might include equations relating all internal compositions and flow rates to the reboiler duty, with the side product temperature being provided as a function of the composition. Because most nonlinear state-space models are based on heat, mass and momentum balances, each state has a physical interpretation. States may also be generated as mathematically convenient intermediate variables of an input output process model. In the examples discussed in this chapter, the states have definite physical significance; however, the principles of MPC are applicable in either case. As a special class of input-output model, we should mention that artificial neural network models are becoming important in many engineering applications, including model predictive control, with a number of researchers providing important new results [20]. This chapter does not specifically discuss neural networks in MPC, although the above cited works make clear that artificial neural networks may be successfully used in MPC.

• Frequency domain or time domain: Frequency domain models in continuous time are based on the Laplace transform of continuous linear systems and are not used for model predictive control, except for the linear unconstrained problem. Depending on the context, these models may be described as input-output or discrete-time frequency domain models. Since few of the concepts of frequency domain analysis are applicable for nonlinear systems, frequency domain concepts play little or no role in nonlinear MPC.

• First-principles or "black box": Models that are derived from heat, mass and momentum balances are frequently called "first-principles" or "fundamental" models, in contrast to other modeling schemes that fit a data set to an arbitrary form. Both approaches have been used successfully in MPC applications. An example of the "black box" approach is that used in model identification using step or impulse response models, in which model coefficients are fitted to process data using statistical methods without regard to underlying physical principles. Basic principles of chemistry were applied to arrive at a model of the chemical reactions involved and mass balances were used to create an overall reactor model. The clearest tradeoff between the first-principles and black-box models lies in our confidence in predictions made by the model. A first principles model presumably can be used to predict over a wide range of conditions, even without prior operating experience, provided that the basic assumptions of the model remain valid. On the other hand, a model based on (for example) artificial neural networks has almost no predictive value outside the range of operating conditions where data has been collected. The best features of both approaches can perhaps be incorporated into a hybrid approach in which fundamental modeling is used for those portions of a process where the physical phenomena are well understood, and an input-output model used for the remainder of the process model. Each of these model characteristics has an impact on the implementation of MPC. Since MPC is an on-line control method, the speed of the computer algorithm used is essential. Linear models are very well suited to MPC because they may be solved quickly and the optimization problem may be posed as linear or quadratic programming problems, for which robust and reliable software is available. Using continuous models in MPC also impacts the speed of solution because it usually requires significantly more variables to represent a discretized model. Distributed parameter models require more time to solve because of the additional difficulty in solving partial differential equation models. From the class of input-output models, step or impulse response models have been favored for industrial application. In view of recent results concerning the use of linear state-space models [55], there appears to be little advantage in continuing with this practice. An equivalent state-space model is no less easy to identify, contains fewer unknown coefficients, can be evaluated more quickly by computer and can represent a wider range of processes. For the nonlinear case, the decision to use first-principles or input-output models is less clear. For small systems with well-understood physical phenomena, fundamental modeling is preferable, because of the ability of the model to predict beyond the range of existing operating data. On the other hand, model identification is easier for larger systems using black-box models. In conventional industrial MPC, installation of an MPC controller usually begins with identification of a linear input-output model in discrete time. Little or no effort is made to determine a first-principles model based on mass, energy or momentum balances. The objective of the modeling process is to determine a model that can be numerically evaluated quickly and that adequately describes the process dynamics in a neighborhood of some desired steady-state operating point. MPC based on these models has been extraordinarily successful in industry, but it is only applicable for stable processes that operate continuously near the nominal steady-state values. Although recent research results have extended the use of linear models for unstable processes [55], the range of operating conditions remains limited to those near the nominal steady state. MPC using nonlinear models is especially suited for batch or semi-batch operations in which process conditions can vary significantly over the course of a batch, or for continuous processes that are expected to experience widely varying operating conditions. In the past, the expense of computation often prevented even the consideration of the use of nonlinear models. As the power of computers available for on-line computations continues to increase, it has become feasible to consider using more complex models of various kinds. Today, the principle limitations in applying nonlinear models for on-line process control are model identification and robustness of control software and algorithms.

The models which are applied to MPC are often the models for showing the behavior of a complex dynamic system. Predicator control algorithm increases complexity of the system and is not suitable for simple system easily controlled by PID control. The main idea of MPC control is shown in Fig. 2.

4.1 Using Predictable Control For Perusing The Rout

Suppose that for a system like $x(k) = f(x(\mathrm{k}), \mathrm{u}(\mathrm{k}))$ the purpose is to obtain u(k) in a way that x determines favorable rout of xd in every moment.

For this purpose, we consider the cost function as below:

$$J(K) = \sum_{i=0}^{Np} ([x(k+i|k) - x_d(k+i)]^T Q[x(k+i|k) - x_d(k+i)] + u^T(k+i)Ru(k+i)) \qquad (4)$$

Where Q and R are the two positive matrix known as design parameters. Np is the predication and $x(k+i|k)$ means x evaluation (the robot location) in the k+i moment which is define as:

$$x(k+i|k) = \begin{bmatrix} x(k+i) \\ y(\mathrm{k+i}) \end{bmatrix} \qquad (5)$$

Moreover, xd means evaluation in the moment k+i based on information which we have had until the moment k and is equal to:

$$x_d(k+i) = \begin{bmatrix} x(k+i) \\ y(\mathrm{k+i}) \end{bmatrix} \qquad (6)$$

Also, the condition as fallows are applied for adjusting of the distance between the robot to the obstacle and pursuing the rout as well as the maximum values of outputs:

$$v_{\min} \le v \le v_{\max}$$
$$w_{\min} \le w \le w_{\max}$$
(7)

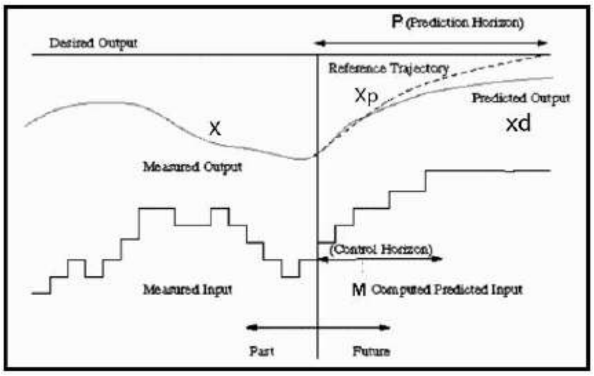

Fig 2: Control strategy of MPC

5.THE STRUCTURE OF GENETIC ALGORITHM

Fig. 3 shows the complete genetic algorithm which is developed in a way that the value of optimizer argument is actually a cost function. Basically, the method of optimization is used for minimizing of a cost function. The cost function considered for appropriate goal in this paper should be in a way that achieving the goal as well as avoiding obstacle are obtained.

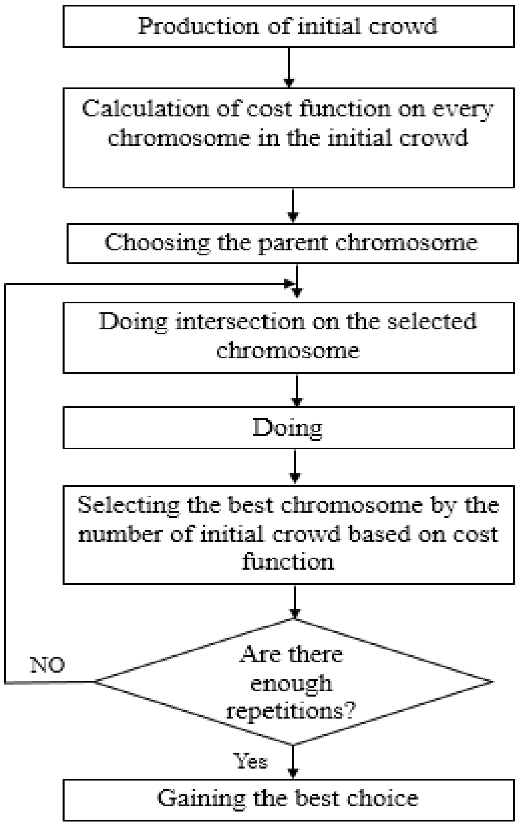

Fig 3: The complete genetic algorithm

6.THE DESIGNING OF THE ROUT THROUGH PREDICTION

On top of everything else, we should move the robot on the designed rout. It is necessary to obtain an appropriate rout for the robot. The purpose of solving the problem is to ignore dynamic obstacle and arriving to the destination. It is supposed that the robot is able to measure the distance of itself to the obstacle and destination. To achieving to this, we consider below rules:

- If the robot is near to the destination—it means that this distance is less than threshold value of Rth—control inputs should be equal to zero.
- If the distance of the robot to the obstacle is high, the robot remains on secure spot through moving the robot on the direct line and the appropriate value Xd should be on the location on the line joining the robot and the destination (Fig. 4).

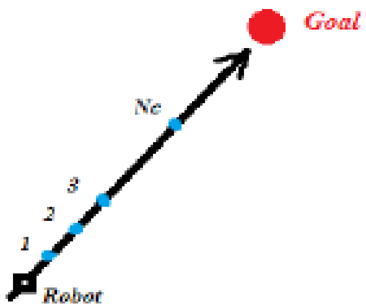

Fig 4: Designing of the rout in the secure spot

- If the obstacle is near to the robot or is approaching to the robot—it means that the robot is approaching to the obstacle—the appropriate rout is considered the same as part of a circle with a radius of Rsafe and the center is the obstacle (Fig. 5). This case is appeared when the stationary obstacle has less velocity than the robot.

Finally, the robot stays on this rout until it is placed in a secure spot or have a state that move on a direct rout on the condition that it does not face to the obstacle.

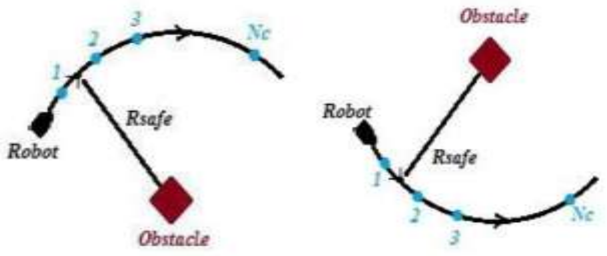

Fig: 5. Designed rout for ignoring obstacle

It is necessary to have knowledge about moving on a direct line or avoiding obstacles. Prediction is in a way that in every step, the next locations of the obstacle must be computed. The distance from obstacle is the minimum distance of the robot from the predicted location for the obstacle in Nc next steps:

$$D_{\min,k} = \min(\sqrt{[x_r(k+i|k)^2 - x_0^{est}(k+i|k)]^2} \quad (8)$$

Which the location of robot placing is defined as:

$$x_r(k+i|k) = \begin{bmatrix} x_r(k+i) \\ y_r(\mathrm{k+i}) \end{bmatrix} \quad (9)$$

Also, the location of obstacle placing is defined as:

$$x_0^{est}(k+i|k) = \begin{bmatrix} x_0(k+i) \\ y_0(\mathrm{k+i}) \end{bmatrix} \quad (10)$$

Through having above value in every moment and placing them to the above equations, in every step, we have $D_{\min,k}$ explaining the distance of robot to the obstacle and defines as:

$$D_{\min,k} = \begin{bmatrix} D_x \\ D_y \end{bmatrix} \quad (11)$$

ISSN-2347-4890
Volume 4 Issue 12 December, 2016

Now, $D_{\min,k}$ should be compare to the stationary value (Rsafe) to show that if the above equation is higher than Rth, it means that the robot is in the secure zone. But if it is less than Rth, the value of Xd should be design in a way that both the robot is not avoided too much from the main rout and the robot is placed in a secure spot. Fig. 6 shows the complete privacy of the operation.

7. SIMULATION RESULTS

In this part the approach is simulated. In the simulations, it will be shown that how the robot moves and the appropriate rout will be discussed. For showing the appropriate operation, different scenarios are considered. Our goal is to achieve the goal (blue ball) and avoiding obstacle (black ball).

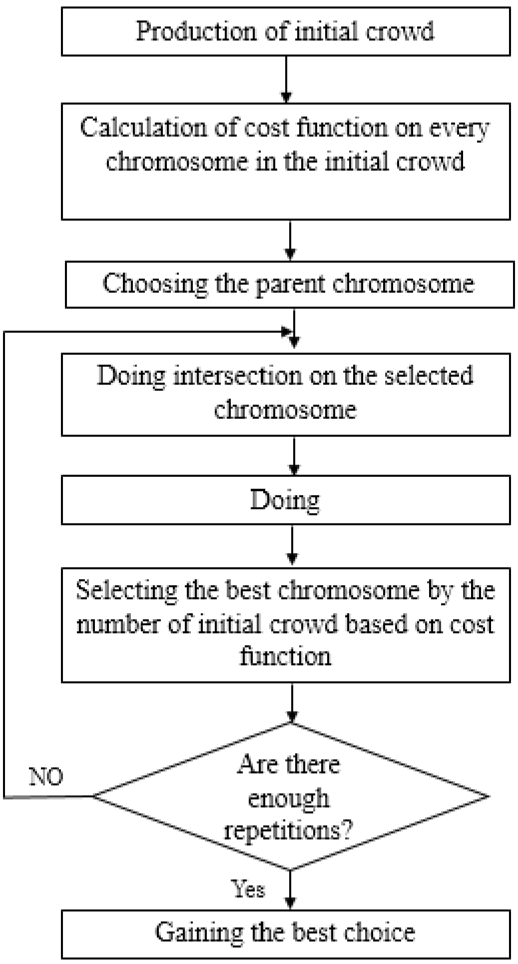

Fig 6: Complete privacy of the operation

In the first state, the obstacle is considered stationary and the location of obstacle is in a way that the obstacle changes its rout and the appropriate rout in a special moment select once that it is favorable for that time. Fig. 7 shows this point. In the next scenario the number of the obstacles is equal two in order to investigate the behave of the robot in this state. In this case, the robot is in the location of (0,0) facing to the destination and obstacles. First, the robot selects its rout like a straight line and then moves. In the next scenario, the obstacles are placed like the figure 9 and again it is investigated. The purpose of this scenario is that the behavior of the robot will be measured when the obstacles are in a horizon line and placed like a barrier in front of the robot.

In the next scenario the number of obstacles are increased into 3 and two different states of it will be discussed. In the Fig. 10, there are 3 obstacles and it shows how the robot moves along side of them.

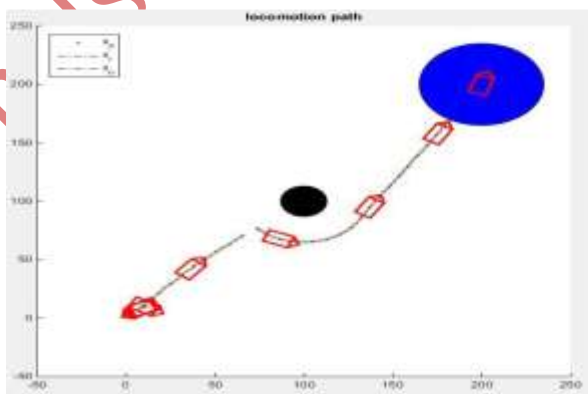

Fig 7: The rout of the robot in the presence of a stationary obstacle

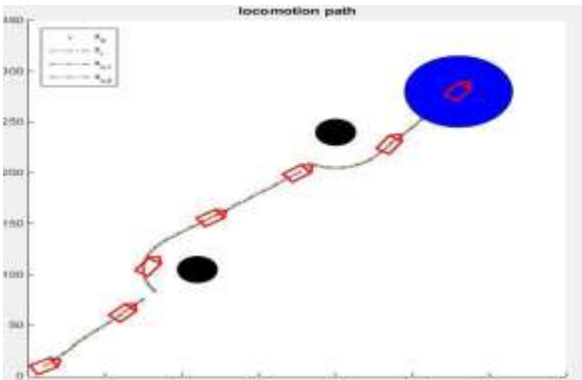

Fig 8: The rout of the robot in the presence of two stationary obstacles.

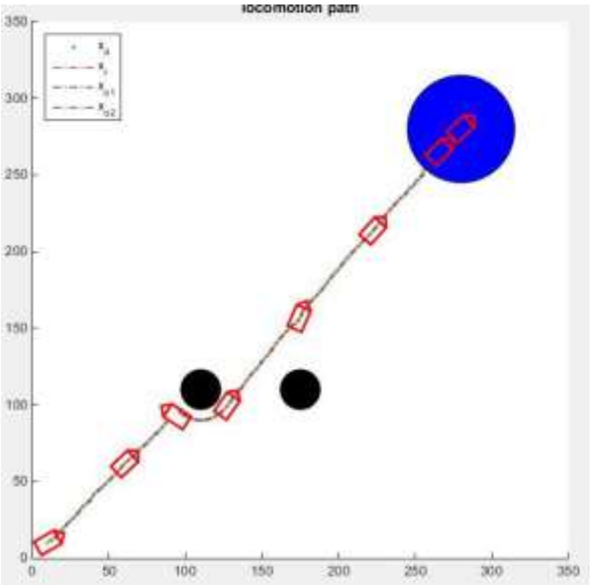

Fig 9: The rout of the robot in the presence of a stationary obstacle in a horizon line.

As it shows, the robot selects the shortest way also it considers the obstacles not to meet the obstacles.

7.1 Investigation of Algorithm in The Presence of Mobile Obstacles

In this part the algorithm is investigated through mobile obstacles and we discussed the behavior of the system.

In the first scenario, both obstacle and robot move until they arrive at (280, 100) that the robot senses the obstacle and then the robot change the line and goes into a straight line in the end. This description is shown in the Fig. 11.

In this case, there are two obstacles that one of them has a circular rout and the second one goes diagonally. This scenario is one of the most important in the investigation of the robot movement in the presence of mobile obstacles. According to the Fig. 12, it is obvious that the robot could have an appropriate decision and recognize every obstacle. Now, the waveform of output signals for the robot for the case of Fig. 12are shown in the Fig. 14, 15, 16, 17, 18, 19.

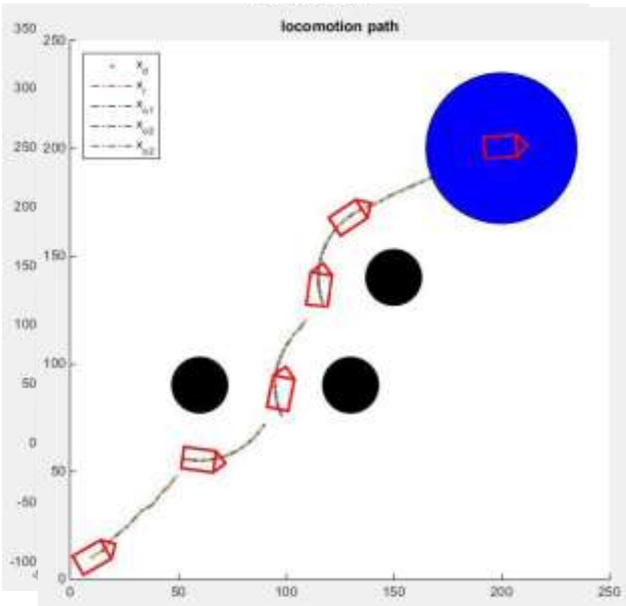

Fig 10: The rout of the robot in the presence of three obstacles.

In the next state, we change the obstacles and investigate the behavior of the robot. Fig. 11 shows this change.

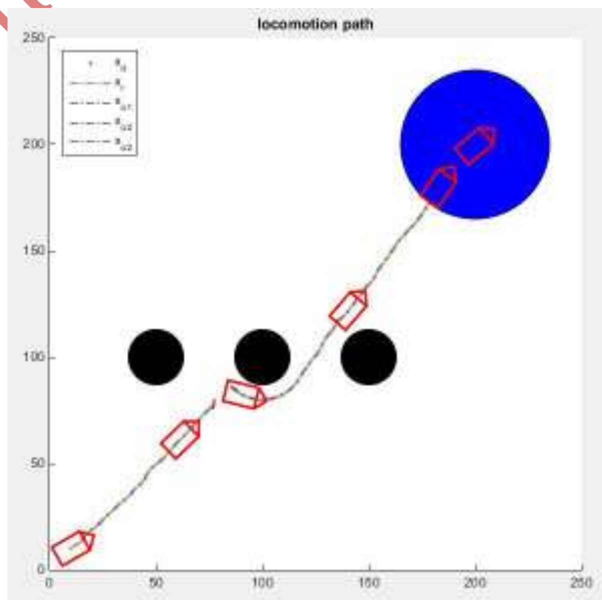

Fig 10:The rout of the robot in the presence of three obstacles which are barrier in front of the robot.

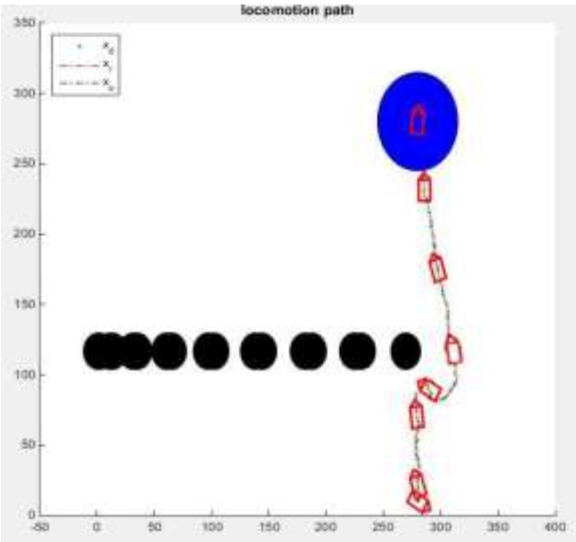

Fig 11: The rout of moving for the robot in the presence of a mobile obstacle

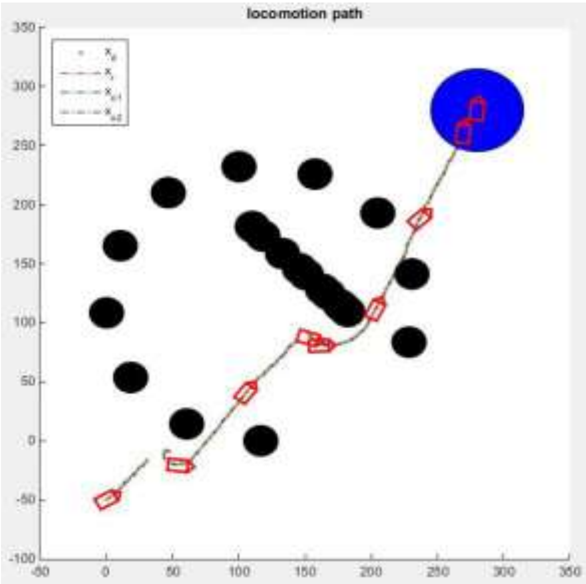

Fig 12: The rout of movement for robot in the presence of two mobile obstacles.

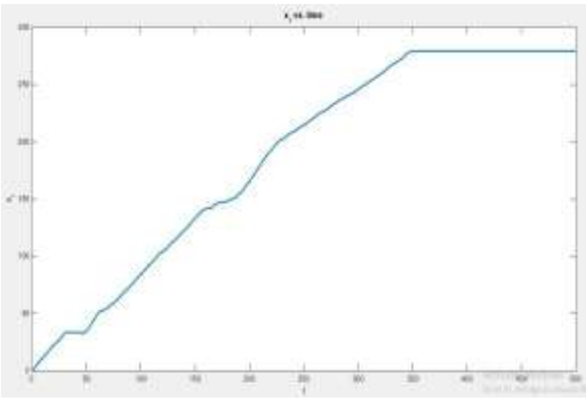

fig 13: the waveform of output signals for the robot related to the Xr

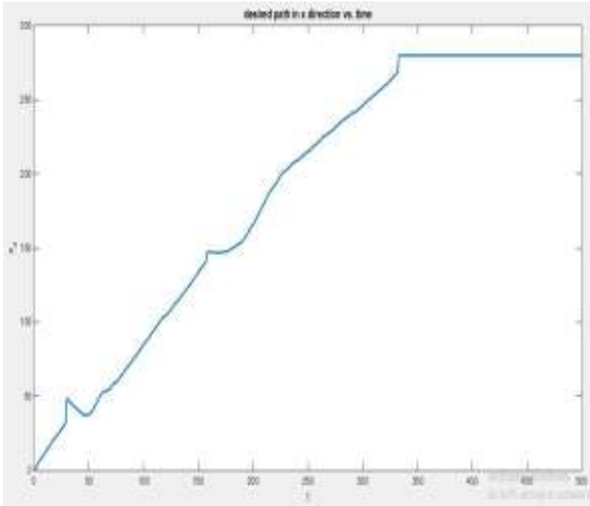

Fig 14: the waveform of output signals for the robot related to the Xd

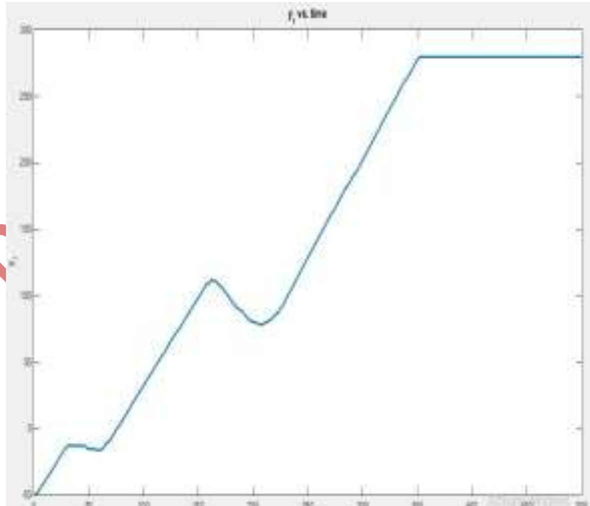

Fig 15: the waveform of output signals for the robot related to the yr

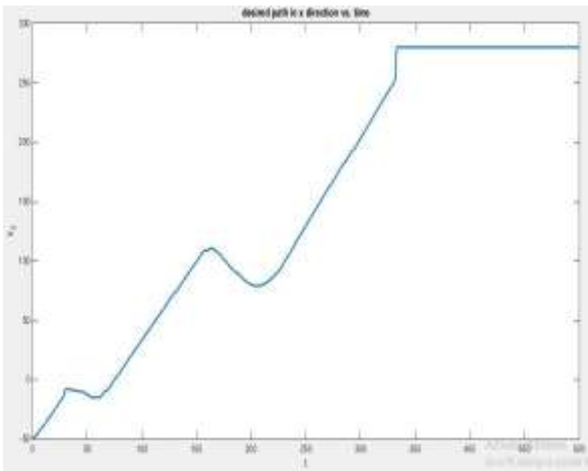

Fig 16: the waveform of output signals for the robot related to the yd

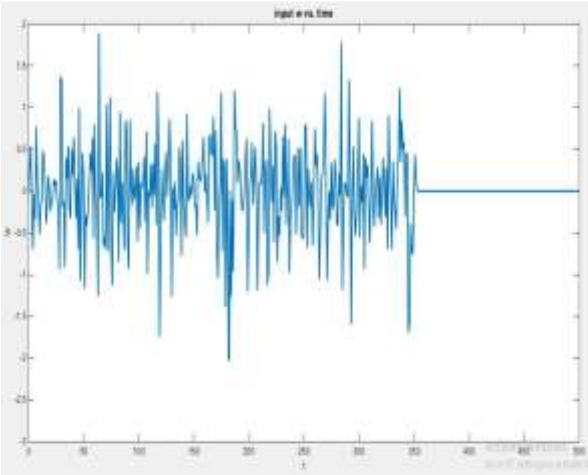

Fig 17: the waveform of output signals for the robot related to the w

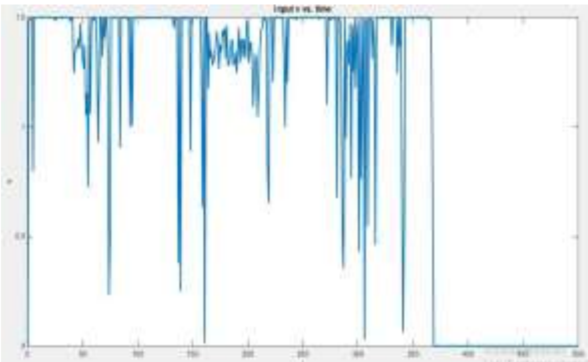

Fig 18: the waveform of output signals for the robot related to the v

8. COMPARISON

In this part we want to compare the results of this paper by the article [44]. The main idea of the article and its difference with this paper is that orbiting rout of the obstacle is predetermined. It means that once the robot senses the obstacle, it turns clockwise or counter clockwise. However, in the article the answer should be consider as an optimized solution and we have conditions in this paper that are necessary:

$$|x - x_0| > x_{safe}$$
$$|y - y_0| > y_{safe}$$
(12)

Where y_{safe} and x_{safe} are the range of secureness for x and y. for having the same working space, we changed its dynamic. The article considered one-dimension of the robot's movement and the dynamic was tri variate. Also, the optimization of the cost function was considered without lagrange factors.

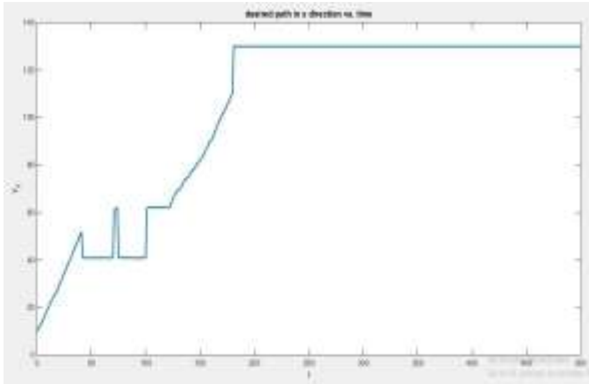

Fig 19: The diagram of yd related to the robot's behavior in the presence of two stationary obstacles for the article.

As said before, the article has more consideration on the rout that causes that the robot's behavior would be precise, nevertheless, the selected line in both methods are the same in this scenario.

Another disadvantage is that in the case of other mobile obstacle, above method has no response and never has solution because of the fact that the obstacle moves.

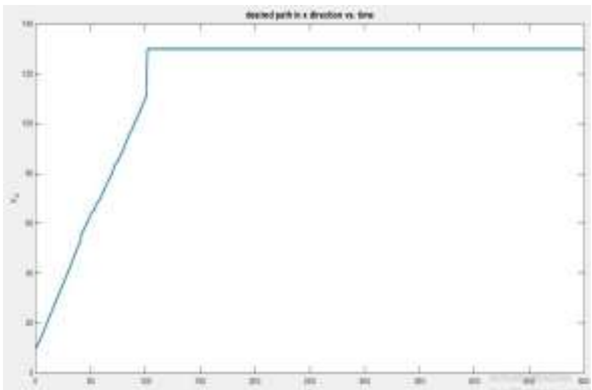

Fig 20: The diagram of yd related to the robot's behavior in the presence of two stationary obstacles for this paper.

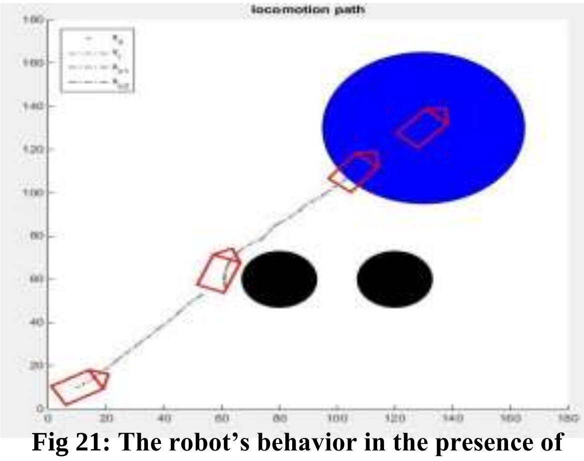

Fig 21: The robot's behavior in the presence of two stationary obstacles in the article which is more precisely because of solving the problem of optimization.

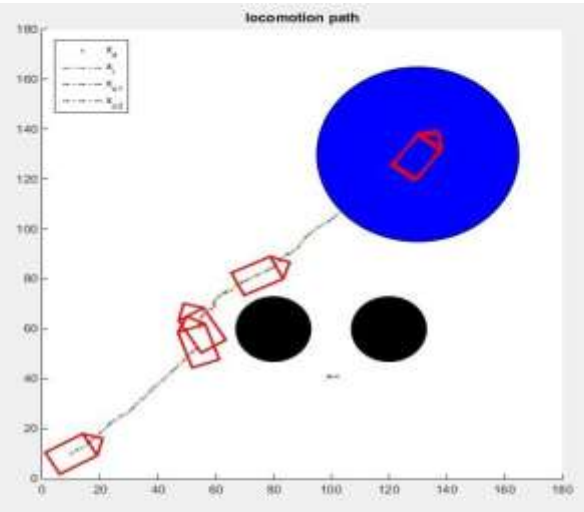

Fig 21: The robot's behavior in the presence of two stationary obstacles in the article which is weaker than the previous solution (the article)

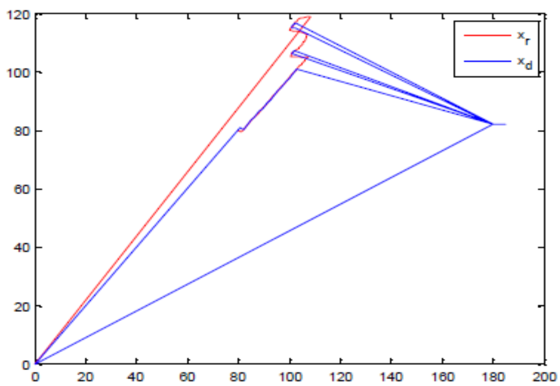

Fig 23: Related to the robot's behavior in the presence of mobile obstacle in the article that leads to no results.

In the end, it should be said that in both methods in the presence of mobile obstacle, the robot has no collision with the obstacle that this is common between them.

8.CONCLUSION

In this paper the movement of the robots and control has been investigated. The problem is that if the robot is placed in an unknown location, it should be reaches to the predetermined goal. In the rout there are obstacles that the robot should not have collision with it. The proposed method for this purpose is to use prediction technique. In fact, the problem is divided into two parts. In the first part, the designing of the rout is done with prediction. Thereby, the robot determines a relative rout that movement in it causes that the robot can reached to the destination without any collision. In the second part, using common technique of prediction control, inputs are design in a way that the robot can move on the appropriate line.

REFERENCE

[1]. Franzè, G., & Lucia, W. (2015), "An obstacle avoidance model predictive control scheme for mobile robots subject to nonholonomic constraints: A sum-of-squares approach." Journal of the Franklin Institute (Vol. 352, pp. 2358-2380)

[2]. Ramírez, D. R., Limón, D., Gomez-Ortega, J., & Camacho, E. F. (1999), "Nonlinear MBPC for mobile robot navigation using genetic algorithms. InRobotics and Automation, 1999." Proceedings. 1999 IEEE International Conference on (Vol. 3, pp. 2452-2457). IEEE.

[3]. Vougioukas, S. G. (2007, April), "Reactive trajectory tracking for mobile robots based on non linear model predictive control." In Robotics and Automation, 2007 IEEE International Conference on (pp. 3074-3079). IEEE.

[4]. Ramirez, D.R (2001), "Genetic algorithms based predictive control for mobile robot navigation in changing environments," Control Conference (ECC), 2001 European(pp. 3179 – 3184) . IEEE.

[5]. Yoon, Y., Shin, J., Kim, H. J., Park, Y., & Sastry, S. (2009). "Model-predictive active steering and obstacle avoidance for

autonomous ground vehicles." Control Engineering Practice (**Vol**. 17, pp. 741-750).

[6]. Kunwar, F., Sheridan, P. K., & Benhabib, B. (2010). "Predictive guidance-based navigation for mobile robots: A novel strategy for target interception on realistic terrains." Journal of Intelligent & Robotic Systems (**vol.** 59 , pp. 367-398).

[7]. Simmons, R. (1996, April), "The curvature-velocity method for local obstacle avoidance. In Robotics and Automation," 1996. Proceedings., 1996 IEEE International Conference on (Vol. 4, pp. 3375-3382). IEEE.

[8]. Fox, D., Burgard, W., & Thrun, S. (1997), "The dynamic window approach to collision avoidance." IEEE Robotics & Automation Magazine (Vol. 4, pp. 23-33).

[9]. Feiten, W., Bauer, R., & Lawitzky, G. (1994, May), "Robust obstacle avoidance in unknown and cramped environments." In Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on (pp. 2412-2417). IEEE.

[10]. Hu, H., & Brady, M. (1994), "A Bayesian approach to real-time obstacle avoidance for a mobile robot." Autonomous Robots (Vol. 1, pp. 69-92).

[11]. Fox, D., Burgard, W., Thrun, S., & Cremers, A. B. (1998, May), "A hybrid collision avoidance method for mobile robots." In Robotics and Automation," 1998. Proceedings. 1998 IEEE International Conference on (Vol. 2, pp. 1238-1243). IEEE.

[12]. Deng, M., Inoue, A., & Sekiguchi, K. (2008), "Lyapunov function-based obstacle avoidance scheme for a two-wheeled mobile robot." Journal of Control Theory and Applications (Vol. 6, pp. 399-404).

[13]. Gao, Y., Lin, T., Borrelli, F., Tseng, E., & Hrovat, D. (2010, January), "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads." In ASME 2010 Dynamic Systems and Control Conference (pp. 265-272). American Society of Mechanical Engineers

[14]. Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006), "Robot modeling and control." (Vol. 3). New York: Wiley.

[15]. de Wit, C. C., Siciliano, B., & Bastineds, G. (1996), "Theory of Robot Control." Springer-Verlag. Belin, Germany.

[16]. Borenstein, J., Feng, L., & Everett, H. R. (1996), "Navigating mobile robots: systems and techniques." AK Peters, Ltd.

[17]. Sciavicco, L., & Siciliano, B. (2012), "Modelling and control of robot manipulators." Springer Science & Business Media.

[18]. Craig, J. J. (2005). "Introduction to robotics: mechanics and control." (Vol. 3). Upper Saddle River: Pearson Prentice Hall.