Real-time Dynamic Obstacle Avoidance For A Non-holonomic Mobile Robot

Mukhtar Sani, Ahmad Hably, Bogdan Robu, Jonathan Dumon, Nacim Meslem

Real-time Dynamic Obstacle Avoidance For A Non-holonomic Mobile Robot

Mukhtar Sani, Ahmad Hably, Bogdan Robu, Jonathan Dumon, Nacim Meslem
Univ. Grenoble Alpes, CNRS, Grenoble-INP
GIPSA-lab, F-38000, Grenoble, France
ahmad.hably@gipsa-lab.grenoble-inp.fr

***Abstract*—This paper presents the experimental validation of a real-time nonlinear model predictive control algorithm developed to deal with dynamic and static obstacle avoidance for a non-holonomic wheeled mobile robot. Unlike state-of-the-art techniques, the speed of the dynamic obstacle is unknown to the controller. The developed controller autonomously drives the robot away from the obstacle by calculating the minimal distance from which the avoidance maneuver starts. Several real-time experimental results for stabilizing a mobile robot in presence of dynamic and static obstacles are presented.**

I. Introduction

Control of autonomous mobile robots is an active area of research with several industrial, military and civil applications. These systems can be programmed to perform a wide range of tasks in complex environments where humans can be at risk, including exploration, interception of dangerous objects, surveillance, and many others. A robot's or any moving object's autonomy cannot be completed without the ability to avoid obstacles and some adversaries. As a result, it is critical to investigate control techniques that allow mobile robots to carry out complex missions while navigating autonomously. These control techniques should be capable of driving the system away from any obstacles that it may encounter. Several approaches have been used in the literature to pilot mobile robots autonomously. They include dynamic feedback linearization [1], Lyapunov control [2], smooth time-varying control [3] and piece-wise continuous feedback control [4]. However, these approaches do not incorporate constraints on the mobile robots states, which are very important for physical implementation and exploitation as well as for obstacles avoidance.

Sequel to its ability in handling constraints and ability to compute optimal control inputs for nonlinear systems, Nonlinear Model Predictive Control (NMPC) becomes the most appropriate technique to deal with control problems for mobile robots [5], [6]. It has been used to address stabilization problems, path following [7], [8] and both point stabilization and tracking problems without incorporating obstacle avoidance [9].

To ensure safety of autonomous vehicles, active collision avoidance system has become a research hot-spot. Static obstacle avoidance, where the position and the size of the obstacle is required have been dealt with in [10] for tracking problems, in [11] for pursuit-evasion games, in [12] for point-stabilization and in [13], [14] for path following problems. In [15], dynamic collision avoidance among multiple mobile robots has been considered. However, these works have assumed full knowledge of the obstacle speed and size, which are either obtained by measurement and sensors or additional computation. While measurement requires embedding additional sensors, thus increasing the computation time. Following these drawbacks, it is therefore interesting to work on another obstacle avoidance method that does not require measuring or predicting the movement of the obstacles.

In this paper, we propose a dynamic obstacle avoidance model predictive control (MPC) algorithm that needs the instantaneous position of the obstacles for a point-stabilization problem. Different from the work of [16], our MPC algorithm does not require additional computation for predicting the obstacle's speed, which could lead to higher computational time and wrong prediction in the case of "tricky" obstacles. This method, initially developed in our previous [17], incorporates obstacle avoidance as a constraint while solving the optimal control problem. The performance of the discretization methods as well as the effect of prediction horizon on the computational time is also studied. The second contribution of this paper is that, the proposed approach could be used to simultaneously handle both static and dynamic obstacle avoidance. Finally, the third contribution of this paper is that an experimental validation of the proposed control algorithm using physical robots is presented.

The remaining part of the paper is organized as follows. The system's modeling is presented in Section II followed by the dynamic obstacle avoidance MPC formulation in Section III. The experimental results are presented in Section IV. Finally, in Section V, conclusions and future perspectives are presented.

II. System Modeling

The mobile robot studied in this paper is a differential drive robot (Fig.1). Its kinematic model governs how wheel speeds map to robot velocities, while its dynamic model governs how the torques map to robot acceleration. In this paper, we will only focus on the kinematic model. The kinematic model of the nonholonomic wheeled mobile robot of unicycle type can be obtained from [18] and is represented by eq.1:

This work has been partially supported by ROBOTEX 2.0 (Grants ROBOTEX ANR-10-EQPX-44-01 and TIRREX ANR-21-ESRE-0015) funded by the French program Investissements d'avenir.

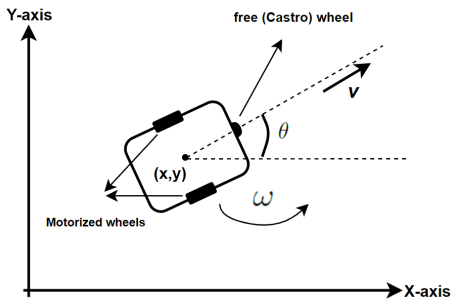


Fig. 1: Schematic diagram of differential drive robot.

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

where the state variable $\mathbf{x} = [x, y, \theta]^T$ denotes the position of the robot in the 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):

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \tag{2}$$

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 less than the number of states [19].

In this present paper, we employ the 4^{th} order Runge-Kutta ($RK4$) discretization. Let us represent the kinematic model of non-holonomic robots (1) as an initial value problem (4).

$$\dot{\mathbf{x}} = f(t, \mathbf{x}) \tag{3}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \tag{4}$$

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

$$\mathbf{x}_{k+1} = \frac{1}{6}T_s(s_1 + 2s_2 + 2s_3 + s_4) \tag{5}$$

such that

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

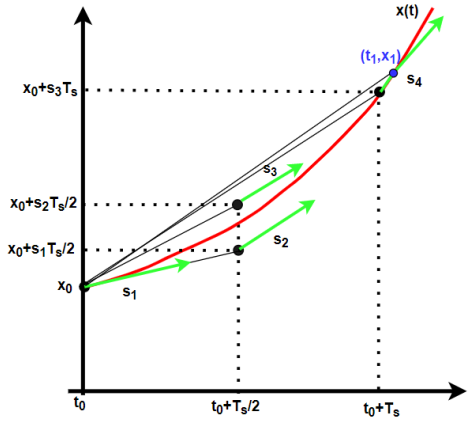


Fig. 2: Runge-Kutta Discretization

The first slope s_1 is at the beginning of the interval. 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. After discretization, the kinematic model of the non-holonomic mobile robot is approximated by the following equation.

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), T_s) \tag{6}$$

where T_s is the sampling time. This kinematic equation is used for the high-level controller, assuming that the low-level controller achieves the desired linear and angular velocities reasonably fast.

III. Dynamic Obstacle Avoidance With Model Predictive Control

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, [20], [21]. Since the model of the system to control is the heart of the MPC design, we discretize the nonlinear model of the non-holonomic robot using the $RK4$ method and formulate an MPC by solving the following optimal control problem (OCP):

$$\min_{\mathbf{u}\in\mathbb{R}^{n_\mathbf{u}\times N}} J_N(\mathbf{x_0}, \mathbf{u}) \tag{7}$$

Subject to

$$\begin{cases} \mathbf{x}(0) = \mathbf{x_0}, \\ \mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)); \quad k \in \{0, 1, ...N-1\}, \\ ||\mathbf{x}(k) - \mathbf{x}_{obs}(k)|| \geq r_{rob} + r_{obs} \\ \mathbf{x}_{min} \leq \mathbf{x}(k) \leq \mathbf{x}_{max} \quad k \in \{1, 2, ...N\}, \\ \mathbf{u}_{min} \leq \mathbf{u}(k) \leq \mathbf{u}_{max} \quad k \in \{0, 1, ...N-1\} \end{cases} \tag{8}$$

where:

$$J_N(\mathbf{x_0}, \mathbf{u}) = \sum_{k=0}^{N-1} V(\mathbf{x}(k), \mathbf{u}(k)) + W(\mathbf{x}(N)) \quad (9)$$

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 can be defined as:

$$V(\mathbf{x}(k), \mathbf{u}(k)) = ||\mathbf{x}(k) - \mathbf{x}^r(k)||^2_Q + ||\mathbf{u}(k) - \mathbf{u}^r(k)||^2_R \quad (10)$$

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. 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 [22]. 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:

$$W(\mathbf{x}(N)) = ||\mathbf{x}(N) - \mathbf{x}^r(N)||^2_P \quad (11)$$

where $P \in \mathbb{R}^{n \times n}$ is a positive definite weighting matrix. The solution of the optimal control problem (7) is the optimal control sequence of the form:

$$\mathbf{u}^* = (\mathbf{u}^*(0), \mathbf{u}^*(1), ..., \mathbf{u}^*(N-1)) \quad (12)$$

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

The schematic diagram of the control architecture is presented in Figure 3. Matlab/Simulink software is used for computing the optimal control. The algorithm is coded using an open-source symbolic framework for automatic differentiation and optimal control software, CasADi, [23]. The optimal control problem is 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 is interfaced with an Interior Point OPTimizer (IPOPT), an open-source software, to provide the solution. The output of the controller is sent to the system for an update, whereas the state measurement, that is the reference position, and the obstacles parameters are fed to the controller at each sampling instant for re-computation of the new control strategy. For real-time validation of the proposed control algorithms with real-time measurements and real transmission of information to the actual system, two non-holonomic mobile robots depicted in Figures 4(a) and 4(b) are fabricated at Gipsa-lab and named Robot 1 and Robot 2 respectively. Each robot consists of two controlled wheels at

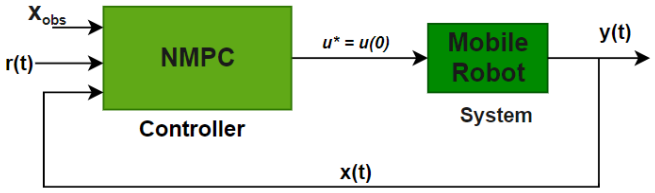
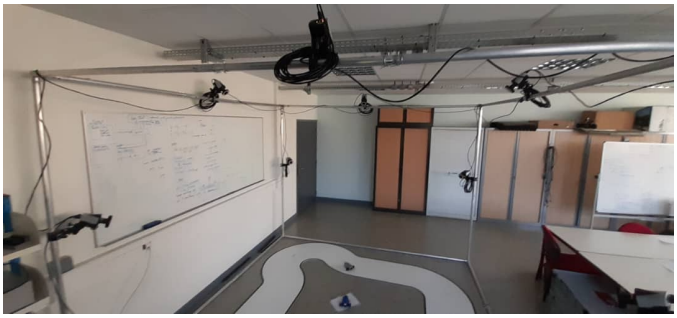


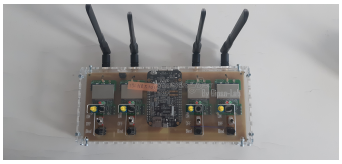
Fig. 3: Control system architecture

(a) Robot 1

(b) Robot 2



(c) Motion capture system



(d) Communication system

Fig. 4: Robots, communication and motion capture systems.

the left and right sides of the robot and a freewheel at the front to support the robot. Two continuous rotation servo motors are coupled to the left and right wheels. Each motor takes the speed command signal (ω_r or ω_l) which controls the movement of the robot (speed and direction). The speed commands of each servo-motor are simultaneously received wirelessly from an Infra Red Communication (IRC) board of Figure 4(d) which communicates with the robot's Spectrum DSMX receiver. The communication between Matlab/Simulink and the IRC board is done via the UDP-protocol block in Simulink.

The motion capture system used in this work is composed of 8 Miqus M3 cameras from the Qualisys company, which is used as a sensor to track the movement of the robots. The cameras, depicted in Figure 4(c) are synchronized using an integrated software which communicates with a Raspberry-pi board. The states of both robots are then sent to Simulink in real-time via the UDP-protocol.

The controllers then use the states and other information to compute the control inputs v and ω which are converted to angular speeds of the left (ω_l) and the right (ω_r) wheels by the following equation:

$$\begin{cases} \omega_r = (2v + \omega R_{rob})/2r \\ \omega_l = (2v - \omega R_{rob})/2r \end{cases} \tag{13}$$

where R_{rob} is the length of the robot's base from the center and r is the radius of the robot's wheels. Robot 2 is considered as an obstacle, while Robot 1 has a task of moving from its initial condition and stabilizes at the referent posture while avoiding the collision with the obstacle. For static obstacle avoidance, Robot 2 is approximated as a circle whose only the position and size is known.

IV. Results and Discussion

This section presents the experimental validation of the proposed controller. The weight matrices are tuned and the best values that stabilize the controller are chosen. The Q and R matrices are designed to be diagonal matrices with diagonal elements respectively equal to $(1, 1, 0.001)$ and $(1, 1)$. The weight on the terminal penalty cost is calculated to be $1000 * Q$.

The camera system depicted in Figure 4(c) is used to detect the instantaneous position and size of the obstacle denoted by ($\mathbf{x} = x_{obs}, y_{obs}, r_{obs}$). With the movement of the obstacle unknown to the controller, the current parameter of the obstacle is sent to the controller which then computes the minimal distance to avoid collision with it. The constraint (14) is added during the formulation of the MPC for obstacles avoidance, as expanded in the following equation.

$$\sqrt{(x_{rob} - x_{obs})^2 + (y_{rob} - y_{obs})^2} \geq (r_{obs} + r_{rob}) \tag{14}$$

where x_{rob}, y_{rob} and r_{rob} are the positions and the radius of the robot respectively. This constraint forces the robot to move tangentially around the surface of the obstacle and then return to its original path after completely avoiding the obstacle.

In Figs. 5 - 8, the experimental results for a point stabilization control problem with static obstacle avoidance is presented [1]. It can be seen that the robot starts from a given initial pose marked at $\mathbf{x}_0 = [-0.9m, -0.7m, \frac{\pi}{2}rad]$ and autonomously moved towards the reference pose marked at $\mathbf{x}_r = [1m, 1m, \frac{\pi}{4}rad]$ where it encountered a static obstacle, which is the second robot approximated as a circle with the diameter of $0.2m$, placed at $(0.15m, 0.15m)$. With the sampling time of $0.1sec$ and a prediction horizon $N = 20$ and the robot's actuator saturation limits obtained from the maximum revolution of the wheels, the linear velocity v ranges from -0.06m/s to 0.06m/s and the angular velocity ω ranges from $-\frac{\pi}{4}$ rad/s to $\frac{\pi}{4}$ rad/s. The state error vector shown on Figure 6 has asymptotically goes to zero for all the states variables with the exception of the variable θ. The slight variation of the reference orientation angle occurred due to experimental errors due to the delay between the commanded controls and the actual controls.

[1]A video on the experiments can be viewed on the following link: video

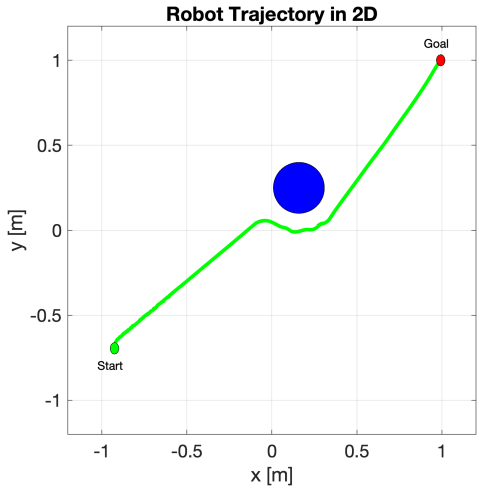


Fig. 5: Point stabilization with static obstacles avoidance.

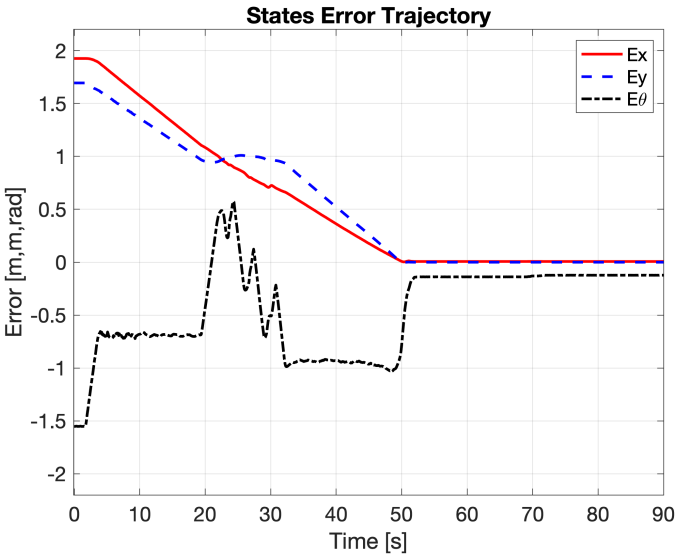


Fig. 6: Point stabilization with static obstacles avoidance: The state error vector.

As presented on Figs 7-8, comparison between the computed (commanded) and real (actual) control profile overlapped. The control profiles have respected the saturation limits given. However small oscillations occurred in the control signal due to small measurement noise present in the data acquisition system.

Secondly, experimental results for a point stabilization problem with dynamic obstacle avoidance is presented in Figure

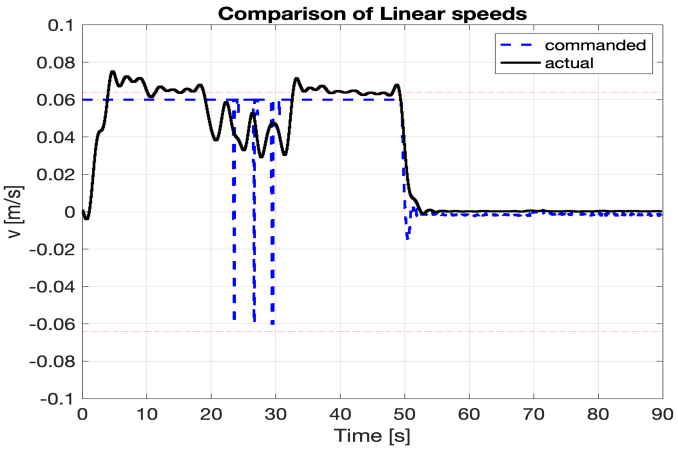


Fig. 7: Point stabilization with static obstacles avoidance: Linear speeds profiles.

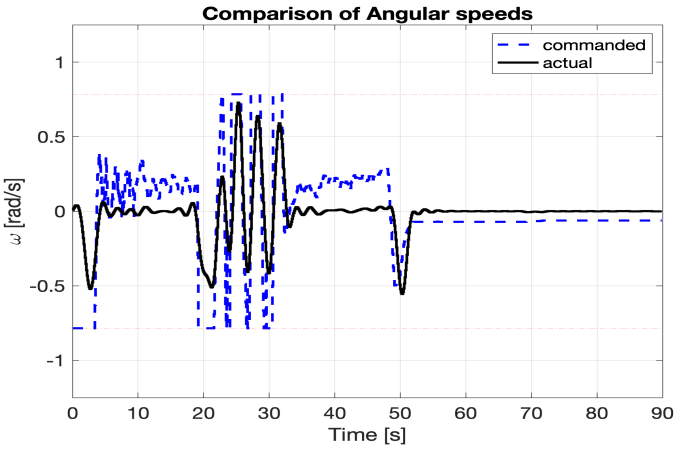


Fig. 8: Point stabilization with static obstacles avoidance: Angular speed profiles.

9. It can be seen that the robot starts from a given initial pose marked at $[-0.9m, -0.7m, \frac{\pi}{2}rad]$ and autonomously moved towards the reference pose marked at $[1, 1, 0]$ while encountering a moving obstacle along the way. Similarly, the second robot is considered as the obstacle which is approximated as a circle with the diameter of $0.2m$ moving in a counter-clockwise direction with a speed of 0.04m/s and turning rate of $\pi/20$ rad/s. The trajectories of the robot and the obstacles at four-time instances are depicted in Figure 9. At time $t = 20secs$, the robot encountered the moving obstacle (shaded circle), and autonomously avoided collision with the it despite the fact that the controller does not know the information about the movement of the obstacle. The robot finally stabilized at the exact reference pose. The state error vector is shown on Figure 10.

As presented in Figs.11-12, comparison between the computed (commanded) and real (actual) control profile is seen to overlap with smooth the control profiles. Here again the control profiles have respected the imposed saturation limits. It is important to note that the commanded speed is the speed

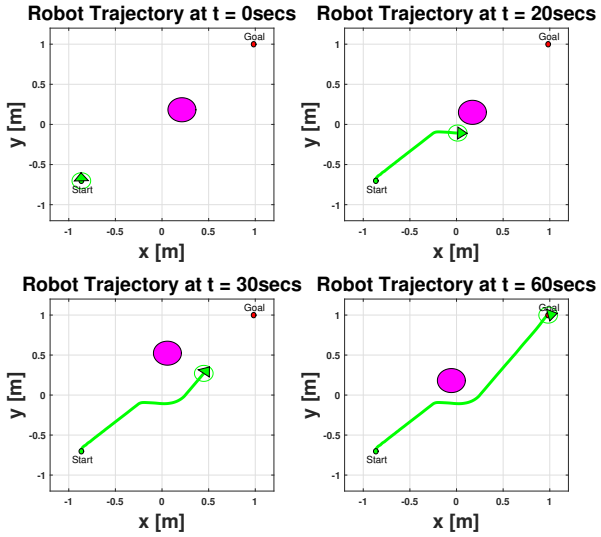


Fig. 9: Point stabilization with dynamic obstacles avoidance: States trajectories.

computed form the controller while the actual speed is the measured obtained from the real data.

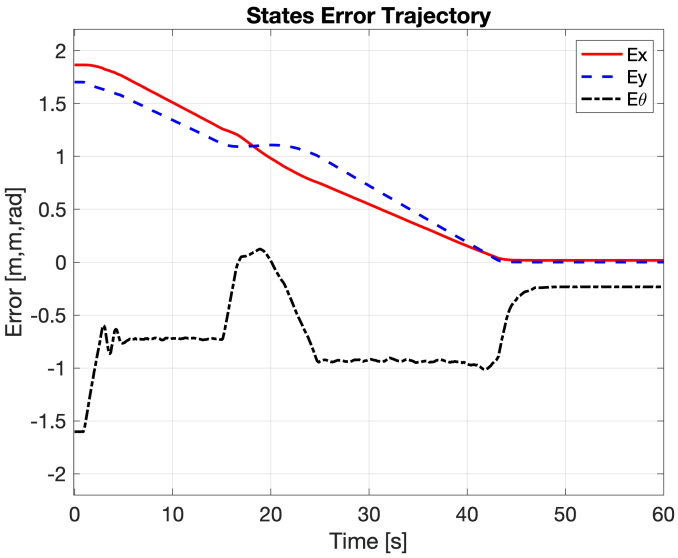


Fig. 10: Point stabilization with dynamic obstacles avoidance: The state error vector.

V. Conclusion

In this paper, we presented a model predictive control-based strategy for dynamic obstacle avoidance for a non-holonomic mobile robot. The proposed controller realized simultaneous online optimal trajectory planning and control. The optimal control inputs are obtained by minimizing the tracking error. The speed profile of the dynamic obstacle was unknown to the controller, thus saved the cost of purchasing and integration of an additional measurement device. The obstacle avoidance was

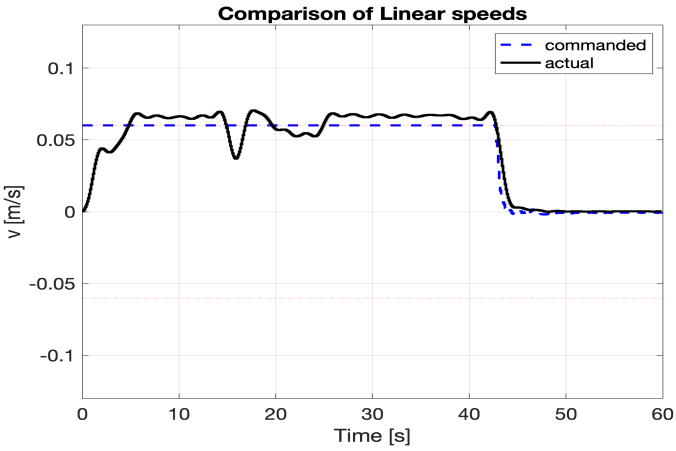


Fig. 11: Point stabilization with dynamic obstacles avoidance: Linear speeds profiles.

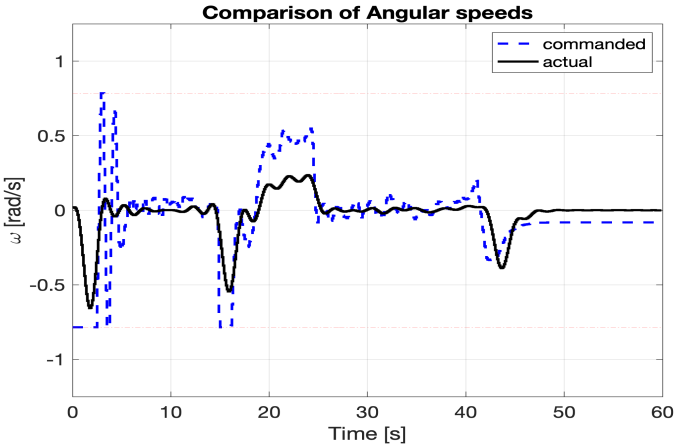


Fig. 12: Point stabilization with dynamic obstacles avoidance: Angular speeds profiles.

integrated as an inequality constraint during the formulation of the MPC which is stabilized by adding a terminal cost function. The experimental platform developed allowed for localization and wireless transmission of the command signals. The experimental results showed that the MPC-based dynamic obstacle avoidance controller exhibits good performance.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] Seung-Mok Lee and Hyun Myung. Receding horizon particle swarm optimisation-based formation control with collision avoidance for nonholonomic mobile robots. *IET Control Theory & Applications*, 9(14):2075–2083, 2015.

[7] T. Faulwasser, B. Kern, and R. Findeisen. Model predictive path-following for constrained nonlinear systems. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 8642–8647, 2009.

[8] 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.

[9] Mohamed W. Mehrez, George K. I. Mann, and Raymond G. Gosine. Comparison of stabilizing nmpc designs for wheeled mobile robots: An experimental study. In *2015 Moratuwa Engineering Research Conference (MERCon)*, pages 130–135, 2015.

[10] 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.

[11] Mukhtar Sani, Bogdan Robu, and Ahmad Hably. Limited information model predictive control for pursuit-evasion games. In *60th Conference on Decision and Control (CDC2021)*, 2021.

[12] Gowtham Garimella, Matthew Sheckells, Joseph L. Moore, and M. Kobilarov. Robust obstacle avoidance using tube nmpc. In *Robotics: Science and Systems*, 2018.

[13] 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.

[14] 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.

[15] 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.

[16] Shaosong Li, Zheng Li, Zhixin Yu, Bangcheng Zhang, and Niaona Zhang. Dynamic trajectory planning and tracking for autonomous vehicle with obstacle avoidance based on model predictive control. *IEEE Access*, 7:132074–132086, 2019.

[17] Mukhtar Sani, Bogdan Robu, and Ahmad Hably. Dynamic obstacles avoidance using nonlinear model predictive control. In *47th Annual Conference of the IEEE Industrial Electronics Society (IES)*, 2021.

[18] L. Jualin. *Mobile Robotics*. Dover Books on Mathematics. Elsevier Publications, 2015.

[19] Xie Feng. *Model Predictive Control of Non-holonomic Mobile Robots*. PhD thesis, Oklahoma State university, USA, 2004.

[20] Mazen Alamir. *Stabilization of Nonlinear Systems Using Receding-horizon Control Schemes: A Parametrized Approach for Fast Systems*. Springer, 2006.

[21] J. B. Rawlings, D. Q. Mayne, and M.M. Diehl. *Model Predictive Control: Theory,Computation and Design*. 2nd Edition. Nob Hill Publishing, LLC, 2019.

[22] 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.

[23] 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.