



Dynamic Obstacles Avoidance Using Nonlinear Model Predictive Control

Mukhtar Sani, Bogdan Robu, Ahmad Hably

► To cite this version:

Mukhtar Sani, Bogdan Robu, Ahmad Hably. Dynamic Obstacles Avoidance Using Nonlinear Model Predictive Control. IECON 2021 - The 47th Annual Conference of the IEEE Industrial Electronics Society, Oct 2021, Toronto (virtual), Canada. 10.1109/IECON48115.2021.9589658 . hal-03410202

HAL Id: hal-03410202

<https://hal.archives-ouvertes.fr/hal-03410202>

Submitted on 31 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dynamic Obstacles Avoidance Using Nonlinear Model Predictive Control

Mukhtar Sani^{1,2}, Bogdan Robu¹, Ahmad Hably¹

¹ Univ. Grenoble Alpes, CNRS, Grenoble INP*, GIPSA-Lab, 38000 Grenoble, France

*Institute of Engineering Univ. Grenoble Alpes

{mukhtar.sani,bogdan.robust,ahmad.hably}@gipsa-lab.grenoble-inp.fr

² Kano University of Science and Technology, Wudil, Nigeria

mukhtarsani@kustwudil.edu.ng

Abstract—In this paper, a Nonlinear Model Predictive Control (NMPC) has been employed to solve point-stabilization problems with static and dynamic obstacles avoidance. The algorithm was implemented on a mobile robot with two differential drive wheels. In NMPC, a cost function is formulated to minimize an error between the reference and the current state of the system subject to constraints. The major drawback of NMPC is the computation time, which results from predicting the system's state over a horizon. However, in this work, the resulting optimal control problem is converted to a discrete nonlinear programming problem using a recently developed toolkit. Dynamic obstacles avoidance is incorporated as a time-varying constraint and can be affected by a short prediction horizon. On the other hand, a long prediction horizon affects the computation time. For this, a terminal state penalty is added to the cost function to guarantee the stability of the control using a relatively shorter prediction horizon. The performance of the proposed controller achieving both static and dynamic obstacles avoidance is verified using several simulation scenarios.

Index Terms—NMPC, Obstacles Avoidance, Mobile Robots

I. INTRODUCTION

Due to the rapid technological development in the twenty-first century, a wide range of applications for mobile robots in various sectors have attracted great interest from researchers. Unmanned vehicles comprising of aerial, ground, and underwater vehicles have been utilized for many applications such as mine clearance [31], patrolling mission [32], surveillance [2], search and rescue mission [6] and educational research purposes [5]. However, several unmanned vehicles in use today are not fully autonomous because of the presence of human-in-the-loop. The human's natural intelligence and skills are being utilized to pilot the robot's navigation. The problem of human-in-the-loop systems is the reliability of the communication system between the robots and the human in the base station. Problems such as delay, bandwidth limitation, jamming, and loss of signals are critical to the reliability of communication systems. In order to remove the human from the loop, the systems should be converted to partial or fully autonomous so as to solve or minimize the effects of the communication issues and also to automatically avoid obstacles and hazardous conditions. A suitable automatic control algorithm would then be employed to pilot the robot.

Several approaches have been used in the literature to pilot mobile robots autonomously. They include dynamic feedback linearization [30], Lyapunov control [17], smooth time-varying control [25] and piece-wise continuous feedback control [10]. However, these approaches do not incorporate constraints on the mobile robots states which are pertinent especially for obstacles avoidance.

Therefore, the natural candidate for the control of autonomous systems is the Model Predictive Control due to its flexibility, its ability to handle both soft and hard constraints as well as its ability to compute optimal control inputs for nonlinear systems. Several works such as [15], [19], [38] have dealt with stabilization problems for nonholonomic mobile robots. Others such as [12], [16], [21], [33] solve trajectory tracking and path following problems using MPC. Many others such as [27], [28], [39] consider solving point stabilization and tracking problems simultaneously without incorporating obstacles avoidance.

On the other hand, safety has become the eternal theme of autonomous vehicles [35]. Active collision avoidance system has become a research hot-spot in the field of automotive due to its ability to effectively improve traffic safety [20]. Static obstacles avoidance have been dealt with in [1], [22] for tracking problems, in [36] for pursuit-evasion games, in [14] for point-stabilization and in [35], [40] for path following problems. Dynamic collision avoidance among multiple mobile robots has been considered in [29]. The literature [11] on dynamic obstacle avoidance has dealt with for an Unmanned Aerial Vehicles (UAV) while [20] has presented dynamic trajectory planning and tracking with dynamic obstacles for an Unmanned Ground Vehicle (UGV). These works incorporated the obstacles avoidance in the cost function and have to make additional computations to predict the trends of the obstacles. This prediction is pertinent because the LIDAR sensor, which is used for detecting the obstacle can only provide the current information of the obstacles. It is therefore interesting to work on another obstacle avoidance method that would not require predicting the movement of the obstacles.

In this paper, we design an NMPC based dynamic obstacles avoidance algorithm that only needs the instantaneous

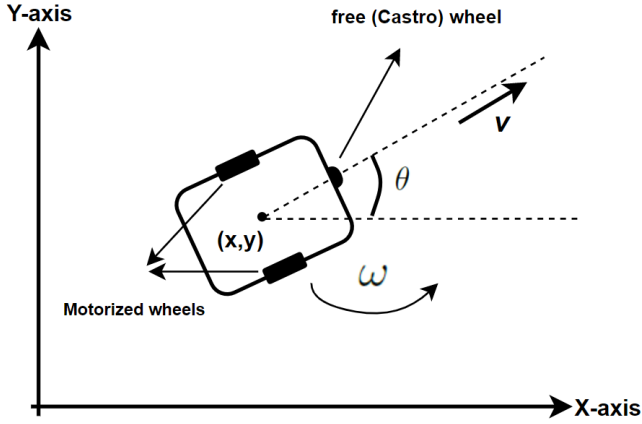


Fig. 1. Schematic diagram of differential drive robot

position of the obstacles for a point-stabilization problem. Different from the work in [20], our algorithm does not require additional computation for predicting the speed of the obstacles which could lead to higher computation time and wrong prediction in the case of intelligent obstacles. This method incorporates obstacle avoidance as a constraint to be considered while solving the optimal control problem. The second contribution of this paper is that, the proposed approach could be used to simultaneously handle both static and dynamic obstacles avoidance. Finally, the third contribution of this paper is that we have compared the performance of two discretization methods, the 4th order Runge-Kutta and the Euler method.

The remaining part of the paper is organized as follows: in Section II, we presented the theoretical background on the kinematic model of the mobile robot, the discretization method, and the obstacles detection and avoidance techniques. Section III presents the controller design and implementation. The simulation results are presented in Section IV while in Section V we draw conclusions and future perspectives.

II. PRELIMINARIES

This paper primarily addresses the control problem and dynamic obstacles avoidance of the differential drive robot class of nonholonomic mobile robots. In this section, the kinematic model, discretization methods, and obstacles avoidance techniques are presented.

A. Kinematic Model of nonholonomic mobile robots

A kinematic model of a mobile robot governs how wheel speeds map to robot velocities, while a dynamic model governs how the torques map to robot acceleration. In this paper, we will focus on the kinematic model and ignore the dynamic model. The kinematic model of the nonholonomic wheeled

mobile robot of unicycle type in Figure 1 can be represented in (1). The details can be obtained from [18], [24].

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (1)$$

where the state variable $\mathbf{x} = [x, y, \theta]^T$ denotes the position of the robot in chassis frame of reference and the heading angle. The control variable $\mathbf{u} = [v, \omega]^T$ denotes the linear speed and the angular speed.

The kinematic models are said to be nonholonomic because with slight manipulation we can obtain a differential constraint in (2):

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0 \quad (2)$$

Since linear approximations are usually regarded as the first step for the analysis and control design of a nonlinear system. Thus if the linearized system is controllable, then the original nonlinear system is at least locally controllable and feedback stabilizable. However, the linearized model is not controllable, because the rank of the controllability matrix is 2 instead of the number of states (3); See [13] for a detailed explanation. The driftless form of nonholonomic robots (3) was investigated for weaker controllability (sufficient) conditions from nonlinear systems theory. Eventually, it was proved using Brockett's theorem [9] that the system 1 is small-time locally controllable [8], [13]

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} + \omega \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3)$$

B. Runge-Kutta Discretization Method

Several works involving discrete NMPC for controlling nonholonomic robots utilized the Euler method (also called Euler forward method [22], [36]. The resulting discrete model is given in (4). However, the Euler method does not give good approximate solutions of nonlinear ordinary differential equations for a larger sampling time (T_s) because it only uses one slope. It is reasonable to assume that using more estimates of slopes at the mid-point and the end of the interval would result in more accuracy, see [23] for more details.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + T_s f(t_k, \mathbf{x}_k) \quad (4)$$

In this present paper, we employ the classic Runge-Kutta method (*RK4*) which is the most widely known member of the Runge-Kutta family. Let us represent the kinematic model of nonholonomic robots (1) as an initial value problem (5).

$$\begin{cases} \dot{\mathbf{x}} = f(t, \mathbf{x}) \\ \mathbf{x}(t_0) = \mathbf{x}_0 \end{cases} \quad (5)$$

The *RK4* approximation of $\mathbf{x}(t_{k+1})$ is \mathbf{x}_{k+1} which depends on the current value \mathbf{x}_k and some weighted average of four increments as depicted in figure 2. Each increment is a function of the sampling time and an estimated slope specified

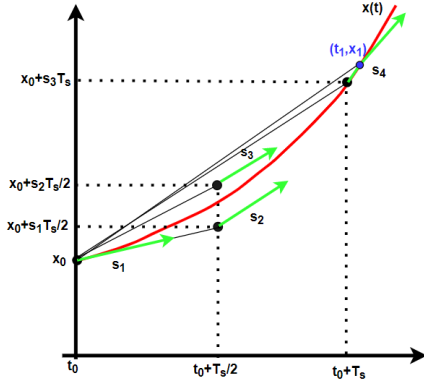


Fig. 2. Runge-Kutta Discretization

by a function of the right-hand side of the differential equation. The mathematical representation of RK4 is given as:

$$\mathbf{x}_{k+1} = \frac{1}{6}T_s(s_1 + 2s_2 + 2s_3 + s_4) \quad (6)$$

Such that:

$$\begin{cases} s_1 = f(t_k, \mathbf{x}_k) \\ s_2 = f(t_k + \frac{T_s}{2}, \mathbf{x}_k + T_s \frac{s_1}{2}) \\ s_3 = f(t_k + \frac{T_s}{2}, \mathbf{x}_k + T_s \frac{s_2}{2}) \\ s_4 = f(t_k + T_s, \mathbf{x}_k + T_s s_3) \end{cases} \quad (7)$$

The first slope s_1 is at the beginning of the interval (this is Euler's method). The slopes s_2 and s_3 are both at the mid-point of the interval while the s_4 is at the end of the interval.

C. Obstacle's Detection and Avoidance

To incorporate obstacle avoidance in our NMPC formulation, the first problem is the detection method. In mobile robotics, LIDAR (acronyms for light detection and ranging) sensor is usually employed to detect the position and the size of an obstacle. Having detected the positions (x_{obs}, y_{obs}) and the diameter of the obstacle d_{obs} , the next is writing a suitable program so that the robot would avoid the obstacle. Several methods of avoiding obstacles were summarized in [7]. The most suitable for NMPC is the bug-type algorithm. In Bug algorithms, the robot moves on the shortest path from its initial position towards the goal position until when it encounters an obstacle. The algorithm forces the robot to move tangentially around the surface of the obstacle and then return to its original path after completely dodging the obstacle. Therefore the constraint in (8) is added during the formulation of the NMPC for obstacles avoidance.

$$\sqrt{(x_{rob} - x_{obs})^2 + (y_{rob} - y_{obs})^2} \geq (r_{obs} + r_{rob}) \quad (8)$$

where x_{rob}, y_{rob} and r_{rob} are the positions and the radius of the robot respectively.

III. CONTROL DESIGN AND IMPLEMENTATION

In this section, we present the mathematical formulation of NMPC, the solution and coding approach and the simulation environment for implementing the controller.

A. Nonlinear Model Predictive Control Design

Model Predictive Control (MPC) is a feedback implementation of optimal control using finite prediction horizon and online optimization. MPC is also known as Receding Horizon Control (RHC) where a future control sequence minimizing an objective function is minimized over a finite horizon. The advantages of MPC comprise its ability to: control multi-variable coupled dynamical systems, handle constraints on the states, handle constraints on control inputs, handle nonlinearities in the systems model conceptually. In addition, MPC have systematic design approach and has a well understood tuning parameters, i.e prediction horizon length and weighting matrices [3], [34], [37].

Since the model of the system to control is central to the design of MPC, we discretize the nonlinear model of the nonholonomic robot using the RK4 method and formulate discrete NMPC by solving the following Optimal Control Problem (OCP):

$$\min_{\mathbf{u} \in \mathbb{R}^n \times N} J_N(\mathbf{x}_0, \mathbf{u}) \quad (9)$$

Subject to

$$\begin{cases} \mathbf{x}(0) = \mathbf{x}_0, \\ \mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)); \quad k \in \{0, 1, \dots, N-1\}, \\ \|\mathbf{x}(k) - \mathbf{x}_{obs}(k)\| \geq r_{rob} + r_{obs} \\ \mathbf{x}_{min} \leq \mathbf{x}(k) \leq \mathbf{x}_{max} \quad k \in \{1, 2, \dots, N\}, \\ \mathbf{u}_{min} \leq \mathbf{u}(k) \leq \mathbf{u}_{max} \quad k \in \{0, 1, \dots, N-1\} \end{cases} \quad (10)$$

where:

$$J_N(\mathbf{x}_0, \mathbf{u}) = \sum_{k=0}^{N-1} V(\mathbf{x}(k), \mathbf{u}(k)) + W(\mathbf{x}(N)) \quad (11)$$

The term $V(\mathbf{x}(k), \mathbf{u}(k))$ is called the running cost which can be computed by penalizing the deviation of the system's state $\mathbf{x}(k)$ and control input $\mathbf{u}(k)$ from the reference state $\mathbf{x}^r(k)$ and reference control input $\mathbf{u}^r(k)$ respectively. Generally, the running cost are defined as:

$$V(\mathbf{x}(k), \mathbf{u}(k)) = \|\mathbf{x}(k) - \mathbf{x}^r(k)\|_Q^2 + \|\mathbf{u}(k) - \mathbf{u}^r(k)\|_R^2 \quad (12)$$

where $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ are positive definite symmetric weighting matrices. N is the prediction horizon assuming that the length of the prediction and control horizon is the same. As previously explained, in point stabilization problem, the state reference $\mathbf{x}^r(k)$ is a fixed value, thus the control, reference $\mathbf{u}^r(k) = 0$. In the case of trajectory tracking problem, the state reference $\mathbf{x}^r(k)$ is time varying, therefore the deviation of control input from the reference can be penalized due to computational advantages such as rendering the optimal control problem easier, avoiding control values with expensive energy [26].

The term $W(\mathbf{x}(N))$ is referred to as terminal cost which is used for stability purpose. It can be computed by penalizing the last entry from the state prediction $\mathbf{x}(N)$ from its reference $\mathbf{x}^r(N)$. Terminal cost can be defined as:

$$W(\mathbf{x}(N)) = \|\mathbf{x}(N) - \mathbf{x}^r(N)\|_P^2 \quad (13)$$

where $P \in \mathbb{R}^{n \times n}$ is a positive definite weighting matrix. The solution of the optimal control problem (9) is the optimal control sequence of the form:

$$\mathbf{u}^* = (\mathbf{u}^*(0), \mathbf{u}^*(1), \dots, \mathbf{u}^*(N-1)) \quad (14)$$

The first part ($\mathbf{u}^* = \mathbf{u}^*(0)$) is applied to the robot, while the rest are discarded.

The weight matrices are tuned and the best values that stabilize the controller are chosen. The Q and R matrices are found to be diagonal matrices with diagonal elements defined as (1, 1, 0.001) and (1, 1) respectively. The weight on the terminal penalty cost is found to be $10000 * Q$.

B. Simulation Environment

In this work, the NMPC algorithm was coded using an open-source symbolic framework for automatic differentiation and optimal control software, CasADi [4]. The optimal control problem was converted to a nonlinear programming problem using a multiple shooting approach (where both the states and the control variables are considered as optimization parameters). The CasADi toolkit was interfaced with an Interior Point OPTimizer (IPOPT), also an open-access software, to provide the solution. Simulink environment in Matlab was used to set up the control system. The output of the controller is sent to the system for an update whereas the state measurement, the reference position, and the obstacles parameters are feed to the controller at each sampling instant for re-computation of the new control strategy.

IV. RESULTS AND DISCUSSION

This section presents the simulation results of our control objective under static and dynamic obstacles.

A. Static obstacles Avoidance

To achieve point stabilization with static obstacles avoidance using discrete NMPC, the robot starts from an initial pose $\mathbf{x}_0 = [-1m, -1m, -\frac{\pi}{4}rad]$ and aimed to reach a goal pose $\mathbf{x}_g = [1m, 1m, \frac{\pi}{4}rad]$ encountering two static obstacles at $(0m, 0m)$ and $(0.8m, 0.6m)$. The diameter of each obstacle is $0.3m$ while the diameter of the robot is $0.04m$. The sampling time is chosen to be $0.1sec$ and a prediction horizon $N = 20$ was selected leading to prediction horizon time $T = 2secs$. The robot's actuator saturation limits are selected randomly such that the linear velocity v ranges from 0 to $0.04m/s$ and the angular velocity ω ranges from $-\frac{\pi}{8}rad/s$ to $\frac{\pi}{8}rad/s$.

The result using the Euler discretization method is presented in Figure 3. Although the controller was able to move from its initial pose to the goal position, it should be noted that the controller's output was not smooth with the Euler method, which is not suitable for physical implementation on robots. We then perform the same experiment with the RK4 discretization method. The result is presented in Figure 4(a) and shows a smooth trajectory of the control profiles much better than the Euler method.

To examine the effects of the length of prediction horizon on the computation time, we present the result of the same

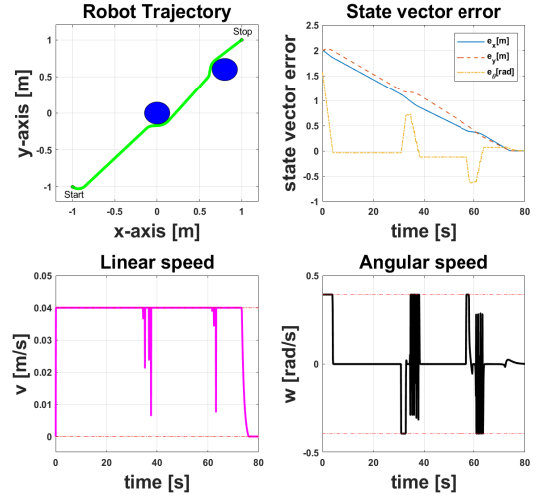


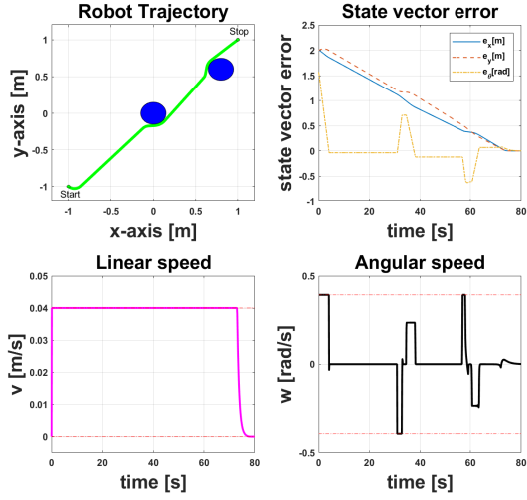
Fig. 3. Euler method with $N=20$

problem in Figure 4(b) and 4(c) which uses a shorter prediction horizon. It can be seen that a longer prediction horizon gives a smoother control profile and consequently the higher the computation time, as shown in Figure 5. It is pertinent to note in 4(c) that despite using a prediction horizon of $N = 5$, the RK4 method still gives better results than Euler with a prediction horizon of $N = 20$.

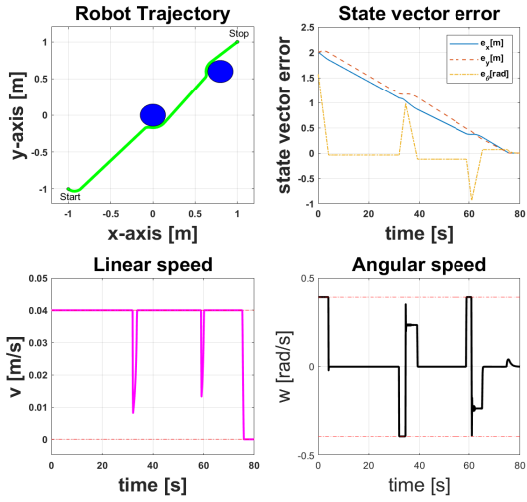
B. Dynamic Obstacle Avoidance

In the case of dynamic obstacle avoidance, a larger simulation environment is required so that the robot must encounter the two obstacles moving at a different speed. The robot's actuator saturation limits are selected randomly such that the linear velocity v ranges from 0 to $0.4m/s$ and the angular velocity ω ranges from $-\frac{\pi}{4}rad/s$ to $\frac{\pi}{4}rad/s$. The diameter of the robot and the obstacles are the same as in the case of static obstacles. The initial position and the speed of the two obstacles are respectively given as $(-0.3m, 2m)$ and $0.05m/s$ for the first and $(-2m, 0m)$ and $0.12m/s$ for the second obstacle, both moving from left to right to block the path of the robot at different positions. The result of dynamic obstacles avoidance is presented in Figure 6. The robot navigated from its initial pose $[-3m, -2m, -\frac{\pi}{4}]$ and aimed to reach its goal pose $[1, 3, \frac{\pi}{4}]$ while encountering two moving obstacles along the way.

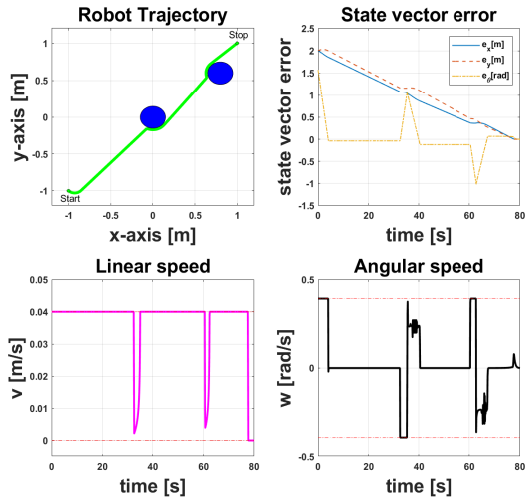
The trajectory of the robot and the obstacles at four-time instances is depicted in Figure 6(a). At $t = 0$ the initial positions of the robot and the obstacles are marked. We can see that it then encounters a moving obstacle (shaded circle) at $t = 7secs$. The robot avoided collision with the obstacle by trying to get around it then moved towards its goal. It then encounters another moving obstacle (the shaded circle) at $t = 15secs$, avoided it and to reach its final destination around $t = 23secs$.



(a) RK4 method with $N = 20$



(b) RK4 with $N = 10$



(c) RK4 method with $N = 5$

Fig. 4. Static Obstacles Avoidance

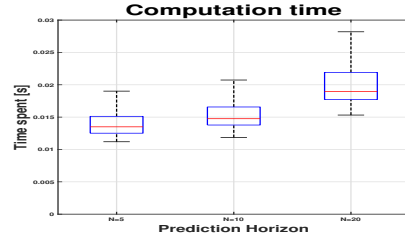
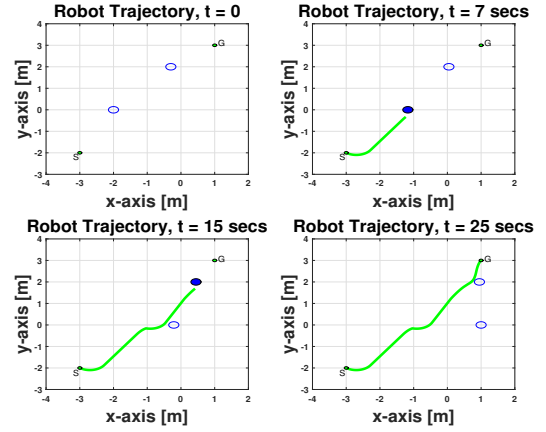
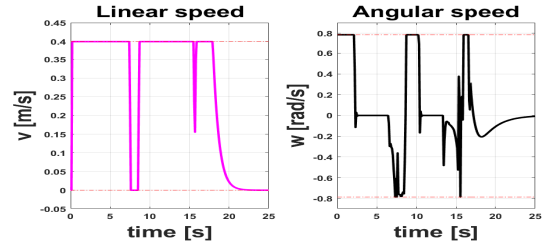


Fig. 5. Computation Time for Static Obstacle Avoidance



(a) States Trajectories



(b) Control Trajectories

Fig. 6. Dynamic Obstacle Avoidance

V. CONCLUSION

This paper dealt with static and dynamic obstacles avoidance for point stabilization problems using discrete NMPC. Two discretization methods are compared the Euler and RK4 method. The latter achieved better performance than the former. The obstacle avoidance was integrated as an inequality constraint during the formulation of the NMPC. Stability was achieved using terminal cost inclusion. The effect of a long prediction horizon on the controller computation time was analyzed, showing that a higher prediction horizon leads to higher computation cost. The results obtained showed that our approach can be used to handle both static and dynamic obstacles without the need for additional computation. As for future work, we will try to adapt our controller to a complex tracking and path following problems and then deploy it on physical robots.

REFERENCES

- [1] Muhammad Awais Abbas, Ruth Milman, and J. Mikael Eklund. Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles. *Canadian Journal of Electrical and Computer Engineering*, 40(1):12–22, 2017.
- [2] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero. Cooperative perimeter surveillance with a team of mobile robots under communication constraints. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5067–5072, 2013.
- [3] Mazen Alamir. *Stabilization of Nonlinear Systems Using Receding-horizon Control Schemes: A Parametrized Approach for Fast Systems*. Springer, 2006.
- [4] Joel A. E. Andersson, Joris Gillis, Greg Horn, James B. Rawlings, and Moritz Diehl. Casadi: A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11, 2019.
- [5] T. Arai, E. Pagello, and L.E. Parker. Guest editorial advances in multirobot systems. *IEEE Transactions on Robotics and Automation*, 18(5):655–661, 2002.
- [6] Markus Bernard, Konstantin Kondak, Ivan Maza, and Anibal Ollero. Autonomous transportation and deployment with aerial robots for search and rescue missions. *Journal of Field Robotics*, 28(6):914–931, 2011.
- [7] Vayeda Anshav Bhavesh. Comparison of various obstacle avoidance algorithms. *International Journal of Engineering Research and Technology (IJERT)*, 2015.
- [8] A. M. Bloch. *Nonholonomic Mechanics*, pages 207–276. Springer New York, New York, NY, 2003.
- [9] R. W. Brockett. Nonlinear control theory and differential geometry,. In *International Congress of Mathematics*, pages 1357–1368, 1983.
- [10] C. Canudas de Wit and O.J. Sordalen. Exponential stabilization of mobile robots with nonholonomic constraints. In *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*, pages 692–697 vol.1, 1991.
- [11] Manuel Castillo-Lopez, Seyed Amin Sajadi-Alamdari, Jose Luis Sanchez-Lopez, Miguel A. Olivares-Mendez, and Holger Voos. Model predictive control for aerial collision avoidance in dynamic environments. In *2018 26th Mediterranean Conference on Control and Automation (MED)*, pages 1–6, 2018.
- [12] T. Faulwasser, B. Kern, and R. Findeisen. Model predictive path-following for constrained nonlinear systems. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 8642–8647, 2009.
- [13] Xie Feng. *Model Predictive Control of Non-holonomic Mobile Robots*. PhD thesis, Oklahoma State university, USA, 2004.
- [14] Gowtham Garimella, Matthew Sheckells, Joseph L. Moore, and M. Kobilarov. Robust obstacle avoidance using tube nmppc. In *Robotics: Science and Systems*, 2018.
- [15] Dongbing Gu and Huosheng Hu. A stabilizing receding horizon regulator for nonholonomic mobile robots. *IEEE Transactions on Robotics*, 21(5):1022–1028, 2005.
- [16] Dongbing Gu and Huosheng Hu. Receding horizon tracking control of wheeled mobile robots. *IEEE Transactions on Control Systems Technology*, 14(4):743–749, 2006.
- [17] G. Indiveri. Kinematic time-invariant control of a 2d nonholonomic vehicle. In *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, volume 3, pages 2112–2117 vol.3, 1999.
- [18] L. Jualin. *Mobile Robotics*. Dover Books on Mathematics. Elsevier Publications, 2015.
- [19] F. Kuhne, W.F. Lages, and J.M.G. da Silva. Point stabilization of mobile robots with nonlinear model predictive control. In *IEEE International Conference Mechatronics and Automation*, 2005, volume 3, pages 1163–1168 Vol. 3, 2005.
- [20] Shaosong Li, Zheng Li, Zhixin Yu, Bangcheng Zhang, and Niaoona Zhang. Dynamic trajectory planning and tracking for autonomous vehicle with obstacle avoidance based on model predictive control. *IEEE Access*, 7:132074–132086, 2019.
- [21] Heonyoung Lim, Yeonsik Kang, Changwhan Kim, Jongwon Kim, and Bum-Jae You. Nonlinear model predictive controller design with obstacle avoidance for a mobile robot. In *2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, pages 494–499, 2008.
- [22] Heonyoung Lim, Yeonsik Kang, Changwhan Kim, Jongwon Kim, and Bum-Jae You. Nonlinear model predictive controller design with obstacle avoidance for a mobile robot. In *2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, pages 494–499, 2008.
- [23] Ruitao Lin, Chan Wang, Fang Liu, and Yueliang Xu. A new numerical method of nonlinear equations by four order runge-kutta method. In *2010 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 1295–1299, 2010.
- [24] K.M. Lynch and F.C. Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.
- [25] R.T. M'Closkey and R.M. Murray. Exponential stabilization of driftless nonlinear control systems using homogeneous feedback. *IEEE Transactions on Automatic Control*, 42(5):614–628, 1997.
- [26] M. W. Mehrez. *Optimization Based Solutions for Control and State Estimation in Non-holonomic Mobile Robots: Stability, Distributed Control, and Relative Localization*. PhD thesis, Memorial University of Newfoundland, Canada, 2017.
- [27] M. W. Mehrez, G. K. I. Mann, and R. G. Gosine. Stabilizing nmppc of wheeled mobile robots using open-source real-time software. In *2013 16th International Conference on Advanced Robotics (ICAR)*, pages 1–6, 2013.
- [28] Mohamed W. Mehrez, George K. I. Mann, and Raymond G. Gosine. Comparison of stabilizing nmppc designs for wheeled mobile robots: An experimental study. In *2015 Moratuwa Engineering Research Conference (MERCon)*, pages 130–135, 2015.
- [29] Ryuhei Mikumo and Hiroyuki Ichihara. Dynamic collision avoidance among multiple mobile robots: A model predictive control approach. In *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 1136–1137, 2017.
- [30] G. Oriolo, A. De Luca, and M. Vendittelli. Wmr control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, 10(6):835–852, 2002.
- [31] David Portugal, Lino Marques, and Manuel Armada. Deploying field robots for humanitarian demining: Challenges, requirements and research trends. pages 649–656, 09 2014.
- [32] David Portugal and Rui Rocha. A survey on multi-robot patrolling algorithms. In Luis M. Camarinha-Matos, editor, *Technological Innovation for Sustainability*, pages 139–146, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [33] Guilherme V. Raffo, Guilherme K. Gomes, Julio E. Normey-Rico, Christian R. Kelber, and Leandro B. Becker. A predictive controller for autonomous vehicle path tracking. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):92–102, 2009.
- [34] J. B. Rawlings, D. Q. Mayne, and M.M. Diehl. *Model Predictive Control: Theory, Computation and Design*. 2nd Edition. Nob Hill Publishing, LLC, 2019.
- [35] Ugo Rosolia, Stijn De Bruyne, and Andrew G. Alleyne. Autonomous vehicle control: A nonconvex approach for obstacle avoidance. *IEEE Transactions on Control Systems Technology*, 25(2):469–484, 2017.
- [36] M. Sani, B. Robu, and A. Hably. Pursuit-evasion game for nonholonomic mobile robots with obstacle avoidance using nmppc. In *2020 28th Mediterranean Conference on Control and Automation (MED)*, pages 978–983, 2020.
- [37] Liuping Wang. *Model Predictive Control Systems Design and Implementation Using MATLAB*. Springer, 2009.
- [38] Karl Worthmann, Mohamed W. Mehrez, Mario Zanon, George K. I. Mann, Raymond G. Gosine, and Moritz Diehl. Model predictive control of nonholonomic mobile robots without stabilizing constraints and costs. *IEEE Transactions on Control Systems Technology*, 24(4):1394–1406, 2016.
- [39] Feng Xie and Rafael Fierro. First-state contractive model predictive control of nonholonomic mobile robots. In *2008 American Control Conference*, pages 3494–3499, 2008.
- [40] Jingjun Zhang, Di Wei, Ruizhen Gao, and Ziqiang Xia. A trajectory tracking and obstacle avoidance approach for nonholonomic mobile robots based on model predictive control. In *2020 IEEE 16th International Conference on Control Automation (ICCA)*, pages 1038–1043, 2020.